Communication Lower Bounds via Query Complexity

by

Mika Göös

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy,
Graduate Department of Computer Science,
University of Toronto

Abstract

Communication Lower Bounds via Query Complexity

Mika Göös
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto, 2016

The goal of this thesis is to prove lower bounds in ***communication complexity*** by exploiting new connections to ***query complexity***. Our basic strategy is to first prove a general theorem stating that for a large class of communication problems F, any protocol for F can be efficiently simulated by a decision tree solving a related problem f, and then rule out efficient decision trees for f. We use this approach to resolve several open problems (some well-known):

(1): We prove superlogarithmic lower bounds on the conondeterministic communication complexity of the *Clique vs. Independent Set* game, answering a question of Yannakakis (STOC 1988). As a corollary, this implies superpolynomial lower bounds for the Alon–Saks–Seymour conjecture in graph theory. Furthermore, we obtain near-optimal deterministic (and even randomised) lower bounds for the Clique vs. Independent Set game. As a corollary, this implies new lower bounds for the log-rank conjecture of Lovász and Saks (FOCS 1988).

(2): We show that deterministic (and even randomised) communication complexity can be superlogarithmic in the partition number of the associated communication matrix. This answers a basic question of Yao (STOC 1979), also posed by Nisan and Kushilevitz (1997).

(3): We prove near-optimal randomised communication lower bounds for the recursive AND-OR tree. This strengthens the classical decision tree results of Saks and Wigderson (FOCS 1986) and Santha (RSA 1995), and answers a question of Beame and Lawry (STOC 1992).

(4): We exhibit an n-variable monotone function in NP with near-maximal monotone circuit depth complexity $\Omega(n/\log n)$. This improves on the previous record of $\Omega(\sqrt{n})$ proved for the perfect matching function by Raz and Wigderson (JACM 1992).

(5): We exhibit an n-node graph whose independent set polytope requires extended formulations of size exponential in $\Omega(n/\log n)$. Previously, no explicit examples of n-dimensional 0/1-polytopes were known with extension complexity larger than exponential in $\Theta(\sqrt{n})$.

Acknowledgements

It is a truth universally acknowledged that students tend to acquire, through osmosis, characteristics of their supervisors. Therefore I think it makes for an interesting exercise to list some traits that I have ***not*** yet succeeded in copying from Toni Pitassi: In research, her focus is often on the most important questions of the field, fuelled by optimism that such hard problems are indeed soluble. (I still sometimes secretly think about less important contraband questions, which Toni specifically advised against.) In non-research matters, despite concerted efforts, I never managed to make her lose her patience over my bureaucratic incompetencies. I have heard multiple outsiders describe her as "chill", a word whose meaning I have yet to look up. I remain in enormous debt to Toni for pulling all the relevant strings that enabled many wonderful funding/internship/workshop opportunities. In addition to Toni's leadership, much of this thesis resulted from join investigations with my associate Dr. (Tom) Watson. We've spent $\aleph_0$ many hours in the trenches together, peering at whiteboards. The key character trait differentiating us is that Tom thinks before he speaks. However, I have not let this get in the way of our science, nor his rather quaint insistence that one should only prove theorems that are true. Needless to say, all the remaining errors in this thesis are his. I'm only joking! Part of the blame (and thanks) goes to my other coauthors: David, Jayram, Raghu, Rahul, Shachar. I am grateful to Ran Raz for serving as the external examiner for this thesis, as well as to the rest of my PhD committee: Ben, Faith, Sasho, Stephen, Vassos. This thesis work was partially supported by a grant from the Simons Foundation (#360891).

I feel privileged to have been able to intern in the few remaining pockets of industry where it is still possible to pursue this kind of theoretical research. In summer 2014, at Microsoft Research Silicon Valley, I was awestruck with its intense and massively collaborative environment. Whilst there, I tricked my versatile mentor Raghu Meka to work on a problem that resulted in maybe the most important chapter of this thesis. The MSR internship was literally too good to be true: weeks after my internship, Microsoft decided to shut down its Silicon Valley lab. In summer 2015, at IBM Almaden, I was met with a more chill (I looked up the word just now) and distractionless atmosphere. This time my mentor T.S. Jayram (the 'J' in [BJKS04]!) was the one proposing research topics, resulting in two thesis chapters. I thank my co-interns for excellent company; at MSR: Ilya, Li-Yang, Mohsen; at IBM: Dimitris, Ilya (again?), Zhao.

Before Toronto, I spent a year researching distributed computing under Jukka Suomela at University of Helsinki. I hope that some remnants of his meticulous style of exposition can be recognised in parts of this thesis, although I admit that I have become "lazier" (to use his choice of words) in this. For reminding me that there is more to TCS than complexity theory, I thank my contemporaries: David(s), Dhinakaran, Jai, Jimmy, Joel, Kaveh, Lalla, Robert, Trevor, Tyrone, Venkatesh, Yuval. For attempts to persuade me that there is more to life than TCS, I thank my fellow grads at Chestnut (especially the old guard: Ahmed, James, Justin) and my Finnish sauna crew (usual suspects: Janne(s), Jere, Jussa, Taneli).

This thesis is dedicated to my parents.

Dedication

Isälle ja äidille.

Contents

1 Overview 1
- 1.1 Query vs. communication . . . 2
- 1.2 Our contributions . . . 3
- 1.3 Subsequent developments . . . 9
- 1.4 What is not included? . . . 10

2 Rectangles Are Nonnegative Juntas 11
- 2.1 Introduction . . . 11
- 2.2 Proof of the Junta Theorem . . . 19
- 2.3 Definitions of models . . . 30
- 2.4 Proof of the Simulation Theorem . . . 34
- 2.5 Applications of the Simulation Theorem . . . 37
- 2.6 Unrestricted–restricted equivalences . . . 41
- 2.7 Additional proofs . . . 46

3 Lower Bounds for Clique vs. Independent Set 50
- 3.1 Introduction . . . 50
- 3.2 Definitions . . . 53
- 3.3 Warm-up discussion . . . 53
- 3.4 Projective plane example . . . 55
- 3.5 Recursive composition . . . 57

4 Deterministic Communication vs. Partition Number 61
- 4.1 Introduction . . . 61
- 4.2 Query Separations . . . 64
- 4.3 Raz–McKenzie Simulation . . . 66

5 Randomised Communication vs. Partition Number 75
- 5.1 Introduction . . . 75
- 5.2 Complexity measures . . . 77
- 5.3 Overview . . . 79

5.4 Decision tree lower bound 84
5.5 AND-composition lemmas 87
5.6 Basic lemmas 94

6 A Composition Theorem for Conical Juntas 97
6.1 Introduction 97
6.2 Definitions and examples 100
6.3 Statement of the Composition Theorem 103
6.4 Proof of the Composition Theorem 104
6.5 Approximate degree lower bounds 108
6.6 Communication lower bounds 110

7 Lower Bounds via Critical Block Sensitivity 112
7.1 Introduction 112
7.2 Versatile gadgets 121
7.3 Communication lower bound 126
7.4 Critical block sensitivity lower bounds 130
7.5 Monotone CSP-SAT 134
7.6 Applications: Proof complexity 136

8 Extension Complexity of Independent Set Polytopes 143
8.1 Introduction 143
8.2 KW/EF connection 146
8.3 Tseitin problem 148
8.4 Reductions 150
8.5 Our gadget 153
8.6 Communication lower bound 156
8.7 Query lower bound 169

9 Open problems 173

Bibliography 175

Chapter 1

Overview

The goal of this thesis is to prove lower bounds in ***communication complexity*** by exploiting new connections to ***query complexity***. While interesting in their own right, communication lower bounds find many applications in (and even beyond) computational complexity theory. In this thesis alone, we will encounter applications to graph theory, combinatorial optimisation (linear programming formulations), monotone circuit complexity, and proof complexity.

In **query complexity**, the objects of study are *decision trees*, one of the simplest and most basic models of computation. A decision tree algorithm evaluates a n-bit boolean function $f\colon \{0,1\}^n \to \{0,1\}$ on an unknown input $x \in \{0,1\}^n$ by repeatedly querying individual input variables. In each step, the algorithm specifies a coordinate $i \in [n]$ and gets to learn $x_i \in \{0,1\}$.

How many queries are needed (in the worst case) to evaluate f?

This basic template can be instantiated for many different types of computation: deterministic, nondeterministic, randomised, etc. The holy grail of complexity theory is to prove separations between different types of computation in the Turing machine model (e.g., $\mathsf{P} \neq \mathsf{NP}$)—while these questions remain hopelessly out of reach, query complexity has its historical roots in separating complexity classes relative to *oracles*; see Vereshchagin [Ver99] for an exposition. Standard references for query complexity include Buhrman and de Wolf [BdW02] and Jukna [Juk12].

In **communication complexity**, the objects of study are *communication protocols*. In the basic model (introduced by Yao [Yao79]), two players, Alice and Bob, share the input to a function $f\colon \{0,1\}^n \to \{0,1\}$ according to some fixed bipartition of the variables, e.g., Bob gets the first half $x \in \{0,1\}^{n/2}$ and Alice gets the second half $y \in \{0,1\}^{n/2}$ of the input. Their goal is to engage in a dialogue over a shared communication channel in order to compute $f(x, y)$.

How many bits must be transmitted between the players to evaluate f?

Again, this basic template can be instantiated for many different types of computation. As with query complexity, classical complexity classes often come with natural communication complexity analogues (introduced by Babai, Frankl, and Simon [BFS86]). Standard references for communication complexity include Kushilevitz and Nisan [KN97] and Jukna [Juk12]. Moreover, Rao and Yehudayoff are working on a new exciting textbook [RY16]!

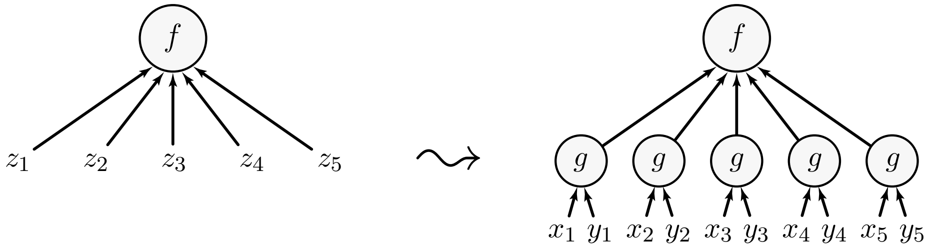


Figure 1.1: In this thesis, we study the communication complexity composed functions $f \circ g^n$ where $f\colon \{0,1\}^n \to \{0,1\}$ is an arbitrary n-bit boolean function and $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is a carefully chosen two-party gadget.

1.1 Query vs. communication

Query $\rightsquigarrow$ Communication. Communication protocols are at least as powerful as decision trees. Indeed, a decision tree that computes f using at most d queries can be simulated by a protocol that communicates at most d bits: a query to the i-th coordinate is simulated by having the player who knows x_i send that bit to the other player. Consequently, lower-bound results against communication protocols are *stronger* than corresponding lower-bound results against decision trees.

Communication $\rightsquigarrow$ Query. A unifying theme in this thesis is to find situations where the above simulation can be *reversed.* More specifically, our basic strategy for proving a communication lower bound is as follows.

Step 1. Prove a general *communication-to-query simulation theorem* stating that for a large class of communication problems F, any protocol for F can be efficiently simulated by a decision tree solving a related problem f.

Step 2. Rule out efficient decision trees for f.

In this thesis, we will focus on the case where F is a *composed* (or *lifted*) function of the form $f \circ g^n$ where $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is some carefully chosen two-party function, often called a *gadget*; see Figure 1.1. Here Alice and Bob are given inputs $x \in \mathcal{X}^n$ and $y \in \mathcal{Y}^n$, respectively. Their goal is to compute

$$F(x,y) \;\coloneqq\; f(g(x_1,y_1),\ldots,g(x_n,y_n)).$$

Intuitively, the difficulty in computing $F \coloneqq f \circ g^n$ stems from the fact that for any i, the i-th input bit $z_i \coloneqq g(x_i,y_i)$ to f remains unknown to either party until they decide to communicate enough information about x_i and y_i. A simulation theorem aims to formalise this intuition: there is no better way for a protocol to compute the composed function $f \circ g^n$ other than to behave like a decision tree querying input bits of f. Provided that the gadget g itself is "small", this relates the communication and query complexities of $f \circ g^n$ and f, sometimes even up to

Reference	Query model	Communication model	Method of lifting
[RM99], Ch. 4	deterministic	deterministic	gadget composition
[SZ09, She11a]	polynomial degree	rank	gadget composition
Chapter 2	conical junta degree	nonnegative rank	gadget composition
[CLRS13]	Sherali–Adams	LP extension complexity	random embedding
[LRS15]	sum-of-squares	SDP extension complexity	random embedding

Table 1.1: Notable communication-to-query simulation theorems at a glance. The first three are formulated in the language of boolean functions (as described in Section 1.1); the last two are formulated in the language of combinatorial optimisation.

constant factors.

One of the main technical contributions of this thesis is a new simulation theorem that applies for certain types of randomised and/or nondeterministic computations. Another result that makes a prominent appearance is a deterministic simulation theorem due to Raz and McKenzie [RM99]. Even in situations where we cannot quite yet prove a fully general simulation theorem, we can nevertheless benefit from intuition garnered from studying the decision tree analogue f of a communication problem $f \circ g^n$ under consideration. We emphasise that exploring the interplay between query and communication complexity is nothing new: it is a recurring theme exploited by many works in the literature, e.g., [NW95, RM99, SZ09, She11a, HN12, CLRS13, LRS15]; see also Table 1.1.

1.2 Our contributions

Chapter 2: Rectangles Are Nonnegative Juntas

The silent workhorse behind many of our contributions is a novel communication-to-query simulation theorem for composed functions $f \circ g^n$ where f is arbitrary and the gadget g is chosen carefully. We show that every randomised bounded-error protocol for $f \circ g^n$ that communicates d bits can be simulated by a degree-$O(d)$ *conical junta* computing f. A *conical junta* is a nonnegative analogue of a multivariate polynomial, namely, a nonnegative combination of conjunctions of literals (input bits or their negations); its degree is the maximum number of literals in any conjunction. (Conical juntas have also been studied under such names as the (one-sided) *partition bound* for query complexity [JK10] and *query complexity in expectation* [KLdW15].) Here are two examples of conical juntas, one computing the two-bit OR function $\mathsf{OR}\colon \{0,1\}^2 \to \{0,1\}$ and another computing the three-bit majority function $\mathsf{Maj}_3\colon \{0,1\}^3 \to \{0,1\}$:

$$\begin{aligned} h_1(x) &= \tfrac{1}{2}x_1 + \tfrac{1}{2}x_2 + \tfrac{1}{2}\bar{x}_1 x_2 + \tfrac{1}{2}x_1\bar{x}_2, \\ h_2(y) &= \tfrac{1}{3}y_1y_2 + \tfrac{1}{3}y_2y_3 + \tfrac{1}{3}y_1y_3 + \tfrac{2}{3}\bar{y}_1y_2y_3 + \tfrac{2}{3}y_1\bar{y}_2y_3 + \tfrac{2}{3}y_1y_2\bar{y}_3. \end{aligned} \tag{1.1}$$

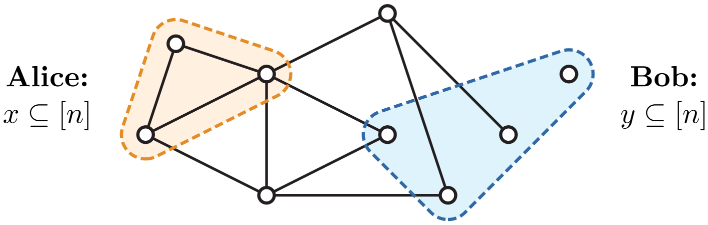
The upshot of the simulation theorem is that a lower bound on the randomised communication complexity of $f \circ g^n$ can now be proved by showing a degree lower bound for conical juntas that (approximately) compute f. More generally, the simulation theorem allows us to characterize the communication complexity of $f \circ g^n$ in all known one-sided (i.e., not closed under complement) *zero-communication* models (defined in Chapter 2) by a corresponding query complexity measure of f. In particular, this includes nondeterministic models, i.e., query and communication analogues of NP. As immediate applications, we also resolve some open problems from prior work [Kla03, BGM06, KMSY14].

This chapter is based on the following publication:

[GLM$^+$15]: Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. In *Proceedings of the 47th Symposium on Theory of Computing (STOC)*, pages 257–266. ACM, 2015. doi:10.1145/2746539.2746596

Chapter 3: Lower Bounds for Clique vs. Independent Set

Since our simulation theorem applies for nondeterministic computations, it suggests an approach to attack an old question of Yannakakis [Yan91] concerning the conondeterministic communication complexity of the *Clique vs. Independent Set* problem. This problem is defined relative to an undirected n-node graph $G = ([n], E)$ as follows: Alice holds a clique $x \subseteq [n]$ in G, Bob holds an independent set $y \subseteq [n]$ in G, and their goal is to decide whether x and y intersect. As the underlying graph enforces $|x \cap y| \in \{0, 1\}$, we may define a boolean function by $\mathsf{CIS}_G(x, y) \coloneqq |x \cap y|$.



It is easy to nondeterministically certify that $x \cap y \neq \emptyset$ by just guessing a node in the intersection—this is $\lceil \log n \rceil$ bits of nondeterministic communication (on any graph G). Yannakakis asked whether the complement of the problem is any harder: can we also certify that $x \cap y = \emptyset$ with just $O(\log n)$ bits of nondeterministic communication (on any graph G)? An upper bound (holding even for deterministic protocols) of $O(\log^2 n)$ was given by Yannakakis.

In Chapter 3 we answer this question by proving an $\omega(\log n)$ lower bound on the conondeterministic communication complexity of the CIS_G problem (for some graph G). Our approach is to first exhibit a query complexity separation for the decision tree analogue of the UP vs. coNP question—namely, unambiguous DNF width vs. CNF width—and then apply our nondeterministic simulation theorem.

The Clique vs. Independent Set problem is interesting partly because it admits so many equivalent formulations, as recently explored by Bousquet, Lagoutte, and Thomassé [BLT14]. Let us mention one consequence here: Our lower-bound result refutes a certain "polynomial" version of the *Alon–Saks–Seymour* conjecture in graph theory. The original conjecture (formulated around 1990) stated that $\mathrm{chr}(G) \le \mathrm{bp}(G) + 1$, i.e., that the chromatic number of G can be bounded in terms of the *biclique partition number* of G (minimum number of complete bipartite graphs needed to partition the edges of G). After several decades of it being open, Huang and Sudakov [HS12] disproved the original conjecture by showing that $\mathrm{chr}(G)$ can be polynomially larger than $\mathrm{bp}(G)$. Our result implies that the gap can be superpolynomial.

This chapter is based on the following publication:

[Göö15]: Mika Göös. Lower bounds for clique vs. independent set. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, pages 1066–1076. IEEE, 2015. doi:10.1109/FOCS.2015.69

Chapter 4: Deterministic Communication vs. Partition Number

Perhaps the most basic observation in communication complexity is that a deterministic protocol of communication cost d that computes a function $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ partitions the domain $\mathcal{X} \times \mathcal{Y}$ into at most 2^d *monochromatic rectangles*. A *rectangle* is a set $R := A \times B$ where $A \subseteq \mathcal{X}$, $B \subseteq \mathcal{Y}$, and we say that R is *monochromatic* if F is constant on R. Hence, the logarithm of the *partition number* $\chi(F)$, defined as the least number of monochromatic rectangles needed to partition $\mathcal{X} \times \mathcal{Y}$, is a lower bound on deterministic communication complexity. It was already asked by Yao [Yao79] to determine the relationship between $\log \chi(F)$ and the deterministic communication complexity of F. For upper bounds, Aho, Ullman, and Yannakakis [AUY83] showed that $O(\log^2 \chi(F))$ bits of deterministic communication suffice to compute F.

In Chapter 4 we show that deterministic communication complexity can be superlogarithmic in $\chi(F)$ by giving an example of an F with deterministic complexity $\tilde{\Omega}(\log^{1.5} \chi(F))$. We also obtain near-optimal $\tilde{\Omega}(\log^2 n)$ deterministic communication lower bounds for the Clique vs. Independent Set problem. (Note that this is incomparable to the conondeterministic result discussed above.) In particular, this yields new lower bounds for the famous *log-rank conjecture* [LS88], which postulates that deterministic communication complexity is polynomially related to the logarithm of the rank (over the reals) of F (here F is viewed as a boolean matrix indexed by x, y). We exhibit an F with deterministic communication complexity $\tilde{\Omega}(\log^2 \mathrm{rank}(F))$, while the best previous bound was $\Omega(\log^{1.63} \mathrm{rank}(F))$ due to Kushilevitz [Kus94].

Our approach is again to exploit a communication-to-query simulation theorem: we prove analogous results in query complexity and then invoke (a small adaptation of) a deterministic simulation theorem due to Raz and McKenzie [RM99].

This chapter is based on the following publication:

[GPW15]: Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, pages 1077–1088. IEEE, 2015. doi:10.1109/FOCS.2015.70

Chapter 5: Randomised Communication vs. Partition Number

Given the deterministic lower-bound results discussed above, an obvious follow-up question is: what happens in the randomised case? Of course, for randomised protocols, $\log\chi(F)$ is no longer a lower bound on the bounded-error randomised communication complexity of F. For example, the n-bit *equality* function $\mathsf{EQ}\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ defined by $\mathsf{EQ}(x,y) = 1$ iff $x = y$ is such that $\log\chi(\mathsf{EQ}) = n$, but the randomised complexity is well-known to be $O(\log n)$ (or even $O(1)$ if we allow public randomness).

In Chapter 5 we strengthen the results of Chapter 4 to the setting of randomised protocols. We show that randomised communication complexity of a function F can be $\tilde{\Omega}(\log^{1.5}\chi(F))$. Furthermore, we obtain near-optimal $\tilde{\Omega}(\log^2 n)$ randomised lower bounds for the CIS_G problem.

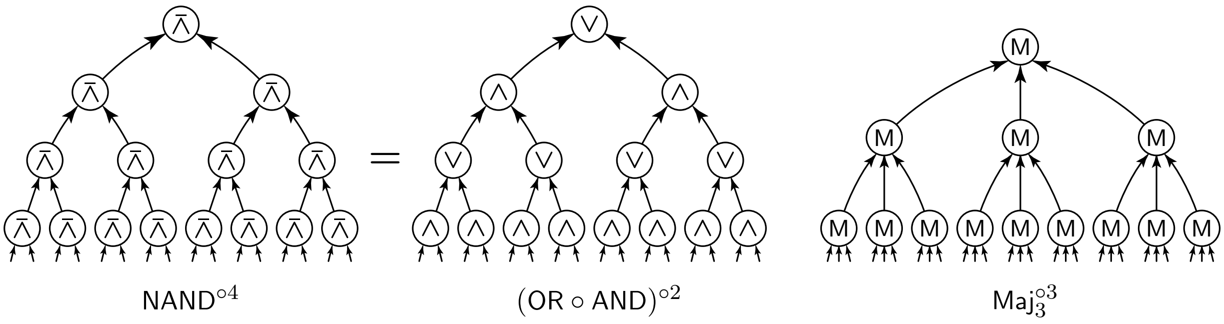
Our separation between $\log\chi(F)$ and randomised communication complexity is technically interesting for a few reasons. First, such a separation cannot be obtained using the standard rectangle-based lower-bound methods, as catalogued by Jain and Klauck [JK10]. Second, there is no known simulation theorem for usual bounded-error randomised computations (BPP). In particular, our junta-based simulation result falls within the purview of rectangle-based methods and hence is subject to their limitations. Our solution is to *mix techniques*: we use our junta-based simulation theorem together with *information complexity* techniques (which are amongst the most powerful lower-bound methods in communication complexity) to achieve our separation.

This chapter is based on the following publication:

[GJPW15]: Mika Göös, T.S. Jayram, Toniann Pitassi, and Thomas Watson. Randomized communication vs. partition number. Technical Report TR15-169, Electronic Colloquium on Computational Complexity (ECCC), 2015. URL: http://eccc.hpi-web.de/report/2015/169/

Chapter 6: A Composition Theorem for Conical Juntas

What else can we apply our junta-based simulation theorem to? A line of work [JKS03, JKR09, LS10, JKZ10] has studied the randomised communication complexity of AND-OR trees, i.e., functions computable by formulas consisting of alternating levels of unbounded fan-in AND and OR gates. For the purposes of communication complexity, it is natural to assume that these formulas have gates of binary fan-in next to the inputs so that Alice gets the first bit of every bottom gate, and Bob gets the second bit of every bottom gate. For example, the ubiquitous set-disjointness function $\mathsf{OR}_n \circ \mathsf{AND}_2^n$ is of this form. For balanced AND-OR trees of height k, a



$\mathsf{NAND}^{\circ 4}$ $(\mathsf{OR} \circ \mathsf{AND})^{\circ 2}$ $\mathsf{Maj}_3^{\circ 3}$

Figure 1.2: Examples of recursively defined boolean functions studied in Chapter 6.

randomised lower bound of $\Omega(n/2^{O(k)})$ was proved in two independent works [LS10, JKR09]. While this lower bound is tight when $k = O(1)$, the bound becomes trivial for the extreme case of binary trees of height $k = \log n$ (left side of Figure 1.2). This shortcoming was partially addressed by [JKZ10] who showed, via a reduction from set-disjointness, a lower bound of $\Omega(\sqrt{n})$ for such AND-OR trees, independently of the height. For randomised *query* complexity, a tight lower bound of $\Omega(n^{0.753\ldots})$ for height $k = \log n$ is a well-known classical result [SW86, San95].

In Chapter 6 we develop a general method of proving degree lower bounds for conical juntas that compute such recursively defined boolean functions. In particular, when applied to binary AND-OR trees, we get a near-optimal $\tilde{\Omega}(n^{0.753\ldots})$ randomised communication lower bound, which answers a question posed by Beame and Lawry [BL92, Law93]. In addition, we apply this method to analyse three-bit majority trees of height k (right side of Figure 1.2): we show an $\Omega(2.59^k)$ randomised communication lower bound, which improves the state-of-the-art even for *query* complexity [JKS03, LNPV06, Leo13, MNS$^+$15].

This chapter is based on the following publication:

[GJ16]: Mika Göös and T.S. Jayram. A composition theorem for conical juntas. In *Proceedings of the 31st Conference on Computational Complexity (CCC)*, pages 5:1–5:16. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.CCC.2016.5

Chapter 7: Lower Bounds via Critical Block Sensitivity

A major drawback with our junta-based simulation theorem (and also with that of [RM99]) is that it requires the gadget $g\colon \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$ to be of logarithmic size, $b = \Theta(\log n)$. For some applications, this is prohibitively large; one would ideally hope for $b = O(1)$. For example, we are unable to use our simulation theorem to reproduce the famous $\Omega(n)$ randomised lower bound for set-disjointness $\mathsf{OR}_n \circ \mathsf{AND}_2^n$ [KS92, Raz92, BJKS04]. Our simulation theorem *does* yield an $\Omega(n)$ bound for the composed function $\mathsf{OR}_n \circ g^n$, but rewriting the gadget g itself as an OR of AND's incurs an exponential-in-b factor blow-up in the number of input variables.

In Chapter 7 we study *critical block sensitivity*, a query complexity measure introduced by Huynh and Nordström [HN12], that is better suited to proving lower bounds for communication problems lifted with constant-size gadgets. For starters, we give a simple new proof (via a reduction from set-disjointness) of the following central result of [HN12]: if S is a search problem with critical block sensitivity d, then every randomised protocol solving the composed search problem $S \circ g^n$ (where g is a certain constant-size gadget) requires $\Omega(d)$ bits of communication. Besides simplicity, our proof has the advantage of generalising to the multi-party setting. We obtain the following applications.

– *Monotone circuit depth:* We exhibit a monotone function in NP on n variables whose monotone circuits require depth $\Omega(n/\log n)$; previously, a bound of $\Omega(\sqrt{n})$ was known for an explicit function [RW92]. Moreover, we prove an $\Omega(\sqrt{n})$ monotone depth bound for a function in monotone P.

– *Proof complexity:* We prove new rank lower bounds as well as obtain the first length–space lower bounds for semi-algebraic proof systems, including Lovász–Schrijver and Lasserre (SOS) systems. In particular, these results extend and simplify the works [BPS07, HN12].

This chapter is based on the following publication:

[GP14]: Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the 46th Symposium on Theory of Computing (STOC)*, pages 847–856. ACM, 2014. doi:10.1145/2591796.2591838

Chapter 8: Extension Complexity of Independent Set Polytopes

Our final contribution is to the study of linear programming formulations for polytopes, or *extended formulations* for short. An *extended formulation* is a description of a given polytope $P \subseteq \mathbb{R}^n$ as the projection of a higher dimensional polytope $E \subseteq \mathbb{R}^e$:

$$P \;=\; \{x \in \mathbb{R}^n : (x, y) \in E \text{ for some } y\}.$$

The *extension complexity* of P is the minimum number of facets in an extended formulation E for P. Extended formulations are useful for solving combinatorial optimization problems: instead of optimizing a linear function over P, we can optimize it over E—this may be more efficient since the runtime of LP solvers often depends on the number of facets. A celebrated work of Fiorini et al. [FMP$^+$15] was the first to prove extension complexity lower bounds for explicit 0/1-polytopes (convex hulls of subsets of $\{0,1\}^n$) of relevance to combinatorial optimisation. In fact, this settled another question from Yannakakis's seminal work [Yan91]. Yannakakis characterised the extension complexity of a polytope P as the *nonnegative rank* (which can be viewed as a communication analogue of conical junta degree) of a certain slack matrix of P. Extension complexity is thus "really" a communication complexity measure, and hence amenable to analysis using the tools featured in this thesis.

In Chapter 8 we exhibit an n-node graph G whose independent set polytope (convex hull of the indicator vectors of independent sets of G) requires extended formulations of size exponential in $\Omega(n/\log n)$. Previously, no explicit examples of n-dimensional 0/1-polytopes were known with extension complexity larger than exponential in $\Theta(\sqrt{n})$.

As the reader might already deduce from the above numerology, our result is related to the $\Omega(n/\log n)$ monotone depth lower bound from Chapter 7. Namely, our approach is to *strengthen* the monotone depth lower bound into an extension complexity lower bound. Our construction is inspired by a relatively little-known connection between extended formulations and (monotone) circuit depth [Hru12, Raz90]. For this result, we are not able to use any known simulation theorems; however, we still benefit from intuitions drawn from query complexity. In fact, as a bonus, we prove an optimal $\Omega(n)$ bound for a certain query complexity analogue of the extension complexity question (which incidentally answers a question of Lovász et al. [LNNW95]). If the junta-based simulation could be made to work with constant-size gadgets, we would immediately get an optimal exponential-in-$\Omega(n)$ extension complexity bound for independent set polytopes.

This chapter is based on the following publication:

[GJW16]: Mika Göös, Rahul Jain, and Thomas Watson. Extension complexity of independent set polytopes. In *Proceedings of the 57th Symposium on Foundations of Computer Science (FOCS)*, 2016. To appear. URL: http://eccc.hpi-web.de/report/2016/070/

1.3 Subsequent developments

Because of *Step 2* of our communication-to-query strategy outlined in Section 1.1, our proofs often yield as a by-product new results in query complexity. For example, in Chapter 4 we construct a boolean function with the largest (to-date) gap between its deterministic decision tree complexity and polynomial degree (the previous best was by Kushilevitz [Kus94]). We are delighted to report that this particular construction has been subsequently extended by other researchers. Most notably, a conjecture of Saks and Wigderson [SW86] stated that the largest possible separation between deterministic and zero-sided randomised query complexities should be a power roughly 1.326 (witnessed by the binary AND-OR tree)—generalising our function, Ambainis et al. [ABB+16a] disproved this conjecture by giving a *quadratic* separation. Other separations obtained in [ABB+16a] include a quadratic separation between zero-sided and one-sided randomised query complexities (no gap was known previously), as well as a power 4 separation between deterministic and quantum query complexities (decision tree analogue of BQP). The paper [MS15] independently showed that our function from Chapter 4 already disproves the Saks–Wigderson conjecture.

Another paper that was inspired by our construction was that of [ABK16] who exhibited a function witnessing a power 2.5 separation between decision tree analogues of BPP and BQP, as well as a function witnessing a 4th power separation between quantum query complexity and approximate polynomial degree.

Our power 1.5 separation between deterministic communication complexity and $\log \chi(F)$ has been improved to $2 - o(1)$ by Ambainis, Kokainis, and Kothari [AKK15]. This is essentially optimal in light of the quadratic upper bound [AUY83].

In [ABB+16b] (which is left out of this thesis) we have extended the results of Chapter 5 by giving an essentially optimal power $2 - o(1)$ separation between randomised communication complexity and $\log \chi(F)$. The construction in that paper is a communication version of the analogous query complexity result from [AKK15]. Another result from [ABB+16b] is a power 2.5 separation between the communication analogues of BPP and BQP, which was inspired by the aforementioned result in query complexity [ABK16].

1.4 What is not included?

Several papers that I have co-authored during my PhD studies have not made their way into this thesis. Some of the results do not quite fit into our *communication-to-query* storyline: In [GPW16b] we studied the power of Arthur–Merlin communication protocols by asking whether information complexity techniques can be used to prove lower bounds against this model (surprisingly, *no*). In [GW14, GPW16a] we explored some structural properties of communication complexity classes. The work [ABB+16b] was mentioned above.

Outside of communication complexity, I have worked on separating monotone circuits computing linear boolean operators [FGJ+16]. Lastly, none of my work in distributed computing is included [GS16, GHS14a, GHS14b, GS13, FGK+14, FGKS13, GHL+16].

Chapter 2

Rectangles Are Nonnegative Juntas

Overview. In this chapter, we develop a new method to prove communication lower bounds for composed functions of the form $f \circ g^n$ where f is any boolean function on n inputs and g is a sufficiently "hard" two-party gadget. Our main structure theorem states that each rectangle in the communication matrix of $f \circ g^n$ can be simulated by a *nonnegative combination of juntas*. This is a new formalization for the intuition that each low-communication randomised protocol can only "query" few inputs of f as encoded by the gadget g. Consequently, we characterize the communication complexity of $f \circ g^n$ in all known one-sided (i.e., not closed under complement) zero-communication models by a corresponding query complexity measure of f. These models in turn capture important lower bound techniques such as corruption, smooth rectangle bound, relaxed partition bound, and extended discrepancy. This chapter is based on the following publication:

[GLM+15]: Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. In *Proceedings of the 47th Symposium on Theory of Computing (STOC)*, pages 257–266. ACM, 2015. doi:10.1145/2746539.2746596

2.1 Introduction

Many functions studied in communication complexity (e.g., equality, set-disjointness, inner-product, gap-hamming; see [KN97, Juk12]) are *composed functions* of the form $f \circ g^n$ where $f\colon \{0,1\}^n \to \{0,1,*\}$ is a partial function and $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is some small two-party function, often called a *gadget*. Here Alice and Bob are given inputs $x \in \mathcal{X}^n$ and $y \in \mathcal{Y}^n$, respectively; we think of the inputs as being partitioned into *blocks* $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ for $i \in [n]$. Their goal is to compute

$$(f \circ g^n)(x, y) \;\coloneqq\; f(g(x_1, y_1), \ldots, g(x_n, y_n)).$$

Intuitively, the difficulty in computing $f \circ g^n$ stems from the fact that for any i, the i-th input $z_i := g(x_i, y_i)$ to f remains unknown to either party until they decide to communicate enough information about x_i and y_i. Indeed, an educated guess is that—assuming g is chosen carefully—the communication complexity of $f \circ g^n$ should be explained by some *query* measure of f.

This chapter is about formalizing the above intuition. Our main result is the following.

Simulation Theorem (Theorem 2.2, informally)**.** *Many types of randomised protocols for $f \circ g^n$ can be simulated by a corresponding type of randomised decision tree for f.*

This result makes it easy to prove strong lower bounds for $f \circ g^n$ in all known one-sided (and some two-sided) *zero-communication* models. Here a zero-communication protocol is understood in the sense of [KLL+15] as a probability distribution over (labelled) rectangles $R = X \times Y$ (where $X \subseteq \mathcal{X}^n$ and $Y \subseteq \mathcal{Y}^n$) together with some acceptance criterion (and hence no communication is needed for Alice and Bob to select a rectangle, since it can be sampled with public randomness). Such models can be used to capture all known rectangle-based lower bound techniques used in communication complexity. This includes widely studied measures such as corruption [Yao83, BFS86, Raz92, Kla03, BPSW06, She12a, GW14], smooth rectangle bound [JK10, Kla10, CKW12, JY12, HJ13, KMSY14], relaxed partition bound [KLL+15], and extended discrepancy [Kla03, GL14]; see [JK10] for an extensive catalog. The Simulation Theorem applies to all these measures: it reduces the task of understanding a specific communication complexity measure of $f \circ g^n$ to the task of understanding a corresponding query complexity measure of f, which is typically a far easier task.

2.1.1 Main structural result: Junta Theorem

In order to motivate our approach (and to introduce notation), we start by reviewing some previous influential work in communication complexity.

Prior work: Approximation by polynomials. A long line of prior work has developed a framework of *polynomial approximation* to analyze the communication complexity of composed functions. Building on the work of Razborov [Raz03], a general framework was introduced by Sherstov [She09, She11a] (called the pattern matrix method) and independently by Shi and Zhu [SZ09] (called the block-composition method). See also the survey [She08]. Both methods have since been studied in the two-party setting [LZ10, RS10, She11b] and also the multiparty setting [LS09b, AC08, Cha08, She12b, She14b, RY15].

One way to phrase the approach taken in these works (a "primal" point of view championed in [She12b]) is as follows. Let Π be a randomised protocol and let $\mathrm{acc}_\Pi(x, y)$ denote the probability that Π accepts an input (x, y). For example, if Π computes a two-party function F with error at most $1/4$, then $\mathrm{acc}_\Pi(x, y) \in [3/4, 1]$ for every 1-input $(x, y) \in F^{-1}(1)$ and $\mathrm{acc}_\Pi(x, y) \in [0, 1/4]$ for every 0-input $(x, y) \in F^{-1}(0)$. When $F := f \circ g^n$ is a composed function, we can define $\mathrm{acc}_\Pi(z)$ for $z \in \operatorname{dom} f$ (domain of f) meaningfully as the probability that Π accepts a *random two-party encoding* of z. More specifically, letting $\mathbf{E}$ denote expectation and $\mathcal{U}_z$ the uniform distribution over $(g^n)^{-1}(z)$ we define

$$\mathrm{acc}_\Pi(z) \;\coloneqq\; \mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{U}_z} \mathrm{acc}_\Pi(\boldsymbol{x},\boldsymbol{y}).$$

The centerpiece in the framework is the following type of structure theorem: assuming g is chosen carefully, for any cost-c protocol Π there is a degree-$O(c)$ multivariate polynomial $p(z)$ such that $\mathrm{acc}_\Pi(z) \approx p(z)$. Here the approximation error is typically measured point-wise. Consequently, if f cannot be approximated point-wise with a low-degree polynomial, one obtains lower bounds against any bounded-error protocol computing $f \circ g^n$.

A technical convenience that will be useful for us is that since randomised protocols are essentially linear combinations of 0/1-labelled rectangles R, it suffices to study the acceptance probability of each individual rectangle R. More formally, it suffices to understand $\mathrm{acc}_R(z)$, defined as the probability that $(\boldsymbol{x},\boldsymbol{y}) \in R$ for a random encoding $(\boldsymbol{x},\boldsymbol{y}) \sim \mathcal{U}_z$ of z. Put succinctly,

$$\mathrm{acc}_R(z) \;\coloneqq\; \mathcal{U}_z(R).$$

An important feature of the polynomial framework is that it often yields tight lower bounds for *two-sided* (i.e., closed under complement) randomised models. However, polynomials are not always the most precise modeling choice when it comes to understanding *one-sided* (i.e., not closed under complement) randomised models, such as randomised generalizations of NP and measures like nonnegative rank.

This chapter: Approximation by conical juntas. In this chapter, we show that randomised protocols for composed functions can be simulated by *conical juntas*, a nonnegative analog of polynomials. Let $h\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}$ be a function. We say that h is a *d-junta* if it only depends on at most d of its input bits—we stress that all juntas in this thesis are nonnegative by definition. More generally, we call h a *conical d-junta* if it lies in the nonnegative cone generated by d-juntas, i.e., if we can write $h = \sum_i a_i h_i$ where $a_i \geq 0$ are nonnegative coefficients and h_i are d-juntas. Equivalently, a conical d-junta can be viewed as a nonnegative combination of width-d conjunctions (i.e., functions of the form $(\ell_1 \wedge \cdots \wedge \ell_w)$ where $w \leq d$ and each ℓ_i is an input variable or its negation). Note that a conical d-junta is, in particular, a polynomial of degree at most d.

For concreteness, we state and prove our results for logarithmic-size inner-product gadgets. That is, throughout this chapter, we restrict our attention to the following setting of parameters:

- The gadget is given by $g(x, y) \coloneqq \langle x, y\rangle \bmod 2$, where $x, y \in \{0,1\}^b$.
- The block length $b = b(n)$ satisfies $b(n) \geq 100 \log n$.

(†)

(However, our results hold more generally whenever g is a sufficiently strong two-source extractor; see Remark 1. Further, lower bounds for inner-product gadget as above can be used to get lower bounds for other gadgets with worse parameters. See Section 2.1.4 for more discussion.)

We are now ready to state our key structural result. The result essentially characterizes the computational power of a single rectangle in the communication matrix of $f \circ g^n$. Note that the theorem makes no reference to f.

Theorem 2.1 (Junta Theorem)**.** *Assume (†). For any $d \geq 0$ and any rectangle R in the domain of g^n there exists a conical d-junta h such that, for all $z \in \{0,1\}^n$,*

$$\mathrm{acc}_R(z) \;\in\; (1 \pm 2^{-\Theta(b)}) \cdot h(z) \;\pm\; 2^{-\Theta(db)}. \tag{2.1}$$

Discussion. Theorem 2.1 is similar in spirit to the approach taken by Chan et al. [CLRS13]. They gave a black-box method for converting Sherali–Adams lower bounds into size lower bounds for extended formulations. A key step in their proof is to approximate a single nonnegative rank-1 matrix with a single junta. In our approach, we approximate a single rectangle with a whole nonnegative combination of juntas. This allows us to achieve better error bounds that yield tight characterizations for many communication models (as discussed in Section 2.1.2 below). In the language of communication complexity, the lower bounds of [CLRS13] went up to about $\Omega(\log^2 n)$. See [CLRS13, §3.1] for more discussion.

The additive error $2^{-\Theta(db)}$ in Theorem 2.1 is essentially optimal, and the same additive error appears in the polynomial approximation framework. The multiplicative error $(1 \pm 2^{-\Theta(b)})$ is new: this is the cost we end up incurring for using juntas instead of polynomials. Such multiplicative error does not appear in the polynomial approximation framework. Whether one can achieve better multiplicative accuracy in Theorem 2.1 is left as an open problem (see Section 2.1.4).

Maybe the biggest drawback with Theorem 2.1 is that our proof assumes block length $b = \Omega(\log n)$ (cf. the pattern matrix method works even when $b = \Theta(1)$). Whether Theorem 2.1 (or some relaxed form of it) is true for $b = \Theta(1)$ is left as an open problem.

2.1.2 Communication versus query: Simulation Theorem

The most intuitive way to formalize our Simulation Theorem is in terms of different randomised models of computation rather than in terms of different lower bound measures. Indeed, we consider several models originally introduced in the context of Turing machine complexity theory: for any such model $\mathcal{C}$ one can often associate, in a canonical fashion, a communication model $\mathcal{C}^{\mathsf{cc}}$ and a decision tree model $\mathcal{C}^{\mathsf{dt}}$. We follow the convention of using names of models as

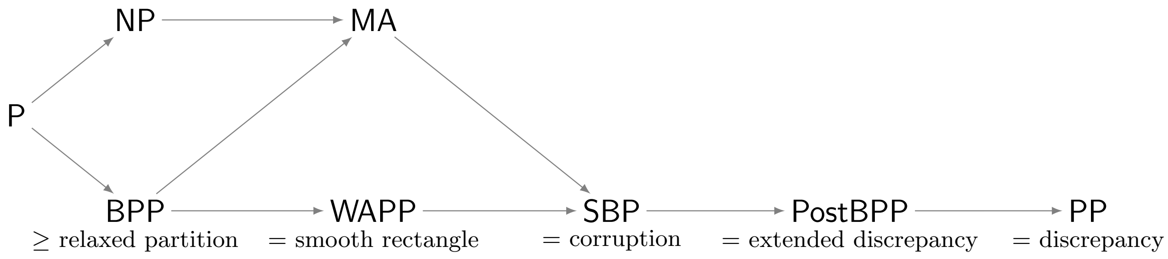


Figure 2.1: Models and lower bound methods at a glance. Arrows denote class inclusions.

complexity measures so that $\mathcal{C}^{\mathsf{cc}}(F)$ denotes the communication complexity of F in model $\mathcal{C}^{\mathsf{cc}}$, and $\mathcal{C}^{\mathsf{dt}}(f)$ denotes the query complexity of f in model $\mathcal{C}^{\mathsf{dt}}$. In this thesis, we further identify $\mathcal{C}^{\mathsf{cc}}$ with the class of partial functions F with $\mathcal{C}^{\mathsf{cc}}(F) \leq \mathrm{poly}(\log n)$. We stress that our complexity classes consist of partial functions (i.e., promise problems)—for total functions many surprising collapses are possible (e.g., $\mathsf{NP}^{\mathsf{cc}} \cap \mathsf{coNP}^{\mathsf{cc}} = \mathsf{P}^{\mathsf{cc}}$ for total functions [KN97, §2.3]).

Our methods allow us to accurately analyze the models listed below (see also Figure 2.1). Our discussion in this introduction is somewhat informal; see Section 2.3 for precise definitions.

- NP: *Nondeterminism.* We view an NP computation as a randomised computation where 1-inputs are accepted with non-zero probability and 0-inputs are accepted with zero probability. The communication analog $\mathsf{NP}^{\mathsf{cc}}$ was formalized in the work of Babai et al. [BFS86] that introduced communication complexity analogs of classical complexity classes.

- WAPP: *Weak Almost-Wide* PP [BGM06]. A WAPP computation is a randomised computation such that 1-inputs are accepted with probability in $[(1-\epsilon)\alpha, \alpha]$, and 0-inputs are accepted with probability in $[0, \epsilon\alpha]$ where $\alpha = \alpha(n) > 0$ is arbitrary and $\epsilon < 1/2$ is a constant. The communication analog $\mathsf{WAPP}^{\mathsf{cc}}$ is equivalent to the (one-sided) *smooth rectangle bound* of Jain and Klauck [JK10] and also to *approximate nonnegative rank* by a result of Kol et al. [KMSY14]. We also study a two-sided model $\mathsf{WAPP} \cap \mathsf{coWAPP}$ whose communication analog corresponds to the two-sided smooth rectangle bound, which was called the *relaxed partition bound* by [KLL$^+$15].

- SBP: *Small Bounded-Error Probability* [BGM06]. An SBP computation is a randomised computation such that 1-inputs are accepted with probability in $[\alpha, 1]$ and 0-inputs are accepted with probability in $[0, \alpha/2]$ where $\alpha = \alpha(n) > 0$ is arbitrary. The communication analog $\mathsf{SBP}^{\mathsf{cc}}$ is equivalent to the (one-sided) *corruption bound* originally defined in [Yao83] (see [GW14]).

- $\mathsf{PostBPP}$: *Postselected* BPP [Aar05]. (Equivalent to $\mathsf{BPP}_{\mathsf{path}}$ [HHT97].) A $\mathsf{PostBPP}$ computation is a randomised computation that may sometimes output $\perp$ (representing "abort" or "don't know"), but conditioned on not outputting $\perp$ the output is correct with probability at least 3/4. The communication analog $\mathsf{PostBPP}^{\mathsf{cc}}$ was first studied in [Kla03] (under the name “approximate majority covers”) and subsequently in [GL14] (under the generic name “zero-communication protocols”) where the term *extended discrepancy* was coined for the dual characterization of $\mathsf{PostBPP}^{\mathsf{cc}}$.

We apply the Junta Theorem to show that when $\mathcal{C}$ is one of the above models, any $\mathcal{C}^{\mathsf{cc}}$ protocol for $f \circ g^n$ can be converted into a corresponding $\mathcal{C}^{\mathsf{dt}}$ decision tree for f. Roughly speaking, this is because such a protocol can be formulated as a distribution over (labelled) rectangles, and each rectangle can be converted (via the Junta Theorem) into a distribution over conjunctions. Hence lower bounds on $\mathcal{C}^{\mathsf{cc}}(f \circ g^n)$ follow in a black-box way from lower bounds on $\mathcal{C}^{\mathsf{dt}}(f)$.

Theorem 2.2 (Simulation Theorem)**.** *Assume (†). For any partial $f\colon \{0,1\}^n \to \{0,1,*\}$,*

$$\begin{aligned} \mathcal{C}^{\mathsf{cc}}(f \circ g^n) &= \Theta(\mathcal{C}^{\mathsf{dt}}(f) \cdot b) \qquad \textit{for } \mathcal{C} \in \{\mathsf{NP}, \mathsf{WAPP}, \mathsf{SBP}\}, \\ \mathcal{C}^{\mathsf{cc}}(f \circ g^n) &\geq \Omega(\mathcal{C}^{\mathsf{dt}}(f) \cdot b) \qquad \textit{for } \mathcal{C} = \mathsf{PostBPP}. \end{aligned}$$

(Here we crucially ignore constant factors in the error parameter ϵ for $\mathcal{C} = \mathsf{WAPP}$.)

Naturally, the upper bounds in Theorem 2.2 follow from the fact that a communication protocol for $f \circ g^n$ can simulate the corresponding decision tree for f: when the decision tree queries the i-th input of f, the protocol spends $b+1$ bits of communication to figure out $z_i = g(x_i, y_i)$ in a brute-force manner. (There is one subtlety concerning the two-sided model $\mathsf{PostBPP}$; see Remark 3.)

We also mention that the result for the simplest model $\mathcal{C} = \mathsf{NP}$ does not require the full power of the Junta Theorem: It is possible to prove it using only a proper subset of the ideas that we present for the other randomised models. We will prove this special case as a warm-up for the Junta Theorem in Section 2.2.

2.1.3 Applications

Using the Simulation Theorem we can resolve several questions from prior work.

SBP and corruption. Our first application is the following.

Theorem 2.3. $\mathsf{SBP}^{\mathsf{cc}}$ *is not closed under intersection.*

We prove this theorem by first giving an analogous lower bound for query complexity: there exists a partial f such that $\mathsf{SBP}^{\mathsf{dt}}(f) \leq O(1)$, but $\mathsf{SBP}^{\mathsf{dt}}(f_\wedge) \geq n^{\Omega(1)}$, where $f_\wedge\colon \{0,1\}^{2n} \to \{0,1,*\}$ is defined by $f_\wedge(z,z') \coloneqq f(z) \wedge f(z')$. This query separation alone yields via standard diagonalization (e.g., [AW09, §5]) an oracle relative to which the classical complexity class SBP is not closed under intersection, solving an open problem posed by [BGM06]. Applying the Simulation Theorem to $f \circ g^n$ and $f_\wedge \circ g^{2n} = (f \circ g^n)_\wedge$ we then obtain Theorem 2.3.

Theorem 2.3 has consequences for Arthur–Merlin communication ($\mathsf{MA}^{\mathsf{cc}}$, $\mathsf{AM}^{\mathsf{cc}}$) which has been studied in [Kla03, RS04, AW09, GS10, Kla11, GR15, GPW16b]. Namely, Klauck [Kla03] asked (using the language of uniform threshold covers) whether the known inclusion $\mathsf{MA}^{\mathsf{cc}} \subseteq \mathsf{SBP}^{\mathsf{cc}}$ is strict. (This was also re-asked in [GW14].) Put differently, is corruption a complete lower bound method for $\mathsf{MA}^{\mathsf{cc}}$ up to polynomial factors? Since $\mathsf{MA}^{\mathsf{cc}}$ is closed under intersection, we conclude that the answer is “no”.

Corollary 2.4. $\mathsf{SBP}^{\mathsf{cc}} \not\subseteq \mathsf{MA}^{\mathsf{cc}}$.

Proving explicit lower bounds for $\mathsf{AM}^{\mathsf{cc}}$ remains one of the central challenges in communication complexity. Motivated by this [GPW16b] studied a certain unambiguous restriction of $\mathsf{AM}^{\mathsf{cc}}$, denoted $\mathsf{UAM}^{\mathsf{cc}}$, as a stepping stone towards $\mathsf{AM}^{\mathsf{cc}}$. They asked whether $\mathsf{UAM}^{\mathsf{cc}} \subseteq \mathsf{SBP}^{\mathsf{cc}}$. In other words, does corruption give lower bounds against $\mathsf{UAM}^{\mathsf{cc}}$ in a black-box fashion? They showed that the answer is “no” for query complexity. Using the Simulation Theorem it is now straightforward to convert this result into an analogous communication separation.

Corollary 2.5. $\mathsf{UAM}^{\mathsf{cc}} \not\subseteq \mathsf{SBP}^{\mathsf{cc}}$.

Intriguingly, we still lack $\mathsf{UAM}^{\mathsf{cc}}$ lower bounds for set-disjointness. Corollary 2.5 implies that such lower bounds cannot be blindly derived from Razborov’s corruption lemma [Raz92].

WAPP and nonnegative rank. Kol et al. [KMSY14] asked whether the error in the definition of WAPP can be efficiently *amplified*, i.e., whether the parameter ϵ can be *reduced* without blowing up the cost too much. It is known that such amplification is possible for the closely related two-sided model AWPP, *Almost-Wide* PP (related to smooth discrepancy and approximate rank), using “amplification polynomials”; see [Fen03, §3] (or [LS09a, §3.2] and [Alo03] for approximate rank). In [KMSY14] it was shown that no one-sided analog of amplification polynomials exists, ruling out one particular approach to amplification.

We show unconditionally that $\mathsf{WAPP}^{\mathsf{cc}}$ (and hence rank^+_ϵ, approximate nonnegative rank) does not admit efficient error amplification in the case of partial functions. For total functions, this at least shows that no “point-wise” method can be used to amplify ϵ, since such methods would also work for partial functions. We write $\mathsf{WAPP}^{\mathsf{cc}}_\epsilon$ for the measure corresponding to error ϵ.

Theorem 2.6. *For all constants $0 < \epsilon < \delta < 1/2$ there exists a two-party partial function F such that $\mathsf{WAPP}^{\mathsf{cc}}_\delta(F) \leq O(\log n)$ but $\mathsf{WAPP}^{\mathsf{cc}}_\epsilon(F) \geq \Omega(n)$.*

Corollary 2.7. *For all constants $0 < \epsilon < \delta < 1/2$ there exists a partial boolean matrix F such that $\mathrm{rank}^+_\delta(F) \leq n^{O(1)}$ but $\mathrm{rank}^+_\epsilon(F) \geq 2^{\Omega(n)}$.*

In order to conclude Corollary 2.7 from Theorem 2.6 we actually need a stronger equivalence of $\mathsf{WAPP}^{\mathsf{cc}}$ and approximate nonnegative rank than the one proved by Kol et al. [KMSY14]: they showed the equivalence for total functions while we need the equivalence for partial functions. The extension to partial functions is nontrivial, and is related to the issue of “unrestricted” vs. “restricted” models of communication.

Unrestricted vs. restricted models. So far we have discussed "restricted" communication models. We can also define their "unrestricted" counterparts in analogy to the well-studied pair of classes $\mathsf{PP}^{\mathsf{cc}}$ (a.k.a. discrepancy [Kla07, §8]) and $\mathsf{UPP}^{\mathsf{cc}}$ (a.k.a. sign-rank [PS86]). Recall that a PP computation is a randomised computation such that 1-inputs are accepted with probability in $[1/2+\alpha,1]$, and 0-inputs are accepted with probability in $[0,1/2-\alpha]$ where $\alpha=\alpha(n)>0$ is arbitrary. In the unrestricted model $\mathsf{UPP}^{\mathsf{cc}}$ the parameter $\alpha>0$ can be arbitrarily small (consequently, the model is defined using private randomness), whereas in the restricted model $\mathsf{PP}^{\mathsf{cc}}$ the cost of a protocol with parameter α is defined as the usual communication cost plus $\log(1/\alpha)$. It is known that $\mathsf{PP}^{\mathsf{cc}} \subsetneq \mathsf{UPP}^{\mathsf{cc}}$ where the separation is exponential [BVdW07].

One can analogously ask whether the unrestricted–restricted distinction is relevant for the models considered in this chapter. (The question was raised and left unresolved for SBP in [GW14].) In fact, the separation of [BVdW07] already witnesses $\mathsf{PostBPP}^{\mathsf{cc}} \subsetneq \mathsf{UPostBPP}^{\mathsf{cc}}$ where the latter is the unrestricted version of the former. By contrast, we prove that the distinction is immaterial for WAPP and SBP, even for partial functions: the unrestricted models $\mathsf{UWAPP}^{\mathsf{cc}}$ and $\mathsf{USBP}^{\mathsf{cc}}$ (see Section 2.3 for definitions) are essentially no more powerful than their restricted counterparts. Consequently, the Simulation Theorem can be applied to analyze these unrestricted models, too—but the equivalences are also interesting in their own right.

Theorem 2.8. $\mathsf{SBP}^{\mathsf{cc}}(F) \leq O(\mathsf{USBP}^{\mathsf{cc}}(F)+\log n)$ *for all* F.

Theorem 2.9. $\mathsf{WAPP}^{\mathsf{cc}}_{\delta}(F) \leq O(\mathsf{UWAPP}^{\mathsf{cc}}_{\epsilon}(F)+\log(n/(\delta-\epsilon)))$ *for all* F *and* $0<\epsilon<\delta<1/2$.

The seemingly more powerful models $\mathsf{USBP}^{\mathsf{cc}}$ and $\mathsf{UWAPP}^{\mathsf{cc}}$ admit characterizations in terms of the nonnegative rank of matrices: instead of rectangles, the protocols compute using nonnegative rank-1 matrices. In particular, $\mathsf{UWAPP}^{\mathsf{cc}}$ turns out to capture $\mathrm{rank}^{+}_{\epsilon}$; it is Theorem 2.9 that will be used in the proof of Corollary 2.7 above.

2.1.4 Open problems

Our main open question is whether Theorem 2.1 continues to hold for $b=O(1)$. If true, such a result would be very useful as inner-product on b bits can be simulated by most other gadgets on blocks of length roughly 2^b (which would be $O(1)$ again). This in turn would give new and more unified proofs of important communication complexity lower bounds such as Razborov's corruption lower bound for set-disjointness [Raz92] and the lower bound for gap-hamming [CR12, She12a, Vid13]. A first hurdle in understanding the case $b=O(1)$ seems to be Lemma 2.13—does some version of it hold for $b=O(1)$? In particular, using notions from Section 2.2.2, we can ask the following concrete question as a starting point: For b a sufficiently big constant, g the inner-product gadget, and two independent 0.9-dense sources $\boldsymbol{X},\boldsymbol{Y}$ over $(\{0,1\}^b)^n$, does $g^n(\boldsymbol{X},\boldsymbol{Y})$ have full support over $\{0,1\}^n$?

The following are some other relevant open problems.

- Can the multiplicative accuracy in Theorem 2.1 be improved? This issue seems to be what is preventing us from quantitatively improving on the lower bounds obtained by [CLRS13] for the LP extension complexity of approximating Max-Cut.
- Raz and McKenzie [RM99] (see also Chapter 4) obtained a simulation theorem that converts deterministic communication protocols for $f \circ g^n$ into deterministic decision trees for f, where g a certain polynomial-size gadget. Can our methods be used to simplify their proof, or to extend their result to other g's?
- Our focus in this chapter has been on partial functions. It remains open whether $\mathsf{SBP}^{\mathsf{cc}} = \mathsf{MA}^{\mathsf{cc}}$ for total functions, or whether efficient error amplification exists for $\mathsf{WAPP}^{\mathsf{cc}}$ for total functions.

2.1.5 Notational conventions

We always write random variables in bold (e.g., $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$). Capital letters X, Y are reserved for subsets of inputs to $G = g^n$ (so all rectangles R are of the form $X \times Y$). We identify such sets with flat distributions: we denote by $\boldsymbol{X}$ the random variable that is uniformly distributed on X. Given a distribution $\mathcal{D}$ and an event E we denote by $(\mathcal{D} \mid E)$ the conditional distribution of $\mathcal{D}$ given E, specifically, $(\mathcal{D} \mid E)(\,\cdot\,) := \mathcal{D}(\,\cdot \cap E)/\mathcal{D}(E)$. We also use the shorthand $\mathcal{D}(\,\cdot \mid E) := (\mathcal{D} \mid E)(\,\cdot\,)$.

2.2 Proof of the Junta Theorem

In this section we prove Theorem 2.1, restated here for convenience.

Theorem 2.1 (Junta Theorem)**.** *Assume (†). For any $d \geq 0$ and any rectangle R in the domain of g^n there exists a conical d-junta h such that, for all $z \in \{0,1\}^n$,*

$$\operatorname{acc}_R(z) \;\in\; (1 \pm 2^{-\Theta(b)}) \cdot h(z) \;\pm\; 2^{-\Theta(db)}. \tag{2.1}$$

2.2.1 Proof overview

We write $G := g^n$ for short. Fix $d \geq 0$ and a rectangle $L \subseteq \operatorname{dom} G$. Our goal is to approximate $\operatorname{acc}_L(z)$ by some conical d-junta $h(z)$. (We are going to use the symbol L for the "main" rectangle so as to keep the symbol R free for later use as a more generic rectangle.) The high-level idea in our proof is extremely direct: to find a suitable h we partition—or at least almost partition—the rectangle L into subrectangles $R \subseteq L$ that behave like width-d conjunctions.

Definition 2.10 (Conjunction rectangles)**.** A rectangle R is a (d, ϵ)-*conjunction* if there exists a width-d conjunction $h_R\colon \{0,1\}^n \to \{0,1\}$ (i.e., h_R can be written as $(\ell_1 \wedge \cdots \wedge \ell_w)$ where $w \leq d$ and each ℓ_i is an input variable or its negation) such that $\operatorname{acc}_R(z) \in (1 \pm \epsilon) \cdot a_R h_R(z)$ for some $a_R \geq 0$ and all $z \in \{0,1\}^n$.

The proof is split into three subsections.

(§ 2.2.2) **Block-wise density:** We start by discussing a key property that is a sufficient condition for a subrectangle $R \subseteq L$ to be a conjunction rectangle.

(§ 2.2.3) **Warm-up: Nondeterministic simulation:** Before proceeding with the full proof of the Junta Theorem, we first demonstrate the usefulness of block-wise density by giving a direct proof of a special case: the simulation theorem for NP. (This special case is all that is needed in Chapter 3.)

(§ 2.2.4) **Reduction to a packing problem:** Continuing with the full proof, instead of partitioning L into conjunctions, we show that it suffices to find a packing (disjoint collection) of conjunction subrectangles of L that cover most of L relative to a given distribution over inputs. This will formalize our main technical task: solving a type of packing-with-conjunctions problem.

(§ 2.2.5) **Solving the packing problem:** This is the technical heart of the proof: we describe an algorithm to find a good packing for L.

2.2.2 Block-wise density

In this subsection we introduce a central notion that will allow us to extract close to uniform output from sufficiently random inputs to $G = g^n \colon \{0,1\}^{bn} \times \{0,1\}^{bn} \to \{0,1\}^n$. Recall that in the setting of two-source extractors (e.g., [Vad12]), one considers a pair of independent random inputs $\boldsymbol{x}$ and $\boldsymbol{y}$ that have high *min-entropy*, defined by $\mathbf{H}_\infty(\boldsymbol{x}) \coloneqq \min_x \log(1/\mathbf{Pr}[\,\boldsymbol{x} = x\,])$. In our setting we think of $G = g^n$ as a *local* two-source extractor: each of the n output bits depends only on few of the input bits. Hence we need a stronger property than high min-entropy on $\boldsymbol{x}$ and $\boldsymbol{y}$ to guarantee that $\boldsymbol{z} \coloneqq G(\boldsymbol{x},\boldsymbol{y})$ will be close to uniform. This property we call *block-wise density*. Below, for $I \subseteq [n]$, we write $\boldsymbol{x}_I$ for the restriction of $\boldsymbol{x}$ to the *blocks* determined by I.

Definition 2.11 (Block-wise density)**.** A random variable $\boldsymbol{x} \in \{0,1\}^{bn}$ is δ*-dense* if for all $I \subseteq [n]$ the blocks $\boldsymbol{x}_I$ have *min-entropy rate* at least δ, that is, $\mathbf{H}_\infty(\boldsymbol{x}_I) \geq \delta b|I|$.

Definition 2.12 (Multiplicative uniformity)**.** A distribution $\mathcal{D}$ on $\{0,1\}^m$ is ϵ*-uniform* if $\mathcal{D}(z) \in (1 \pm \epsilon) \cdot 2^{-m}$ for all outcomes z.

Lemma 2.13. *Assume* (†)*. If $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent and* 0.6*-dense, then $G(\boldsymbol{x},\boldsymbol{y})$ is $2^{-b/20}$-uniform.*

Proof. Let $\boldsymbol{z} \coloneqq G(\boldsymbol{x},\boldsymbol{y})$. First observe that for any $I \subseteq [n]$ the parity of the output bits $\boldsymbol{z}_I$ is simply $\langle \boldsymbol{x}_I, \boldsymbol{y}_I \rangle \bmod 2$. We use the fact that inner-product is a good two-source extractor to argue that this parity is close to an unbiased random bit. Indeed, by 0.6-density we have $\mathbf{H}_\infty(\boldsymbol{x}_I) + \mathbf{H}_\infty(\boldsymbol{y}_I) \geq 1.2 \cdot b|I|$ and this implies by a basic theorem of Chor and Goldreich [CG88, Theorem 9] that for $I \neq \emptyset$,

$$\big|\, \mathbf{Pr}[\, \langle \boldsymbol{x}_I, \boldsymbol{y}_I \rangle \bmod 2 \,=0\,] - 1/2 \,\big| \;\leq\; 2^{-0.1 \cdot b|I|+1}. \tag{2.2}$$

This bound is enough to yield ϵ-uniformity for $\epsilon := 2^{-b/20}$, as we next verify using standard Fourier analysis (see, e.g., [O'D14]).[1] Let $\mathcal{D}$ be the distribution of $\boldsymbol{z}$. We think of $\mathcal{D}$ as a function $\{0,1\}^n \to [0,1]$ and write it in the Fourier basis as

$$\mathcal{D}(z) \;=\; \sum_{I\subseteq[n]} \widehat{\mathcal{D}}(I)\chi_I(z)$$

where $\chi_I(z) := (-1)^{\sum_{i\in I} z_i}$ and $\widehat{\mathcal{D}}(I) := 2^{-n}\sum_z \mathcal{D}(z)\chi_I(z) = 2^{-n}\cdot \mathbf{E}_{\boldsymbol{z}\sim\mathcal{D}}[\,\chi_I(\boldsymbol{z})\,]$. Note that $\widehat{\mathcal{D}}(\emptyset) = 2^{-n}$ because $\mathcal{D}$ is a distribution. In this language, property (2.2) says that, for all $I \neq \emptyset$, $2^n \cdot |\widehat{\mathcal{D}}(I)| = |\,\mathbf{E}[\,(-1)^{\langle \boldsymbol{x}_I, \boldsymbol{y}_I\rangle}\,]| \leq 2^{-0.1\cdot b|I|+2}$, which is at most $\epsilon 2^{-2|I|\log n}$ by our definition of b and ϵ. Hence,

$$2^n \sum_{I\neq\emptyset} |\widehat{\mathcal{D}}(I)| \;\leq\; \epsilon \sum_{I\neq\emptyset} 2^{-2|I|\log n} \;=\; \epsilon \sum_{k=1}^{n} \binom{n}{k} 2^{-2k\log n} \;\leq\; \epsilon \sum_{k=1}^{n} 2^{-k\log n} \;\leq\; \epsilon.$$

We use this to show that $|\mathcal{D}(z) - 2^{-n}| \leq \epsilon 2^{-n}$ for all $z \in \{0,1\}^n$, which proves the lemma. To this end, let $\mathcal{U}$ denote the uniform distribution (note that $\widehat{\mathcal{U}}(I) = 0$ for all $I \neq \emptyset$) and let $\mathbb{1}_z$ denote the indicator for z defined by $\mathbb{1}_z(z) = 1$ and $\mathbb{1}_z(z') = 0$ for $z' \neq z$ (note that $|\widehat{\mathbb{1}}_z(I)| = 2^{-n}$ for all I). We can now calculate

$$\begin{aligned} |\mathcal{D}(z) - 2^{-n}| \;&=\; |\langle \mathbb{1}_z, \mathcal{D}\rangle - \langle \mathbb{1}_z, \mathcal{U}\rangle| \;=\; |\langle \mathbb{1}_z, \mathcal{D} - \mathcal{U}\rangle| \;=\; 2^n \cdot |\langle \widehat{\mathbb{1}}_z, \widehat{\mathcal{D}} - \widehat{\mathcal{U}}\rangle| \\ &\leq\; 2^n \cdot \textstyle\sum_{I\neq\emptyset} |\widehat{\mathbb{1}}_z(I)| \cdot |\widehat{\mathcal{D}}(I)| \;=\; \textstyle\sum_{I\neq\emptyset} |\widehat{\mathcal{D}}(I)| \;\leq\; \epsilon 2^{-n}. \quad \square \end{aligned}$$

Remark 1. The only properties of inner-product we needed in the above proof were that it is a strong two-source extractor and that it satisfies an XOR-lemma. However, all sufficiently strong two-source extractors have the latter property automatically [Sha03], so we could have fixed g to be any such extractor in Theorem 2.1. It is known [LSŠ08] that an XOR-lemma holds even under the weaker assumption of g having low discrepancy (not necessarily under the uniform distribution over $\operatorname{dom} g$). Hence it is plausible that Theorem 2.1 could be extended to handle such g, as well.

We have the following corollary; here we write $\bar{I} := [n] \smallsetminus I$ for short.

Corollary 2.14. *Assume* (†). *Let $R = X \times Y$ and suppose there is an $I \subseteq [n]$ such that $\boldsymbol{X}_I$ and $\boldsymbol{Y}_I$ are fixed while $\boldsymbol{X}_{\bar{I}}$ and $\boldsymbol{Y}_{\bar{I}}$ are* 0.6*-dense. Then R is an* $(|I|, O(2^{-b/20}))$*-conjunction.*

Proof. Let $\boldsymbol{z} := G(\boldsymbol{X}, \boldsymbol{Y})$ and note that $\boldsymbol{z}_I$ is fixed. Write $\epsilon := 2^{-b/20}$ for short. Applying Lemma 2.13 to $\boldsymbol{x} = \boldsymbol{X}_{\bar{I}}$ and $\boldsymbol{y} = \boldsymbol{Y}_{\bar{I}}$ ($\boldsymbol{x}$ and $\boldsymbol{y}$ are 0.6-dense) shows that $|G^{-1}(z) \cap R|/|R| \in (1 \pm \epsilon)\cdot 2^{-|\bar{I}|}$ whenever $z_I = \boldsymbol{z}_I$ (and 0 otherwise). If g were perfectly balanced, then we would have $|G^{-1}(z)|/2^{2bn} = 2^{-n}$ for all $z \in \{0,1\}^n$; instead, since g is only approximately balanced

[1] This fact resembles the classic "Vazirani XOR-Lemma" [Vaz86], except that the latter only guarantees the distribution is close to uniform in statistical distance, and it assumes a single bound on the bias of all parities (whereas we assume a bound that depends on the size of the parity).

($|g^{-1}(1)|, |g^{-1}(0)| \in 2^{2b-1} \pm 2^{b-1}$), it can be seen by direct calculation that $|G^{-1}(z)|/2^{2bn} \in (1 \pm \epsilon) \cdot 2^{-n}$ for all $z \in \{0,1\}^n$ (though this can also be seen by another application of Lemma 2.13—to uniform $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^{bn}$, which are 1-dense). Therefore $\mathrm{acc}_R(z) = |G^{-1}(z) \cap R|/|G^{-1}(z)| \in (1 \pm O(\epsilon)) \cdot 2^{|I|-2bn}|R|$ if $z_I = \boldsymbol{z}_I$ and $\mathrm{acc}_R(z) = 0$ if $z_I \neq \boldsymbol{z}_I$. This is of the form $(1 \pm O(\epsilon)) \cdot a_R h_R(z)$ (where $h_R(z) = 1$ iff $z_I = \boldsymbol{z}_I$), as required. □

2.2.3 Warm-up: Nondeterministic simulation

For expository purposes, we use the concept of block-wise density to give a direct proof of Theorem 3.4 for the case of $\mathcal{C} = \mathsf{NP}$, namely:

Theorem 2.15. *Assume (†). For all $f\colon \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{NP}^{\mathsf{cc}}(f \circ G) \;\geq\; \Omega(\mathsf{NP}^{\mathsf{dt}}(f) \cdot b). \tag{2.3}$$

In words, we prove that any covering Π of the 1-inputs of $f \circ G$ using 2^d rectangles implies an $O(d/b)$-width DNF representation for f. Our goal will be to argue that for every $z \in f^{-1}(1)$ there is a small-width conjunction that certifies "$f(z) = 1$". More precisely, we are looking for a conjunction h such that *(i)* h has width $O(d/b)$, *(ii)* h accepts z, and *(iii)* h only accepts 1-inputs of f. To find such a conjunction, we find a subrectangle $R \subseteq L$ of some $L \in \Pi$ such that $G(R)$ equals the set of inputs accepted by some h satisfying the properties prescribed above. More precisely, *(i)* $G(R)$ is a subcube of codimension $O(d/b)$, that is, $G(R)$ is fixed on $O(d/b)$ many coordinates and has full support elsewhere, in fact, we find an R that is an $(O(d/b), 1/2)$-conjunction rectangle, *(ii)* $z \in G(R)$, and *(iii)* $G(R) \subseteq f^{-1}(1)$. Note that *any* $R \subseteq L \in \Pi$ contains only 1-inputs of $f \circ G$ so that $G(R)$ contains only 1-inputs of f, that is, property *(iii)* will hold automatically.

Finding $\boldsymbol{R}$. Fix $z \in f^{-1}(1)$ and let $\boldsymbol{xy}$ be uniformly distributed on $G^{-1}(z)$. Using the fact that Π covers the whole support of $\boldsymbol{xy}$, we can find some $L \in \Pi$ such that $\mathbf{Pr}[\,\boldsymbol{xy} \in L\,] \geq 2^{-d}$. Denote by $(\boldsymbol{xy} \mid \boldsymbol{xy} \in L)$ the variables $\boldsymbol{xy}$ conditioned on the event "$\boldsymbol{xy} \in L$".

Let $I \subseteq [n]$ be a maximum-size subset for which 0.8-density is violated for $(\boldsymbol{xy} \mid \boldsymbol{xy} \in L)$, and let α be an outcome witnessing this: $\mathbf{Pr}[\,\boldsymbol{x}_I\boldsymbol{y}_I = \alpha \mid \boldsymbol{xy} \in L\,] > 2^{-1.6b|I|}$. We claim that conditioning further on the event "$\boldsymbol{x}_I\boldsymbol{y}_I = \alpha$", the remaining blocks indexed by $\bar{I} := [n] \smallsetminus I$ are 0.8-dense. Write $\boldsymbol{x}'\boldsymbol{y}' := (\boldsymbol{xy} \mid \boldsymbol{xy} \in L,\; \boldsymbol{x}_I\boldsymbol{y}_I = \alpha)$ for short.

Claim 2.16. *$\boldsymbol{x}'_{\bar{I}}\boldsymbol{y}'_{\bar{I}}$ is 0.8-dense.*

Proof. Suppose not: there is some nonempty set $J \subseteq \bar{I}$ and a string β such that $\mathbf{Pr}[\,\boldsymbol{x}_J\boldsymbol{y}_J = \beta \mid \boldsymbol{xy} \in L,\; \boldsymbol{x}_I\boldsymbol{y}_I = \alpha\,] > 2^{-1.6b|J|}$. Now $\mathbf{Pr}[\,\boldsymbol{x}_I\boldsymbol{y}_I = \alpha \text{ and } \boldsymbol{x}_J\boldsymbol{y}_J = \beta \mid \boldsymbol{xy} \in L\,] = \mathbf{Pr}[\,\boldsymbol{x}_I\boldsymbol{y}_I = \alpha \mid \boldsymbol{xy} \in L\,] \cdot \mathbf{Pr}[\,\boldsymbol{x}_J\boldsymbol{y}_J = \beta \mid \boldsymbol{xy} \in L,\; \boldsymbol{x}_I\boldsymbol{y}_I = \alpha\,] > 2^{-1.6b|I|} \cdot 2^{-1.6b|J|} = 2^{-1.6b|I \cup J|}$. But this means that $(\boldsymbol{xy} \mid \boldsymbol{xy} \in L)$ violates 0.8-density on $I \cup J$, which contradicts the maximality of I. □

Claim 2.17. $|I| \leq O(d/b)$.

Proof. Our gadget is almost balanced: $|g^{-1}(1)|, |g^{-1}(0)| \geq 2^{2b}/4 = 2^{2(b-1)}$. It follows that $\mathbf{H}_\infty(\boldsymbol{x}_I \boldsymbol{y}_I) \geq 2(b-1)|I|$ for all $I \subseteq [n]$. On the one hand, $\mathbf{H}_\infty(\boldsymbol{x}_I \boldsymbol{y}_I \mid \boldsymbol{xy} \in L) \geq \mathbf{H}_\infty(\boldsymbol{x}_I \boldsymbol{y}_I) - \log(1/\mathbf{Pr}[\boldsymbol{xy} \in R]) \geq 2(b-1)|I| - d$, where we used the fact that conditioning on an event with probability p lowers the min-entropy by at most $\log(1/p)$. On the other hand, $\mathbf{H}_\infty(\boldsymbol{x}_I \boldsymbol{y}_I \mid \boldsymbol{xy} \in L) < 1.6b|I|$ as $(\boldsymbol{xy} \mid \boldsymbol{xy} \in L)$ violates 0.8-density on I. These two bounds imply the claim. □

In summary, $\boldsymbol{x}'\boldsymbol{y}'$ is fixed on $O(d/b)$ many blocks I, and 0.8-dense on the remaining blocks $\bar{I}$. To apply Lemma 2.13 we need a pair of independent random variables, and currently $\boldsymbol{x}'$ and $\boldsymbol{y}'$ are highly correlated (e.g., $G(\boldsymbol{x}', \boldsymbol{y}') = z$). Let $\boldsymbol{x}''$ and $\boldsymbol{y}''$ be independent copies of $\boldsymbol{x}'$ and $\boldsymbol{y}'$.

Claim 2.18. *$\boldsymbol{x}''_{\bar{I}}\boldsymbol{y}''_{\bar{I}}$ is 0.6-dense.*

Proof. Let $J \subseteq \bar{I}$. We want to show $\mathbf{H}_\infty(\boldsymbol{x}''_J \boldsymbol{y}''_J) \geq 1.2b|J|$. We calculate $\mathbf{H}_\infty(\boldsymbol{x}'_J) \geq \mathbf{H}_\infty(\boldsymbol{x}'_J \boldsymbol{y}'_J) - b|J| \geq 1.6b|J| - b|J| = 0.6b|J|$, where the first inequality follows since $b|J|$ is an upper bound on the support size of $\boldsymbol{y}'_J$ (measured in bits) and the second inequality uses the 0.8-density of $\boldsymbol{x}'_{\bar{I}}\boldsymbol{y}'_{\bar{I}}$. The same bound holds for $\mathbf{H}_\infty(\boldsymbol{y}'_J)$. Hence $\mathbf{H}_\infty(\boldsymbol{x}''_J \boldsymbol{y}''_J) = \mathbf{H}_\infty(\boldsymbol{x}''_J) + \mathbf{H}_\infty(\boldsymbol{y}''_J) = \mathbf{H}_\infty(\boldsymbol{x}'_J) + \mathbf{H}_\infty(\boldsymbol{y}'_J) \geq 0.6b|J| + 0.6b|J| = 1.2b|J|$. □

The subrectangle $R \subseteq L$ we are looking for is now defined as the support of $\boldsymbol{x}''\boldsymbol{y}''$. We can apply Lemma 2.13 to $\boldsymbol{x}''_{\bar{I}}\boldsymbol{y}''_{\bar{I}}$ to deduce that $G(\boldsymbol{x}'', \boldsymbol{y}'') \in G(R)$ has full support on $\bar{I}$ (in fact, lemma states that $G(\boldsymbol{x}'', \boldsymbol{y}'')$ is even multiplicatively uniform on this support). Moreover, by construction, $G(R)$ is fixed on I and $z \in G(R)$. This concludes the proof of Theorem 2.15.

2.2.4 Reduction to a packing problem

Let us continue with the proof of the full Junta Theorem. The purpose of this subsection is to massage the statement of the Junta Theorem into an alternative form in order to uncover its main technical content. We will end up with a certain type of packing problem, formalized in Theorem 2.20 at the end of this subsection.

Fix some "multiplicative" error bound $\epsilon := 2^{-\Theta(b)}$ for the purposes of the following discussion. Whenever $\mathscr{C}$ is a packing (disjoint collection) of (d, ϵ)-conjunction subrectangles of L we let

$$h_{\mathscr{C}} \ := \ \sum_{R \in \mathscr{C}} a_R h_R.$$

Write $\cup\mathscr{C} := \cup_{R\in\mathscr{C}} R$ for short. Then $\mathrm{acc}_{\cup\mathscr{C}} := \sum_{R\in\mathscr{C}} \mathrm{acc}_R$ is multiplicatively approximated by the conical d-junta $h_{\mathscr{C}}$ in the sense that $\mathrm{acc}_{\cup\mathscr{C}}(z) \in (1 \pm \epsilon) \cdot h_{\mathscr{C}}(z)$. Hence if we could find a $\mathscr{C}$ that partitioned $L = \cup\mathscr{C}$, we would have proved the theorem—without incurring any additive error.

Unfortunately, there are a few obstacles standing in the way of finding a perfect partition $\mathscr{C}$. One unavoidable issue is that we cannot multiplicatively approximate a tiny rectangle L with a low-degree conical junta. This is why we allow a small additive error and only multiplicatively approximate the acceptance probabilities of those z that have large enough $\operatorname{acc}_L(z)$. Indeed, we set

$$Z \;\coloneqq\; \{\, z \in \{0,1\}^n : \operatorname{acc}_L(z) \geq 2^{-db/20} \,\},$$

and look for a $\mathscr{C}$ that covers most of each of the sets $G^{-1}(z) \cap L$ for $z \in Z$. More precisely, suppose for a moment that we had a packing $\mathscr{C}$ such that for each $z \in Z$,

$$\mathcal{U}_z(\cup\mathscr{C} \mid L) \;\geq\; 1 - \epsilon, \tag{2.4}$$

where $\mathcal{U}_z(\cup\mathscr{C} \mid L) = \operatorname{acc}_{\cup\mathscr{C}}(z)/\operatorname{acc}_L(z)$ by definition. Indeed, assuming (2.4) we claim that

$$(1-\epsilon)\cdot h_{\mathscr{C}}(z) \;\leq\; \operatorname{acc}_L(z) \;\leq\; (1+O(\epsilon))\cdot h_{\mathscr{C}}(z) + 2^{-\Theta(db)}. \tag{2.5}$$

In particular, $h_{\mathscr{C}}$ achieves the desired approximation (2.1). For the first inequality, since $\cup\mathscr{C} \subseteq L$ we never multiplicatively overestimate acc_L, that is, we have $\operatorname{acc}_L \geq \operatorname{acc}_{\cup\mathscr{C}} \geq (1-\epsilon)\cdot h_{\mathscr{C}}$. For the second inequality, for $z \in Z$ we have $\operatorname{acc}_L(z) \leq (1-\epsilon)^{-1}\cdot\operatorname{acc}_{\cup\mathscr{C}}(z) \leq (1-\epsilon)^{-1}\cdot(1+\epsilon)\cdot h_{\mathscr{C}}(z) \leq (1+O(\epsilon))\cdot h_{\mathscr{C}}(z)$, and for $z \notin Z$ we have simply $\operatorname{acc}_L(z) < 2^{-\Theta(db)}$ by the definition of Z.

Unfortunately, we do not know how to construct a packing $\mathscr{C}$ satisfying (2.4) either. Instead, we show how to find a *randomised* packing $\boldsymbol{\mathscr{C}}$ that guarantees (2.4) *in expectation*. More precisely, our construction goes through the following primal/dual pair of statements that are equivalent by the minimax theorem.

Primal:	$\exists$ distribution $\mathcal{C}$ over $\mathscr{C}$'s	$\forall\, z \in Z$	$\mathbf{E}_{\boldsymbol{\mathscr{C}}\sim\mathcal{C}}\,\mathcal{U}_z(\cup\boldsymbol{\mathscr{C}} \mid L) \geq 1-\epsilon$
Dual:	$\forall$ distribution μ over Z	$\exists\,\mathscr{C}$	$\mathbf{E}_{\boldsymbol{z}\sim\mu}\,\mathcal{U}_{\boldsymbol{z}}(\cup\mathscr{C} \mid L) \;\geq 1-\epsilon$

Suppose the primal statement holds for some $\mathcal{C}$. Then we claim that the convex combination $h \coloneqq \mathbf{E}_{\boldsymbol{\mathscr{C}}\sim\mathcal{C}}\, h_{\boldsymbol{\mathscr{C}}}$ achieves the desired approximation. The right side of (2.5) can be reformulated as

$$h_{\boldsymbol{\mathscr{C}}}(z) \;\geq\; (1 - O(\epsilon + \boldsymbol{\epsilon}_z))\cdot(\operatorname{acc}_L(z) - 2^{-\Theta(db)}) \tag{2.6}$$

where $\boldsymbol{\epsilon}_z \coloneqq 1 - \mathcal{U}_z(\cup\boldsymbol{\mathscr{C}} \mid L)$ is a random variable depending on $\boldsymbol{\mathscr{C}}$ (so $\mathbf{E}_{\boldsymbol{\mathscr{C}}\sim\mathcal{C}}[\,\boldsymbol{\epsilon}_z\,] \leq \epsilon$). Applying linearity of expectation to (2.6) shows (along with the left side of (2.5)) that h satisfies (2.1).

Therefore, to prove Theorem 2.1 it remains to prove the dual statement. This will preoccupy us for the whole of Section 2.2.5 where, for convenience, we will prove a slightly more general claim formalized below.

Definition 2.19 (Lifted distributions)**.** A distribution $\mathcal{D}$ on the domain of G is said to be a *lift* of a distribution μ on the codomain of G if $\mathcal{D}(x,y) = \mu(z)/|G^{-1}(z)|$ where $z \coloneqq G(x,y)$. Note that a lifted distribution is a convex combination of distributions of the form $\mathcal{U}_z$.

Theorem 2.20 (Packing with conjunctions)**.** *Assume* (†)*. Let $d \geq 0$ and let L be a rectangle. There is an $\epsilon \coloneqq 2^{-\Theta(b)}$ such that for any lifted distribution $\mathcal{D}$ with $\mathcal{D}(L) \geq 2^{-db/20}$ there exists a packing $\mathscr{C}$ consisting of (d,ϵ)-conjunction subrectangles of L such that $\mathcal{D}(\cup\mathscr{C} \mid L) \geq 1-\epsilon$.*

The dual statement can be derived from Theorem 2.20 as follows. We need to check that for any distribution μ on Z there is some lifted distribution $\mathcal{D}$ such that $\mathcal{D}(L) \geq 2^{-db/20}$ and $\mathcal{D}(\,\cdot \mid L) = \mathcal{E}(\,\cdot\,)$ where $\mathcal{E}(\,\cdot\,) \coloneqq \mathbf{E}_{\boldsymbol{z}\sim\mu}\mathcal{U}_{\boldsymbol{z}}(\,\cdot \mid L)$ is the probability measure relevant to the dual statement. For intuition, a seemingly natural candidate would be to choose $\mathcal{D} = \mathbf{E}_{\boldsymbol{z}\sim\mu}\mathcal{U}_{\boldsymbol{z}}$; however, this does not ensure $\mathcal{D}(\,\cdot \mid L) = \mathcal{E}(\,\cdot\,)$ as conditioning on L may not commute with taking convex combinations of $\mathcal{U}_z$'s. This is why we instead define a slightly different distribution $\mu'(z) \coloneqq \gamma\mu(z)/\mathcal{U}_z(L)$ where $\gamma \coloneqq (\mathbf{E}_{\boldsymbol{z}\sim\mu}\, 1/\mathcal{U}_{\boldsymbol{z}}(L))^{-1}$ is a normalizing constant. If we now choose $\mathcal{D} \coloneqq \mathbf{E}_{\boldsymbol{z}\sim\mu'}\,\mathcal{U}_{\boldsymbol{z}}$ the conditioning on L works out. Indeed, noting that $\gamma = \mathcal{D}(L)$ we have $\mathcal{D}(\,\cdot \mid L) = \mathcal{D}(L)^{-1}\mathcal{D}(\,\cdot \cap L) = \gamma^{-1}\sum_z \mu'(z)\mathcal{U}_z(\,\cdot \cap L) = \sum_z \mu(z)\mathcal{U}_z(\,\cdot \cap L)/\mathcal{U}_z(L) = \mathbf{E}_{\boldsymbol{z}\sim\mu}\mathcal{U}_{\boldsymbol{z}}(\,\cdot \mid L) = \mathcal{E}(\,\cdot\,)$, as desired. Also note that $\mathcal{D}(L) = \mathbf{E}_{\boldsymbol{z}\sim\mu'}\,\mathcal{U}_{\boldsymbol{z}}(L) \geq \mathbf{E}_{\boldsymbol{z}\sim\mu'}\, 2^{-db/20} = 2^{-db/20}$ since μ' is supported on Z.

2.2.5 Solving the packing problem

In this section we prove Theorem 2.20. Fix an error parameter $\epsilon \coloneqq 2^{-b/100}$.

Notation. In the course of the argument, for any rectangle $R = X \times Y$, we are going to associate a bipartition of $[n]$ into *free* blocks, denoted $\operatorname{free} R$, and *fixed* blocks, denoted $\operatorname{fix} R \coloneqq [n] \smallsetminus \operatorname{free} R$. We will always ensure that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are fixed on the blocks in $\operatorname{fix} R$. However, if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are fixed on some block i, we may or may not put i into $\operatorname{fix} R$; thus the sets $\operatorname{fix} R$ and $\operatorname{free} R$ are not predefined functions of R, but rather will be chosen during the proof of Theorem 2.20. We say that *the free marginals of R are $(\delta,\mathcal{D})$-dense* if for $\boldsymbol{xy} \sim (\mathcal{D} \mid R)$ we have that $\boldsymbol{x}_{\operatorname{free} R}$ and $\boldsymbol{y}_{\operatorname{free} R}$ are δ-dense. Note that if $\mathcal{D} = \mathcal{U}$ is the uniform distribution, then the definition states that $\boldsymbol{X}_{\operatorname{free} R}$ and $\boldsymbol{Y}_{\operatorname{free} R}$ are δ-dense. The following is a rephrasing of Corollary 2.14.

Proposition 2.21. *If the free marginals of R are $(0.6,\mathcal{U})$-dense then R is a $(|\operatorname{fix} R|,\epsilon)$-conjunction.* □

We also use the following notation: if C is a *condition* (e.g., of the form $(x_I = \alpha)$ or $(x_I \neq \alpha)$) we write X_C for the set of $x \in X$ that satisfy C. For example, $X_{(x_I=\alpha)} \coloneqq \{x \in X : x_I = \alpha\}$.

Roadmap. The proof is in two steps. In the first step we find a packing with subrectangles whose free marginals are $(0.8,\mathcal{D})$-dense. In the second step we "prune" these subrectangles so that their free marginals become $(0.6,\mathcal{U})$-dense. These two steps are encapsulated in the following two lemmas.

```
**Packing Algorithm for $L$:**
Initialize $\mathscr{P} := \{L\}$ where $\mathrm{fix}\,L := \emptyset$ and $L$ is labelled *live*
**Repeat** for $n+1$ rounds
    Replace each $R \in \mathscr{P}$ by all the non-*error* subrectangles output by $\textsc{Partition}(R)$
Output $\mathscr{C}' := \mathscr{P}$
```

```
Subroutine $\textsc{Partition}$ (with error parameter $\delta := \epsilon/2n$)

*Input*: A rectangle $R_{\mathsf{in}}$
*Output*: A partition of $R_{\mathsf{in}}$ into *dense*/*live*/*error* subrectangles

Initialize $R := R_{\mathsf{in}}$ with $\mathrm{fix}\,R := \mathrm{fix}\,R_{\mathsf{in}}$
**While** the following two conditions hold
        **(C1)**: $\mathcal{D}(R \mid R_{\mathsf{in}}) > \delta$
        **(C2)**: The free marginals of $R$ are not both $(0.8, \mathcal{D})$-dense
    Let $\boldsymbol{xy} \sim (\mathcal{D} \mid R)$ and let $X$ and $Y$ be such that $R = X \times Y$
    We may assume that $\boldsymbol{x}_{\mathrm{free}\,R}$ is not 0.8-dense (otherwise consider $\boldsymbol{y}_{\mathrm{free}\,R}$)
    Let $I \subseteq \mathrm{free}\,R$ and $\alpha$ be such that $\mathbf{Pr}[\,\boldsymbol{x}_I = \alpha\,] > 2^{-0.8 \cdot b|I|}$
    Let $\mathcal{B} := \left\{\beta : \mathbf{Pr}[\,\boldsymbol{y}_I = \beta \mid \boldsymbol{x}_I = \alpha\,] > \delta \cdot 2^{-b|I|}\right\}$
    **For each** $\beta \in \mathcal{B}$
        Let $R_{\mathsf{out}} := X_{(x_I = \alpha)} \times Y_{(y_I = \beta)}$ with $\mathrm{fix}\,R_{\mathsf{out}} := \mathrm{fix}\,R \cup I$
        Output $R_{\mathsf{out}}$ labelled as *live*
    **End for**
    Output $X_{(x_I = \alpha)} \times Y_{(y_I \notin \mathcal{B})}$ labelled as *error*
    Update $R := X_{(x_I \neq \alpha)} \times Y$ (with the same $\mathrm{fix}\,R$)
**End while**
Output $R$ labelled as *dense* if **(C2)** failed, or as *error* if **(C1)** failed
```

Figure 2.2: Packing algorithm.

Lemma 2.22 (Core packing step)**.** *There is a packing $\mathscr{C}'$ of subrectangles of L such that $\mathcal{D}(\cup\mathscr{C}' \mid L) \geq 1 - \epsilon$ and for each $R \in \mathscr{C}'$ we have $|\mathrm{fix}\,R| \leq d$ and the free marginals of R are $(0.8, \mathcal{D})$-dense (for some choice of the sets* $\mathrm{fix}\,R$ *and* $\mathrm{free}\,R$*).*

Lemma 2.23 (Pruning step)**.** *For each $R \in \mathscr{C}'$ there is a subrectangle $R' \subseteq R$ with $\mathrm{fix}\,R' = \mathrm{fix}\,R$ such that $\mathcal{D}(R' \mid R) \geq 1 - \epsilon$ and the free marginals of R' are $(0.6, \mathcal{U})$-dense.*

Theorem 2.20 follows immediately by stringing together Lemma 2.22, Lemma 2.23, and Proposition 2.21. In particular, the final packing $\mathscr{C}$ will consist of the pruned rectangles R' (which are (d, ϵ)-conjunctions by Proposition 2.21) and we have $\mathcal{D}(\cup\mathscr{C} \mid L) \geq (1-\epsilon)^2 \geq 1 - 2\epsilon$. (We proved the theorem with error parameter 2ϵ instead of ϵ.)

Core packing step

We will now prove Lemma 2.22. The desired packing $\mathscr{C}'$ of subrectangles of L will be found via a packing algorithm given in Figure 2.2.

Informal overview. The principal goal in the algorithm is to find subrectangles $R \subseteq L$ whose free marginals are $(0.8, \mathcal{D})$-dense while keeping $|\operatorname{fix} R|$ small. To do this, we proceed in rounds. The main loop of the algorithm maintains a *pool* $\mathscr{P}$ of disjoint subrectangles of L and in each round we inspect each $R \in \mathscr{P}$ in the subroutine Partition. If we find that R does not have dense free marginals, we partition R further. The output of Partition(R) is a partition of R into subrectangles each labelled as either *dense*, *live*, or *error*. We are simply going to ignore the *error* rectangles, i.e., they do not re-enter the pool $\mathscr{P}$. For the *live* subrectangles $R' \subseteq R$ we will have made progress: the subroutine will ensure that the free marginals of R' will become more dense as compared to the free marginals of R.

The subroutine Partition works as follows. If the input rectangle R_{in} satisfies the density condition on its free marginals, we simply output R_{in} labelled as *dense*. Otherwise we find some subset I of free blocks that violates the density condition on one of the marginals. Then we consider the subrectangle $R_{\mathsf{out}} \subseteq R_{\mathsf{in}}$ that is obtained from R_{in} by fixing the non-dense marginal to its overly-likely value on I and the other marginal to each of its typical values on I. Intuitively, these fixings have the effect of increasing the "relative density" in the remaining free blocks, and so we have found a single subrectangle where we have made progress. We then continue iteratively on the rest of R_{in} until only a $\delta \coloneqq \epsilon/2n$ fraction of R_{in} remains, which we deem as *error*.

Note that, at the end of $n+1$ rounds, each $R \in \mathscr{C}'$ must be labelled *dense* because once a rectangle R reaches $\operatorname{fix} R = [n]$, the density condition on the free marginals is satisfied vacuously. It remains to argue that the other two properties in Lemma 2.22 hold for $\mathscr{C}'$.

Error analysis. We claim that in each run of Partition at most a fraction 2δ of the distribution $(\mathcal{D} \mid R_{\mathsf{in}})$ gets classified as *error*. This claim implies that $\cup\mathscr{C}'$ covers all but an ϵ fraction of $(\mathcal{D} \mid L)$ since the total error relative to $(\mathcal{D} \mid L)$ can be easily bounded by the number of rounds (excluding the last round, which only labels the remaining *live* rectangles as *dense*) times the error in Partition, which is $n \cdot 2\delta = \epsilon$ under our claim.

To prove our claim, we first note that the *error* rectangle output on line 18 contributes a fraction $\leq \delta$ of error relative to $(\mathcal{D} \mid R_{\mathsf{in}})$ by **(C1)**. Consider then *error* rectangles output on line 15. Here we have (using notation from the algorithm) $\mathbf{Pr}[\,\boldsymbol{y}_I \notin \mathcal{B} \mid \boldsymbol{x}_I = \alpha\,] \leq \delta$ by the definition of $\mathcal{B}$ so we only incur $\leq \delta$ fraction of error relative to $(\mathcal{D} \mid R')$ where $R' \coloneqq X_{(x_I = \alpha)} \times Y$. In the subsequent line we redefine $R \coloneqq R \smallsetminus R'$, which ensures that the errors on line 15 do not add up over the different iterations. Hence, altogether, line 15 contributes a fraction $\leq \delta$ of error relative to $(\mathcal{D} \mid R_{\mathsf{in}})$. The total error in Partition is then at most $\delta + \delta = 2\delta$, which was our claim.

Number of fixed blocks. Let $R \in \mathscr{C}'$. We need to show that $|\operatorname{fix} R| \le d$. Let R_i, $i \in [n+1]$, be the unique rectangle in the pool at the start of the i-th round such that $R \subseteq R_i$. Let ℓ be the largest number such that R_ℓ is labelled *live*. Hence $|\operatorname{fix} R| = |\operatorname{fix} R_\ell|$. Let $Q \supseteq R_\ell$ consist of all the inputs that agree with R_ℓ on the fixed coordinates $\operatorname{fix} R$. We claim that

$$\mathcal{D}(Q) \;\le\; 2^{-(2b-2)|\operatorname{fix} R|}, \tag{2.7}$$

$$\mathcal{D}(R_\ell) \;\ge\; 2^{-1.9\cdot b|\operatorname{fix} R| - db/20}. \tag{2.8}$$

Let us first see how to conclude the proof of Lemma 2.22 assuming the above inequalities. Since $\mathcal{D}(Q) \ge \mathcal{D}(R_\ell)$ we can require that (2.7) $\ge$ (2.8) and (taking logarithms) obtain the inequality $-(2b-2)|\operatorname{fix} R| \ge -1.9 \cdot b|\operatorname{fix} R| - db/20$. But this implies $|\operatorname{fix} R| \le d$, as desired.

To prove (2.7), write $\mathcal{D}(Q) = \mathbf{E}_{\boldsymbol{z}\sim\mu}\,\mathcal{U}_{\boldsymbol{z}}(Q)$ for some μ since $\mathcal{D}$ is a lifted distribution. Here for each fixed z we either have $\mathcal{U}_z(Q) = 0$ in case the fixings of Q are inconsistent with z, or otherwise $\mathcal{U}_z(Q) = \prod_{j\in\operatorname{fix} R} 1/|g^{-1}(z_j)| \le 2^{-(2b-2)|\operatorname{fix} R|}$ (where we used the fact that the gadget g is approximately balanced: $|g^{-1}(1)|, |g^{-1}(0)| \ge 2^{2b}/4$). Hence $\mathcal{D}(Q)$ is a convex combination of values that satisfy (2.7).

To prove (2.8), note that $\mathcal{D}(R_\ell) = \mathcal{D}(R_\ell \mid L) \cdot \mathcal{D}(L) \ge \mathcal{D}(R_\ell \mid L) \cdot 2^{-db/20}$. Hence it suffices to show that $\mathcal{D}(R_\ell \mid L) \ge 2^{-1.9\cdot b|\operatorname{fix} R|}$. To this end, write $|\operatorname{fix} R| = \sum_{i=1}^{\ell-1} |I_i|$ where I_i is the set of blocks that were fixed to obtain $R_{i+1} = R_{\mathsf{out}}$ from $R_i = R_{\mathsf{in}}$ and use the following claim inductively.

Claim 2.24. *Each R_{out} output labelled as live (on line 13) satisfies $\mathcal{D}(R_{\mathsf{out}} \mid R_{\mathsf{in}}) \ge 2^{-1.9\cdot b|I|}$.*

Proof. Using notation from the algorithm,

$$\begin{aligned}
\mathcal{D}(R_{\mathsf{out}} \mid R_{\mathsf{in}}) &= \mathcal{D}(R_{\mathsf{out}} \mid R) \cdot \mathcal{D}(R \mid R_{\mathsf{in}}) \\
&\ge \mathcal{D}(R_{\mathsf{out}} \mid R) \cdot \delta && \text{(by } \mathbf{(C1)}\text{)} \\
&= \mathbf{Pr}[\,\boldsymbol{x}_I = \alpha \text{ and } \boldsymbol{y}_I = \beta\,] \cdot \delta \\
&\ge 2^{-0.8\cdot b|I|} \cdot \delta \cdot 2^{-b|I|} \cdot \delta \\
&= 2^{-1.8\cdot b|I| - b/50 - 2\log n - 2} && \text{(definition of } \epsilon, \delta\text{)} \\
&\ge 2^{-1.9\cdot b|I|}.
\end{aligned}$$

□

Pruning step

We will now prove Lemma 2.23. Let $R = X \times Y \in \mathscr{C}'$ and $\boldsymbol{xy} \sim (\mathcal{D} \mid R)$. For notational convenience, we assume that $\operatorname{fix} R = \emptyset$, i.e., we forget about the fixed blocks and think of $\boldsymbol{x}$ and $\boldsymbol{y}$ as 0.8-dense. As will be clear from the proof, if $\operatorname{fix} R$ was non-empty, it would only help us in the ensuing calculations.

We want to find a "pruned" subrectangle $R' := X' \times Y' \subseteq R$ such that

(i) $\mathbf{Pr}[\,\boldsymbol{xy} \in X' \times Y'\,] \ge 1 - \epsilon$,

(ii) $\boldsymbol{X'}$ and $\boldsymbol{Y'}$ are 0.6-dense.

In fact, it is enough to show how to find an $X' \subseteq X$ such that

(i') $\mathbf{Pr}[\,\boldsymbol{x} \in X'\,] \geq 1 - \epsilon/2$,
(ii') $\boldsymbol{X'}$ is 0.6-dense.

Indeed, we can run the argument for *(i',ii')* twice, once for X and once for Y in place of X. The property *(i)* then follows by a union bound.

We will obtain X' by forbidding some outcomes of $\boldsymbol{X}_I$ that are too likely. We build up a set $\mathcal{C}$ of conditions via the following algorithm. We use the notation $X_{\mathcal{C}} = \cap_{C \in \mathcal{C}} X_C$ below.

```
Initialize $\mathcal{C} := \emptyset$
Repeat
    If $X_{\mathcal{C}} = \emptyset$, then halt with a failure
    If $\boldsymbol{X}_{\mathcal{C}}$ is 0.6-dense, then halt with a success
    Otherwise let $I$ and $\alpha$ be such that $\mathbf{Pr}[\,(\boldsymbol{X}_{\mathcal{C}})_I = \alpha\,] > 2^{-0.6 \cdot b|I|}$
    Add the condition $(x_I \neq \alpha)$ to $\mathcal{C}$
End repeat
```

This process eventually halts since $|X_{\mathcal{C}}|$ decreases every time we add a new condition to $\mathcal{C}$. Let $\mathcal{F}$ denote the set of final conditions when the process halts. We show that $X' := X_{\mathcal{F}}$ satisfies *(i',ii')*. Write $\mathcal{F} = \cup_{s \in [n]} \mathcal{F}_s$ where $\mathcal{F}_s$ denotes conditions of the form $(x_I \neq \alpha)$, $|I| = s$, in $\mathcal{F}$.

Claim 2.25. *$|\mathcal{F}_s| \leq 2^{0.7 \cdot bs}$.*

Proof of claim. The effect of adding a new condition $(x_I \neq \alpha)$, $|I| = s$, to $\mathcal{C}$ is to shrink the size of $X_{\mathcal{C}}$ by a factor of $\mathbf{Pr}[\,(\boldsymbol{X}_{\mathcal{C}})_I \neq \alpha\,] < 1 - \delta$ where $\delta := 2^{-0.6 \cdot bs}$. Our initial set has size $|X| \leq 2^{bn}$ and hence we cannot shrink it by such a condition more than $k \geq |\mathcal{F}_s|$ times where k is the smallest number satisfying $|X|(1-\delta)^k < 1$. Solving for k gives $k \leq O(bn/\delta) = O(bn \cdot 2^{0.6 \cdot bs})$, which is at most $2^{0.7 \cdot bs}$ given our definition of b. □

We can now verify *(i')* by a direct calculation:

$$\begin{aligned}
\mathbf{Pr}[\,\boldsymbol{x} \notin X'\,] &= \mathbf{Pr}[\,\boldsymbol{x} \notin X_{\mathcal{F}}\,] \\
&\leq \textstyle\sum_s \mathbf{Pr}[\,\boldsymbol{x} \notin X_{\mathcal{F}_s}\,] \\
&\leq \textstyle\sum_s \sum_{(x_I \neq \alpha) \in \mathcal{F}_s} \mathbf{Pr}[\,\boldsymbol{x}_I = \alpha\,] \\
&\leq \textstyle\sum_s |\mathcal{F}_s| \cdot 2^{-0.8 \cdot bs} && (\mathbf{H}_\infty(\boldsymbol{x}_I) \geq 0.8 \cdot b|I|) \\
&\leq \textstyle\sum_s 2^{-0.1 \cdot bs} && (\text{Claim 2.25}) \\
&\leq \epsilon/2.
\end{aligned}$$

This also proves *(ii')* because the calculation implies that $X' \neq \emptyset$ which means that our process halted with a *success*. This concludes the proof of Lemma 2.23.

2.3 Definitions of models

In Section 2.3.1 we introduce our restricted-by-default communication models, justify why they can be viewed as "zero-communication" models, and explain their relationships to known lower bound techniques. In Section 2.3.2 we define their corresponding unrestricted versions. In Section 2.3.3 we describe the query complexity counterparts of our communication models.

2.3.1 Restricted communication models

We define NP protocols in a slightly nonstandard way as randomised protocols, just for stylistic consistency with the other models. The acronyms WAPP and SBP were introduced in [BGM06] (their communication versions turn out to be equivalent to the smooth rectangle bound and the corruption bound, as argued below). We introduce the acronym $\mathsf{2WAPP}$ (for lack of existing notation) to correspond to a two-sided version of WAPP (which is equivalent to the zero-communication with abort model of [KLL$^+$15]). We use the notation $\mathsf{PostBPP}$ [Aar05] instead of the more traditional $\mathsf{BPP_{path}}$ [HHT97] as it is more natural for communication protocols.

A protocol outputs 0 or 1, and in some of these models it may also output $\perp$ representing "abort" or "don't know". In the following definition, α can be arbitrarily small and should be thought of as a function of the input size n for a family of protocols.

Definition 2.26. For $\mathcal{C} \in \{\mathsf{NP}, \mathsf{2WAPP}_\epsilon, \mathsf{WAPP}_\epsilon, \mathsf{SBP}, \mathsf{PostBPP}\}$ and $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1,*\}$ a partial function, define $\mathcal{C}^{\mathsf{cc}}(F)$ as the minimum over all $\alpha > 0$ and all "α-correct" public-randomness protocols for F of the communication cost plus $\log(1/\alpha)$ (this sum is considered to be the cost), where α-correctness is defined as follows.

NP : If $F(x,y) = 1$ then $\mathbf{Pr}[\,\Pi(x,y) = 1\,] \geq \alpha$, and if $F(x,y) = 0$ then $\mathbf{Pr}[\,\Pi(x,y) = 1\,] = 0$.

$\mathsf{2WAPP}_\epsilon$: The protocol may output $\perp$, and for all $(x,y) \in \operatorname{dom} F$, $\mathbf{Pr}[\,\Pi(x,y) = F(x,y)\,] \geq (1-\epsilon)\alpha$ and $\mathbf{Pr}[\,\Pi(x,y) \neq \perp\,] \leq \alpha$.

WAPP_ϵ : If $F(x,y) = 1$ then $\mathbf{Pr}[\,\Pi(x,y) = 1\,] \in [(1-\epsilon)\alpha, \alpha]$, and if $F(x,y) = 0$ then $\mathbf{Pr}[\,\Pi(x,y) = 1\,] \in [0, \epsilon\alpha]$.[2]

SBP : If $F(x,y) = 1$ then $\mathbf{Pr}[\,\Pi(x,y) = 1\,] \geq \alpha$, and if $F(x,y) = 0$ then $\mathbf{Pr}[\,\Pi(x,y) = 1\,] \leq \alpha/2$.

$\mathsf{PostBPP}$: The protocol may output $\perp$, and for all $(x,y) \in \operatorname{dom} F$, $\mathbf{Pr}[\,\Pi(x,y) \neq \perp\,] \geq \alpha$ and $\mathbf{Pr}[\,\Pi(x,y) = F(x,y) \mid \Pi(x,y) \neq \perp\,] \geq 3/4$.

The "syntactic relationships" among the four models $\mathsf{2WAPP}$, WAPP, SBP, $\mathsf{PostBPP}$ is summarized in the below table. The meaning of the column and row labels is as follows. For the columns, "two-sided" means that the protocol outputs values in $\{0,1,\perp\}$ and conditioned on not

[2] The definition of WAPP in [BGM06] uses ϵ in a different way: $\frac{1}{2}+\epsilon$ and $\frac{1}{2}-\epsilon$ instead of $1-\epsilon$ and ϵ.

outputting $\perp$, the output is correct with high probability. A "one-sided" protocol outputs values in $\{0,1\}$, and we measure its probability of outputting 1 and compare it against the correctness parameter $\alpha > 0$. For the rows, "bounded" means that the *non-abort* probability—that is, the probability of not outputting $\perp$ for two-sided models, or the probability of outputting 1 for one-sided models—is uniformly upper bounded by α, whereas "unbounded" means that the non-abort probability need not be upper bounded by α.

	Two-sided	One-sided
Bounded non-abort	2WAPP	WAPP
Unbounded non-abort	PostBPP	SBP

It is straightforward to see that the relative computational power ("semantic relationships") of the models is as follows (recall Figure 2.1): for all F and all constants $0 < \epsilon < 1/2$, we have $\mathsf{2WAPP}^{\mathsf{cc}}_{\epsilon}(F) \geq \mathsf{WAPP}^{\mathsf{cc}}_{\epsilon}(F) \geq \Omega(\mathsf{SBP}^{\mathsf{cc}}(F)) \geq \Omega(\mathsf{PostBPP}^{\mathsf{cc}}(F))$ and $\mathsf{NP}^{\mathsf{cc}}(F) \geq \mathsf{SBP}^{\mathsf{cc}}(F)$. Furthermore, exponential separations are known for all these relationships: unique-set-intersection is easy for $\mathsf{WAPP}^{\mathsf{cc}}_{0}$ but hard for $\mathsf{2WAPP}^{\mathsf{cc}}_{\epsilon}$ (indeed, for $\mathsf{coSBP}^{\mathsf{cc}}$ [Raz92, GW14]); set-intersection is easy for $\mathsf{SBP}^{\mathsf{cc}}$ (indeed, for $\mathsf{NP}^{\mathsf{cc}}$) but hard for $\mathsf{WAPP}^{\mathsf{cc}}_{\epsilon}$ [Kla10]; set-disjointness is easy for $\mathsf{PostBPP}^{\mathsf{cc}}$ (indeed, for $\mathsf{coNP}^{\mathsf{cc}}$) but hard for $\mathsf{SBP}^{\mathsf{cc}}$ [Raz92, GW14]; equality is easy for $\mathsf{SBP}^{\mathsf{cc}}$ (indeed, for $\mathsf{coRP}^{\mathsf{cc}}$) but hard for $\mathsf{NP}^{\mathsf{cc}}$. Moreover, $\mathsf{WAPP}^{\mathsf{cc}}$ is a one-sided version of $\mathsf{2WAPP}^{\mathsf{cc}}$ in the sense that $\mathsf{2WAPP}^{\mathsf{cc}}_{\epsilon}(F) \leq O(\mathsf{WAPP}^{\mathsf{cc}}_{\epsilon/2}(F) + \mathsf{coWAPP}^{\mathsf{cc}}_{\epsilon/2}(F))$ (so the classes would satisfy $\mathsf{2WAPP}^{\mathsf{cc}} = \mathsf{WAPP}^{\mathsf{cc}} \cap \mathsf{coWAPP}^{\mathsf{cc}}$ if we ignore the precise value of the constant ϵ).

The reason we do not include an ϵ parameter in the $\mathsf{SBP}^{\mathsf{cc}}$ and $\mathsf{PostBPP}^{\mathsf{cc}}$ models is because standard amplification techniques could be used to efficiently decrease ϵ in these models (rendering the exact value immaterial up to constant factors). Another subtlety concerns the behavior of correct protocols on the *undefined* inputs $\{0,1\}^n \times \{0,1\}^n \smallsetminus \operatorname{dom} F$. For example, for $\mathsf{2WAPP}^{\mathsf{cc}}_{\epsilon}$, the corresponding definitions in [KLL+15] also require that for every undefined input (x,y), $\mathbf{Pr}[\,\Pi(x,y) \neq \perp\,] \in [(1-\epsilon)\alpha, \alpha]$. We allow arbitrary behavior on the undefined inputs for stylistic consistency, but our results also hold for the other version. As a final remark, we mention that our definition of $\mathsf{NP}^{\mathsf{cc}}$ is only equivalent to the usual definition within an additive logarithmic term; see Remark 2 below.

Relation to zero-communication models. The following fact shows that protocols in our models can be expressed simply as distributions over (labelled) rectangles; thus these models can be considered "zero-communication" since Alice and Bob can each produce an output with no communication, and then have the output of the protocol be a simple function of their individual outputs.[3]

Fact 2.27. *Without loss of generality, in each of the five models from Definition 2.26, for each outcome of the public randomness the associated deterministic protocol is of the following form.*

[3] Admittedly, for Alice and Bob themselves to know the output of this simple function, they would need to use a constant amount of communication.

$\mathsf{NP},\ \mathsf{WAPP}_\epsilon,\ \mathsf{SBP}$: *There exists a rectangle R such that the output is 1 iff the input is in R.*

$2\mathsf{WAPP}_\epsilon,\ \mathsf{PostBPP}$: *There exists a rectangle R and a bit b such that the output is b if the input is in R and is $\perp$ otherwise.*

Proof. Consider a protocol Π in one of the models from Definition 2.26, and suppose it has communication cost c and associated $\alpha > 0$, so the cost is $c + \log(1/\alpha)$. We may assume that each deterministic protocol has exactly 2^c possible transcripts. Transform Π into a new protocol Π' that operates as follows on input (x, y): Sample an outcome of the public randomness of Π, then sample a uniformly random transcript with associated rectangle R and output-value b, then execute the following.

$$\text{If } (x,y) \in R \text{ then output } b \text{, otherwise output } \begin{cases} 0 & \text{if } \mathsf{NP},\ \mathsf{WAPP}_\epsilon,\ \mathsf{SBP} \\ \perp & \text{if } 2\mathsf{WAPP}_\epsilon,\ \mathsf{PostBPP} \end{cases}.$$

We have $\mathbf{Pr}[\,\Pi'(x,y) = 1\,] = 2^{-c}\,\mathbf{Pr}[\,\Pi(x,y) = 1\,]$, and for $2\mathsf{WAPP}_\epsilon,\ \mathsf{PostBPP}$ we also have $\mathbf{Pr}[\,\Pi'(x,y) = 0\,] = 2^{-c}\,\mathbf{Pr}[\,\Pi(x,y) = 0\,]$. Thus in all cases Π' is $(2^{-c}\alpha)$-correct. Formally, it takes two bits of communication to check whether $(x, y) \in R$, so the cost of Π' is $2 + \log(1/2^{-c}\alpha)$, which is the cost of Π plus 2. □

Relation to lower bound measures. Using Fact 2.27 it is straightforward to see that, ignoring the $+2$ cost of checking whether the input is in a rectangle, $2\mathsf{WAPP}^{\mathsf{cc}}_\epsilon$ is exactly equivalent to the relaxed partition bound of [KLL+15] (with the aforementioned caveat about undefined inputs) and $\mathsf{WAPP}^{\mathsf{cc}}_\epsilon$ is exactly equivalent to the (one-sided) smooth rectangle bound[4], denoted srec^1 [JK10]. For completeness, the definition of srec^1 and the proof of the following fact appear in Section 2.7.1.

Fact 2.28. *$\mathsf{srec}^1_\epsilon(F) \le \mathsf{WAPP}^{\mathsf{cc}}_\epsilon(F) \le \mathsf{srec}^1_\epsilon(F) + 2$ for all F and all $0 < \epsilon < 1/2$.*

It was shown in [GW14] that $\mathsf{SBP}^{\mathsf{cc}}$ is equivalent (within constant factors) to the (one-sided) corruption bound. We remark that by a simple application of the minimax theorem, $\mathsf{PostBPP}^{\mathsf{cc}}$ also has a dual characterization analogous to the corruption bound.[5]

2.3.2 Unrestricted communication models

For all the models described above, we can define their unrestricted versions, denoted by prepending U to the acronym (not to be confused with complexity classes where U stands for "unambiguous"). The distinction is that the restricted versions charge $+\log(1/\alpha)$ in the cost, whereas the unrestricted versions do not charge anything for α in the cost (and hence they are

[4] The paper that introduced this bound [JK10] defined it as the optimum value of a certain linear program, but following [KMSY14] we define it as the log of the optimum value.

[5] $\mathsf{PostBPP}^{\mathsf{cc}}(F)$ is big-Θ of the maximum over all distributions μ over $\{0,1\}^n \times \{0,1\}^n$ of the minimum $\log(1/\mu(R))$ over all rectangles R that are unbalanced in the sense that $\mu(R \cap F^{-1}(1))$ and $\mu(R \cap F^{-1}(0))$ are not within a factor of 2 of each other. In the corruption bound, the maximum is only over balanced μ, and R is considered unbalanced if $\mu(R \cap F^{-1}(1))$ is more than some constant factor greater than $\mu(R \cap F^{-1}(0))$.

defined using private randomness; otherwise every function would be computable with constant cost.)

Definition 2.29. For $\mathcal{C} \in \{\mathsf{NP}, \mathsf{2WAPP}_\epsilon, \mathsf{WAPP}_\epsilon, \mathsf{SBP}, \mathsf{PostBPP}\}$ and $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1,*\}$ a partial function, define $\mathsf{U}\mathcal{C}^{\mathsf{cc}}(F)$ as the minimum over all $\alpha > 0$ and all "α-correct" private-randomness protocols for F of the communication cost, where the α-correctness criteria are as in Definition 2.26.

Standard sparsification of randomness (à la Newman's Theorem [New91], [KN97, Theorem 3.14]) can be used to show that the unrestricted models are essentially at least as powerful as their restricted versions for all F: for $\mathcal{C} \in \{\mathsf{NP}, \mathsf{SBP}, \mathsf{PostBPP}\}$ we have $\mathsf{U}\mathcal{C}^{\mathsf{cc}}(F) \leq O(\mathcal{C}^{\mathsf{cc}}(F) + \log n)$, and for $\mathcal{C} \in \{\mathsf{2WAPP}, \mathsf{WAPP}\}$ we have $\mathsf{U}\mathcal{C}^{\mathsf{cc}}_\delta(F) \leq O(\mathcal{C}^{\mathsf{cc}}_\epsilon(F) + \log(n/(\delta - \epsilon)))$ where $0 < \epsilon < \delta$. (The additive logarithmic terms come from converting public randomness to private.)

Remark 2. We note that $\mathsf{UNP}^{\mathsf{cc}}$ is actually equivalent to the standard definition of nondeterministic communication complexity, while our $\mathsf{NP}^{\mathsf{cc}}$ from Definition 2.26 is only equivalent within an additive logarithmic term. It is fair to call this an abuse of notation, but it does not affect our communication–query equivalence for NP since we consider block length $b = \Omega(\log n)$ anyway.

$\mathsf{UWAPP}^{\mathsf{cc}}$ and nonnegative rank. Of particular interest to us will be $\mathsf{UWAPP}^{\mathsf{cc}}$ which turns out to be equivalent to *approximate nonnegative rank*. Recall that for M a nonnegative matrix, the *nonnegative rank* $\operatorname{rank}^+(M)$ is defined as the minimum r such that M can be written as the sum of r nonnegative rank-1 matrices, or equivalently, $M = UV$ for nonnegative matrices U, V with inner dimension r for the multiplication. Below, we view a partial function $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1,*\}$ as a $2^n \times 2^n$ partial boolean matrix.

Definition 2.30 (Approximate nonnegative rank)**.** For partial F, $\operatorname{rank}^+_\epsilon(F)$ is defined as the minimum $\operatorname{rank}^+(M)$ over all nonnegative matrices M such that $M_{x,y} \in F(x,y) \pm \epsilon$ for all $(x,y) \in \operatorname{dom} F$ (in other words, $\|F - M\|_\infty \leq \epsilon$ on $\operatorname{dom} F$).

For completeness, the straightforward proof of the following fact appears in Section 2.7.2.

Fact 2.31. $\log \operatorname{rank}^+_\epsilon(F) \leq \mathsf{UWAPP}^{\mathsf{cc}}_\epsilon(F) \leq \lceil \log \operatorname{rank}^+_{\epsilon/2}(F) \rceil + 2$ *for all* F *and all* $0 < \epsilon < 1/2$.

2.3.3 Query models

A randomised decision tree $\mathcal{T}$ is a probability distribution over deterministic decision trees, and the query cost is the maximum height of a decision tree in the support.

Definition 2.32. For $\mathcal{C} \in \{\mathsf{NP}, \mathsf{2WAPP}_\epsilon, \mathsf{WAPP}_\epsilon, \mathsf{SBP}, \mathsf{PostBPP}\}$ and $f\colon \{0,1\}^n \to \{0,1,*\}$ a partial function, define $\mathcal{C}^{\mathsf{dt}}(f)$ as the minimum over all $\alpha > 0$ and all "α-correct" randomised decision trees for f of the query cost, where the α-correctness criteria are as in Definition 2.26 (but where protocols $\Pi(x,y)$ are replaced with randomised decision trees $\mathcal{T}(z)$).

Completely analogously to how the zero-communication models can be viewed w.l.o.g. as distributions over (labelled) rectangles (Fact 2.27), their query counterparts can be viewed w.l.o.g. as distributions over (labelled) conjunctions.

Fact 2.33. *Without loss of generality, in each of the five models from Definition 2.32, for each outcome of the randomness the associated deterministic decision tree is of the following form.*

$\mathsf{NP},\ \mathsf{WAPP}_\epsilon,\ \mathsf{SBP}$: *There exists a conjunction h such that the output is 1 iff the input is in $h^{-1}(1)$.*

$2\mathsf{WAPP}_\epsilon,\ \mathsf{PostBPP}$: *There exists a conjunction h and a bit b such that the output is b if the input is in $h^{-1}(1)$ and is $\perp$ otherwise.*

Proof. Consider a randomised decision tree $\mathcal{T}$ in one of the models from Definition 2.32, and suppose it has query cost d and associated $\alpha > 0$. We may assume that each deterministic decision tree has a full set of 2^d leaves and the queries along each root-to-leaf path are distinct. Hence each leaf is associated with a width-d conjunction that checks whether the input is consistent with the queries made in its root-to-leaf path. Transform $\mathcal{T}$ into a new randomised decision tree $\mathcal{T}'$ that operates as follows on input z: Sample an outcome of the randomness of $\mathcal{T}$, then sample a uniformly random leaf with associated conjunction h and output-value b, then execute the following.

$$\text{If } h(z) = 1 \text{ then output } b\text{, otherwise output } \begin{cases} 0 & \text{if } \mathsf{NP},\ \mathsf{WAPP}_\epsilon,\ \mathsf{SBP} \\ \perp & \text{if } 2\mathsf{WAPP}_\epsilon,\ \mathsf{PostBPP} \end{cases}.$$

We have $\mathbf{Pr}[\mathcal{T}'(z) = 1] = 2^{-d}\,\mathbf{Pr}[\mathcal{T}(z) = 1]$, and for $2\mathsf{WAPP}_\epsilon$, $\mathsf{PostBPP}$ we also have $\mathbf{Pr}[\mathcal{T}'(z) = 0] = 2^{-d}\,\mathbf{Pr}[\mathcal{T}(z) = 0]$. Thus in all cases $\mathcal{T}'$ is $(2^{-d}\alpha)$-correct, and $\mathcal{T}'$ also has query cost d. □

We defined our query models without charging anything for α, i.e., α is unrestricted. This means that deriving communication upper bounds for $f \circ g^n$ in restricted models from corresponding query upper bounds for f is nontrivial; this is discussed in Section 2.4.2. Nevertheless, we contend that Definition 2.26 and Definition 2.32 are the "right" definitions that correspond to one another. The main reason is because in the "normal forms" (Fact 2.27 and Fact 2.33), all the cost in the communication version comes from α, and all the cost in the query version comes from the width of the conjunctions—and when we apply the Junta Theorem in Section 2.4.1, the communication α directly determines the conjunction width.

2.4 Proof of the Simulation Theorem

In this section we derive the Simulation Theorem (Theorem 2.2) from the Junta Theorem (Theorem 2.1). The proof is in two parts: Section 2.4.1 for lower bounds and Section 2.4.2 for upper bounds.

2.4.1 Communication lower bounds

The Junta Theorem implies that for functions lifted with our hard gadget g, every distribution $\mathcal{R}$ over rectangles can be transformed into a distribution $\mathcal{H}$ over conjunctions such that for every $z \in \{0,1\}^n$, the acceptance probability under $\mathcal{H}$ is related in a simple way to the acceptance probability under $\mathcal{R}$ averaged over all two-party encodings of z. This allows us to convert zero-communication protocols (which are distributions over (labelled) rectangles by Fact 2.27) into corresponding decision trees (which are distributions over (labelled) conjunctions by Fact 2.33).

More precisely, let $\mathcal{R}$ be a distribution over rectangles in the domain of $G = g^n$. First, apply the Junta Theorem to each R in the support of $\mathcal{R}$ to get an approximating conical d-junta h_R. Now we can approximate the convex combination

$$\mathrm{acc}_{\mathcal{R}}(z) = \mathop{\mathbf{E}}_{\boldsymbol{R}\sim\mathcal{R}} \mathrm{acc}_{\boldsymbol{R}}(z) \in \mathop{\mathbf{E}}_{\boldsymbol{R}\sim\mathcal{R}} \Big((1\pm o(1))\cdot h_{\boldsymbol{R}}(z) \pm 2^{-\Theta(db)}\Big) \subseteq (1\pm o(1))\cdot\Big(\mathop{\mathbf{E}}_{\boldsymbol{R}\sim\mathcal{R}} h_{\boldsymbol{R}}(z)\Big) \pm 2^{-\Theta(db)}$$

by the conical d-junta $\mathbf{E}_{\boldsymbol{R}\sim\mathcal{R}}\, h_{\boldsymbol{R}}$ with the same parameters as in the Junta Theorem (we settle for multiplicative error $(1\pm o(1))$ since it suffices for the applications). But conical d-juntas are—up to scaling—convex combinations of width-d conjunctions. Specifically, we may write any conical d-junta as $\mathrm{acc}_{\mathcal{H}}(z)/a$ where $a > 0$ is some constant of proportionality and $\mathrm{acc}_{\mathcal{H}}(z) \coloneqq \mathbf{E}_{\boldsymbol{h}\sim\mathcal{H}}\, \boldsymbol{h}(z)$ where $\mathcal{H}$ is a distribution over width-d conjunctions. Finally, we rearrange the approximation so the roles of $\mathrm{acc}_{\mathcal{H}}(z)$ and $\mathrm{acc}_{\mathcal{R}}(z)$ are swapped, since it is more convenient for the applications. Hence we arrive at the following reformulation of the Junta Theorem.

Corollary 2.34 (Junta Theorem—reformulation). *Assume (†). For any $d \geq 0$ and any distribution $\mathcal{R}$ over rectangles in the domain of g^n there exists a distribution $\mathcal{H}$ over width-d conjunctions and a constant of proportionality $a > 0$ such that, for all $z \in \{0,1\}^n$,*

$$\mathrm{acc}_{\mathcal{H}}(z) \;\in\; a\cdot\big((1\pm o(1))\cdot \mathrm{acc}_{\mathcal{R}}(z) \;\pm\; 2^{-\Theta(db)}\big). \tag{2.9}$$

We will now prove the lower bounds in Theorem 2.2. Here the error parameters for WAPP are made more explicit.

Theorem 2.35. *Assume (†). For any $f\colon \{0,1\}^n \to \{0,1,*\}$ and constants $0 < \epsilon < \delta < 1/2$,*

$$\begin{aligned} \mathcal{C}^{\mathsf{cc}}(f\circ g^n) \;&\geq\; \Omega(\mathcal{C}^{\mathsf{dt}}(f)\cdot b) \qquad \textit{for} \quad \mathcal{C}\in\{\mathsf{NP},\mathsf{SBP},\mathsf{PostBPP}\},\\ \mathcal{C}^{\mathsf{cc}}_{\epsilon}(f\circ g^n) \;&\geq\; \Omega(\mathcal{C}^{\mathsf{dt}}_{\delta}(f)\cdot b) \qquad \textit{for} \quad \mathcal{C}\in\{\mathsf{2WAPP},\mathsf{WAPP}\}. \end{aligned}$$

Proof. For convenience of notation we let $\mathcal{C}^{\mathsf{cc}} \coloneqq \mathcal{C}^{\mathsf{cc}}_{\epsilon}$ and $\mathcal{C}^{\mathsf{dt}} \coloneqq \mathcal{C}^{\mathsf{dt}}_{\delta}$ in the $\mathcal{C}\in\{\mathsf{2WAPP},\mathsf{WAPP}\}$ cases. Given an α-correct cost-c $\mathcal{C}^{\mathsf{cc}}$ protocol Π for $f\circ g^n$ assumed to be in the "normal form" given by Fact 2.27, we convert it into a cost-$O(c/b)$ $\mathcal{C}^{\mathsf{dt}}$ decision tree $\mathcal{T}$ for f.

For $\mathcal{C}\in\{\mathsf{NP},\mathsf{WAPP},\mathsf{SBP}\}$, Π is a distribution over rectangles, so applying Corollary 2.34 with $d \coloneqq O(c/b)$ so that $2^{-\Theta(db)} \leq o(2^{-c}) = o(\alpha)$, there exists a distribution $\mathcal{T}$ over width-d conjunctions and an $a > 0$ such that for all $z\in\{0,1\}^n$, $\mathrm{acc}_{\mathcal{T}}(z) \in a\cdot\big((1\pm o(1))\cdot\mathrm{acc}_{\Pi}(z) \pm o(\alpha)\big)$.

Note that $\mathrm{acc}_\Pi(z)$ obeys the α-correctness criteria of f since it obeys the α-correctness criteria of $f \circ g^n$ for each encoding of z. Hence $\mathrm{acc}_{\mathcal{T}}(z)$ obeys the $(a\alpha')$-correctness criteria for some $\alpha' \in \alpha \cdot (1 \pm o(1))$. (For $\mathcal{C} = \mathsf{SBP}$ slight amplification may be needed. Also, for $\mathcal{C} = \mathsf{NP}$ we need to ensure that $\mathrm{acc}_{\mathcal{T}}(z) = 0$ whenever $\mathrm{acc}_\Pi(z) = 0$, but this is implicit in the proof of the Junta Theorem; see the left side of (2.5).) In conclusion, $\mathcal{T}$ is a cost-d $\mathcal{C}^{\mathsf{dt}}$ decision tree for f.

For $\mathcal{C} \in \{\mathsf{2WAPP}, \mathsf{PostBPP}\}$, Π can be viewed as a convex combination $\pi_0\Pi_0 + \pi_1\Pi_1$ where Π_0 is a distribution over 0-labelled rectangles and Π_1 is a distribution over 1-labelled rectangles. Applying the above argument to Π_0 and Π_1 separately, we may assume the scaling factor a is the same for both, by assigning some probability to a special "contradictory" conjunction that accepts nothing. We get a distribution over labelled width-d conjunctions $\mathcal{T} \coloneqq \pi_0\mathcal{T}_0 + \pi_1\mathcal{T}_1$ such that $\mathbf{Pr}[\,\mathcal{T}(z) = 0\,] = \pi_0\,\mathrm{acc}_{\mathcal{T}_0}(z) \in \pi_0 a \cdot \big((1 \pm o(1)) \cdot \mathrm{acc}_{\Pi_0}(z) \pm o(\alpha)\big) \subseteq a \cdot \big((1 \pm o(1)) \cdot \mathbf{Pr}[\,\Pi(z) = 0\,] \pm o(\alpha)\big)$ where we use the shorthand $\mathbf{Pr}[\,\Pi(z) = 0\,] \coloneqq \mathbf{E}_{\boldsymbol{xy} \sim \mathcal{U}_z}\,\mathbf{Pr}[\,\Pi(\boldsymbol{x}, \boldsymbol{y}) = 0\,]$. An analogous property holds for outputting 1 instead of 0. Note that $\mathbf{Pr}[\,\Pi(z) = 0\,]$ and $\mathbf{Pr}[\,\Pi(z) = 1\,]$ obey the α-correctness criteria since they do for each encoding of z. Hence $\mathbf{Pr}[\,\mathcal{T}(z) = 0\,]$ and $\mathbf{Pr}[\,\mathcal{T}(z) = 1\,]$ obey the $(a\alpha')$-correctness criteria for some $\alpha' \in \alpha \cdot (1 \pm o(1))$. (For $\mathcal{C} = \mathsf{PostBPP}$ slight amplification may be needed.) In conclusion, $\mathcal{T}$ is a cost-d $\mathcal{C}^{\mathsf{dt}}$ decision tree for f. □

2.4.2 Communication upper bounds

Theorem 2.36. *Let $\mathcal{C} \in \{\mathsf{NP}, \mathsf{2WAPP}_\epsilon, \mathsf{WAPP}_\epsilon, \mathsf{SBP}\}$. For any partial $f\colon \{0,1\}^n \to \{0,1,*\}$ and any gadget $g\colon \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$, we have $\mathcal{C}^{\mathsf{cc}}(f \circ g^n) \le O(\mathcal{C}^{\mathsf{dt}}(f) \cdot (b + \log n))$.*

Proof. On input (x, y) the communication protocol just simulates the randomised decision tree on input $z \coloneqq g^n(x, y)$, and when the decision tree queries the i-th bit of z, the communication protocol evaluates $z_i \coloneqq g(x_i, y_i)$ by brute force. This has communication cost $\mathcal{C}^{\mathsf{dt}}(f) \cdot (b + 1)$, and it inherits the α parameter from the randomised decision tree. The nontrivial part is that the query models allow arbitrarily small α, which could give arbitrarily large $+\log(1/\alpha)$ cost to the communication protocol. For these particular query models, it turns out that we can assume without loss of generality that $\log(1/\alpha) \le O(\mathcal{C}^{\mathsf{dt}}(f) \cdot \log n)$. We state and prove this for $\mathsf{SBP}^{\mathsf{dt}}$ below. (The other three models are no more difficult to handle.) □

Proposition 2.37. *Every partial function f admits an α-correct $\mathsf{SBP}^{\mathsf{dt}}$ decision tree of query cost $d \coloneqq \mathsf{SBP}^{\mathsf{dt}}(f)$ where $\alpha \ge 2^{-d}\binom{n}{d}^{-1} \ge 2^{-O(d \cdot \log n)}$.*

Proof. Consider an α'-correct cost-d $\mathsf{SBP}^{\mathsf{dt}}$ decision tree for f in the "normal form" given by Fact 2.33. We may assume each deterministic decision tree in the support is a conjunction with exactly d literals (and there are $2^d\binom{n}{d}$ many such conjunctions). The crucial observation is that it never helps to assign a probability larger than α' to any conjunction: if some conjunction appears with probability $p > \alpha'$, we may replace its probability with α' and assign the leftover probability $p - \alpha'$ to a special "contradictory" conjunction that accepts nothing. This modified randomised decision tree is still α'-correct for f. Finally, remove all probability from the contradictory conjunction and scale the remaining probabilities (along with α') to sum up to 1. Let α be the scaled version of α'. Now we have that α is greater than or equal to each of $2^d\binom{n}{d}$ many probabilities, and hence α must be at least the reciprocal of this number. □

Remark 3. In the case of $\mathsf{PostBPP}^{\mathsf{dt}}$ we cannot assume w.l.o.g. that $\log(1/\alpha) \leq \operatorname{poly}(d, \log n)$. The canonical counterexample is a *decision list* function $f\colon \{0,1\}^n \to \{0,1\}$ defined relative to a binary vector $(a_1, \dots, a_n) \in \{0,1\}^n$ so that $f(x) \coloneqq a_i$ where $i \in [n]$ is the smallest number such that $x_i = 1$, or $f(x) \coloneqq 0$ if no such i exists. Each decision list admits a cost-1 $\mathsf{PostBPP}^{\mathsf{dt}}$ decision tree, but for some decision lists the associated α must be exponentially small in n; see, e.g., [BVdW07] for more details. Indeed, two-party lifts of decision lists have been used in separating unrestricted communication models from restricted ones as we will discuss in Section 2.6.

2.5 Applications of the Simulation Theorem

In this section we use the Simulation Theorem to derive our applications. We prove Theorem 2.3 and Theorem 2.6 in Section 2.5.1 and Section 2.5.2, respectively. Throughout this section we use $o(1)$ to denote a quantity that is upper bounded by some sufficiently small constant, which may be different for the different instances of $o(1)$. (For example, $a \leq o(b)$ formally means there exists a constant $\epsilon > 0$ such that $a \leq \epsilon \cdot b$.)

2.5.1 Nonclosure under intersection

Recall that $f_\wedge(z, z') \coloneqq f(z) \wedge f(z')$. Here $f_\wedge$ is *not* to be thought of as a two-party function; we study the query complexity of $f_\wedge$, whose input we happen to divide into two halves called z and z'. We start with the following lemma.

Lemma 2.38. *There exists a partial f such that $\mathsf{SBP}^{\mathsf{dt}}(f) \leq O(1)$, but $\mathsf{SBP}^{\mathsf{dt}}(f_\wedge) \geq \Omega(n^{1/4})$.*

Let $k \coloneqq o(\sqrt{n})$ and define a partial function $f\colon \{0,1\}^n \to \{0,1,*\}$ by

$$f(z) \;\coloneqq\; \begin{cases} 1 & \text{if } |z| \geq k \\ 0 & \text{if } |z| \leq k/2 \\ * & \text{otherwise} \end{cases}$$

where $|z|$ denotes the Hamming weight of z.

In proving the lower bound in Lemma 2.38 we make use of the following duality principle for $\mathsf{SBP}^{\mathsf{dt}}$, which we phrase abstractly in terms of a collection $\mathscr{H}$ of "basic functions" over some finite set of inputs Z. In our concrete case $\mathscr{H}$ consists of decision trees of height d, or equivalently width-d conjunctions by Fact 2.33, and $Z \subseteq \{0,1\}^n$ is the domain of the partial function f. We state the duality principle for acceptance gap $[0, \alpha/2)$-vs-$(\alpha, 1]$ rather than $[0, \alpha/2]$-vs-$[\alpha, 1]$ as this implicitly ensures $\alpha > 0$. The slight difference in the multiplicative gap, (>2)-vs-(≥ 2), is immaterial as the gap can be efficiently amplified for SBP affecting only constant factors.

Fact 2.39. *For all $\mathscr{H} \subseteq \{0,1\}^Z$ and non-constant $f\colon Z \to \{0,1\}$, the following are equivalent.*

(i) There exists a distribution $\mathcal{H}$ over $\mathscr{H}$ such that for all $(z_1, z_0) \in f^{-1}(1) \times f^{-1}(0)$,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{h}\sim\mathcal{H}}[\,\boldsymbol{h}(z_1) = 1\,] \;>\; 2 \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{h}\sim\mathcal{H}}[\,\boldsymbol{h}(z_0) = 1\,]. \tag{2.10}$$

(ii) For each pair of distributions (μ_1, μ_0) over $f^{-1}(1)$ and $f^{-1}(0)$ there is an $h \in \mathscr{H}$ with

$$\mathop{\mathbf{Pr}}_{\boldsymbol{z_1}\sim\mu_1}[\,h(\boldsymbol{z_1}) = 1\,] \;>\; 2 \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{z_0}\sim\mu_0}[\,h(\boldsymbol{z_0}) = 1\,]. \tag{2.11}$$

The direction *(i)* ⇒ *(ii)* is trivial and is all we need for our proof, but it is interesting that the converse direction *(ii)* ⇒ *(i)* also holds, by a slightly non-standard argument. We include a full proof in Section 2.7.4.

We also use the following basic calculation (given in Section 2.7.3 for completeness).

Fact 2.40. *Let $h\colon \{0,1\}^n \to \{0,1\}$ be a width-d conjunction with i positive literals. Then h accepts a uniformly random string of Hamming weight w with probability $\in (w/n)^i \cdot (1 \pm o(1))$ provided $w \leq o(\sqrt{n})$ and $d \leq o(\sqrt{w})$.*

Proof of Lemma 2.38. Let f and $f_\wedge$ be as above. We have $\mathsf{SBP}^{\mathsf{dt}}(f) = 1$ via the decision tree $\mathcal{T}$ that picks a random coordinate and accepts iff the coordinate is 1. For the lower bound on $\mathsf{SBP}^{\mathsf{dt}}(f_\wedge)$, we use the contrapositive of *(i)* ⇒ *(ii)*. Let $\mathscr{H}$ consist of all conjunctions of width $o(n^{1/4})$. Let $\mathcal{Z}_w$ denote the uniform distribution over n-bit strings of weight w, intended to be used as either the first input z or the second input z' to $f_\wedge$. We construct a hard pair of distributions (μ_1, μ_0) over $f_\wedge^{-1}(1)$ and $f_\wedge^{-1}(0)$, respectively, by

$$\mu_1 \;\coloneqq\; \mathcal{Z}_k \times \mathcal{Z}_k, \qquad \mu_0 \;\coloneqq\; \frac{1}{2}(\mathcal{Z}_{k/2} \times \mathcal{Z}_{2k}) + \frac{1}{2}(\mathcal{Z}_{2k} \times \mathcal{Z}_{k/2}).$$

Here × denotes concatenation of strings, e.g., $(\boldsymbol{z}, \boldsymbol{z'}) \sim \mu_1$ is such that $\boldsymbol{z}, \boldsymbol{z'} \sim \mathcal{Z}_k$ and $\boldsymbol{z}$ and $\boldsymbol{z'}$ are independent. For intuition why the pair (μ_1, μ_0) is hard, consider the natural decision tree $\mathcal{T}_\wedge$ attempting to compute $f_\wedge$ that runs $\mathcal{T}$ (defined above) twice, once for z and once for z', accepting iff both runs accept. Since $\mathcal{T}$ accepts $\mathcal{Z}_k$ with probability k/n, we have that $\mathcal{T}_\wedge$ accepts μ_1 with probability k^2/n^2. Similarly, $\mathcal{T}_\wedge$ accepts μ_0 with probability $\frac{1}{2}(k/2n) \cdot (2k/n) + \frac{1}{2}(2k/n) \cdot (k/2n) = k^2/n^2$. Hence $\mathcal{T}_\wedge$ fails to distinguish between μ_1 and μ_0. More generally, we make a similar calculation for any width-$o(n^{1/4})$ conjunction. Indeed, let $h\colon \{0,1\}^{2n} \to \{0,1\}$ be an arbitrary conjunction in $\mathscr{H}$, and suppose h has i positive literals in z and j positive literals in z'. Then by Fact 2.40 we have

$$\frac{\mathbf{Pr}_{(\boldsymbol{z},\boldsymbol{z'})\sim\mu_1}[\,h(\boldsymbol{z},\boldsymbol{z'}) = 1\,]}{\mathbf{Pr}_{(\boldsymbol{z},\boldsymbol{z'})\sim\mu_0}[\,h(\boldsymbol{z},\boldsymbol{z'}) = 1\,]} \;\in\; \frac{(k/n)^i \cdot (k/n)^j}{\frac{1}{2} \cdot (k/2n)^i \cdot (2k/n)^j + \frac{1}{2} \cdot (2k/n)^i \cdot (k/2n)^j} \cdot (1 \pm o(1))$$

$$
\begin{aligned}
&= \frac{1}{\frac{1}{2}\cdot 2^{j-i} + \frac{1}{2}\cdot 2^{i-j}} \cdot (1 \pm o(1)) \\
&\leq 1 \cdot (1 \pm o(1)) \\
&\leq 2.
\end{aligned}
$$

This means that $\neg$*(ii)* and hence $\neg$*(i)*. Therefore $f_\wedge$ has no cost-$o(n^{1/4})$ $\mathsf{SBP}^{\mathsf{dt}}$ decision tree. □

We can now prove Theorem 2.3, restated here from the introduction.

Theorem 2.3. $\mathsf{SBP}^{\mathsf{cc}}$ *is not closed under intersection.*

Proof. Let f and $f_\wedge$ be as above. Define $F := f \circ g^n$ and $F_\wedge := f_\wedge \circ g^{2n} = (f \circ g^n)_\wedge$ where $g\colon \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$, $b = \Theta(\log n)$, is our hard gadget from (†). Then by the Simulation Theorem (Theorem 2.2), we have $\mathsf{SBP}^{\mathsf{cc}}(F_\wedge) \geq \Omega(\mathsf{SBP}^{\mathsf{dt}}(f_\wedge) \cdot b) \geq \Omega(n^{1/4} \cdot b)$ which is not polylogarithmic in the input length so that $F_\wedge \notin \mathsf{SBP}^{\mathsf{cc}}$. Furthermore, we have $\mathsf{SBP}^{\mathsf{cc}}(F) \leq O(\mathsf{SBP}^{\mathsf{dt}}(f) \cdot b) \leq O(b)$ which is logarithmic in the input length. Thus $F \in \mathsf{SBP}^{\mathsf{cc}}$, which implies that $F_\wedge$ is the intersection of two functions in $\mathsf{SBP}^{\mathsf{cc}}$ (one that evaluates F on the first half of the input, and one that evaluates F on the second half). □

2.5.2 Unamplifiability of error

Our next application of the Simulation Theorem shows that the error parameter ϵ for $\mathsf{WAPP}^{\mathsf{cc}}$ cannot be efficiently amplified. Combining this with the results illustrated in Figure 2.4 (in particular, the fact that the equivalence holds for partial functions) shows that also for approximate nonnegative rank, ϵ cannot be efficiently amplified.

Theorem 2.6. *For all constants $0 < \epsilon < \delta < 1/2$ there exists a two-party partial function F such that $\mathsf{WAPP}^{\mathsf{cc}}_\delta(F) \leq O(\log n)$ but $\mathsf{WAPP}^{\mathsf{cc}}_\epsilon(F) \geq \Omega(n)$.*

Proof. Let c/d be a rational (expressed in lowest terms) such that $(1-2\delta)/(1-\delta) \leq c/d < (1-2\epsilon)/(1-\epsilon)$. Note that such c, d exist (since $\epsilon < \delta$) and that they are constants depending only on ϵ and δ. Define a partial function $f\colon \{0,1\}^n \to \{0,1,*\}$ by

$$
f(z) := \begin{cases} 1 & \text{if } |z| \in \{c, d\} \\ 0 & \text{if } |z| = 0 \\ * & \text{otherwise} \end{cases}
$$

where $|z|$ denotes the Hamming weight of z. By the Simulation Theorem (Theorem 2.35 and Theorem 2.36), it suffices to prove that $\mathsf{WAPP}^{\mathsf{dt}}_\delta(f) \leq O(1)$ and $\mathsf{WAPP}^{\mathsf{dt}}_\epsilon(f) \geq \Omega(n)$.

Upper bound. Consider a cost-1 decision tree $\mathcal{T}'$ that picks a random coordinate and accepts iff the coordinate is 1. Then $\mathrm{acc}_{\mathcal{T}'}(z) = |z|/n$. Let $\alpha := d/n$ and define $\mathcal{T}$ as follows: on input z accept with probability $\delta\alpha$, reject with probability $\delta(1-\alpha)$, and run $\mathcal{T}'(z)$ with the remaining probability $(1-\delta)$. Now $\mathrm{acc}_{\mathcal{T}}(z)$ behaves as plotted on the left side of Figure 2.3: if

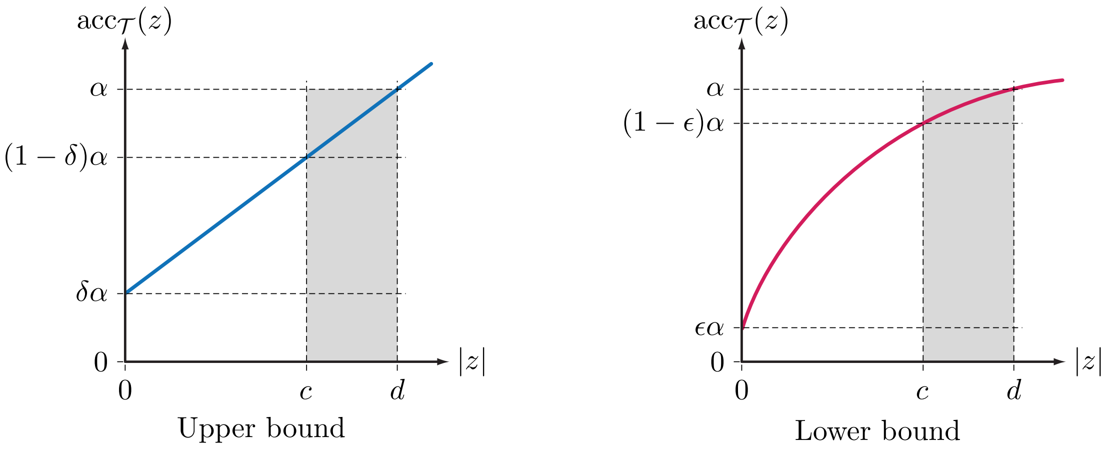


Figure 2.3: Illustration for the proof of Theorem 2.6.

$|z| = 0$ then $\mathrm{acc}_{\mathcal{T}}(z) = \delta\alpha$, if $|z| = d$ then $\mathrm{acc}_{\mathcal{T}}(z) = \delta\alpha + (1-\delta)d/n = \alpha$, and if $|z| = c$ then $\mathrm{acc}_{\mathcal{T}}(z) = \delta\alpha + (1-\delta)c/n$ which is at most α and at least $\delta\alpha + (1-\delta)d(1-2\delta)/((1-\delta)n) = \delta\alpha + (1-2\delta)\alpha = (1-\delta)\alpha$. In particular, $\mathcal{T}$ is an α-correct $\mathsf{WAPP}_{\delta}^{\mathsf{dt}}$ decision tree for f.

Lower bound. The $\mathsf{WAPP}_{\delta}^{\mathsf{dt}}$ decision tree designed above is "tight" for f in the following sense: If we decrease the error parameter from δ to ϵ, there is no longer any convex function of $|z|$ that would correspond to the acceptance probability of an α-correct $\mathsf{WAPP}_{\epsilon}^{\mathsf{dt}}$ decision tree for f. This is suggested on the right side of Figure 2.3: only a non-convex function of $|z|$ can satisfy the α-correctness requirements for f. We show that the acceptance probability of any low-cost $\mathsf{WAPP}_{\epsilon}^{\mathsf{dt}}$ decision tree can indeed be accurately approximated by a convex function, which then yields a contradiction.

We now give the details. Suppose for contradiction that $\mathcal{T}$ is a distribution over width-$o(n)$ conjunctions (by Fact 2.33) forming an α-correct $\mathsf{WAPP}_{\epsilon}^{\mathsf{dt}}$ decision tree for f, for some arbitrary $\alpha > 0$. Consider the function $Q\colon \{0,c,d\} \to [0,1]$ defined by $Q(w) \coloneqq \mathbf{E}_{\boldsymbol{z}\,:\,|\boldsymbol{z}|=w}\,\mathrm{acc}_{\mathcal{T}}(\boldsymbol{z})$ where the expectation is over a uniformly random string of Hamming weight w. Note that $Q(0) \in [0,\epsilon\alpha]$ and $Q(w) \in [(1-\epsilon)\alpha, \alpha]$ for $w \in \{c,d\}$ by the correctness of $\mathcal{T}$. A function $R\colon \{0,c,d\} \to \mathbb{R}$ is convex iff $(R(c)-R(0))/c \le (R(d)-R(0))/d$. Note that Q is non-convex since $((1-\epsilon)\alpha - \epsilon\alpha)/c > (\alpha - \epsilon\alpha)/d$. In fact, this shows that there cannot exist a convex function R that point-wise multiplicatively approximates Q within $1 \pm o(1)$. However, we claim that there exists such an R, which provides the desired contradiction.

We now argue the claim. For a width-$o(n)$ conjunction h, let $Q_h\colon \{0,c,d\} \to [0,1]$ be defined by $Q_h(w) \coloneqq \mathbf{Pr}_{\boldsymbol{z}\,:\,|\boldsymbol{z}|=w}[\,h(\boldsymbol{z}) = 1\,]$, and note that $Q = \mathbf{E}_{\boldsymbol{h}\sim\mathcal{T}}\, Q_{\boldsymbol{h}}$. We show that for each such h, Q_h can be multiplicatively approximated by a convex function R_h. Hence Q is multiplicatively approximated by the convex function $R \coloneqq \mathbf{E}_{\boldsymbol{h}\sim\mathcal{T}}\, R_{\boldsymbol{h}}$.

Let $\ell \le o(n)$ denote the number of literals in h, and let i denote the number of positive literals in h. If $i > c$, we have $Q_h(0) = Q_h(c) = 0$ and thus Q_h is convex and we can take $R_h \coloneqq Q_h$. Henceforth suppose $i \le c$. Using the notation $(t)_m$ for the falling factorial $t(t-1)\cdots(t-m+1)$, for $w \in \{c, d\}$ we have $Q_h(w) = \binom{n-\ell}{w-i}/\binom{n}{w} = (w)_i(n-\ell)_{w-i}/(n)_w$.

Suppose $i = 0$. Then $Q_h(0) = 1$, and for $w \in \{c, d\}$ we have $Q_h(w) = (n-\ell)_w/(n)_w \geq (1-o(1))^w \geq 1-o(1)$ (since $\ell \leq o(n)$). Thus we can let R_h be the constant 1 function. Now suppose $1 \leq i \leq c$. Then $Q_h(0) = 0$, and for $w \in \{c, d\}$ we denote the "0 to w slope" as $s_w \coloneqq (Q_h(w) - Q_h(0))/w = (w-1)_{i-1}(n-\ell)_{w-i}/(n)_w$. We have

$$\frac{s_c}{s_d} = \frac{(c-1)_{i-1}}{(d-1)_{i-1}} \cdot \frac{(n-\ell)_{c-i}}{(n-\ell)_{d-i}} \cdot \frac{(n)_d}{(n)_c} = \frac{(c-1)_{i-1}}{(d-1)_{i-1}} \cdot \frac{(n-c)_{d-c}}{(n-\ell-c+i)_{d-c}}.$$

The second multiplicand on the right side is at least 1 and at most $(1+o(1))^{d-c} \leq 1+o(1)$ since $\ell \leq o(n)$. Now we consider two subcases. If $2 \leq i \leq c$ then the first multiplicand on the right side is at most $1-\Omega(1)$ since $c < d$; hence $s_c/s_d \leq 1$ and thus Q_h is convex and we can take $R_h \coloneqq Q_h$. Suppose $i = 1$. Then the first multiplicand on the right side is 1, and hence $s_c/s_d \in 1 \pm o(1)$. This means Q_h is approximately linear. More precisely, defining $R_h(w) \coloneqq s_c \cdot w$, we have $R_h(0) = Q_h(0)$, $R_h(c) = Q_h(c)$, and $R_h(d) = Q_h(d) \cdot s_c/s_d \in Q_h(d) \cdot (1 \pm o(1))$. □

Corollary 2.7. *For all constants $0 < \epsilon < \delta < 1/2$ there exists a partial boolean matrix F such that* $\mathrm{rank}^+_\delta(F) \leq n^{O(1)}$ *but* $\mathrm{rank}^+_\epsilon(F) \geq 2^{\Omega(n)}$.

Proof sketch. Theorem 2.6 together with Theorem 2.9 (proved in the next section) imply that for all $0 < \epsilon < \delta < 1/2$ there is a partial F such that $\mathsf{UWAPP}^{\mathsf{cc}}_\delta(F) \leq O(\log n)$ and $\mathsf{UWAPP}^{\mathsf{cc}}_\epsilon(F) \geq \Omega(n)$. Unfortunately, there is a slight problem with applying Fact 2.31 to conclude a similar separation for rank^+_ϵ as this direct simulation loses a factor of 2 in the error parameter ϵ. This loss results from the following asymmetry between the measures $\mathsf{UWAPP}^{\mathsf{cc}}_\epsilon$ and rank^+_ϵ: the acceptance probabilities of 1-inputs are in $[(1-\epsilon)\alpha, \alpha]$ in the former, whereas 1-entries can be approximated with values in $[1-\epsilon, 1+\epsilon]$ in the latter. However, this annoyance is easily overcome by considering modified versions of $\mathsf{WAPP}^{\mathsf{cc}}_\epsilon$ and $\mathsf{UWAPP}^{\mathsf{cc}}_\epsilon$ where the acceptance probability on 1-inputs is allowed to lie in $[(1-\epsilon)\alpha, (1+\epsilon)\alpha]$. It can be verified that under such a definition Theorem 2.6, Theorem 2.9, and Fact 2.31 continue to hold, and the "new" Fact 2.31 does not lose the factor 2 in the error. □

2.6 Unrestricted–restricted equivalences

In this section we prove our unrestricted–restricted equivalence results, Theorem 2.8 and Theorem 2.9, restated below. In Section 2.6.1 we prove a key "Truncation Lemma", and in Section 2.6.2 we use the lemma to prove the equivalences.

As already alluded to in the introduction, Buhrman et al. [BVdW07] exhibited a function F with $\mathsf{UPostBPP}^{\mathsf{cc}}(F) \leq O(\log n)$ and $\mathsf{PP}^{\mathsf{cc}}(F) \geq \Omega(n^{1/3})$. This simultaneously gives an exponential separation between $\mathsf{PostBPP}^{\mathsf{cc}}$ and $\mathsf{UPostBPP}^{\mathsf{cc}}$ and between $\mathsf{PP}^{\mathsf{cc}}$ and $\mathsf{UPP}^{\mathsf{cc}}$. For our other models, we will show that the unrestricted and restricted versions are essentially equivalent. We state and prove this result only for $\mathsf{SBP}^{\mathsf{cc}}$ and $\mathsf{WAPP}^{\mathsf{cc}}$ as the result for $\mathsf{2WAPP}^{\mathsf{cc}}$ is very similar.

$$\begin{tikzcd}
\mathsf{WAPP}^{\mathsf{cc}} \arrow[r, phantom, "\equiv" description, "\text{Fact 2.28}"] \arrow[d, phantom, "\rotatebox{90}{$\equiv$}" description, "\text{Theorem 2.9, all } F"'] & \mathsf{srec}^1 \arrow[d, phantom, "\rotatebox{90}{$\equiv$}" description, "\text{[KMSY14], total } F"] \\
\mathsf{UWAPP}^{\mathsf{cc}} \arrow[r, phantom, "\equiv" description, "\text{Fact 2.31}"'] & \log \operatorname{rank}^+_\epsilon
\end{tikzcd}$$

Figure 2.4: Summary of equivalences.

Theorem 2.8. $\mathsf{SBP}^{\mathsf{cc}}(F) \leq O(\mathsf{USBP}^{\mathsf{cc}}(F) + \log n)$ *for all* F.

Theorem 2.9. $\mathsf{WAPP}^{\mathsf{cc}}_\delta(F) \leq O(\mathsf{UWAPP}^{\mathsf{cc}}_\epsilon(F) + \log(n/(\delta - \epsilon)))$ *for all* F *and* $0 < \epsilon < \delta < 1/2$.

Hence, roughly speaking, $\mathsf{SBP}^{\mathsf{cc}}$ and $\mathsf{USBP}^{\mathsf{cc}}$ are equivalent and $\mathsf{WAPP}^{\mathsf{cc}}$ and $\mathsf{UWAPP}^{\mathsf{cc}}$ are equivalent. Here "equivalence" is ignoring constant factors and additive logarithmic terms in the cost, but much more significantly it is ignoring constant factors in ϵ (for $\mathsf{WAPP}^{\mathsf{cc}}$), which is important as we know that ϵ cannot be efficiently amplified (Theorem 2.6).

Discussion of Theorem 2.8. The equivalence of $\mathsf{SBP}^{\mathsf{cc}}$ and $\mathsf{USBP}^{\mathsf{cc}}$ implies an alternative proof of the lower bound $\mathsf{USBP}^{\mathsf{cc}}(\text{Disj}) \geq \Omega(n)$ for set-disjointness from [GW14] without using information complexity. Indeed, that paper showed that $\mathsf{SBP}^{\mathsf{cc}}(\text{Disj}) \geq \Omega(n)$ follows from Razborov's corruption lemma [Raz92]. It was also noted in [GW14] that the greater-than function Gt (defined by $\text{Gt}(x, y) := 1$ iff $x > y$ as n-bit numbers) satisfies $\mathsf{USBP}^{\mathsf{cc}}(\text{Gt}) = \Theta(1)$ and $\mathsf{SBP}^{\mathsf{cc}}(\text{Gt}) = \Theta(\log n)$, and thus the $+\log n$ gap in Theorem 2.8 is tight. Our proof of Theorem 2.8 shows, in some concrete sense, that Gt is the "only" advantage $\mathsf{USBP}^{\mathsf{cc}}$ has over $\mathsf{SBP}^{\mathsf{cc}}$. Theorem 2.8 is analogous to, but more complicated than, Proposition 2.37 since both say that without loss of generality α is not too small in the SBP models.

Discussion of Theorem 2.9. The equivalence of $\mathsf{WAPP}^{\mathsf{cc}}$ and $\mathsf{UWAPP}^{\mathsf{cc}}$ implies the equivalence of the smooth rectangle bound (see Fact 2.28 below) and approximate nonnegative rank (see Fact 2.31 below), which was already known for total functions [KMSY14]. Our Theorem 2.9 implies that the equivalence holds even for partial functions, which was crucially used in the proof of Corollary 2.7. The situation is summarized in Figure 2.4.

2.6.1 The Truncation Lemma

The following lemma is a key component in the proofs of Theorem 2.8 and Theorem 2.9.

Definition 2.41. For a nonnegative matrix M, we define its *truncation* $\overline{M}$ to be the same matrix but where each entry > 1 is replaced with 1.

Lemma 2.42 (Truncation Lemma)**.** *For every $2^n \times 2^n$ nonnegative rank-1 matrix M and every d there exists a $O(d + \log n)$-communication public-randomness protocol Π such that for every (x, y) we have $\operatorname{acc}_\Pi(x, y) \in \overline{M}_{x,y} \pm 2^{-d}$.*

We describe some intuition for the proof. We can write $M_{x,y} = u_x v_y$ where $u_x, v_y \geq 0$. First, note that if all entries of M are at most 1, then $\operatorname{acc}_\Pi(x, y) = M_{x,y}$ can be achieved in a zero-communication manner: scaling all u_x's by some factor and scaling all v_y's by the inverse factor, we may assume that all $u_x, v_y \leq 1$; then Alice can accept with probability u_x and Bob can independently accept with probability v_y. Truncation makes all the entries at most 1 but may destroy the rank-1 property. Also note that in general, for the non-truncated entries there may be no "global scaling" for which the zero-communication approach works: there may be some entries with $u_x v_y < 1$ but $u_x > 1$, and other entries with $u_x v_y < 1$ but $v_y > 1$. Roughly speaking, we instead think in terms of "local scaling" that depends on (x, y).

As a starting point, consider a protocol where Alice sends u_x to Bob, who then declares acceptance with probability $\min(u_x v_y, 1)$. We cannot afford to communicate u_x exactly, so we settle for an approximation. We express u_x and v_y in "scientific notation" with an appropriate base and round the mantissa of u_x to have limited precision. The exponent of u_x, however, may be too expensive to communicate, but since u_x, v_y are multiplied, all that matters is the sum of their exponents. Determining the sum of the exponents exactly may be too expensive, but the crux of the argument is that we only need to consider a limited number of cases. If the sum of the exponents is small, then the matrix entry is very close to 0 and we can reject without knowing the exact sum. If the sum of the exponents is large, then the matrix entry is guaranteed to be truncated and we can accept. Provided the base is large enough, there are only a few "inbetween" cases. Determining which case holds can be reduced to a greater-than problem, which can be solved with error exponentially small in d using communication $O(d + \log n)$.

We now give the formal proof.

Proof of Lemma 2.42. Let $M_{x,y} = u_x v_y$ where $u_x, v_y \geq 0$, and define $\delta \coloneqq 2^{-d}/2$ and $B \coloneqq 1/\delta$.

Henceforth we fix an input (x, y). For convenience we let all notation be relative to (x, y), so we start by defining $u \coloneqq u_x$ and $v \coloneqq v_y$, and note that $\overline{M}_{x,y} = \min(uv, 1)$. Assuming $u > 0$, define $i \coloneqq \lceil \log_B u \rceil$ (so $u \in (B^{i-1}, B^i]$) and $a \coloneqq u/B^i$ (so $a \in (\delta, 1]$). Similarly, assuming $v > 0$, define $j \coloneqq \lceil \log_B v \rceil$ (so $v \in (B^{j-1}, B^j]$) and $b \coloneqq v/B^j$ (so $b \in (\delta, 1]$). Note that $uv = abB^{i+j} \in (B^{i+j-2}, B^{i+j}]$. The protocol Π is as follows. (Line 4 is underspecified but we will address that later.)

```
1: If $u = 0$ or $v = 0$ then reject
2: Alice sends Bob $\widetilde{a} \in a \pm \delta^2$ (and ensuring $\widetilde{a} \leq 1$) using $O(d)$ bits
3: Bob computes $p \coloneqq \widetilde{a} \cdot b$
4: Determine with probability at least $1 - \delta$ which of the following four cases holds:
5: If $i + j < 0$ then reject
6: If $i + j = 0$ then accept with probability $p$
7: If $i + j = 1$ then accept with probability $\min(pB, 1)$
8: If $i + j > 1$ then accept
```

We first argue correctness. Assume $u, v > 0$. We have $ab \in (\widetilde{a} \pm \delta^2)b \subseteq p \pm \delta^2$ (using $b \leq 1$) and thus $uv \in (p \pm \delta^2)B^{i+j}$. Pretending for the moment that line 4 succeeds with probability 1, we can verify that in all four cases the acceptance probability would be $\in \overline{M}_{x,y} \pm \delta$:

5: If $i + j < 0$ then $0 \in \overline{M}_{x,y} \pm \delta$ since $uv \leq B^{i+j} \leq \delta$.
6: If $i + j = 0$ then $p \in \overline{M}_{x,y} \pm \delta$ since $uv \in (p \pm \delta^2)B^{i+j} \subseteq p \pm \delta$.
7: If $i + j = 1$ then $\min(pB, 1) \in \overline{M}_{x,y} \pm \delta$ since $uv \in (p \pm \delta^2)B^{i+j} \subseteq pB \pm \delta$.
8: If $i + j > 1$ then $1 = \overline{M}_{x,y}$ since $uv > B^{i+j-2} \geq 1$.

The error probability of line 4 only affects the overall acceptance probability by $\pm\delta$, so $\mathrm{acc}_\Pi(x, y) \in \overline{M}_{x,y} \pm 2\delta \subseteq \overline{M}_{x,y} \pm 2^{-d}$.

The communication cost is $O(d)$ except for line 4. Line 4 can be implemented with three tests: $i + j \geq 0$, $i + j \geq 1$, $i + j \geq 2$, each having error probability $\delta/3$. These tests are implemented in the same way as each other, so we just describe how to test whether $i + j \geq 0$. In other words, if we let T denote the indicator matrix for $i + j \geq 0$, then we want to compute T with error probability $\delta/3$ and communication $O(d + \log n)$. If we assume the rows are sorted in decreasing order of u and the columns are sorted in decreasing order of v, then each row and each column of T consists of 1's followed by 0's. To compute T, we may assume without loss of generality it has no duplicate rows and no duplicate columns, in which case it is a greater-than matrix (of size at most $2^n \times 2^n$) with the 1's in the upper-left triangle, possibly with the all-0 row deleted and/or the all-0 column deleted. The greater-than function can be computed with any error probability $\gamma > 0$ and communication $O(\log(n/\gamma))$ by running the standard protocol [KN97, p. 170] for $O(\log(n/\gamma))$ many steps. □

Remark 4. We note that the $O(d + \log n)$ communication bound in Lemma 2.42 is optimal, assuming $n \geq d$. Indeed, define a nonnegative rank-1 matrix M by $M_{x,y} \coloneqq (2^{-d})^{x-y}$ where x and y are viewed as nonnegative n-bit integers. Consider any protocol Π with $\mathrm{acc}_\Pi(x, y) \in \overline{M}_{x,y} \pm 2^{-d}$, and note that it determines with error probability $2^{-(d-1)}$ whether $x \leq y$. The latter is known to require $\Omega(\log n)$ communication (even for constant d) [Vio15]. Also, by a union bound there exists an outcome of the randomness for which Π determines whether $x \leq y$ for all pairs $x, y < 2^{d/2-1}$ (of which there are 2^{d-2}), which requires $\Omega(d)$ communication by the deterministic lower bound for greater-than on $(d/2 - 1)$-bit integers.

2.6.2 Proofs of unrestricted–restricted equivalences

We now give the (very similar) proofs of Theorem 2.8 and Theorem 2.9 using the Truncation Lemma. We make use of the following basic fact.

Fact 2.43. *Given a private-randomness protocol* Π *of communication cost* c*, label the accepting transcripts as* $\tau \in \{1, 2, \ldots, 2^c\}$*. Then for each accepting transcript* τ *there exists a nonnegative rank-*1 *matrix* N^τ *such that the following holds. For each* (x, y)*, the probability of getting transcript* τ *on input* (x, y) *is* $N^\tau_{x,y}$*, and thus* $\mathrm{acc}_\Pi(x, y) = \sum_{\tau=1}^{2^c} N^\tau_{x,y}$.

For both proofs, the goal is to show that any protocol (of type $\mathsf{USBP^{cc}}$ or $\mathsf{UWAPP}^{\mathsf{cc}}_\epsilon$) can be converted into another protocol (of type $\mathsf{SBP^{cc}}$ or $\mathsf{WAPP}^{\mathsf{cc}}_\delta$, respectively) of comparable cost. We transform an α-correct protocol of cost c, where α might be prohibitively small, into a (roughly) 2^{-c}-correct protocol without increasing the communication by too much. We use Fact 2.43 to express the acceptance probabilities as a sum of nonnegative rank-1 matrices. The basic intuition is to divide everything by α to get a “1-correct” matrix sum; however, this new sum may not correspond to acceptance probabilities of a protocol. To achieve the latter, we truncate each summand (which does not hurt the correctness, and which makes each summand correspond to acceptance probabilities from the Truncation Lemma), then multiply each summand by 2^{-c} (which essentially changes the correctness parameter from 1 to 2^{-c}, and which corresponds to picking a uniformly random summand).

Proof of Theorem 2.8. Fix a cost-c $\mathsf{USBP^{cc}}$ protocol Π for F with associated $\alpha > 0$ and associated matrices N^τ from Fact 2.43. Thus $\sum_\tau N^\tau_{x,y}$ is $\geq \alpha$ if $F(x, y) = 1$ and $\leq \alpha/2$ if $F(x, y) = 0$. We claim that the following public-randomness protocol Π' witnesses $\mathsf{SBP^{cc}}(F) \leq O(c + \log n)$:

```
Pick $\tau \in \{1, 2, \ldots, 2^c\}$ uniformly at random
Run the protocol from Lemma 2.42 with $M^\tau := \frac{1}{\alpha} N^\tau$ and $d := c + 3$
```

We first argue correctness. We have $\mathrm{acc}_{\Pi'}(x, y) \in 2^{-c} \sum_\tau \big(\overline{M}^\tau_{x,y} \pm 2^{-d}\big) = 2^{-c}\big(\sum_\tau \overline{M}^\tau_{x,y} \pm 2^{-3}\big)$. If $F(x, y) = 0$ then $\sum_\tau \overline{M}^\tau_{x,y} \leq \sum_\tau \frac{1}{\alpha} N^\tau_{x,y} \leq 1/2$ and thus $\mathrm{acc}_{\Pi'}(x, y) \leq (5/8)2^{-c}$. Now suppose $F(x, y) = 1$. If $M^\tau_{x,y} \leq 1$ for all τ then $\sum_\tau \overline{M}^\tau_{x,y} = \sum_\tau \frac{1}{\alpha} N^\tau_{x,y} \geq 1$, and if not then we also have $\sum_\tau \overline{M}^\tau_{x,y} \geq \max_\tau \overline{M}^\tau_{x,y} = 1$. In either case, $\mathrm{acc}_{\Pi'}(x, y) \geq (7/8)2^{-c}$. Since there is a constant factor gap between the acceptance probabilities on 1-inputs and 0-inputs, we can use and-amplification in a standard way [GW14] to bring the gap to a factor of 2 while increasing the cost by only a constant factor. Since the communication cost of Π' is $O(d + \log n) = O(c + \log n)$, and the associated α' value is $2^{-O(c)}$, the overall cost is $O(c + \log n)$. □

Proof of Theorem 2.9. Fix a cost-c $\mathsf{UWAPP}^{\mathsf{cc}}_\epsilon$ protocol Π for F with associated $\alpha > 0$ and associated matrices N^τ from Fact 2.43. Thus $\sum_\tau N^\tau_{x,y}$ is $\in [(1 - \epsilon)\alpha, \alpha]$ if $F(x, y) = 1$ and $\in [0, \epsilon\alpha]$ if $F(x, y) = 0$. We claim that the following public-randomness protocol Π' witnesses $\mathsf{WAPP}^{\mathsf{cc}}_\delta(F) \leq O(c + \log(n/\Delta))$ where $\Delta := (\delta - \epsilon)/2$:

```
Pick $\tau \in \{1, 2, \dots, 2^c\}$ uniformly at random
Run the protocol from Lemma 2.42 with $M^\tau := \frac{1}{\alpha} N^\tau$ and $d := c + \lceil \log(1/\Delta) \rceil$
```

We first argue correctness. We have $\mathrm{acc}_{\Pi'}(x, y) \in 2^{-c} \sum_\tau \big(\overline{M}^\tau_{x,y} \pm 2^{-d}\big) \subseteq 2^{-c}\big(\sum_\tau \overline{M}^\tau_{x,y} \pm \Delta\big)$. Define $\alpha' := 2^{-c}(1 + \Delta)$. If $F(x, y) = 0$ then $\sum_\tau \overline{M}^\tau_{x,y} \leq \sum_\tau \frac{1}{\alpha} N^\tau_{x,y} \leq \epsilon$ and thus $\mathrm{acc}_{\Pi'}(x, y) \in [0, 2^{-c}(\epsilon + \Delta)] \subseteq [0, \delta\alpha']$. Now suppose $F(x, y) = 1$. Then $M^\tau_{x,y} \leq 1$ for all τ (otherwise $\mathrm{acc}_\Pi(x, y) = \sum_\tau \alpha M^\tau_{x,y} > \alpha$). Hence $\sum_\tau \overline{M}^\tau_{x,y} = \sum_\tau \frac{1}{\alpha} N^\tau_{x,y} \in [1 - \epsilon, 1]$, and thus $\mathrm{acc}_{\Pi'}(x, y) \in [2^{-c}(1 - \epsilon - \Delta), 2^{-c}(1 + \Delta)] \subseteq [(1 - \delta)\alpha', \alpha']$. So Π' is a $\mathsf{WAPP}^{\mathsf{cc}}_\delta$ protocol for F of cost $O(d + \log n) + \log(1/\alpha') \leq O(c + \log(n/\Delta))$. □

Remark 5. In the proof of Theorem 2.9, note that if F is total then Lemma 2.42 is not needed: The entries of each M^τ are all bounded by 1, and thus $M^\tau_{x,y}$ can be written as $u_x v_y$ where $u_x, v_y \in [0, 1]$. Hence to accept with probability $M^\tau_{x,y}$, Alice can accept with probability u_x and Bob can accept with probability v_y. This incurs no loss in the ϵ parameter and has communication cost 2, witnessing that $\mathsf{WAPP}^{\mathsf{cc}}_\epsilon(F) \leq \mathsf{UWAPP}^{\mathsf{cc}}_\epsilon(F) + 2$ if F is total.

2.7 Additional proofs

2.7.1 Proof of Fact 2.28

$\mathsf{srec}^1_\epsilon(F)$ is defined as the log of the optimum value of the following linear program, which has a variable w_R for each rectangle R.

$$\begin{array}{rll} \text{minimize} & \sum_R w_R & \\ \text{subject to} & \sum_{R\,:\,(x,y)\in R} w_R \in [1-\epsilon, 1] & \forall (x,y) \in F^{-1}(1) \\ & \sum_{R\,:\,(x,y)\in R} w_R \in [0, \epsilon] & \forall (x,y) \in F^{-1}(0) \\ & w_R \geq 0 & \forall R \end{array}$$

We first show the first inequality. Given a cost-c $\mathsf{WAPP}^{\mathsf{cc}}_\epsilon$ protocol for F, put it in the "normal form" given by Fact 2.27 so that $\alpha = 2^{-c}$ and each outcome of the randomness is a rectangle. For each rectangle R, let $w_R := p_R/\alpha$ where p_R is the probability of R in the normal form protocol. This is a feasible solution with objective value $1/\alpha$, so $\mathsf{srec}^1_\epsilon(F) \leq \log(1/\alpha) = c$. We now show the second inequality. Given an optimal solution, let $\alpha := 1/\sum_R w_R$ and consider a protocol that selects rectangle R with probability αw_R. This is an α-correct $\mathsf{WAPP}^{\mathsf{cc}}_\epsilon$ protocol for F of cost $2 + \mathsf{srec}^1_\epsilon(F)$.

2.7.2 Proof of Fact 2.31

We first show the first inequality. Fix a cost-c $\mathsf{UWAPP}^{\mathsf{cc}}_\epsilon$ protocol Π for F with associated $\alpha > 0$ and associated matrices N^τ from Fact 2.43. Thus $\sum_\tau N^\tau_{x,y}$ is $\in [(1 - \epsilon)\alpha, \alpha]$ if $F(x, y) = 1$

and $\in [0, \epsilon\alpha]$ if $F(x, y) = 0$. Hence letting $M \coloneqq \sum_\tau \frac{1}{\alpha} N^\tau$, we have $M_{x,y} \in F(x, y) \pm \epsilon$ for all $(x, y) \in \operatorname{dom} F$ and $\operatorname{rank}^+(M) \leq 2^c$.

We now show the second inequality. Suppose M is such that $M_{x,y} \in F(x, y) \pm \epsilon/2$ for all $(x, y) \in \operatorname{dom} F$ and $r \coloneqq \operatorname{rank}^+(M)$ is witnessed by $M = UV$, and let t be the maximum entry in U, V. We claim that the following private-randomness protocol Π witnesses $\mathsf{UWAPP}^{\mathsf{cc}}_\epsilon(F) \leq \lceil \log r \rceil + 2$:

```
Alice picks $i \in \{1, 2, \ldots, r\}$ uniformly at random and sends it to Bob
Alice accepts with probability $U_{x,i}/t$ and sends her decision to Bob
Bob accepts with probability $V_{i,y}/t$ and sends his decision to Alice
Accept iff both Alice and Bob accept
```

We have $\operatorname{acc}_\Pi(x, y) = \frac{1}{r} \sum_i U_{x,i} V_{i,y}/t^2 = M_{x,y}/rt^2$. Let $\alpha \coloneqq (1 + \epsilon/2)/rt^2$. If $F(x, y) = 1$ then $\operatorname{acc}_\Pi(x, y) \in [(1 - \epsilon/2)/rt^2, (1 + \epsilon/2)/rt^2] \subseteq [(1 - \epsilon)\alpha, \alpha]$. If $F(x, y) = 0$ then $\operatorname{acc}_\Pi(x, y) \in [0, (\epsilon/2)/rt^2] \subseteq [0, \epsilon\alpha]$. Thus the protocol is correct with respect to α.

2.7.3 Proof of Fact 2.40

We use the notation $(t)_m$ for the falling factorial $t(t - 1) \cdots (t - m + 1)$. The acceptance probability is

$$\frac{\binom{n-d}{w-i}}{\binom{n}{w}} = \frac{(n - d)_{w-i}}{(w - i)!} \cdot \frac{w!}{(n)_w} = \frac{(w)_i}{(n)_w \,/\, (n - d)_{w-i}}.$$

We claim that

(i) $w^i \cdot (1 - o(1)) \leq (w)_i \leq w^i$,
(ii) $n^w \cdot (1 - o(1)) \leq (n)_w \leq n^w$,
(iii) $n^{w-i} \cdot (1 - o(1)) \leq (n - d)_{w-i} \leq n^{w-i}$.

Then the acceptance probability is in

$$\frac{w^i}{n^w \,/\, n^{w-i}} \cdot (1 \pm o(1)) = (w/n)^i \cdot (1 \pm o(1)).$$

The three upper bounds are trivial. For the lower bound in (i), we have

$$\begin{aligned} (w)_i &= w^i \cdot (1 - \tfrac{0}{w})(1 - \tfrac{1}{w}) \cdots (1 - \tfrac{i-1}{w}) \\ &\geq w^i \cdot 4^{-0/w} 4^{-1/w} \cdots 4^{-(i-1)/w} \\ &= w^i \cdot 4^{-i(i-1)/2w} \\ &\geq w^i \cdot (1 - o(1)) \end{aligned}$$

since $i \leq d \leq o(\sqrt{w})$. The lower bound in (ii) follows similarly using $w \leq o(\sqrt{n})$. For (iii), we have

$$(n-d)_{w-i} \;\geq\; (n-d)^{w-i} \cdot (1-o(1)) \;=\; n^{w-i} \cdot (1-o(1)) \cdot (1-d/n)^{w-i}$$

as above using $w - i \leq o(\sqrt{n-d})$, and we have $(1-d/n)^{w-i} \geq (4^{-d/n})^w \geq 1 - o(1)$ since $d < w \leq o(\sqrt{n})$.

2.7.4 Proof of Fact 2.39

We first prove *(i)* ⇒ *(ii)*. Assume *(i)*, and consider μ_1 distributed over $f^{-1}(1)$ and μ_0 distributed over $f^{-1}(0)$. We have for $\boldsymbol{h} \sim \mathcal{H}$ and $\boldsymbol{z_i} \sim \mu_i$ that

$$\begin{aligned}
\mathbf{E}_{\boldsymbol{h}}\, \mathbf{Pr}_{\boldsymbol{z_1}}[\, \boldsymbol{h}(\boldsymbol{z_1}) = 1\,] \;&=\; \mathbf{Pr}_{\boldsymbol{h},\boldsymbol{z_1}}[\, \boldsymbol{h}(\boldsymbol{z_1}) = 1\,] \\
&\geq\; \min\nolimits_{z_1 \in f^{-1}(1)} \mathbf{Pr}_{\boldsymbol{h}}[\, \boldsymbol{h}(z_1) = 1\,] \\
&>\; 2 \cdot \max\nolimits_{z_0 \in f^{-1}(0)} \mathbf{Pr}_{\boldsymbol{h}}[\, \boldsymbol{h}(z_0) = 1\,] \\
&\geq\; 2 \cdot \mathbf{Pr}_{\boldsymbol{h},\boldsymbol{z_0}}[\, \boldsymbol{h}(\boldsymbol{z_0}) = 1\,] \\
&=\; \mathbf{E}_{\boldsymbol{h}}\;\, 2 \cdot \mathbf{Pr}_{\boldsymbol{z_0}}[\, \boldsymbol{h}(\boldsymbol{z_0}) = 1\,].
\end{aligned}$$

If $\mathbf{Pr}_{\boldsymbol{z_1}}[\, h(\boldsymbol{z_1}) = 1\,] \leq 2 \cdot \mathbf{Pr}_{\boldsymbol{z_0}}[\, h(\boldsymbol{z_0}) = 1\,]$ for all h, then the above would be false.

We now prove *(ii)* ⇒ *(i)*. Assume *(ii)*, and define α_{μ_1,μ_0} to be the maximum of $\mathbf{Pr}_{\boldsymbol{z_1} \sim \mu_1}[\, h(\boldsymbol{z_1}) = 1\,]$ over all h such that $\mathbf{Pr}_{\boldsymbol{z_1} \sim \mu_1}[\, h(\boldsymbol{z_1}) = 1\,] > 2 \cdot \mathbf{Pr}_{\boldsymbol{z_0} \sim \mu_0}[\, h(\boldsymbol{z_0}) = 1\,]$. It is not difficult to see that the function $(\mu_1, \mu_0) \mapsto \alpha_{\mu_1,\mu_0}$ is lower semi-continuous, since if we change (μ_1, μ_0) infinitesimally then $\mathbf{Pr}_{\boldsymbol{z_1} \sim \mu_1}[\, h(\boldsymbol{z_1}) = 1\,] > 2 \cdot \mathbf{Pr}_{\boldsymbol{z_0} \sim \mu_0}[\, h(\boldsymbol{z_0}) = 1\,]$ still holds for the (previously) optimum h, and the left side of the inequality only changes infinitesimally (but another h may become "available" and raise the value of α_{μ_1,μ_0}, hence the function is not upper semi-continuous). It is a basic fact of analysis that a lower semi-continuous function on a compact set attains its infimum. Since the set of (μ_1, μ_0) pairs is compact, and since $\alpha_{\mu_1,\mu_0} > 0$ for all (μ_1, μ_0), we have $\inf_{\mu_1,\mu_0} \alpha_{\mu_1,\mu_0} > 0$. Let α^* be any real such that $0 < \alpha^* < \inf_{\mu_1,\mu_0} \alpha_{\mu_1,\mu_0}$. Hence we have $\alpha_{\mu_1,\mu_0} > \alpha^*$ for all (μ_1, μ_0).

Let M be the matrix with rows indexed by Z and columns indexed by $\mathscr{H}$, such that $M_{z,h} := h(z)$. Then for every (μ_1, μ_0) there exists an h such that $\mathbf{E}_{\boldsymbol{z_1} \sim \mu_1} M_{\boldsymbol{z_1},h} > \alpha^*$ and $\mathbf{E}_{\boldsymbol{z_1} \sim \mu_1} M_{\boldsymbol{z_1},h} > 2 \cdot \mathbf{E}_{\boldsymbol{z_0} \sim \mu_0} M_{\boldsymbol{z_0},h}$. Let M' be the matrix with rows indexed by Z and (infinitely-many) columns indexed by $\mathscr{H} \times [0,1]$, such that $M'_{z,(h,s)} := s \cdot h(z)$. Then for every (μ_1, μ_0) there exists a (h, s) such that $\mathbf{E}_{\boldsymbol{z_1} \sim \mu_1} M'_{\boldsymbol{z_1},(h,s)} > \alpha^*$ and $\mathbf{E}_{\boldsymbol{z_0} \sim \mu_0} M'_{\boldsymbol{z_0},(h,s)} < \alpha^*/2$ (by choosing s to be slightly greater than $\alpha^* / \mathbf{E}_{\boldsymbol{z_1} \sim \mu_1} M_{\boldsymbol{z_1},h}$). Let $A \colon \mathbb{R} \to \mathbb{R}$ be the affine transformation $A(x) := (1-x) \cdot \frac{\alpha^*}{1-\alpha^*/2}$. Let M'' be the matrix indexed like M', such that $M''_{z,(h,s)} := M'_{z,(h,s)}$ if $f(z) = 1$, and $M''_{z,(h,s)} := A\big(M'_{z,(h,s)}\big)$ if $f(z) = 0$. Then for every (μ_1, μ_0) there exists a (h, s) such that $\mathbf{E}_{\boldsymbol{z_1} \sim \mu_1} M''_{\boldsymbol{z_1},(h,s)} > \alpha^*$ and, by linearity of expectation, $\mathbf{E}_{\boldsymbol{z_0} \sim \mu_0} M''_{\boldsymbol{z_0},(h,s)} = A\big(\mathbf{E}_{\boldsymbol{z_0} \sim \mu_0} M'_{\boldsymbol{z_0},(h,s)}\big) > \big(1 - \alpha^*/2\big) \cdot \frac{\alpha^*}{1-\alpha^*/2} = \alpha^*$.

We claim that for every distribution μ over Z there exists a (h, s) such that $\mathbf{E}_{\boldsymbol{z} \sim \mu} M''_{\boldsymbol{z},(h,s)} > \alpha^*$.

If $\mu(f^{-1}(1)) > 0$ and $\mu(f^{-1}(0)) > 0$ then this follows from the above using $\mu_1 = (\mu \mid f^{-1}(1))$ and $\mu_0 = (\mu \mid f^{-1}(0))$. Otherwise if, say, $\mu(f^{-1}(0)) = 0$ (similarly if $\mu(f^{-1}(1)) = 0$) then we can let $\mu_1 = \mu$ and μ_0 be an arbitrary distribution over $f^{-1}(0)$, and apply the above.

Now by the minimax theorem (a continuous version as used in [TTV09]) the two-player zero-sum game given by M'' (with payoffs to the column player) has value $> \alpha^*$, and thus there exists a distribution $\mathcal{H}'$ over $\mathscr{H} \times [0,1]$ such that for all $z \in Z$, $\mathbf{E}_{(\boldsymbol{h},\boldsymbol{s})\sim\mathcal{H}'}\, M''_{z,(\boldsymbol{h},\boldsymbol{s})} > \alpha^*$. Thus for all $z_1 \in f^{-1}(1)$ we have $\mathbf{E}_{(\boldsymbol{h},\boldsymbol{s})\sim\mathcal{H}'}\, M'_{z_1,(\boldsymbol{h},\boldsymbol{s})} > \alpha^*$, and for all $z_0 \in f^{-1}(0)$ by linearity of expectation we have $\mathbf{E}_{(\boldsymbol{h},\boldsymbol{s})\sim\mathcal{H}'}\, M'_{z_0,(\boldsymbol{h},\boldsymbol{s})} = A^{-1}\big(\mathbf{E}_{(\boldsymbol{h},\boldsymbol{s})\sim\mathcal{H}'}\, M''_{z_0,(\boldsymbol{h},\boldsymbol{s})}\big) < 1 - \alpha^* \cdot \frac{1-\alpha^*/2}{\alpha^*} = \alpha^*/2$.

For $h \in \mathscr{H}$, if we define p_h to be the expectation under $\mathcal{H}'$ of the function that outputs s on inputs (h, s) and outputs 0 otherwise, then for all z we have $\mathbf{E}_{(\boldsymbol{h},\boldsymbol{s})\sim\mathcal{H}'}\, M'_{z,(\boldsymbol{h},\boldsymbol{s})} = \sum_h p_h \cdot M_{z,h}$. Finally, we define the distribution $\mathcal{H}$ over $\mathscr{H}$ so the probability of h is p_h/P where $P \coloneqq \sum_h p_h$. Then for all z we have $\mathbf{Pr}_{\boldsymbol{h}\sim\mathcal{H}}[\,\boldsymbol{h}(z) = 1\,] = \frac{1}{P} \cdot \mathbf{E}_{(\boldsymbol{h},\boldsymbol{s})\sim\mathcal{H}'}\, M'_{z,(\boldsymbol{h},\boldsymbol{s})}$. Thus for all $z_1 \in f^{-1}(1)$ we have $\mathbf{Pr}_{\boldsymbol{h}\sim\mathcal{H}}[\,\boldsymbol{h}(z_1) = 1\,] > \alpha^*/P$, and for all $z_0 \in f^{-1}(0)$ we have $\mathbf{Pr}_{\boldsymbol{h}\sim\mathcal{H}}[\,\boldsymbol{h}(z_0) = 1\,] < \alpha^*/2P$, and hence *(i)* holds.

Chapter 3

Lower Bounds for Clique vs. Independent Set

Overview. In this chapter we prove an $\omega(\log n)$ lower bound on the conondeterministic communication complexity of the Clique vs. Independent Set problem introduced by Yannakakis [Yan91]. As a corollary, this implies superpolynomial lower bounds for the Alon–Saks–Seymour conjecture in graph theory. Our approach is to first exhibit a query complexity separation for the decision tree analogue of the UP vs. coNP question—namely, unambiguous DNF width vs. CNF width—and then "lift" this separation over to communication complexity using the simulation theorem from Chapter 2. This chapter is based on the following publication:

[Göö15]: Mika Göös. Lower bounds for clique vs. independent set. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, pages 1066–1076. IEEE, 2015. doi:10.1109/FOCS.2015.69

3.1 Introduction

Yannakakis's [Yan91] *Clique vs. Independent Set* problem, associated with an undirected n-node graph $G = ([n], E)$, is the following: Alice holds a clique $x \subseteq [n]$ in G, Bob holds an independent set $y \subseteq [n]$ in G, and their goal is to decide whether x and y intersect. As the underlying graph enforces $|x \cap y| \in \{0, 1\}$, we may define a boolean function by $\mathsf{CIS}_G(x, y) := |x \cap y|$.

Upper bounds. For every G there is an $\lceil \log n \rceil$-bit nondeterministic communication protocol for CIS_G that guesses the name of the unique node in the intersection $x \cap y$. Recall (e.g., [KN97, Juk12]) that, combinatorially, this means that the 1-entries of the communication matrix of CIS_G can be covered with n rectangles. We write this fact as $\mathsf{NP}^{\mathsf{cc}}(\mathsf{CIS}_G) \leq \lceil \log n \rceil$. Yannakakis further proved that $\mathsf{P}^{\mathsf{cc}}(\mathsf{CIS}_G) \leq O(\log^2 n)$, where P^{cc} stands for deterministic communication

Measure	Lower bound	Reference
P^{cc}	$2 \cdot \log n$	Kushilevitz, Linial, and Ostrovsky [KLO99]
$\mathsf{coNP}^{\mathsf{cc}}$	$6/5 \cdot \log n$	Huang and Sudakov [HS12]
$\mathsf{coNP}^{\mathsf{cc}}$	$3/2 \cdot \log n$	Amano [Ama14b]
$\mathsf{coNP}^{\mathsf{cc}}$	$2 \cdot \log n$	Shigeta and Amano [SA15]
$\mathsf{coNP}^{\mathsf{cc}}$	$\omega(\log n)$	This chapter

Table 3.1: Lower bounds on the deterministic (P^{cc}) and conondeterministic ($\mathsf{coNP}^{\mathsf{cc}}$) communication complexities of the Clique vs. Independent Set problem.

complexity. In particular, we have the same upper bound on the conondeterministic complexity:

$$\mathsf{coNP}^{\mathsf{cc}}(\mathsf{CIS}_G) \;\leq\; O(\log^2 n).$$

Lower bounds. It has been a relatively long-standing open problem to prove (for some choice of G) superlogarithmic lower bounds on $\mathsf{P}^{\mathsf{cc}}(\mathsf{CIS}_G)$, let alone on $\mathsf{coNP}^{\mathsf{cc}}(\mathsf{CIS}_G)$. See, for instance, the textbooks by Kushilevitz and Nisan [KN97, Exercise 1.8] and Jukna [Juk12, Research Problem 4.15]. See also Table 3.1 for a summary of previous bounds, and the works of Kushilevitz and Weinreb [KW09a, KW09b] for indirect attacks on the problem.

Our main result is to obtain such superlogarithmic lower bounds.

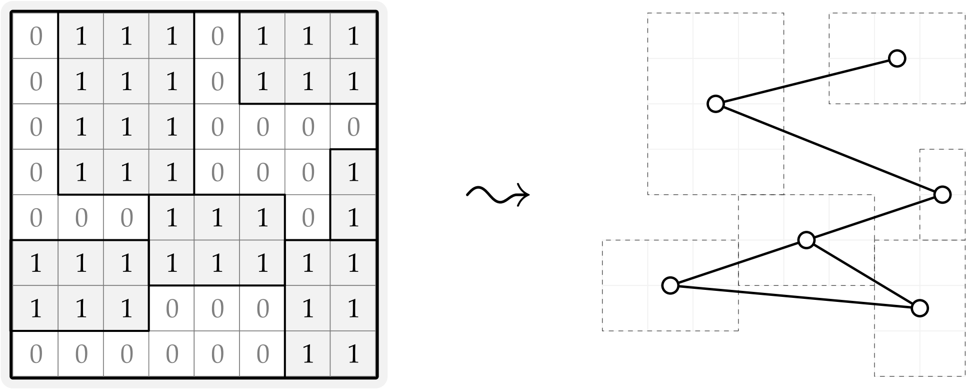
Theorem 3.1. *There is a family of graphs G such that*

$$\mathsf{coNP}^{\mathsf{cc}}(\mathsf{CIS}_G) \;\geq\; \Omega(\log^{1.128} n).$$

Implications. The CIS problem admits several equivalent formulations, as explored in-depth by Bousquet, Lagoutte, and Thomassé [BLT14]. In particular, Theorem 3.1 refutes a certain "polynomial" version of the Alon–Saks–Seymour conjecture [BLT14, Conjecture 16]. The original conjecture stated that $\chi(G) \leq \mathrm{bp}(G) + 1$, that is, that the chromatic number of G can be bounded in terms of the *biclique partition number* of G (minimum number of complete bipartite graphs needed to partition the edges of G). Huang and Sudakov [HS12] disproved the original conjecture by showing that $\chi(G)$ can be polynomially larger than $\mathrm{bp}(G)$. Theorem 3.1 implies that the gap can be superpolynomial. See [BLT14] for more details about this and other connections.

3.1.1 Our approach

Unambiguity. The canonical $\lceil \log n \rceil$-cost nondeterministic protocol for CIS_G outlined above has the additional property of being *unambiguous*: on each input there is at most one accepting nondeterministic computation. That is, combinatorially, the rectangles covering the 1-entries do not overlap. Thus $\lceil \log n \rceil$ is an upper bound on the unambiguous communication complexity



0	1	1	1	0	1	1	1
0	1	1	1	0	1	1	1
0	1	1	1	0	0	0	0
0	1	1	1	0	0	0	1
0	0	0	1	1	1	0	1
1	1	1	1	1	1	1	1
1	1	1	0	0	0	1	1
0	0	0	0	0	0	1	1

Partition of $F^{-1}(1)$

Graph G

Figure 3.1: Reduction $F \leq \mathsf{CIS}_G$ [Yan91, Lemma 1]: Fix an optimal partition of the 1-inputs of F using $2^{\mathsf{UP}^{\mathsf{cc}}(F)}$ rectangles. The nodes of the graph G are the rectangles, and two nodes are connected by an edge if the corresponding rectangles share a row. We have $F \leq \mathsf{CIS}_G$ via the following map: Alice maps her input x to the set of all rectangles intersecting row x, and Bob maps his input y to the set of all rectangles intersecting column y.

of CIS_G, which we write as $\mathsf{UP}^{\mathsf{cc}}(\mathsf{CIS}_G) \leq \lceil \log n \rceil$. In fact, it is known that the CIS_G family of problems is *complete* for unambiguous communication: for every two-party function $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ there is a graph G on $n = 2^{\mathsf{UP}^{\mathsf{cc}}(F)}$ nodes such that F appears as a subproblem of CIS_G. See Figure 3.1 for an illustration. In particular, we have $\mathsf{UP}^{\mathsf{cc}}(F) = \mathsf{UP}^{\mathsf{cc}}(\mathsf{CIS}_G) = \log n$. Given this structural perspective, our goal becomes to exhibit a total[1] function with a large $\mathsf{UP}^{\mathsf{cc}}$ vs. $\mathsf{coNP}^{\mathsf{cc}}$ gap.

The following is a rephrasing of Theorem 3.1.

Theorem 3.2. *There is an F such that $\mathsf{coNP}^{\mathsf{cc}}(F) \geq \mathsf{UP}^{\mathsf{cc}}(F)^{\beta}$ where $\beta > 1.128$ is a constant.*

Query complexity. Instead of attacking Theorem 3.2 head-on, we first show an analogous separation in the simpler-to-understand world of decision tree complexity [BdW02]. Here one deals with plain boolean functions $f\colon \{0,1\}^n \to \{0,1\}$ (different n than above) without any two-party structure. We use the superscript "dt" for query complexity measures. For example, $\mathsf{NP}^{\mathsf{dt}}(f)$ denotes the nondeterministic decision tree complexity of f, or equivalently, the minimum k such that f can be written as a k-DNF. We also set $\mathsf{coNP}^{\mathsf{dt}}(f) := \mathsf{NP}^{\mathsf{dt}}(\neg f)$. A DNF is *unambiguous* if on every input at most one of its terms (a.k.a. certificates) evaluates to true. We define $\mathsf{UP}^{\mathsf{dt}}(f)$ as the minimum k such that f can be written as an unambiguous k-DNF.

The following is the query analogue of Theorem 3.2.

Theorem 3.3. *There is an f such that $\mathsf{coNP}^{\mathsf{dt}}(f) \geq \mathsf{UP}^{\mathsf{dt}}(f)^{\alpha}$ where $\alpha > 1.128$ is a constant.*

[1] It is easy to give a partial function (promise problem) with an exponential $\mathsf{UP}^{\mathsf{cc}}$ vs. $\mathsf{coNP}^{\mathsf{cc}}$ gap: the *unique set-intersection* function UINTER satisfies $\mathsf{UP}^{\mathsf{cc}}(\text{UINTER}) \leq O(\log n)$ but $\mathsf{coNP}^{\mathsf{cc}}(\text{UINTER}) \geq \Omega(n)$ [Raz92, KW14].

From query to communication. Given an f as above, we then apply the simulation theorem from Chapter 2 that allows us to convert f into a communication problem $f \circ g^n$ while preserving its conondeterministic complexity. We use the following special case (proved in Section 2.2.3) of the full junta-based simulation theorem.

Theorem 3.4. *There is a gadget g on $b = \Theta(\log n)$ bits so that for all $f\colon \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{coNP}^{\mathsf{cc}}(f \circ g^n) \;\geq\; \Omega(\mathsf{coNP}^{\mathsf{dt}}(f) \cdot b). \tag{3.1}$$

This establishes the lower bound on $\mathsf{coNP}^{\mathsf{cc}}(F)$. For the upper bound on $\mathsf{UP}^{\mathsf{cc}}(F)$, we use the simple fact that a protocol can always simulate a corresponding type of decision tree:

$$\mathsf{UP}^{\mathsf{cc}}(f \circ g^n) \;\leq\; O(\mathsf{UP}^{\mathsf{dt}}(f) \cdot b). \tag{3.2}$$

To conclude, Theorem 3.3 together with (3.1) and (3.2) imply a gap of $\mathsf{coNP}^{\mathsf{cc}}(F) \geq \Omega(\mathsf{UP}^{\mathsf{cc}}(F)^{\alpha} \cdot \log^{1-\alpha} n)$. Our F is actually going to satisfy $\mathsf{UP}^{\mathsf{cc}}(F) \geq n^{\Omega(1)}$ so the previous bound can be written as $\mathsf{coNP}^{\mathsf{cc}}(F) \geq \mathsf{UP}^{\mathsf{cc}}(F)^{\alpha - o(1)}$. This proves Theorem 3.2 for any $\beta < \alpha$.

The rest of this chapter is devoted to proving Theorem 3.3.

3.2 Definitions

A size-k *certificate* for "$P(x)$"—where $P(\,\cdot\,)$ is a predicate and $x \in \{0,1\}^n$ is an input—is a partial assignment C to inputs, setting at most k variables, such that x agrees with C (we also say that C accepts x) and for every y agreeing with C it holds that $P(y)$. The *certificate complexity* of "$P(x)$" is the least size of a certificate for "$P(x)$". A nondeterministic ($\mathsf{NP}^{\mathsf{dt}}$) decision tree $\mathcal{T}$ of cost k for "$P(\,\cdot\,)$" is a collection of size-k certificates such that for every x, $P(x)$ holds iff there exists an $C \in \mathcal{T}$ that accepts x. We say that $\mathcal{T}$ is *unambiguous* ($\mathsf{UP}^{\mathsf{dt}}$), if on every input x there is at most one certificate $C \in \mathcal{T}$ that accepts x. We define $\mathsf{NP}^{\mathsf{dt}}(f)$ (resp. $\mathsf{UP}^{\mathsf{dt}}(f)$) as the least cost of a $\mathsf{NP}^{\mathsf{dt}}$ (resp. $\mathsf{UP}^{\mathsf{dt}}$) decision tree for "$f(\,\cdot\,) = 1$". Also let $\mathsf{coNP}^{\mathsf{dt}}(f) := \mathsf{NP}^{\mathsf{dt}}(\neg f)$. Finally, $\mathsf{P}^{\mathsf{dt}}(f)$ denotes the deterministic decision tree complexity of f.

3.3 Warm-up discussion

The purpose of this section is to develop a "feel" for the $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ question, and motivate some of the choices that are made in the upcoming sections.

Upper bound on gap. A well-known result for decision trees states that $\mathsf{P}^{\mathsf{dt}}(f) \leq \mathsf{NP}^{\mathsf{dt}}(f) \cdot \mathsf{coNP}^{\mathsf{dt}}(f)$; see [Juk12, §14.2]. A similar argument shows that $\mathsf{P}^{\mathsf{dt}}(f) \leq \mathsf{UP}^{\mathsf{dt}}(f)^2$. In particular, $\mathsf{coNP}^{\mathsf{dt}}(f) \leq \mathsf{UP}^{\mathsf{dt}}(f)^2$, and hence the $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ gap cannot be too large (as in communication complexity). This argument is illuminating, so we present it here.

Proposition 3.5. $\mathsf{P}^{\mathsf{dt}}(f) \leq \mathsf{UP}^{\mathsf{dt}}(f)^2$.

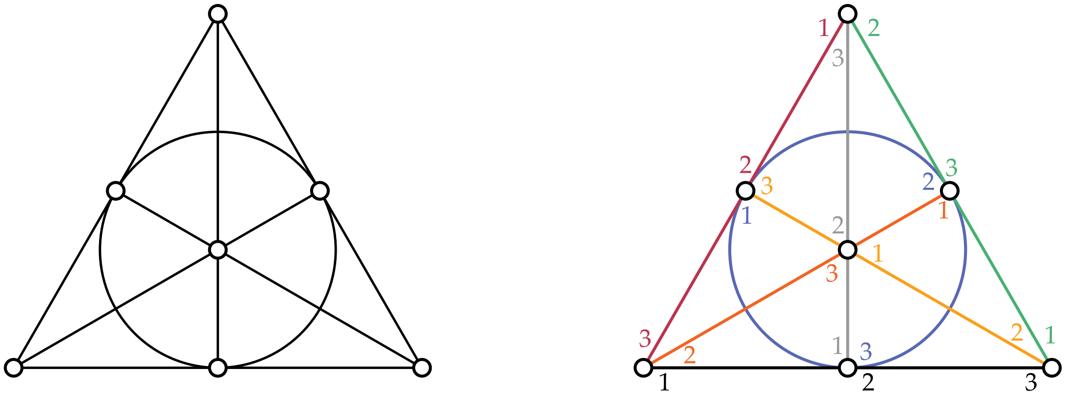
Proof. Let $\mathcal{T}$ be a $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f. A key observation is that any two certificates of $\mathcal{T}$ must intersect in variables—otherwise we could construct an input that satisfied two distinct certificates of $\mathcal{T}$, which contradicts unambiguity. The P^{dt} decision tree is this: Choose any certificate $C \in \mathcal{T}$ and query all its variables (at most $\mathsf{UP}^{\mathsf{dt}}(f)$ many). Let ρ be a partial assignment recording the values revealed, and let f_ρ be the restriction of f by ρ. If f_ρ is constant, we can output the corresponding value. Otherwise recursively run a decision tree for f_ρ. Here $\mathsf{UP}^{\mathsf{dt}}(f_\rho) \leq \mathsf{UP}^{\mathsf{dt}}(f) - 1$ as every remaining certificate $C' \neq C$ of $\mathcal{T}$ must intersect C in variables, so ρ already fixes at least one variable from each C'. □

Recursive composition. It is easy to exhibit a constant-size function with *some* $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ gap. For example, define a function on 3 bits by $f(x) = 1$ iff x has Hamming weight 1 or 2. We have $\mathsf{UP}^{\mathsf{dt}}(f) = 2$ since $\{x_1\bar{x}_2,\ x_2\bar{x}_3,\ x_3\bar{x}_1\}$ is an unambiguous collection of certificates for f. We also have $\mathsf{coNP}^{\mathsf{dt}}(f) = 3$ since the function is sensitive to flipping any of the bits on the all-0 input $\vec{0}$.

A standard trick in boolean function complexity [NS94, NW95, Sav02] is to take a constant-size function, such as f above, and to recursively compose it with itself: $f^{i+1}(\,\cdot\,) := f(f^i(\,\cdot\,), f^i(\,\cdot\,), f^i(\,\cdot\,)))$. The hope is that at every level of recursion the gap between the two complexity measures of interest increases by some constant factor (e.g., 3/2 in the case of f). However, this approach works straightforwardly only when the complexity measures are two-sided, that is, they do not distinguish between f and $\neg f$. In our case of $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ we are searching for a function whose 1-inputs are easy, but 0-inputs are hard. An obstacle associated with a recursively defined function f^i is that the only way to certify "$f^i(\,\cdot\,) = 1$"—which should be easy—often involves recursively certifying that "$f^{i-1}(\,\cdot\,) = 0$"—which should be hard!

Large alphabets. Our construction will employ recursion. In order to avoid the obstacle described above, we are going to enlarge the input/output alphabets of our functions. With large alphabets we assume a decision tree model where queries are still atomic, and reading a symbol from any input costs one query. We will study functions of the form $f\colon (\{0\} \cup \Sigma)^n \to \{0\} \cup \Sigma$, where the symbols in Σ are intuitively thought of as "easy to certify" as the output of f. The symbol 0 will be "hard to certify" as the output of f, and consequently we will always study the $\mathsf{coNP}^{\mathsf{dt}}$ complexity relative to the all-0 input $\vec{0}$. This will play nicely with recursive composition—a hard input for f^i will evaluate to $\vec{0}$ on each level of the construction. On the other hand, we get an efficient unambiguous decision tree for "$f^i(\,\cdot\,) = \sigma$" where $\sigma \in \Sigma$ as we only need to recursively certify the same type of easy statements: "$f^{i-1}(\,\cdot\,) = \sigma'$" where $\sigma' \in \Sigma$. That is, we make sure that an $\mathsf{UP}^{\mathsf{dt}}$ decision tree will never need to certify hard statements of the form "$f^i(\,\cdot\,) = 0$". Such $\mathsf{UP}^{\mathsf{dt}}$ decision trees will be called 0-*avoiding* later.

Note that if we have a function $f\colon \Sigma^n \to \{0, 1\}$ with a large input alphabet, we may always convert it to a boolean function by composing it with n copies of some surjection



(a) (b)

Figure 3.2: (a) Fano plane with 7 nodes and 7 hyperedges, and (b) a symmetric incidence ordering.

$g\colon \{0,1\}^{\lceil \log|\Sigma| \rceil} \to \Sigma$. The following naïve bounds will suffice for our purposes:

$$\mathcal{C}(f) \;\leq\; \mathcal{C}(f \circ g^n) \;\leq\; \mathcal{C}(f) \cdot \lceil \log|\Sigma| \rceil \qquad \text{for all } \; \mathcal{C} \in \{\mathsf{UP}^{\mathsf{dt}}, \mathsf{coNP}^{\mathsf{dt}}\}. \tag{3.3}$$

3.4 Projective plane example

In this section we describe an example witnessing a constant-factor $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ gap. The example is based on projective planes, and it will be suitable for recursive composition later.

Finite projective planes. For our purposes, a finite projective plane H is a k-uniform hypergraph with $n = k^2 - k + 1$ nodes and n hyperedges. Each node is incident to k edges, and each edge is incident to k nodes. Moreover, the edges are pairwise intersecting. See Figure 3.2a for an example of the case $k = 3$. It is well known that finite projective planes exist whenever $k - 1$ is a prime power.

Symmetric incidence ordering. Given an H as above, we fix for each node $v \in V(H)$ an ordering of its incident edges, and for each edge $e \in E(H)$ an ordering of its incident nodes. We make sure that the two orderings are *symmetric*: e is the i-th incident edge of $v \in e$ iff v is the i-th incident node of e; see Figure 3.2b. Such symmetric orderings are guaranteed to exist by Hall's matching theorem [Die10, §2.1]. (Consider the k-regular bipartite incidence graph of H that has the original nodes $V(H)$ on the left and the original hyperedges $E(H)$ on the right and there is an edge $\{v, e\}$ iff $v \in e$. Then by Hall's theorem the edges of this graph can be partitioned into k disjoint perfect matchings—this encodes the desired orderings.)

Unweighted function. To turn H into a query problem f, we make the nodes of H correspond to input variables that take on values from $\{0\} \cup [k]$. These values are interpreted as *pointers*: we say that a node with input $i \in [k]$ *points* to its i-th incident edge. A node with input 0 does not point anywhere (0 is a *null pointer*). Moreover, we say that an assignment $x\colon V(H) \to \{0\} \cup [k]$ *satisfies* an edge $e \in E(H)$ if all the incident nodes $v \in e$ point to e according to x. Note that each x can satisfy at most a single edge, since every two edges share a node, and this node can only point to one of the two edges.

The function $f\colon (\{0\} \cup [k])^n \to \{0,1\}$ is now defined so that $f(x) = 1$ iff x satisfies an edge. There is an obvious cost-k $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f whose certificates are in one-to-one correspondence with the edges of H. Unfortunately, the certificate complexity of "$f(\vec{0}) = 0$" is not large: it is also k as we can query k null pointers from any fixed edge, which ensures that no edge is satisfied. We will next fix this by assigning *weights* to the inputs.

(*Remark:* There are in fact 0-inputs of f with certificate complexity $> k$. However, these inputs will contain pointers, so they will not help us in the upcoming recursive construction: pointers will be too easy to certify recursively; only the number of 0's read by a certificate counts.)

Input weights. We are going to modify the function f defined above so that querying the inputs x_j to f becomes harder: to decide whether "$x_j = i$" one needs to spend $w(i)$ queries (instead of 1) where $w\colon [k] \to \mathbb{N}$ is a *weighting* function of our choosing. We allow only positive weights, so we agree that $0 \notin \mathbb{N}$. Concretely, this is achieved by considering a composed function $f_w \coloneqq f \circ g_w^n$ where g_w is a gadget that implements the weights w. More specifically, g_w is of the form $(\{0\} \cup [k])^m \to \{0\} \cup [k]$ where $m \coloneqq \max w([k])$ is the maximum weight and $g_w(x)$ is defined as follows: If $x = \vec{0}$, output 0. Otherwise let $x_j = i$ be the first non-0 coordinate of x. If $j \le w(i)$, output i; otherwise output 0.

By construction, for deterministic decision trees and for every $i \in [k]$, it is both necessary and sufficient to make $w(i)$ queries in order to decide whether "$g_w(\,\cdot\,) = i$". Furthermore, we record for later use the following two properties that also follow straightforwardly from the construction.

(P1) The certificate complexity of "$g_w(\vec{0}) \ne i$" is $w(i)$.

(P2) Suppose $\hat{w}(i) \coloneqq \ell \cdot w(i)$ for all i. Then $\mathsf{coNP}^{\mathsf{dt}}(f_{\hat{w}}) = \ell \cdot \mathsf{coNP}^{\mathsf{dt}}(f_w)$ for every f.

Weighted function. Define w by $w(i) \coloneqq i$ and consider $f_w \coloneqq f \circ g_w^n$. We claim that

$$\begin{aligned} \mathsf{UP}^{\mathsf{dt}}(f_w) &\le k(k+1)/2, \\ \mathsf{coNP}^{\mathsf{dt}}(f_w) &\ge k^2 - k + 1. \end{aligned}$$

That is, we have asymptotically a factor 2 separation.

Upper bound. We can devise a $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f_w by simply composing a $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f and optimal deterministic decision trees for the g_w's. A certificate corresponding to an edge $e \in V(H)$ in the $\mathsf{UP}^{\mathsf{dt}}$ decision tree for "$f(\,\cdot\,) = 1$" checks for each $i \in [k]$ that the i-th incident node of e is outputting i, which recursively involves checking that "$g_w(\,\cdot\,) = i$" for the corresponding gadget. For each i this amounts to $w(i) = i$ queries, which is $\sum_{i\in[k]} i = k(k+1)/2$ in total.

Lower bound. Our goal is to lower bound $\mathsf{coNP}^{\mathsf{dt}}(f_w)$ by analysing the 0-input $\vec{0}$. We start by highlighting a special property of the gadget that is afforded by our choice of w. The property states that to certify—on input $\vec{0}$—that none of a set of ℓ pointers appear as the output of g_w, one needs to make at least ℓ queries to the input variables.

Claim 3.6. *Let $S \subseteq [k]$. The certificate complexity of "$g_w(\vec{0}) \notin S$" is at least $|S|$.*

Proof. To certify "$g_w(\vec{0}) \notin S$" it is necessary to certify "$g_w(\vec{0}) \neq i$" where i is the largest number in S. But this latter task needs certificates of size $w(i) = i \geq |S|$ by *(P1)*. □

Our special property actually holds more generally across all the n gadgets. This generalised property states that to certify—on input $\vec{0}$—that none of a set of ℓ pointers (possibly associated with different nodes of H) appear in the output of the n gadgets g_w^n, one needs to make at least ℓ queries to the input variables of g_w^n. To state this more precisely, write $G \coloneqq g_w^n$ and let $G_v(\,\cdot\,)$ denote the output of the gadget corresponding to node $v \in V(H)$.

Claim 3.7. *For each v let $S_v \subseteq [k]$. The certificate complexity of "$\forall v : G_v(\vec{0}) \notin S_v$" is at least $\sum_v |S_v|$.*

Proof. Claim 3.6 proves this for each fixed v, but since the gadgets are defined on disjoint sets of variables, the individual certificate complexities sum up. □

Finally, the lower bound follows from the fact that any certificate for "$f_w(\vec{0}) = 0$" must certify that each edge $e \in E(H)$ lacks at least one pointer. This is $n = k^2 - k + 1$ pointers (and hence queries) in total.

3.5 Recursive composition

In this section we prove Theorem 3.3, restated below for convenience. In short, our idea is to recursively compose the example of Section 3.4.

Theorem 3.3. *There is an f such that $\mathsf{coNP}^{\mathsf{dt}}(f) \geq \mathsf{UP}^{\mathsf{dt}}(f)^{\alpha}$ where $\alpha > 1.128$ is a constant.*

Conventions. In what follows, we will consider pairs (f, w) where $f\colon (\{0\} \cup \Sigma)^n \to \{0, 1\}$ is a function and $w\colon \Sigma \to \mathbb{N}$ assigns weights to the inputs of f. We also define $f_w \coloneqq f \circ g_w^n$ as before. A 0-*avoiding* $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f is one whose certificates contain only values from Σ (i.e., not 0). More generally, a 0-*avoiding* $\mathsf{UP}^{\mathsf{dt}}$ decision tree for a weighted function f_w is a composition of a 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f and a deterministic decision tree for the g_w's; here a decision tree for g_w can—and indeed sometimes must—read 0's from the inputs. We let $\mathsf{UP}^{\mathsf{dt}}_{\star}(f_w)$ stand for the minimum cost of a 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f_w.

We prove our $\mathsf{coNP}^{\mathsf{dt}}$ lower bound by considering the complexity of certifying "$f_w(\vec{0}) = 0$", that is, we only focus on the 0-input $\vec{0}$. Hence we introduce the notation $\mathsf{coNP}^{\mathsf{dt}}_\star(f_w)$ to stand for the certificate complexity of "$f_w(\vec{0}) = 0$".

Overview. Given a pair (f, w), we construct a new pair (f', w') with the following properties. (Here k is again a parameter, chosen later, such that $k - 1$ is a prime power.)

(A1) Unambiguous complexity: $\mathsf{UP}^{\mathsf{dt}}_\star(f'_{w'}) \leq k(k+1)/2 \cdot \mathsf{UP}^{\mathsf{dt}}_\star(f_w)$.
(A2) Conondeterministic complexity: $\mathsf{coNP}^{\mathsf{dt}}_\star(f'_{w'}) \geq (k^2 - k + 1) \cdot \mathsf{coNP}^{\mathsf{dt}}_\star(f_w)$.

We proceed in two steps. In the first step (Section 3.5.1), we transform (f, w) into a pair $(\hat{f}, \hat{w})$ where $\hat{f}$ is a multi-valued function outputting values in $\{0\} \cup [k]$. The key property is the following:

> **If** f_w exhibits a gap between unambiguously certifying "$f_w(\cdot) = 1$" vs. nondeterministically certifying "$f_w(\vec{0}) = 0$", **then** for all $i \in [k]$, $\hat{f}_{\hat{w}}$ exhibits the same gap between unambiguously certifying "$\hat{f}_{\hat{w}}(\cdot) = i$" vs. nondeterministically certifying "$\hat{f}_{\hat{w}}(\vec{0}) \neq i$".

In the second step (Section 3.5.2), we begin thinking of $\hat{f}_{\hat{w}}$ as a gadget outputting pointer values: we plug several copies of $\hat{f}_{\hat{w}}$ as inputs to the projective plane example of Section 3.4. This will amplify the $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ gap further and give us (f', w').

From this construction it is easy to derive Theorem 3.3 (Section 3.5.3).

3.5.1 First step: Multi-valued outputs

From $f\colon (\{0\} \cup \Sigma)^N \to \{0, 1\}$ we will construct $\hat{f}\colon (\{0\} \cup \Sigma \times [k])^N \to \{0\} \cup [k]$. For notation, let $\pi_1\colon \Sigma \times [k] \to \Sigma$ and $\pi_2\colon \Sigma \times [k] \to [k]$ denote the natural projection maps, and extend $\pi_1(0) = \pi_2(0) = 0$ for convenience. For $x \in (\{0\} \cup \Sigma \times [k])^N$ we write $\pi_1(x)$ and $\pi_2(x)$ for the coordinate-wise application of π_1 and π_2 to x.

Definition of $(\hat{f}, \hat{w})$. Fix some optimal 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree $\mathcal{T}$ for f. We define $\hat{f}$ on input $x \in (\{0\} \cup \Sigma \times [k])^N$ as follows: If $f(\pi_1(x)) = 0$, output 0. Otherwise consider the unique certificate of $\mathcal{T}$ that accepts $\pi_1(x)$. This certificate reads some subset $S \subseteq [N]$ of inputs, where $x_j \neq 0$ for all $j \in S$, since $\mathcal{T}$ is 0-avoiding. If there is an $i \in [k]$ such that $\pi_2(x_j) = i$ for all $j \in S$, then output i; otherwise output 0.

The input weights $\hat{w}\colon \Sigma \times [k] \to \mathbb{N}$ are defined by $\hat{w}(\sigma, i) \coloneqq w(\sigma) \cdot i$.

Analysis. We claim that $(\hat{f}, \hat{w})$ satisfies the following.

(B1) There is a 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree for "$\hat{f}_{\hat{w}}(\cdot) = i$" with cost $i \cdot \mathsf{UP}^{\mathsf{dt}}_\star(f_w)$.
(B2) The certificate complexity of "$\hat{f}_{\hat{w}}(\vec{0}) \neq i$" is at least $i \cdot \mathsf{coNP}^{\mathsf{dt}}_\star(f_w)$.

(B1) holds: We can simply modify $\mathcal{T}$ slightly: to certify "$\hat{f}_{\hat{w}}(\,\cdot\,) = i$" the modified decision tree works as before but now additionally checks the new condition on $\pi_2(x)$ where x is the input to $\hat{f}$ (as encoded by $g_{\hat{w}}$). The cost of the modified tree is i times that of $\mathcal{T}$ as the relevant inputs are now i times heavier in the new $\hat{w}$ than in the old w.

(B2) holds: We prove this by a reduction argument. Fix $i \in [k]$. Consider deleting all symbols from the input alphabet of $\hat{f}$ except $\{0\} \cup \Sigma \times \{i\}$ and call the resulting subfunction $\hat{\hat{f}}$. Clearly the certificate complexity of "$\hat{f}_{\hat{w}}(\vec{0}) \neq i$" (which we are interested in) is at least that of "$\hat{\hat{f}}_{\hat{w}}(\vec{0}) \neq i$". Note that the range of $\hat{\hat{f}}$ is just $\{0, i\}$. Moreover, $\hat{\hat{f}}$ is isomorphic to f: identify the input alphabets via the map π_1, and the output alphabets via the map $0 \mapsto 0$, $i \mapsto 1$. Therefore the certificate complexities of "$\hat{\hat{f}}_{\hat{w}}(\vec{0}) \neq i$" and "$f_{\hat{w}}(\vec{0}) = 0$" are the same. (Here we abused notation by identifying the alphabets of f and $\hat{w}$ according to the isomorphism.) But $\hat{w}$ is nothing but w scaled by a factor of i (on the relevant alphabet $\Sigma \times \{i\}$), so by *(P2)* the certificate complexity of "$f_{\hat{w}}(\vec{0}) = 0$" is i times that of "$f_w(\vec{0}) = 0$", namely $i \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$.

3.5.2 Second step: Composition with a projective plane

Take a k-uniform projective plane hypergraph H as in Section 3.4 and let its associated unweighted function be $h\colon (\{0\} \cup [k])^n \to \{0, 1\}$ where $n = k^2 - k + 1$. We define $f' \coloneqq h \circ \hat{f}^n$ together with the old weights $w' \coloneqq \hat{w}$ (so $f'_{w'} = h \circ \hat{f}^n_{\hat{w}}$). We claim that (f', w') satisfies *(A1)* and *(A2)*.

(A1) holds: We do the natural thing: The 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree for $f'_{w'}$ has a certificate for every edge $e \in E(H)$ that, for every $i \in [k]$, recursively checks that the i-th node incident to e points to e. The i-th recursive check involves certifying "$\hat{f}_{\hat{w}}(\,\cdot\,) = i$", which we can do in a 0-avoiding manner using *(B1)*. This has total cost $\sum_{i \in [k]} i \cdot \mathsf{UP}^{\mathsf{dt}}_{\star}(f_w) = k(k+1)/2 \cdot \mathsf{UP}^{\mathsf{dt}}_{\star}(f_w)$, as desired.

(A2) holds: We now begin thinking of the $\hat{f}_{\hat{w}}$'s as "gadgets". Under this nomenclature, we can simply repeat the argument from Section 3.4 verbatim. Indeed, our special property in the case of a single gadget now states the following: to certify—on input $\vec{0}$—that none of a set of ℓ pointers appear as the output of $\hat{f}_{\hat{w}}$, one needs to make at least $\ell \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$ queries to the inputs of $\hat{f}_{\hat{w}}$.

Claim 3.8. *Let $S \subseteq [k]$. The certificate complexity of "$\hat{f}_{\hat{w}}(\vec{0}) \notin S$" is at least $|S| \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$.*

Proof. To certify "$\hat{f}_{\hat{w}}(\vec{0}) \notin S$" it is necessary to certify "$\hat{f}_{\hat{w}}(\vec{0}) \neq i$" where i is the largest number in S. But this latter task needs certificates of size $i \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w) \geq |S| \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$ by *(B2)*. □

As before, because the n gadgets $\hat{f}^n_{\hat{w}}$ are fed disjoint sets of variables, a more general property holds: to certify—on input $\vec{0}$—that none of a set of ℓ pointers (possibly associated with different nodes of H) appear in the output of the n gadgets $\hat{f}^n_{\hat{w}}$, one needs to make at least $\ell \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$ queries to the input variables of $\hat{f}^n_{\hat{w}}$. To state this more precisely, write $G \coloneqq \hat{f}^n_{\hat{w}}$ and let $G_v(\,\cdot\,)$ denote the output of the gadget corresponding to node $v \in V(H)$.

Claim 3.9. *For each v let $S_v \subseteq [k]$. The certificate complexity of "$\forall v : G_v(\vec{0}) \notin S_v$" is at least $\sum_v |S_v| \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$.*

Proof. Claim 3.8 proves this for each fixed v, but since the gadgets are defined on disjoint sets of variables, the individual certificate complexities sum up. □

Finally, the lower bound follows from the fact that any certificate for "$f'_{w'}(\vec{0}) = 0$" must certify that each edge $e \in E(H)$ lacks at least one pointer. This is $n = k^2 - k + 1$ pointers that require $(k^2 - k + 1) \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$ queries in total.

3.5.3 Putting everything together

Define a sequence of pairs (f^i, w^i) as follows: Initially, $f^0 \colon \{0, 1\} \to \{0, 1\}$ is the identity function and $w^0(1) \coloneqq 1$. Clearly the $\mathsf{UP}^{\mathsf{dt}}_{\star}$ and $\mathsf{coNP}^{\mathsf{dt}}_{\star}$ complexities for $f^0_{w^0}$ are both 1. Recursively define (f^{i+1}, w^{i+1}) as the "primed" versions of (f^i, w^i).

By inspecting the construction, we have the following properties.

- Number of inputs to f^i is $(k^2 - k + 1)^i$.
- Maximum weight of w^i is k^i.
- Therefore, the number of inputs to $f^i_{w^i}$ is $n \coloneqq (k^3 - k^2 + k)^i$.
- Input alphabet $\{0\} \cup \Sigma$ of $f^i_{w^i}$ has size $|\{0\} \cup \Sigma| = 1 + k^i$.
- *(A1)* implies: $\mathsf{UP}^{\mathsf{dt}}(f^i_{w^i}) \leq (k(k + 1)/2)^i$.
- *(A2)* implies: $\mathsf{coNP}^{\mathsf{dt}}(f^i_{w^i}) \geq (k^2 - k + 1)^i$.

The last two items say that $\mathsf{coNP}^{\mathsf{dt}}(f^i_{w^i}) \geq \mathsf{UP}^{\mathsf{dt}}(f^i_{w^i})^{\beta}$ for $\beta \coloneqq \log(k^2 - k + 1)/\log(k(k + 1)/2)$. This is maximised at $k = 8$ (among k such that $k - 1$ is a prime power), which yields $\beta = \log_{36} 57 > 1.128$.

As a final step, we need to replace the input alphabet of $f^i_{w^i}$ with a binary encoding. The above parameters reveal that $\log |\{0\} \cup \Sigma| \leq O(\log n)$ while our estimates for both $\mathsf{coNP}^{\mathsf{dt}}(f^i_{w^i})$ and $\mathsf{UP}^{\mathsf{dt}}(f^i_{w^i})$ are $n^{\Theta(1)}$. Hence the loose bounds (3.3) give us the gap in Theorem 3.3 for any $\alpha < \beta$.

Chapter 4

Deterministic Communication vs. Partition Number

Overview. In this chapter we show that deterministic communication complexity can be superlogarithmic in the partition number of the associated communication matrix. We also obtain near-optimal deterministic lower bounds for the Clique vs. Independent Set problem, which in particular yields new lower bounds for the log-rank conjecture. All these results follow from a simple adaptation of a communication-to-query simulation theorem of Raz and McKenzie [RM99] together with lower bounds for the analogous query complexity questions. This chapter is based on the following publication:

[GPW15]: Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, pages 1077–1088. IEEE, 2015. doi:10.1109/FOCS.2015.70

4.1 Introduction

The *partition number* of a two-party function $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is defined by

$$\chi(F) \;\coloneqq\; \chi_1(F) + \chi_0(F)$$

where $\chi_i(F)$ is the least number of rectangles (sets of the form $A \times B$ where $A \subseteq \mathcal{X}$, $B \subseteq \mathcal{Y}$) needed to partition the set $F^{-1}(i)$. Yao [Yao79] observed that $\log \chi(F)$ is a lower bound on the deterministic communication complexity of F and inquired about the exact relationship. For upper bounds, it is known that $O(\log^2 \chi(F))$ bits [AUY83], or even $O(\log^2 \chi_1(F))$ bits [Yan91], suffice.

Our results are as follows—here the notation $\tilde{\Omega}(m)$ hides factors polylogarithmic in m.

Theorem 4.1. *There is an F with deterministic communication complexity $\tilde{\Omega}(\log^{1.5} \chi(F))$.*

Theorem 4.2. *There is an F with deterministic communication complexity $\tilde{\Omega}(\log^2 \chi_1(F))$.*

Theorem 4.1 implies that the logarithm of the partition number does not characterize (up to constant factors) deterministic communication complexity, which solves an old problem [KN97, Open Problem 2.10]. The previous best lower bound in this direction was about $2 \cdot \log \chi(F)$ due to Kushilevitz, Linial, and Ostrovsky [KLO99]. In this chapter, we show—maybe surprisingly—that superlogarithmic lower bounds can be obtained using known techniques!

Theorem 4.2 is essentially tight in view of the upper bound $O(\log^2 \chi_1(F))$ mentioned above. In Chapter 3 we exhibited a different F with *conondeterministic* communication complexity $\Omega(\log^{1.128} \chi_1(F))$; this is quantitatively weaker than Theorem 4.2 and hence the two results are incomparable. The question about the relationship between $\log \chi_1(F)$ and deterministic communication complexity is sometimes referred to as the Clique vs. Independent Set problem; see [Juk12, §4.4] for an excellent overview. In particular, Theorem 4.2 implies that there exists a graph on n nodes for which the Clique vs. Independent Set problem (Alice is given a clique and Bob is given an independent set: do they intersect?) requires $\tilde{\Omega}(\log^2 n)$ communication. (The upper bound $O(\log^2 n)$ holds for all graphs.) Theorem 4.2 also gives improved lower bounds for the log-rank conjecture [LS88] (see [Lov14] for a survey): Viewing rectangles as all-1 submatrices, we have $\chi_1(F) \geq \text{rank}(F)$ where the rank is over the reals. Hence Theorem 4.2 implies a communication lower bound of $\tilde{\Omega}(\log^2 \text{rank}(F))$. The previous record was $\Omega(\log^{1.63} \text{rank}(F))$ due to Kushilevitz [Kus94].

4.1.1 Our approach

Deterministic simulation. We use tools that were introduced already in 1997 by Raz and McKenzie [RM99] (building on [EIRS01]). They proved a simulation theorem that converts a deterministic protocol for $F := f \circ g^n$ (where f is arbitrary but the gadget g is chosen carefully) into a deterministic decision tree for f. Unfortunately, their result was originally formulated only in case f was a certain "structured" search problem (canonical search problem associated with a DNF tautology), and this is how their result has been applied subsequently [BEGJ00, Joh01]. However, we observe that, with minor modifications, their proof actually works without any assumptions on f. Such a simulation theorem (for functions f) was conjectured in [Dru09]. We provide (in Section 4.3) a self-contained and streamlined exposition (including some simplifications) of the following version of the Raz–McKenzie result—here $\mathsf{P}^{\mathsf{cc}}(F)$ denotes the deterministic communication complexity of F and $\mathsf{P}^{\mathsf{dt}}(f)$ denotes the deterministic decision tree complexity of f.

Theorem 4.3 (Simulation Theorem)**.** *There is a gadget $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ where the size of Alice's input is $\log |\mathcal{X}| = \Theta(\log n)$ bits such that for all $f\colon \{0,1\}^n \to \{0,1\}$ we have*

$$\mathsf{P}^{\mathsf{cc}}(f \circ g^n) \;=\; \mathsf{P}^{\mathsf{dt}}(f) \cdot \Theta(\log n).$$

The gadget in the above can be taken to be the usual indexing function $g\colon [m] \times \{0,1\}^m \to \{0,1\}$ where $m \coloneqq \mathrm{poly}(n)$ and $g(x,y) \coloneqq y_x$.

Nondeterministic models. Recall that a *nondeterministic protocol* (e.g., [KN97, Juk12]) is a protocol that is allowed to make guesses—an input is accepted iff there is at least one accepting computation. Combinatorially, a nondeterministic protocol for F of communication cost k can be visualized as a covering of the set $F^{-1}(1)$ using at most 2^k (possibly overlapping) rectangles. Thus, the nondeterministic communication complexity of F, denoted $\mathsf{NP}^{\mathsf{cc}}(F)$, is just the logarithm of the least number of rectangles needed to cover $F^{-1}(1)$. A nondeterministic protocol is *unambiguous* if for each input, there is at most one accepting computation. Combinatorially, this means that the associated rectangles covering $F^{-1}(1)$ do not overlap. We use the notation $\mathsf{UP}^{\mathsf{cc}}(F) \coloneqq \lceil \log \chi_1(F) \rceil$, and also define $\mathsf{coUP}^{\mathsf{cc}}(F) \coloneqq \lceil \log \chi_0(F) \rceil$, and, using the shorthand $\mathsf{2UP} \coloneqq \mathsf{UP} \cap \mathsf{coUP}$, we define the two-sided measure $\mathsf{2UP}^{\mathsf{cc}}(F) \coloneqq \lceil \log \chi(F) \rceil \in \max\{\mathsf{UP}^{\mathsf{cc}}(F), \mathsf{coUP}^{\mathsf{cc}}(F)\} \pm O(1)$.

Analogously, a *nondeterministic decision tree* (e.g., [Juk12, §14.2]) is a decision tree that is allowed to make guesses. Formally, we treat a nondeterministic decision tree for f as a collection of 1-certificates (accepting computations), that is, partial assignments to variables of f that force the output of the function to be 1; the cost is the maximum number of variables fixed by a partial assignment. In other words, a nondeterministic decision tree is just a DNF formula; the cost is the maximum width of its terms. We denote by $\mathsf{NP}^{\mathsf{dt}}(f)$ the minimum cost of a nondeterministic decision tree for f, that is, its DNF width. A nondeterministic decision tree is *unambiguous* if for each input, there is at most one accepting certificate. We denote by $\mathsf{UP}^{\mathsf{dt}}(f)$ the minimum cost of an unambiguous decision tree for f. We also let $\mathsf{coUP}^{\mathsf{dt}}(f) \coloneqq \mathsf{UP}^{\mathsf{dt}}(\neg f)$ and $\mathsf{2UP}^{\mathsf{dt}}(f) \coloneqq \max\{\mathsf{UP}^{\mathsf{dt}}(f), \mathsf{coUP}^{\mathsf{dt}}(f)\}$.

Communication ↔ query. We can rephrase our communication results with the new notation.

Theorem 4.1 (Rephrased). *There is an F such that $\mathsf{P}^{\mathsf{cc}}(F) \geq \tilde{\Omega}(\mathsf{2UP}^{\mathsf{cc}}(F)^{1.5})$.*

Theorem 4.2 (Rephrased). *There is an F such that $\mathsf{P}^{\mathsf{cc}}(F) \geq \tilde{\Omega}(\mathsf{UP}^{\mathsf{cc}}(F)^{2})$.*

Our goal is to prove analogous query complexity separations (in Section 4.2):

Theorem 4.4. *There is an f such that $\mathsf{P}^{\mathsf{dt}}(f) \geq \tilde{\Omega}(\mathsf{2UP}^{\mathsf{dt}}(f)^{1.5})$.*

Theorem 4.5. *There is an f such that $\mathsf{P}^{\mathsf{dt}}(f) \geq \tilde{\Omega}(\mathsf{UP}^{\mathsf{dt}}(f)^{2})$.*

Theorems 4.1–4.2 can now be derived by simply applying Theorem 4.3 and the trivial fact that $\mathcal{C}^{\mathsf{cc}}(f \circ g^n) \leq \mathcal{C}^{\mathsf{dt}}(f) \cdot O(\log n)$ for all $\mathcal{C} \in \{\mathsf{2UP}, \mathsf{UP}\}$. We only add that the functions in Theorems 4.4–4.5 will actually satisfy $\mathsf{P}^{\mathsf{dt}}(f) = n^{\Theta(1)}$ and hence the factor $\Theta(\log n)$ overhead that is introduced by the gadget gets hidden in our $\tilde{\Omega}$-notation.

A few comments about Theorems 4.4–4.5 are in order. Firstly, Savický [Sav02] and Belovs [Bel06] have previously exhibited a function with $\mathsf{P}^{\mathsf{dt}}(f) \geq \Omega(2\mathsf{UP}^{\mathsf{dt}}(f)^{1.261})$. This means that a quantitatively weaker (but still superlogarithmic) version of Theorem 4.1 follows already by combining the Savický–Belovs result with Theorem 4.3. Secondly, it is not hard to see that $\mathsf{UP}^{\mathsf{dt}}(f) \geq \deg(f)$ where $\deg(f)$ is the minimum degree of a multilinear real polynomial that agrees with f on boolean inputs. (The communication analogue of this inequality, namely $\mathsf{UP}^{\mathsf{cc}}(F) \geq \log \operatorname{rank}(F)$, was discussed above.) Consequently, Theorem 4.5 gives the largest known gap between $\mathsf{P}^{\mathsf{dt}}(f)$ and $\deg(f)$. The previous record was $\mathsf{P}^{\mathsf{dt}}(f) \geq \Omega(\deg(f)^{1.63})$ by Kushilevitz [Kus94], and the current best upper bound in this context is $\mathsf{P}^{\mathsf{dt}}(f) \leq O(\deg(f)^3)$ for all f by Midrijānis [Mid04].

4.2 Query Separations

In proving the query complexity separations it is convenient to work with functions $f\colon \Sigma^n \to \{0,1\}$ that have a larger-than-boolean input alphabet Σ. For such functions the understanding is that it still costs one query for a decision tree to learn a particular input variable. At the end, we may always convert such an f back to a boolean function $f \circ h^n$ where $h\colon \{0,1\}^{\lceil \log |\Sigma| \rceil} \to \Sigma$ is some surjection. The following trivial bounds suffice for us:

$$\mathcal{C}^{\mathsf{dt}}(f) \;\leq\; \mathcal{C}^{\mathsf{dt}}(f \circ h^n) \;\leq\; \mathcal{C}^{\mathsf{dt}}(f) \cdot \lceil \log |\Sigma| \rceil, \qquad \forall \mathcal{C} \in \{\mathsf{P}, 2\mathsf{UP}, \mathsf{UP}\}. \tag{4.1}$$

We start with the proof of Theorem 4.5 since the proof of Theorem 4.4 uses Theorem 4.5 (as a black box).

4.2.1 Proof of Theorem 4.5

Motivating example. Let $n := k^2$, and consider the function $f\colon \{0,1\}^{k\times k} \to \{0,1\}$ defined on boolean matrices $M \in \{0,1\}^{k\times k}$ such that $f(M) = 1$ iff M contains a *unique* all-1 column. We claim that

$$\begin{aligned} \mathsf{NP}^{\mathsf{dt}}(f) &\;\leq\; 2k - 1, \\ \mathsf{P}^{\mathsf{dt}}(f) &\;\geq\; k^2. \end{aligned}$$

For the upper bound, consider 1-certificates that read the unique all-1 column and a single 0-entry from each of the other columns. (Note that this collection of certificates is not unambiguous!) For the lower bound, it suffices to give an *adversary argument* (see, e.g., [Juk12, §14]), that is, a strategy to answer queries made by a decision tree such that even after $k^2 - 1$ queries, the output of the function is not yet determined. Here is the strategy: Suppose the decision tree queries M_{ij}. If M_{ij} is the last unqueried entry in the j-th column, answer $M_{ij} = 0$. Otherwise answer $M_{ij} = 1$. It is straightforward to check that this strategy forces the decision tree to query all of the entries.

Actual gap example. We modify the function described above with the goal of establishing

$$\begin{aligned}\mathsf{UP}^{\mathsf{dt}}(f) &\leq 2k-1,\\ \mathsf{P}^{\mathsf{dt}}(f) &\geq k^2.\end{aligned}$$

The modified function, which we still call f, has input variables that take on values from the alphabet $\Sigma := \{0,1\} \times ([k]\times[k] \cup \{\perp\})$. Here $[k]\times[k]\cup\{\perp\}$ is a set of *pointer values*, where we interpret an entry $M_{ij} = (m_{ij}, p_{ij}) \in \Sigma$ as *pointing* to another entry $M_{p_{ij}}$ given that $p_{ij} \neq \perp$. If $p_{ij} = \perp$ then we have a null pointer. We define the function $f\colon \Sigma^{k\times k} \to \{0,1\}$ by describing an unambiguous decision tree computing it. (We give an "algorithmic" definition rather than writing a list of certificates.)

> *Unambiguous decision tree:* Nondeterministically guess a column index $j \in [k]$. Read the entries $M_{ij} = (m_{ij}, p_{ij})$ for $i \in [k]$ while checking that $m_{ij} = 1$ for all i and that $p_{ij} \neq \perp$ for at least one i. Let i be the first index for which $p_{ij} \neq \perp$. Next, iteratively follow pointers for $k-1$ steps starting at $(i_1, j_1) := p_{ij}$. Namely, at the s-th step, read M_{i_s,j_s} and if $s \leq k-2$ then check that $p_{i_s,j_s} \neq \perp$ and define $(i_{s+1}, j_{s+1}) := p_{i_s,j_s}$. Finally, check that the resulting sequence $(i_1,j_1),\ldots,(i_{k-1},j_{k-1})$ visits all but the j-th column (i.e., $\{j_1,\ldots,j_{k-1}\} = [k] \smallsetminus \{j\}$) and that $m_{i_s,j_s} = 0$ for all $s \in [k-1]$.

Thus the upper bound holds by construction. For the lower bound, we use the below strategy; here a query to an entry M_{ij} is called *critical* if M_{ij} is the last unqueried entry in its column.

> *Adversary strategy:* Always answer queries with $(1,\perp)$ unless the query is critical. On the first critical query, answer $(0,\perp)$. On subsequent critical queries, answer $(0,p)$ where $p \in [k]\times[k]$ points to where the previous critical query took place.

The function value remains undetermined after k^2-1 queries, because we can answer the last (k^2-th) query with $(0,\perp)$ to make the function evaluate to 0, or with $(1,p)$, where p is as above, to make the function evaluate to 1. This proves $\mathsf{P}^{\mathsf{dt}}(f) \geq \Omega(\mathsf{UP}^{\mathsf{dt}}(f)^2)$ for a function with a non-boolean alphabet. If we convert f into a boolean function $f' := f \circ h^n$ (where $n := k^2$) as in (4.1) we end up with the claimed gap $\mathsf{P}^{\mathsf{dt}}(f') \geq \tilde{\Omega}(\mathsf{UP}^{\mathsf{dt}}(f')^2)$ since the conversion introduces only some $\lceil \log|\Sigma| \rceil = \Theta(\log n)$ factors.

4.2.2 Proof of Theorem 4.4

Let g be given by Theorem 4.5 such that $\mathsf{P}^{\mathsf{dt}}(g) = \tilde{\Theta}(q^2)$ where $q := \mathsf{UP}^{\mathsf{dt}}(g)$. We define $f := \mathsf{AND} \circ g^q$, that is, $f(z_1,\ldots,z_q) = 1$ iff $g(z_i) = 1$ for all $i \in [q]$. We claim that

$$\begin{aligned}2\mathsf{UP}^{\mathsf{dt}}(f) &\leq \tilde{O}(q^2),\\ \mathsf{P}^{\mathsf{dt}}(f) &\geq \tilde{\Omega}(q^3).\end{aligned}$$

For the upper bound, an unambiguous certificate for an input z will contain unambiguous 1-certificates for $g(z_i) = 1$ for all $i \in [\ell-1]$ where ℓ is the least index such that $g(z_\ell) = 0$, or $\ell \coloneqq q+1$ if no such index exists. If $\ell \leq q$ we also include an unambiguous 0-certificate for $g(z_\ell) = 0$ that just mimics the execution of an optimal decision tree for g on input z_ℓ. In other words, we use the fact that $\mathsf{coUP}^{\mathsf{dt}}(g) \leq \mathsf{P}^{\mathsf{dt}}(g)$. The cost is at most $(\ell - 1) \cdot \mathsf{UP}^{\mathsf{dt}}(g) + \mathsf{P}^{\mathsf{dt}}(g) \leq \tilde{O}(q^2)$. For the lower bound, we have $\mathsf{P}^{\mathsf{dt}}(\mathsf{AND} \circ g^q) = \mathsf{P}^{\mathsf{dt}}(\mathsf{AND}) \cdot \mathsf{P}^{\mathsf{dt}}(g) = q \cdot \tilde{\Theta}(q^2) = \tilde{\Theta}(q^3)$ by the basic fact (e.g., [Sav02, Lemma 3.2]) that P^{dt} behaves multiplicatively with respect to composition.

4.3 Raz–McKenzie Simulation

The goal of this section is to give a self-contained, streamlined, and somewhat simplified proof of the Simulation Theorem that works without any assumptions on the outer function

$$f\colon \{0,1\}^N \to \{0,1\}.$$

(Here we use N for the input length instead of n, which we reserve for later use.) In fact, f can be taken to be anything: a partial function, a search problem (a general relation), or have a non-boolean codomain. However, we stick with the boolean function case for concreteness.

The gadget $g\colon [m] \times \{0,1\}^m \to \{0,1\}$, where $m \coloneqq N^{20}$, is chosen to be the indexing function defined by $g(x, y) \coloneqq y_x$. Recall that for the composed function $F \coloneqq f \circ g^N$, Alice's input is $x \coloneqq (x_1, \ldots, x_N) \in [m]^N$ and Bob's input is $y \coloneqq (y_1, \ldots, y_N) \in (\{0,1\}^m)^N$. We denote by $z_i \coloneqq g(x_i, y_i)$ the i-th input bit of f so that $F(x, y) \coloneqq f(z_1, \ldots, z_N)$.

We prove the nontrivial part of the Simulation Theorem, namely the lower bound

$$\mathsf{P}^{\mathsf{cc}}(f \circ g^N) \;\geq\; \mathsf{P}^{\mathsf{dt}}(f) \cdot \Omega(\log m).$$

4.3.1 High-level overview

Once and for all, we fix a deterministic protocol for $F \coloneqq f \circ g^N$ of communication cost $k \leq o(N \cdot \log m)$. The basic strategy is to use the protocol to build a decision tree of cost $O(k/\log m)$ for evaluating the outer function f on an unknown input $z \in \{0,1\}^N$. The simulation algorithm proceeds in iterations, where in each iteration we either descend one level in the communication protocol tree (by making the protocol send a bit), or descend one level in the decision tree (by querying a bit of z). To show the simulation is correct, we maintain invariants ensuring that when we reach a leaf in the protocol tree, the value it outputs must be the correct value of $f(z)$ (hence we can make the current node in the decision tree a leaf). To show the simulation is efficient, we use a potential function argument showing that in each "communication iteration" the potential increases by at most $O(1)$, and in each "query iteration" the potential decreases by at least $\Omega(\log m)$, and hence the number of query iterations is at most $O(k/\log m)$ since there are at most k communication iterations.

In a little more detail, let R_v denote the rectangle associated with the current node v in the communication protocol tree. The simulation maintains a "cleaned up" subrectangle $A \times B \subseteq R_v$ with the property that the set of all outputs of g^N over points in $A \times B$ is exactly the set of all possible z's that are consistent with the results of the queries made so far. This ensures the correctness when we reach a leaf. The analysis has two key lemmas: the Thickness Lemma helps us update $A \times B$ in a communication iteration, and the Projection Lemma helps us update $A \times B$ in a query iteration.

To determine which type of iteration should be next, we examine, for each unqueried coordinate, how predictable it is (in some sense) from the values of the other unqueried coordinates of g^N within $A \times B$. If no coordinate is too predictable, then it is "safe" to have a communication iteration; the protocol partitions the rectangle R_v into two sides, and we restrict to the side that is "bigger" (from the perspective of the unqueried coordinates), then use the Thickness Lemma to do some further clean-up that restores our invariants. On the other hand, if say the i-th coordinate is too predictable from the others, then its value (within $A \times B$) is in danger of becoming a function of the values of the other coordinates (which would violate our invariants). In this case, we query z_i while we are still able to accommodate either possible value for it (which might become impossible if we delayed querying z_i), and the Projection Lemma allows us to clean up $A \times B$ and restore our invariants.

We describe our notation and state the two key lemmas in Section 4.3.2. Then we describe the simulation algorithm itself in Section 4.3.3 and analyze it in Section 4.3.4. Finally, we provide the proofs of the two key lemmas in Section 4.3.5 and Section 4.3.6.

4.3.2 Notation and lemmas

For a node v in the communication protocol tree, let $R_v \coloneqq X^v \times Y^v$ denote its associated rectangle, let $X^{v,b} \subseteq X^v$ be the set of Alice's inputs on which the bit b would be sent (if Alice sends), and let $Y^{v,b} \subseteq Y^v$ be the set of Bob's inputs on which the bit b would be sent (if Bob sends).

Supposing $A \subseteq [m]^n$ and $B \subseteq (\{0,1\}^m)^n$ for some $n \leq N$, we make the following definitions.

- **Size of sets:** Let $\alpha(A)$ be such that $|A| = 2^{-\alpha(A)} \cdot |[m]^n|$, and let $\beta(B)$ be such that $|B| = 2^{-\beta(B)} \cdot \left|(\{0,1\}^m)^n\right|$ (assuming $|A|, |B| > 0$).
- **Projections:** If $I \subseteq [n]$ then let $A_I \coloneqq \big\{(x_i)_{i\in I} : (x_1, \dots, x_n) \in A \text{ for some } (x_j)_{j\in[n]\smallsetminus I}\big\} \subseteq [m]^{|I|}$ be the projection of A onto the coordinates in I, and similarly $B_I \coloneqq \big\{(y_i)_{i\in I} : (y_1, \dots, y_n) \in B \text{ for some } (y_j)_{j\in[n]\smallsetminus I}\big\} \subseteq (\{0,1\}^m)^{|I|}$.
- **Pruning:** If $U \subseteq [m]$, $V \subseteq \{0,1\}^m$, and $i \in [n]$, then let $A^{i,U} \coloneqq \big\{x \in A : x_i \in U\big\}$ and $B^{i,V} \coloneqq \big\{y \in B : y_i \in V\big\}$.
- **Auxiliary graph:** If $i \in [n]$ then let $\mathrm{Graph}_i(A)$ be the bipartite graph defined as follows. The left nodes are $[m]$, the right nodes are $[m]^{n-1}$, and each tuple $x \coloneqq (x_1, \dots, x_n) \in A$ is viewed as an edge between the left node x_i and the right node $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$.

Note that $A_{[n]\smallsetminus\{i\}}$ is the set of nonzero-degree right nodes.

- **Average/minimum degree:** Let $\mathrm{AvgDeg}_i(A) \coloneqq |A|/|A_{[n]\smallsetminus\{i\}}|$ and $\mathrm{MinDeg}_i(A)$ be, respectively, the average and minimum degrees of a nonzero-degree right node in $\mathrm{Graph}_i(A)$.
- **Thickness:** We say A is *thick* iff $\mathrm{MinDeg}_i(A) \geq m^{17/20}$ for all $i \in [n]$.

The following lemma is helpful for when we need to let the communication protocol send a bit.

Lemma 4.6 (Thickness Lemma)**.** *If $n \geq 2$ and $A \subseteq [m]^n$ is such that $\mathrm{AvgDeg}_i(A) \geq d$ for all $i \in [n]$, then there exists an $A' \subseteq A$ such that*

(1) $\mathrm{MinDeg}_i(A') \geq d/2n$ *for all* $i \in [n]$,
(2) $\alpha(A') \leq \alpha(A) + 1$.

The following lemma is helpful for when we need to have the decision tree query a bit.

Lemma 4.7 (Projection Lemma)**.** *Suppose $n \geq 2$, $A \subseteq [m]^n$ is thick, and $B \subseteq (\{0,1\}^m)^n$ is such that $\beta(B) \leq m^{2/20}$. Then for every $i \in [n]$ and every $b \in \{0,1\}$ there exists a b-monochromatic rectangle $U \times V \subseteq [m] \times \{0,1\}^m$ in g such that*

(1) $A^{i,U}_{[n]\smallsetminus\{i\}}$ *is thick,*
(2) $\alpha\big(A^{i,U}_{[n]\smallsetminus\{i\}}\big) \leq \alpha(A) - \log m + \log \mathrm{AvgDeg}_i(A)$,
(3) $\beta\big(B^{i,V}_{[n]\smallsetminus\{i\}}\big) \leq \beta(B) + 1$.

4.3.3 Description of the simulation algorithm

The Simulation Theorem is witnessed by Algorithm 1, which is a decision tree for f that employs the hypothesized communication protocol for F. Algorithm 1 uses the following variables: v is a node in the communication protocol tree, $I \subseteq [N]$ is the set of unqueried coordinates, $A \subseteq [m]^N$ is a set of inputs to Alice, and $B \subseteq (\{0,1\}^m)^N$ is a set of inputs to Bob. We now exposit what Algorithm 1 is doing, with reference to the high-level overview in Section 4.3.1.

On input $z \in \{0,1\}^N$, the node variable v traces a root-to-leaf path (of length at most k) in the protocol tree, which is used to determine which z_i bits to query, and when. The set $A \times B$ is the "cleaned up" subrectangle of R_v (so we maintain $A \subseteq X^v$ and $B \subseteq Y^v$). We maintain the invariant that every $(x, y) \in A \times B$ is consistent with the results of the queries made so far (i.e., $g^N(x, y)$ agrees with z on queried coordinates), or in other words, $A_{\{i\}} \times B_{\{i\}}$ is z_i-monochromatic in g for $i \in [N] \smallsetminus I$. Thus we never need to worry about any coordinate that has previously been queried. The interesting structure in the sets A and B is what they look like on the unqueried coordinates, i.e., the projections A_I and B_I. Since all $2^{|I|}$ settings of the unqueried bits of z remain possible, we must maintain that all these settings are indeed possible outcomes of $g^{|I|}$ on points in $A_I \times B_I$. In fact we maintain a stronger property that turns out to entail this, namely that A_I is thick ($\mathrm{MinDeg}_i(A_I) \geq m^{17/20}$ for every $i \in I$) and

Algorithm 1: Simulation algorithm for Theorem 4.3

```
   Input: $z \in \{0,1\}^N$
   Output: $f(z)$
1  initialize $v =$ root, $I = [N]$, $A = [m]^N$, $B = (\{0,1\}^m)^N$
2  while $v$ is not a leaf do
3      if $\text{AvgDeg}_i(A_I) \geq m^{19/20}$ for all $i \in I$ then
4          let $v_0, v_1$ be the children of $v$
5          if Alice sends a bit at $v$ then
6              let $b \in \{0,1\}$ be such that $\alpha\big((A \cap X^{v,b})_I\big) \leq \alpha(A_I) + 1$
7              let $A' \subseteq (A \cap X^{v,b})_I$ be such that
8                  (1) $A'$ is thick
9                  (2) $\alpha(A') \leq \alpha\big((A \cap X^{v,b})_I\big) + 1$
10             update $A = \big\{x \in A \cap X^{v,b} : (x_i)_{i \in I} \in A'\big\}$ and $v = v_b$  (so now $A_I = A'$)
11         else if Bob sends a bit at $v$ then
12             let $b \in \{0,1\}$ be such that $\beta\big((B \cap Y^{v,b})_I\big) \leq \beta(B_I) + 1$
13             update $B = B \cap Y^{v,b}$ and $v = v_b$
14         end
15     else if $\text{AvgDeg}_i(A_I) < m^{19/20}$ for some $i \in I$ then
16         query $z_i$
17         let $U \times V \subseteq [m] \times \{0,1\}^m$ be a $z_i$-monochromatic rectangle of $g$ such that
18             (1) $A^{i,U}_{I \smallsetminus \{i\}}$ is thick
19             (2) $\alpha\big(A^{i,U}_{I \smallsetminus \{i\}}\big) \leq \alpha(A_I) - (\log m)/20$
20             (3) $\beta\big(B^{i,V}_{I \smallsetminus \{i\}}\big) \leq \beta(B_I) + 1$
21         update $A = A^{i,U}$, $B = B^{i,V}$, and $I = I \smallsetminus \{i\}$
22     end
23 end
24 output the same value that $v$ does
```

B_I is "large" (as measured by $\beta(B_I)$). The potential function is $\alpha(A_I)$; i.e., we look at the set of all projections of elements of A to the unqueried coordinates, and we consider how large this set is compared to its domain $[m]^{|I|}$. Smaller potential corresponds to a larger set.

We caution that the sets A and B in the statements of the Thickness Lemma and Projection Lemma will not be the $A \subseteq [m]^N$ and $B \subseteq (\{0,1\}^m)^N$ maintained by the algorithm, but rather will be subsets of the projected spaces $([m]^N)_I = [m]^n$ and $((\{0,1\}^m)^N)_I = (\{0,1\}^m)^n$, where n is the size of I.

Lines 2–23 are the main loop, with each iteration being either a "communication iteration" (if line 3 holds) in which we update v, A, B, or a "query iteration" (if line 15 holds) in which we update I, A, B. The type of iteration is determined by $\min_{i \in I} \text{AvgDeg}_i(A_I)$, which is our measure of how much the values of the unqueried coordinates are unpredictable from each other within $A \times B$.

In a communication iteration, there are two subcases depending on whether it is Alice's turn (line 5) or Bob's turn (line 11) to communicate. In either subcase, the bit of communication partitions R_v (and hence $A \times B$) into two sides, and we restrict our attention to the "bigger" side (lines 6 and 12) by having the communication protocol "send" the corresponding bit. Here, "bigger" is actually in terms of the projections A_I and B_I. This ensures the potential does not increase too much if Alice sends, and B_I stays large enough if Bob sends. However, if Alice sends, then the restriction to the bigger side may destroy the thickness invariant, and the Thickness Lemma is used (lines 7–9) to repair this.

In a query iteration, we have the decision tree query a bit z_i for which $\mathrm{AvgDeg}_i(A_I)$ is too small (line 16). Then we can use the Projection Lemma (lines 17–20) to restrict $A \times B$ to a subrectangle on which the i-th output bit of g^N is fixed to z_i (for either possible value of $z_i \in \{0,1\}$); this exploits the fact that $\mathrm{MinDeg}_i(A_I)$ is large by the thickness invariant. Furthermore, the fact that $\mathrm{AvgDeg}_i(A_I)$ is small allows us to ensure a $\Omega(\log m)$ decrease in potential (i.e., the density of A_I increases). (Although the absolute size of A_I decreases, recall that the measure $\alpha(A_I)$ is relative to the current set I; by fixing the i-th coordinate, I becomes $I \smallsetminus \{i\}$, and since we fixed a coordinate of small average degree, the density projected to $I \smallsetminus \{i\}$ will increase a lot.)

4.3.4 Analysis of the simulation algorithm

We now formally argue that Algorithm 1 witnesses the Simulation Theorem (assuming the Thickness Lemma and the Projection Lemma). Assuming lines 7–9 and 17–20 always succeed (which we argue below), in each iteration one of the following three cases occurs.

- If lines 3 and 5 hold, then $\alpha(A_I)$ increases by ≤ 2 and $\beta(B_I)$ stays the same.
- If lines 3 and 11 hold, then $\alpha(A_I)$ stays the same and $\beta(B_I)$ increases by ≤ 1.
- If line 15 holds, then $\alpha(A_I)$ decreases by $\geq (\log m)/20$ and $\beta(B_I)$ increases by ≤ 1.

Since there are at most k iterations in which line 3 holds, and since $\alpha(A_I)$ is initially 0 and always nonnegative, it follows that there are at most $40k/\log m$ iterations in which line 15 holds, and hence the decision tree makes at most $40k/\log m$ queries. Moreover, since there are at most $k + 40k/\log m \leq m^{2/20}$ iterations, and $\beta(B_I)$ is initially 0, at all times we have $\beta(B_I) \leq m^{2/20}$.

Claim 4.8. *Lines 7–9 and 17–20 always succeed, and the following loop invariants are maintained.*

(i) *A_I is thick.*
(ii) *$A \times B \subseteq R_v$.*
(iii) *$g(x_i, y_i) = z_i$ for all $(x,y) \in A \times B$ and all $i \in [N] \smallsetminus I$.*

Proof. The invariants trivially hold initially. Now assume they hold at the beginning of an iteration.

Suppose lines 3 and 5 hold. For all $i \in I$, we have $\mathrm{AvgDeg}_i\big((A \cap X^{v,b})_I\big) = \big|(A \cap X^{v,b})_I\big| \,/\, \big|(A \cap X^{v,b})_{I \smallsetminus \{i\}}\big| \geq (|A_I|/2) \,/\, |A_{I \smallsetminus \{i\}}| = \mathrm{AvgDeg}_i(A_I)/2 \geq m^{19/20}/2$. Thus we may apply the Thickness Lemma with $(A \cap X^{v,b})_I$ (in place of A in the lemma), I identified with $[n]$, and $d := m^{19/20}/2$

(noting that $d/2n \geq m^{19/20}/4m^{1/20} \geq m^{17/20}$) to conclude that lines 7–9 succeed, and hence (i) is maintained. Also, (ii) is maintained by line 10. If lines 3 and 11 hold, then (i) is trivially maintained and (ii) is maintained by line 13. Supposing line 3 holds, in either case (iii) is maintained since the new $A \times B$ is a subset of the old $A \times B$ and I is unchanged.

Now suppose line 15 holds. Since (i) holds and $\beta(B_I) \leq m^{2/20}$,[1] we may apply the Projection Lemma with A_I and B_I (in place of A and B in the lemma), I identified with $[n]$, and $b \coloneqq z_i$ (noting that $-\log m + \log \operatorname{AvgDeg}_i(A_I) \leq -(\log m)/20$) to conclude that lines 17–20 succeed, and hence (i) is maintained. The new $A \times B$ is a subset of the old $A \times B$; therefore, (ii) is maintained since v is unchanged, and (iii) is maintained since $U \times V$ is z_i-monochromatic in g. □

Let v be the leaf reached at termination. We claim that there exists an $(x, y) \in R_v$ such that $g^N(x, y) = z$, and hence the algorithm indeed outputs $f(z) = F(x, y)$. Imagine that instead of terminating, the algorithm continues by executing lines 16–21 repeatedly, once for each remaining coordinate $i \in I$ in arbitrary order until only one coordinate remains unqueried—except that we ignore condition (2) (line 19). In this "extended" execution there are a total of $k + N - 1 \leq m^{2/20}$ iterations, so we have $\beta(B_I) \leq m^{2/20}$ at all times, and thus as in the proof of Claim 4.8, the application of the Projection Lemma always succeeds and invariants (i), (ii), (iii) are maintained.

Consider the state (i.e., v, I, A, B) at the end of this extended execution. Then I is a singleton, say $\{1\}$, and $|A_{\{1\}}| = \operatorname{MinDeg}_1(A_{\{1\}}) \geq m^{17/20}$ by (i), and $|B_{\{1\}}| \geq 2^{-m^{2/20}} \cdot 2^m = 2^{m - m^{2/20}}$. Hence $A_{\{1\}} \times B_{\{1\}}$ is not monochromatic in g, since the largest monochromatic rectangle with rows $A_{\{1\}}$ has at most $2^{m - |A_{\{1\}}|} < |B_{\{1\}}|$ columns. Pick an $(x_1, y_1) \in A_{\{1\}} \times B_{\{1\}}$ such that $g(x_1, y_1) = z_1$, and pick an $(x, y) \in A \times B$ with this value of (x_1, y_1). By (ii) we have $(x, y) \in R_v$, and by (iii) we also have $g(x_i, y_i) = z_i$ for all $i \in [N] \smallsetminus \{1\}$, and thus $g^N(x, y) = z$. The correctness is established.

4.3.5 Proof of the Thickness Lemma

The Thickness Lemma is witnessed by Algorithm 2, which constructs a sequence $A = A^0 \supseteq A^1 \supseteq A^2 \supseteq \cdots$ that converges to the desired set A'.

Algorithm 2: Algorithm for Lemma 4.6

```
1 let $A^0 \coloneqq A$
2 for $j = 0, 1, 2, \ldots$ do
3     if $\operatorname{MinDeg}_i(A^j) \geq d/2n$ for all $i \in [n]$ then stop and output $A' \coloneqq A^j$
4     let $i$ be such that $\operatorname{MinDeg}_i(A^j) < d/2n$, and assume $i = 1$ for convenience of notation
5     let $(x_2^*, \ldots, x_n^*)$ be a nonzero-degree right node in $\operatorname{Graph}_1(A^j)$ with degree $< d/2n$
6     let $A^{j+1} \coloneqq A^j \smallsetminus \{(x_1, x_2^*, \ldots, x_n^*) : x_1 \in [m]\}$
7 end
```

[1] There is no circular reasoning here; in showing that $\beta(B_I) \leq m^{2/20}$ we just needed that lines 7–9 and 17–20 succeeded in all iterations before this one.

If the algorithm terminates, then A' satisfies (1). We just need to argue that it does terminate, moreover with $|A'| \geq |A|/2$ (which is equivalent to (2)). In an iteration, it obtains $\mathrm{Graph}_i(A^{j+1})$ from $\mathrm{Graph}_i(A^j)$ by removing all edges incident to some right node in $A^j_{[n]\smallsetminus\{i\}}$. Hence $\left|A^{j+1}_{[n]\smallsetminus\{i\}}\right| = \left|A^j_{[n]\smallsetminus\{i\}}\right| - 1$, and for every $i' \neq i$, $\left|A^{j+1}_{[n]\smallsetminus\{i'\}}\right| \leq \left|A^j_{[n]\smallsetminus\{i'\}}\right|$. Therefore, the total number of iterations is at most $\sum_{i=1}^n |A_{[n]\smallsetminus\{i\}}| = \sum_{i=1}^n |A|/\operatorname{AvgDeg}_i(A) \leq n \cdot |A|/d$. Since $|A^{j+1}| > |A^j| - d/2n$ in each iteration, in total at most $(n \cdot |A|/d) \cdot (d/2n) = |A|/2$ elements of A can be removed throughout the execution. Thus the algorithm must terminate with $|A'| \geq |A|/2$.

4.3.6 Proof of the Projection Lemma

Assume $i = n$ for convenience of notation, so $A^{i,U}_{[n]\smallsetminus\{i\}} = A^{n,U}_{[n-1]} = \big\{(x_1, \ldots, x_{n-1}) : (x_1, \ldots, x_n) \in A \text{ for some } x_n \in U\big\}$ (which is the set of right nodes in $\mathrm{Graph}_n(A)$ that have a neighbor in U) and $B^{i,V}_{[n]\smallsetminus\{i\}} = B^{n,V}_{[n-1]} = \big\{(y_1, \ldots, y_{n-1}) : (y_1, \ldots, y_n) \in B \text{ for some } y_n \in V\big\}$.

We claim that if we take a uniformly random $U \subseteq [m]$ of size $m^{7/20}$ and let $V \coloneqq \big\{w \in \{0,1\}^m : w_j = b \text{ for all } j \in U\big\}$, then

(0) $A^{n,U}_{[n-1]} = A_{[n-1]}$ with probability greater than $1 - 2^{-m^{3/20}}$,
(1) $A_{[n-1]}$ is thick,
(2) $\alpha(A_{[n-1]}) \leq \alpha(A) - \log m + \log \operatorname{AvgDeg}_n(A)$,
(3) $\beta\big(B^{n,V}_{[n-1]}\big) \leq \beta(B) + 1$ with probability greater than $2^{-m^{3/20}}$.

The Projection Lemma then follows by a union bound. (We mention that our argument for property (3) is substantially different from and simpler than the corresponding part of the proof in [RM99].)

Property (0). For every nonzero-degree right node $(x_1, \ldots, x_{n-1}) \in A_{[n-1]}$ of $\mathrm{Graph}_n(A)$, let $L_{x_1,\ldots,x_{n-1}} \coloneqq \big\{x_n \in [m] : (x_1, \ldots, x_{n-1}, x_n) \in A\big\}$ denote the set of all left nodes adjacent to it. We have $|L_{x_1,\ldots,x_{n-1}}| \geq \operatorname{MinDeg}_n(A) \geq m^{17/20}$, and $(x_1, \ldots, x_{n-1}) \in A^{n,U}_{[n-1]}$ iff U intersects $L_{x_1,\ldots,x_{n-1}}$. Since U has size $m^{7/20}$, the probability U does not intersect $L_{x_1,\ldots,x_{n-1}}$ is at most $(1 - m^{17/20}/m)^{m^{7/20}} \leq e^{-m^{4/20}}$. Since the number of elements $(x_1, \ldots, x_{n-1}) \in A_{[n-1]}$ is at most $m^{n-1} \leq 2^{m^{1/20} \cdot \log m}$, by a union bound the probability that one of them is not in $A^{n,U}_{[n-1]}$ is at most $2^{m^{1/20} \cdot \log m} \cdot e^{-m^{4/20}} < 2^{-m^{3/20}}$.

Property (1). For this it suffices to show that $\operatorname{MinDeg}_j(A_{[n-1]}) \geq \operatorname{MinDeg}_j(A)$ for all $j \in [n-1]$. Assume $j = n-1$ for convenience of notation. For every nonzero-degree right node $(x_1, \ldots, x_{n-2})$ in $\mathrm{Graph}_{n-1}(A_{[n-1]})$, there exists x_{n-1} such that $(x_1, \ldots, x_{n-2}, x_{n-1}) \in A_{[n-1]}$. Thus by the definition of $A_{[n-1]}$ there exists x_n such that $(x_1, \ldots, x_{n-2}, x_{n-1}, x_n) \in A$. Therefore, by the definition of $\operatorname{MinDeg}_{n-1}(A)$ applied to the nonzero-degree right node $(x_1, \ldots, x_{n-2}, x_n)$ of $\mathrm{Graph}_{n-1}(A)$, we have that $(x_1, \ldots, x_{n-2}, x'_{n-1}, x_n) \in A$ holds for at least $\operatorname{MinDeg}_{n-1}(A)$ different elements x'_{n-1}. All these elements satisfy $(x_1, \ldots, x_{n-2}, x'_{n-1}) \in A_{[n-1]}$. Hence, the degree of the right node $(x_1, \ldots, x_{n-2})$ in $\mathrm{Graph}_{n-1}(A_{[n-1]})$ is at least $\operatorname{MinDeg}_{n-1}(A)$.

Property (2). We have $|A_{[n-1]}| = |A|/\operatorname{AvgDeg}_n(A)$ and $|[m]^{n-1}| = |[m]^n|/m$, and hence $\alpha(A_{[n-1]}) = \log\bigl(|[m]^{n-1}|/|A_{[n-1]}|\bigr) = \log\bigl(|[m]^n|/|A|\bigr) - \log\bigl(m/\operatorname{AvgDeg}_n(A)\bigr) = \alpha(A) - \log m + \log \operatorname{AvgDeg}_n(A)$. (Thus (2) holds with equality, but we only needed the inequality.)

Property (3). We first state a claim, whose proof we give later.

Claim 4.9. *For every $W \subseteq \{0,1\}^m$ with $\beta(W) \le m^{11/20}$, we have $\mathbf{Pr}_U[V \cap W \neq \emptyset] \ge 3/4$.*

In particular, for every $W \subseteq \{0,1\}^m$ we have $\mathbf{Pr}_U[V \cap W \neq \emptyset] \ge \frac{3}{4} \cdot |W|/2^m - 2^{-m^{11/20}}$. For every $(y_1, \dots, y_{n-1}) \in (\{0,1\}^m)^{n-1}$, let $W_{y_1,\dots,y_{n-1}} \coloneqq \bigl\{y_n \in \{0,1\}^m : (y_1, \dots, y_{n-1}, y_n) \in B\bigr\}$. Letting $(y_1, \dots, y_{n-1})$ be uniformly random in $(\{0,1\}^m)^{n-1}$, we have

$$\begin{aligned}
\mathbf{E}_U\Bigl[\bigl|B^{n,V}_{[n-1]}\bigr|/2^{m(n-1)}\Bigr] &= \mathbf{E}_{y_1,\dots,y_{n-1}}\ \mathbf{Pr}_U\Bigl[(y_1, \dots, y_{n-1}) \in B^{n,V}_{[n-1]}\Bigr] \\
&= \mathbf{E}_{y_1,\dots,y_{n-1}}\ \mathbf{Pr}_U\bigl[V \cap W_{y_1,\dots,y_{n-1}} \neq \emptyset\bigr] \\
&\ge \mathbf{E}_{y_1,\dots,y_{n-1}}\ \Bigl(\tfrac{3}{4} \cdot \bigl|W_{y_1,\dots,y_{n-1}}\bigr|/2^m - 2^{-m^{11/20}}\Bigr) \\
&= \tfrac{3}{4} \cdot |B|/2^{mn} - 2^{-m^{11/20}} \\
&\ge \tfrac{5}{8} \cdot |B|/2^{mn}
\end{aligned}$$

where the last line follows since $|B|/2^{mn} = 2^{-\beta(B)} \ge 2^{-m^{2/20}}$. It follows that with probability at least $\frac{1}{8} \cdot |B|/2^{mn} > 2^{-m^{3/20}}$ over U, we have $\bigl|B^{n,V}_{[n-1]}\bigr|/2^{m(n-1)} \ge \frac{1}{2} \cdot |B|/2^{mn}$, which is equivalent to (3). This finishes the proof of the Projection Lemma, except for the proof of Claim 4.9.

Recall that $b \in \{0,1\}$ is fixed. For $W \subseteq \{0,1\}^m$ and $j \in [m]$, define $W^j \coloneqq \{w \in W : w_j = b\}$ and $\operatorname{Bad}(W) \coloneqq \{j \in [m] : |W^j| < |W|/4\}$.

Claim 4.10. *For every $W \subseteq \{0,1\}^m$, $|\operatorname{Bad}(W)| \le 6\beta(W)$.*

Proof of Claim 4.10. Let w be a random variable uniformly distributed over W, and let $H(\cdot)$ denote Shannon entropy. There are at most $6\beta(W)$ coordinates j such that $\mathbf{Pr}[w_j = b] < 1/4$, since otherwise $H(w) \le \sum_{j=1}^m H(w_j) < 6\beta(W) \cdot H(1/4) + (m - 6\beta(W)) \cdot 1 \le m - 6\beta(W) \cdot (1 - 0.82) \le m - \beta(W)$, contradicting the fact that $H(w) = \log|W| = m - \beta(W)$. □

Proof of Claim 4.9. Suppose we sample $U \coloneqq \{j_1, \dots, j_{m^{7/20}}\}$ by iteratively picking each $j_{i+1} \in [m] \smallsetminus \{j_1, \dots, j_i\}$ uniformly at random. We write V as V_U, as a reminder that it depends on U. For $i \in \{0, 1, \dots, m^{7/20}\}$, define $W_i \coloneqq \bigl\{w \in W : w_{j_1} = w_{j_2} = \cdots = w_{j_i} = b\bigr\}$, and note that $W_0 = W$, $W_{i+1} = W_i^{j_{i+1}}$, and $W_{m^{7/20}} = V_U \cap W$. Let E_{i+1} denote the event that $j_{i+1} \notin \operatorname{Bad}(W_i)$, and note that if E_{i+1} occurs then $\beta(W_{i+1}) \le \beta(W_i) + 2$. Thus if $E_1 \cap \cdots \cap E_{m^{7/20}}$ occurs then $\beta(V_U \cap W) \le \beta(W) + 2m^{7/20} < \infty$ and hence $V_U \cap W \neq \emptyset$. Conditioned on any particular outcome of $j_1, \dots, j_i$ for which $E_1 \cap \cdots \cap E_i$ occurs, by Claim 4.10 we have $|\operatorname{Bad}(W_i)| \le 6\beta(W_i) \le 6(\beta(W) + 2i)$ and thus

$$\mathbf{Pr}\bigl[E_{i+1} \mid j_1, \dots, j_i\bigr] \ \ge\ 1 - \frac{|\operatorname{Bad}(W_i)|}{m - i} \ \ge\ 1 - \frac{6(\beta(W) + 2i)}{(6/7)m} \ \ge\ e^{-14(\beta(W)+2i)/m}$$

where the last inequality uses the fact that $1 - x \geq e^{-2x}$ if $x \in [0, 1/2]$, applied to $x \coloneqq 6(\beta(W) + 2i)/(6/7)m \leq 7(m^{11/20} + 2m^{7/20})/m \leq 1/2$. We conclude that

$$\begin{aligned}
\mathbf{Pr}[V_U \cap W \neq \emptyset] &\geq \mathbf{Pr}\big[E_1 \cap \cdots \cap E_{m^{7/20}}\big] \\
&= \textstyle\prod_{i=0}^{m^{7/20}-1} \mathbf{Pr}\big[E_{i+1} \mid E_1 \cap \cdots \cap E_i\big] \\
&\geq \textstyle\prod_{i=0}^{m^{7/20}-1} e^{-14(\beta(W)+2i)/m} \\
&= \exp\big(-\textstyle\sum_{i=0}^{m^{7/20}-1} 14(\beta(W) + 2i)/m\big) \\
&= \exp\big(-\tfrac{14}{m}\big(\beta(W)m^{7/20} + (m^{7/20} - 1)m^{7/20}\big)\big) \\
&\geq \exp\big(-14\big(m^{-2/20} + m^{-6/20}\big)\big) \\
&\geq 3/4. \qquad \square
\end{aligned}$$

Chapter 5

Randomised Communication vs. Partition Number

Overview. In this chapter, we show that *randomised* communication complexity can be superlogarithmic in the partition number of the associated communication matrix, and we obtain near-optimal *randomised* lower bounds for the Clique vs. Independent Set problem. These results strengthen the deterministic lower bounds obtained in Chapter 4. This chapter is based on the following publication:

[GJPW15]: Mika Göös, T.S. Jayram, Toniann Pitassi, and Thomas Watson. Randomized communication vs. partition number. Technical Report TR15-169, Electronic Colloquium on Computational Complexity (ECCC), 2015. URL: http://eccc.hpi-web.de/report/2015/169/

5.1 Introduction

In Chapter 4 we exhibited a boolean function $F\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$ whose deterministic communication complexity is superlogarithmic in the *partition number*

$$\chi(F) \;\coloneqq\; \chi_0(F)+\chi_1(F)$$

where $\chi_i(F)$ is the least number of rectangles (sets of the form $A\times B$ where $A\subseteq\mathcal{X}$, $B\subseteq\mathcal{Y}$) needed to partition the set $F^{-1}(i)$. In this chapter, we upgrade the lower-bound results from Chapter 4 to hold against randomised protocols—here the notation $\tilde\Omega(m)$ hides factors polylogarithmic in m.

Theorem 5.1. *There is an F with randomised communication complexity $\tilde\Omega(\log^{1.5}\chi(F))$.*

Theorem 5.2. *There is an F with randomised communication complexity $\tilde\Omega(\log^2\chi_1(F))$.*

5.1.1 Implications

Theorem 5.1: Previously, no examples of F were known with randomised communication complexity larger than $\log \chi(F)$. In fact, such a separation cannot be obtained using the usual rectangle-based lower-bound methods, as catalogued by Jain and Klauck [JK10]. In particular, Theorem 5.1 shows that randomised complexity can be polynomially larger than the *partition bound* [JK10, JLV14], which is one of the most powerful general lower bound methods for randomised communication. (Consequently, our proof of Theorem 5.1 has to exploit another powerful lower-bound method, namely *information complexity.*) Note also that every F has deterministic communication complexity at least $\log \chi(F)$ and at most $O(\log^2 \chi(F))$, where the latter upper bound is a classical result of [AUY83]. Theorem 5.1 shows that the upper bound cannot be improved much even if we allow randomisation.

Theorem 5.2: The relationship between $\chi_1(F)$ and the communication complexity of F can be equivalently formulated in the language of the *Clique vs. Independent Set* game, played on a graph derived from F (Alice holds a clique, Bob holds an independent set: do they intersect?). See [Yan91, §4] or [Juk12, §4.4] for the equivalence. Yannakakis [Yan91] (extending [AUY83]) proved that every F has deterministic communication complexity at most $O(\log^2 \chi_1(F))$. Our Theorem 5.2 shows that this upper bound is essentially tight even if we allow randomised protocols, and it implies that there is a graph on n nodes for which Clique vs. Independent Set requires $\tilde{\Omega}(\log^2 n)$ randomised communication. (The deterministic upper bound $O(\log^2 n)$ holds for all graphs.)

Extension complexity. In fact, we prove Theorem 5.2 by showing that (the negation of) the function F has high *approximate nonnegative rank* (a.k.a. smooth rectangle bound; see Section 5.2 for definitions). One consequence in the field of *extended formulations* (see [Yan91, FMP⁺15] for definitions) is that we obtain a graph G such that the polytope generated by the so-called "clique inequalities" of G has extension complexity $n^{\tilde{\Omega}(\log n)}$. (The slack matrix associated with the clique inequalities is simply (the negation of) the Clique vs. Independent Set game. These inequalities capture the independent set polytope of G when G is perfect—our graph G however is not.) The previous bound in this direction was $n^{\Omega(\log^{0.128} n)}$ from Chapter 3. Technically speaking, the lower bound from Chapter 3 was proved for *nondeterministic* communication complexity, so the full result remains incomparable with Theorem 5.2.

Log-rank conjecture. The famous log-rank conjecture of Lovász and Saks [LS88] postulates that the deterministic communication complexity of F is polynomially related to $\log \operatorname{rank}(F)$. Gavinsky and Lovett [GL14] have shown that the conjecture is equivalent to asking whether the randomised communication complexity of F is polynomially bounded in $\log \operatorname{rank}(F)$. Here our Theorem 5.2 gives at least a near-quadratic separation between the randomised communication complexity of F and $\log \operatorname{rank}(F) \leq \log \chi_1(F)$; the previous best lower bound was $\Omega(\log^{1.63} \operatorname{rank}(F))$ due to Kushilevitz [Kus94]. Furthermore, Troy Lee has pointed out to us that our construction underlying Theorem 5.2 exhibits a nearly a 4th-power separation between the logarithms of *approximate nonnegative rank* and *approximate rank*. This gives lower bounds for the so-called *log-approximate-rank conjecture* [LS07, Conjecture 42], which is the randomised analogue of the log-rank conjecture. The previous best separation was quadratic (as witnessed by the set-disjointness problem).

Subsequent work. The 1.5-th power separation in Theorem 5.1 has been subsequently improved to near-quadratic in [ABB$^+$16b], which is optimal. The work [ABB$^+$16b] builds on the techniques developed in this work. In particular, they iteratively apply our Corollary 5.18 to analyze a communication analogue of a query complexity construction due to Ambainis, Kokainis, and Kothari [AKK15].

5.1.2 Our techniques

The basic strategy in Chapter 4 for obtaining the deterministic versions of Theorems 5.1–5.2 was to first obtain analogous gaps in the easier-to-understand world of query complexity, then "lift" the results to communication complexity using a so-called *simulation lemma*. For getting randomised lower bounds, two obstacles immediately present themselves: (i) The functions studied in Chapter 4 are too easy for randomised protocols (as shown by [MS15]). (ii) There is no known simulation lemma for the bounded-error randomised setting.

To handle obstacle (i), we modify the functions from Chapter 4 in a way that preserves their low partition numbers while eliminating the structure that was exploitable by randomised protocols. (Similar constructions have been given by [ABB$^+$16a, ABK16].) To handle obstacle (ii) for Theorem 5.2, we actually prove a lower bound for a model that is stronger than the standard randomised model, but for which there *is* a known simulation lemma, namely that of Chapter 2. This idea alone does not handle obstacle (ii) for Theorem 5.1, though. For that, we start by giving a proof of the query complexity analogue of Theorem 5.1, then develop a way to *mimic* that argument using communication complexity, by going through information complexity (exploiting machinery from [KLL$^+$12] and [BW15a]). In the process, this yields a corollary (Corollary 5.18) that is of independent interest: information complexity under arbitrary distributions is essentially equivalent to information complexity under distributions that are only over 1-inputs (or only over 0-inputs).

5.2 Complexity measures

We study the following communication complexity models/measures; see Figure 5.1. For any complexity measure $\mathcal{C}$ we write $\mathsf{co}\mathcal{C}(F) \coloneqq \mathcal{C}(\neg F)$ and $2\mathcal{C}(F) \coloneqq \max\{\mathcal{C}(F), \mathsf{co}\mathcal{C}(F)\}$ for short.

- **P^{cc}:** The deterministic communication complexity of F is denoted $\mathsf{P}^{\mathsf{cc}}(F)$.
- **$\mathsf{BPP}^{\mathsf{cc}}$:** The randomised communication complexity of F is denoted $\mathsf{BPP}^{\mathsf{cc}}(F)$.

- **$\mathsf{UP}^{\mathsf{cc}}$:** Recall (e.g., [KN97, Juk12]) that a cost-c nondeterministic protocol for F corresponds to a covering (allowing overlaps) of $F^{-1}(1)$ with 2^c rectangles. A nondeterministic protocol is *unambiguous* if on every 1-input there is a unique accepting computation; combinatorially, this means we have a disjoint covering (partition) of $F^{-1}(1)$. We define $\mathsf{UP}^{\mathsf{cc}}(F) \coloneqq \lceil \log \chi_1(F) \rceil$. Thus $\mathsf{coUP}^{\mathsf{cc}}(F) = \lceil \log \chi_0(F) \rceil$, and $2\mathsf{UP}^{\mathsf{cc}}(F) \in \lceil \log \chi(F) \rceil \pm 1$.

- **$\mathsf{WAPP}^{\mathsf{cc}}$:** Abstractly speaking, a WAPP computation (*Weak Almost-Wide* PP; introduced in [BGM06]) is a randomised computation that accepts 1-inputs with probability in $[(1-\epsilon)\alpha, \alpha]$, and 0-inputs with probability in $[0, \epsilon\alpha]$, where $\epsilon < 1/2$ is an error parameter and $\alpha = \alpha(n) > 0$ is arbitrary.

  Instantiating this for protocols, we define $\mathsf{WAPP}^{\mathsf{cc}}_{\epsilon}(F)$ as the least "cost" of a randomised (public-coin) protocol Π that computes F in the above sense; the "cost" of a protocol Π with parameter α is defined as the usual communication cost (number of bits communicated) plus $\log(1/\alpha)$. In this definition, we may assume w.l.o.g. that Π is *zero-communication* [KLL$^+$12]: Π is simply a probability distribution over rectangles R, and Π accepts an input (x, y) iff $(x, y) \in R$ for the randomly chosen R. Such a protocol Π exchanges only 2 bits to check the condition $(x, y) \in R$, and the rest of the cost is coming from having a tiny α.

  We note that $\mathsf{WAPP}^{\mathsf{cc}}$ corresponds to the (one-sided) *smooth rectangle bound* of [JK10], which is known to be equivalent to *approximate nonnegative rank* [KMSY14]. A consequence of this equivalence is that $\mathsf{WAPP}^{\mathsf{cc}}$ could alternatively be defined without charging anything for $\alpha > 0$, as long as we restrict our protocols to be *private-coin*. Also, $2\mathsf{WAPP}^{\mathsf{cc}}$ is equivalent to the *relaxed partition bound* of [KLL$^+$12] (we elaborate on this in Section 5.5.2). We remark that $\mathsf{WAPP}^{\mathsf{cc}}$ is not amenable to efficient amplification of the error parameter; there can be an exponential gap between $\mathsf{WAPP}^{\mathsf{cc}}_{\epsilon}$ and $\mathsf{WAPP}^{\mathsf{cc}}_{\delta}$ for different constants ϵ and δ, at least for partial functions (Chapter 2).

For a boolean function $f\colon \{0,1\}^n \to \{0,1\}$ we consider the following decision tree models/measures:

- **P^{dt}:** The deterministic decision tree complexity of f is denoted $\mathsf{P}^{\mathsf{dt}}(f)$.

- **$\mathsf{BPP}^{\mathsf{dt}}$:** The randomised decision tree complexity of f is denoted $\mathsf{BPP}^{\mathsf{dt}}(f)$.

- **$\mathsf{UP}^{\mathsf{dt}}$:** A nondeterministic decision tree is a DNF formula. We think of the conjunctions in the DNF formula as *certificates*—partial assignments to inputs that force the function to be 1. The cost is the maximum number of input bits read by a certificate. A nondeterministic decision tree is *unambiguous* if on every 1-input there is a unique accepting certificate. We define $\mathsf{UP}^{\mathsf{dt}}(f)$ as the least cost of an unambiguous decision tree for f. Other works that have studied unambiguous decision trees include [Sav02, Bel06, KRS15].

- **$\mathsf{WAPP}^{\mathsf{dt}}$:** We define $\mathsf{WAPP}^{\mathsf{dt}}_{\epsilon}(f)$ as the least height of a randomised decision tree that accepts 1-inputs with probability in $[(1-\epsilon)\alpha, \alpha]$, and 0-inputs with probability in $[0, \epsilon\alpha]$,

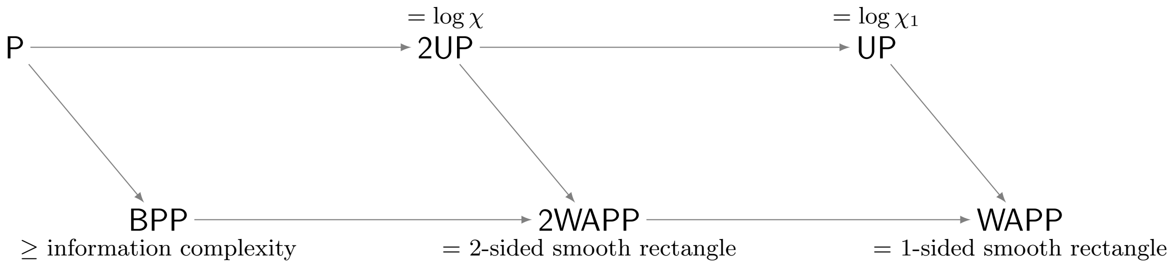


Figure 5.1: Models of computation that can be instantiated for both communication and query complexity. Here $\mathsf{A} \longrightarrow \mathsf{B}$ means that model B can simulate model A without any overhead.

where $\alpha = \alpha(n) > 0$ is arbitrary. (Note that only the number of queries matters; we do not charge for α being small.) Like the communication version, this measure is not amenable to efficient amplification of the error parameter (Chapter 2).

The analogue of a $\mathsf{WAPP}^{\mathsf{cc}}$ protocol being w.l.o.g. a distribution over rectangles is that a $\mathsf{WAPP}^{\mathsf{dt}}$ decision tree is w.l.o.g. a distribution over conjunctions. This implies that we may characterize $\mathsf{WAPP}^{\mathsf{dt}}_{\epsilon}(f)$ using *conical juntas*: A *conical junta* h is a nonnegative linear combination of conjunctions. That is, $h = \sum w_C C$ where the sum ranges over conjunctions $C\colon \{0,1\}^n \to \{0,1\}$ and $w_C \geq 0$ for all C. Then $\mathsf{WAPP}^{\mathsf{dt}}_{\epsilon}(f)$ is the least *degree* (maximum width of a conjunction with positive weight in h) of a conical junta h that ϵ-approximates f in the sense that $h(z) \in [1-\epsilon, 1]$ for all $z \in f^{-1}(1)$, and $h(z) \in [0, \epsilon]$ for all $z \in f^{-1}(0)$. Other works have studied conical juntas under such names as the (one-sided) *partition bound for query complexity* [JK10] and *query complexity in expectation* [KLdW15].

5.3 Overview

In this section we give an outline for obtaining our main results, Theorems 5.1–5.2. For complexity models/measures $\mathcal{C}$ and $\mathcal{C}'$, we informally say "$\mathcal{C}$-vs-$\mathcal{C}'$ gap" to mean the existence of a function whose $\mathcal{C}$ complexity is significantly higher than its $\mathcal{C}'$ complexity. Using the notation defined in Section 5.2, we can rephrase our main results as follows.

Theorem 5.1 ($\mathsf{BPP}^{\mathsf{cc}}$-vs-$\mathsf{2UP}^{\mathsf{cc}}$). *There is an F such that* $\mathsf{BPP}^{\mathsf{cc}}(F) \geq \tilde{\Omega}(\mathsf{2UP}^{\mathsf{cc}}(F)^{1.5})$.

Theorem 5.2 ($\mathsf{BPP}^{\mathsf{cc}}$-vs-$\mathsf{UP}^{\mathsf{cc}}$). *There is an F such that* $\mathsf{BPP}^{\mathsf{cc}}(F) \geq \tilde{\Omega}(\mathsf{UP}^{\mathsf{cc}}(F)^2)$.

(§ 5.3.1) **Tribes-List:** Our starting point is to define *Tribes-List*, a variant of a function introduced in Chapter 4. Its purpose is to witness a BPP-vs-UP gap for query complexity.

(§ 5.3.2) **Composition:** Next, we modify Tribes-List using two types of function composition, which we call *lifting* and AND-*composition*, to obtain candidate functions for BPP-vs-$\mathsf{2UP}$ gaps in both query and communication complexity.

Unambiguous decision tree for TL:

Nondeterministically guess a column index $j \in [k]$. Consider the entries $M_{ij} = (m_{ij}, p_{ij})$ for $i \in [k]$: check that $m_{ij} = 1$ for all i and that $p_{ij} \neq \bot$ for at least one i (this is $\leq 2k$ queries). Let i be the first row index for which $p_{ij} \neq \bot$ and read the full value of p_{ij} (this is $\Theta(k \log k)$ queries). Interpret $p_{ij} \in [k]^{[k] \smallsetminus \{j\}}$ as a *list of pointers*, describing a row index for all columns other than j. For each of these $k-1$ pointed-to entries $M_{i'j'}$, check that $m_{i'j'} = 0$ (this is $k-1$ queries).

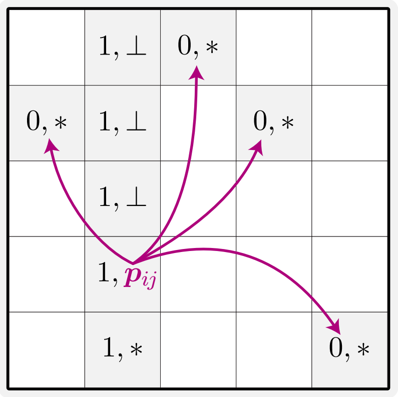


Figure 5.2: The unambiguous decision tree that defines the Tribes-List function.

(§ 5.3.3) **Overview of proofs:** With the candidate functions defined, we outline our strategy to prove the desired communication lower bounds.

5.3.1 Tribes-List

The *Tribes-List* function $\mathsf{TL}\colon \{0,1\}^n \to \{0,1\}$ is defined on $n \coloneqq \Theta(k^3 \log k)$ bits where k is a parameter. We think of the input as a $k \times k$ matrix M with entries M_{ij} taking values from the alphabet $\Sigma \coloneqq \{0,1\} \times ([k]^{k-1} \cup \{\bot\})$. Here each entry is encoded with $\Theta(k \log k)$ bits, and we assume that the encoding of $M_{ij} = (m_{ij}, p_{ij}) \in \Sigma$ is such that a single bit is used to encode the value $m_{ij} \in \{0,1\}$ and another bit is used to encode whether or not $p_{ij} = \bot$. If $p_{ij} \neq \bot$, then we can learn its exact value in $[k]^{k-1}$ by querying all the $\Theta(k \log k)$ bits.

Informally, we have $\mathsf{TL}(M) = 1$ iff M has a unique all-$(1,*)$ column (here $*$ is a wildcard) that also contains an entry with $k-1$ pointers to entries of the form $(0,*)$ in all other columns. More formally, we define TL in Figure 5.2 by describing an unambiguous decision tree of cost $\Theta(k \log k)$ computing it.

5.3.2 Composition

Given a base function witnessing some complexity gap, we will establish a different but related complexity gap by transforming the function into a more complex one via one (or both) of the following operations involving function composition: *lifting* and AND-*composition*. Lifting is used to go from a query complexity gap to an analogous communication complexity gap. AND-composition is used to go from a gap with a UP upper bound to a gap with a $\mathsf{2UP}$ upper bound. To show that an operation indeed converts one gap to another gap, we need two types of results: an observation showing how the relevant upper bounds behave under the operation, and a more difficult lemma showing how the relevant lower bounds behave under the operation.

Lifting. Recall that a decision tree for f generally yields a corresponding type of communication protocol for the composed function $f \circ g^n$:

Observation 5.3. *For all* $f\colon \{0,1\}^n \to \{0,1\}$, $g\colon \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$, *and* $\mathcal{C} \in \{\mathsf{2UP}, \mathsf{UP}\}$, *we have* $\mathcal{C}^{\mathsf{cc}}(f \circ g^n) \leq \mathcal{C}^{\mathsf{dt}}(f) \cdot O(b + \log n)$.

In the converse direction, we use a simulation theorem for WAPP from Chapter 2:

Lemma 5.4 (Simulation for WAPP). *For all* $f\colon \{0,1\}^n \to \{0,1\}$ *and constants* $0 < \epsilon < \delta < 1/2$, *we have* $\mathsf{WAPP}^{\mathsf{dt}}_{\delta}(f) \leq O\big(\mathsf{WAPP}^{\mathsf{cc}}_{\epsilon}(f \circ g^n)/\log n\big)$ *where* $g\colon \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$ *is the inner-product gadget defined as follows:* $b = b(n) := 100 \log n$, *and* $g(x_i, y_i) := \langle x_i, y_i \rangle \bmod 2$.

AND-composition. Given $f\colon \{0,1\}^n \to \{0,1\}$ we can compose it with the k-bit AND function to obtain $\mathsf{AND} \circ f^k\colon (\{0,1\}^n)^k \to \{0,1\}$ defined by $(\mathsf{AND} \circ f^k)(z_1, \ldots, z_k) = 1$ iff $f(z_i) = 1$ for all i. Similarly, given $F\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ we can obtain $\mathsf{AND} \circ F^k\colon \mathcal{X}^k \times \mathcal{Y}^k \to \{0,1\}$ defined by $(\mathsf{AND} \circ F^k)(x, y) = 1$ iff $F(x_i, y_i) = 1$ for all i.

AND-composition converts a UP upper bound into a $\mathsf{2UP}$ upper bound (as in Chapter 4):

Observation 5.5. *For all f and k, we have* $\mathsf{2UP}^{\mathsf{dt}}(\mathsf{AND} \circ f^k) \leq k \cdot \mathsf{UP}^{\mathsf{dt}}(f) + O(\mathsf{UP}^{\mathsf{dt}}(f)^2)$. *Similarly, for all F and k, we have* $\mathsf{2UP}^{\mathsf{cc}}(\mathsf{AND} \circ F^k) \leq k \cdot \mathsf{UP}^{\mathsf{cc}}(F) + O(\mathsf{UP}^{\mathsf{cc}}(F)^2 + \log k)$.

The two parts of Observation 5.5 are analogous, so we describe the idea only in terms of the query complexity part. Since $\mathsf{coUP}^{\mathsf{dt}}(f) \leq \mathsf{P}^{\mathsf{dt}}(f) \leq O(\mathsf{UP}^{\mathsf{dt}}(f)^2)$, it suffices to have $\mathsf{coUP}^{\mathsf{dt}}(f)$ as the second term on the right side. The idea is to let a 1-certificate for $\mathsf{AND} \circ f^k$ be comprised of 1-certificates for each of the k copies of f, and a 0-certificate for $\mathsf{AND} \circ f^k$ be comprised of a 0-certificate for the first copy of f that evaluates to 0, together with 1-certificates for each of the preceding copies of f.

On the other hand, the following lemma (proven in Section 5.5.1) shows that randomised query complexity goes up by a factor of k under AND-composition.

Lemma 5.6. *For all f and k, we have* $\mathsf{BPP}^{\mathsf{dt}}(f) \leq O\big(\mathsf{BPP}^{\mathsf{dt}}(\mathsf{AND} \circ f^k)/k\big)$.

We note that Lemma 5.6 qualitatively strengthens the tight direct sum result for randomised query complexity in [JKS10] since computing the outputs of all k copies of f is at least as hard as computing the AND of the outputs. Similarly, if we could prove an analogue of Lemma 5.6 for communication complexity, it would qualitatively strengthen the notoriously-open tight direct sum conjecture for randomised communication complexity.

5.3.3 Overview of proofs

The following diagram shows how we construct the functions used to witness our gaps. Starting with some f, we can lift it to obtain F, or we can apply AND-composition to obtain f^*. We can obtain F^* by either lifting f^* or equivalently applying AND-composition to F.

$$\begin{tikzcd}
\mathsf{coWAPP}^{\mathsf{cc}}\text{-vs-}\mathsf{UP}^{\mathsf{cc}} & \mathsf{BPP}^{\mathsf{cc}}\text{-vs-}\mathsf{2UP}^{\mathsf{cc}} \\
\boldsymbol{F} \arrow[r, "\mathsf{AND}\text{-composition}"] & \boldsymbol{F^*} \\
\boldsymbol{f} \arrow[u, "\text{lifting}"] \arrow[r, "\mathsf{AND}\text{-composition}"'] & \boldsymbol{f^*} \arrow[u, "\text{lifting}"'] \\
\mathsf{coWAPP}^{\mathsf{dt}}\text{-vs-}\mathsf{UP}^{\mathsf{dt}} & \mathsf{BPP}^{\mathsf{dt}}\text{-vs-}\mathsf{2UP}^{\mathsf{dt}}
\end{tikzcd}$$

Proof of Theorem 5.2. We start by discussing the proof of Theorem 5.2 as it will be used in the proof of Theorem 5.1. We actually prove the following stronger version of Theorem 5.2 that gives a lower bound even against $\mathsf{coWAPP}^{\mathsf{cc}}_{\epsilon}(F) \leq O(\mathsf{BPP}^{\mathsf{cc}}(F))$:

Theorem 5.2* ($\mathsf{coWAPP}^{\mathsf{cc}}$-vs-$\mathsf{UP}^{\mathsf{cc}}$). *There is an F such that* $\mathsf{coWAPP}^{\mathsf{cc}}_{0.04}(F) \geq \tilde{\Omega}(\mathsf{UP}^{\mathsf{cc}}(F)^2)$.

Our proof follows the same outline as in Chapter 4 and only requires us to lift the following analogous result for query complexity (proved in Section 5.4):

Lemma 5.7 ($\mathsf{coWAPP}^{\mathsf{dt}}$-vs-$\mathsf{UP}^{\mathsf{dt}}$)**.** $\mathsf{coWAPP}^{\mathsf{dt}}_{0.05}(\mathsf{TL}) \geq \tilde{\Omega}(\mathsf{UP}^{\mathsf{dt}}(\mathsf{TL})^2)$.

To derive Theorem 5.2*, set $f := \mathsf{TL}$ and $F := f \circ g^n$, where g is the gadget from Lemma 5.4 and n is the input length of f. Recall that $\mathsf{UP}^{\mathsf{dt}}(f) \geq n^{\Omega(1)}$. Thus by Observation 5.3, $\mathsf{UP}^{\mathsf{cc}}(F) \leq \mathsf{UP}^{\mathsf{dt}}(f) \cdot O(\log n) \leq \tilde{O}(\mathsf{UP}^{\mathsf{dt}}(f))$, and by Lemma 5.4, $\mathsf{coWAPP}^{\mathsf{cc}}_{0.04}(F) \geq \Omega(\mathsf{coWAPP}^{\mathsf{dt}}_{0.05}(f) \cdot \log n) \geq \Omega(\mathsf{coWAPP}^{\mathsf{dt}}_{0.05}(f))$. Thus $\mathsf{coWAPP}^{\mathsf{cc}}_{0.04}(F) \geq \tilde{\Omega}(\mathsf{UP}^{\mathsf{cc}}(F)^2)$.

Proof of Theorem 5.1. An "obvious" strategy for Theorem 5.1 would be again to first prove the analogous query complexity result and then lift it to communication complexity. (This is the outline used for the analogous result in Chapter 4.) In other words, we would follow the lower-right path in the above diagram:

Obvious strategy

(a) Start with f witnessing a $\mathsf{BPP}^{\mathsf{dt}}$-vs-$\mathsf{UP}^{\mathsf{dt}}$ gap.
(b) Obtain f^* witnessing a $\mathsf{BPP}^{\mathsf{dt}}$-vs-$\mathsf{2UP}^{\mathsf{dt}}$ gap by applying AND-composition to f.
(c) Obtain F^* witnessing a $\mathsf{BPP}^{\mathsf{cc}}$-vs-$\mathsf{2UP}^{\mathsf{cc}}$ gap by lifting f^*.

We have the tools to complete steps (a) and (b):

Lemma 5.8 ($\mathsf{BPP}^{\mathsf{dt}}$-vs-$\mathsf{2UP}^{\mathsf{dt}}$)**.** *There is an f such that* $\mathsf{BPP}^{\mathsf{dt}}(f) \geq \tilde{\Omega}(\mathsf{2UP}^{\mathsf{dt}}(f)^{1.5})$.

Proof. This is witnessed by $f^* := \mathsf{AND} \circ \mathsf{TL}^k$ where $k := \mathsf{UP}^{\mathsf{dt}}(\mathsf{TL})$. By Observation 5.5, $\mathsf{2UP}^{\mathsf{dt}}(f^*) \leq O(k^2)$, and by Lemmas 5.6–5.7, $\mathsf{BPP}^{\mathsf{dt}}(f^*) \geq \Omega(k \cdot \mathsf{BPP}^{\mathsf{dt}}(\mathsf{TL})) \geq \Omega(k \cdot \mathsf{coWAPP}^{\mathsf{dt}}_{0.05}(\mathsf{TL})) \geq \tilde{\Omega}(k^3)$. □

Unfortunately, we do not know how to carry out step (c), because we currently lack a simulation lemma for BPP. (We believe that such a lemma is true, and it is an interesting open problem to prove this!) We get around this obstacle by reversing the order of steps (b) and (c), that is, we instead follow the upper-left path in the diagram:

Modified strategy

(a′) Start with f witnessing a $\mathsf{coWAPP}^{\mathsf{dt}}$-vs-$\mathsf{UP}^{\mathsf{dt}}$ gap.
(b′) Obtain F witnessing a $\mathsf{coWAPP}^{\mathsf{cc}}$-vs-$\mathsf{UP}^{\mathsf{cc}}$ gap by lifting f.
(c′) Obtain F^* witnessing a $\mathsf{BPP}^{\mathsf{cc}}$-vs-$\mathsf{2UP}^{\mathsf{cc}}$ gap by applying AND-composition to F.

Steps (a′) and (b′) are just Theorem 5.2⋆. For step (c′) it would suffice to have an analogue of Lemma 5.6 for communication complexity. This is open, but fortunately we have some wiggle room since it suffices to have coWAPP_ϵ instead of BPP on the left side of Lemma 5.6. For this, we *can* prove a communication analogue (indeed, with $\mathsf{2WAPP}_\epsilon$ instead of coWAPP_ϵ):

Lemma 5.9. *For all F, k, and constants $0 < \epsilon < 1/2$, we have*

$$\mathsf{2WAPP}^{\mathsf{cc}}_\epsilon(F) \;\le\; O\big(\mathsf{BPP}^{\mathsf{cc}}(\mathsf{AND}\circ F^k)/k + \log \mathsf{BPP}^{\mathsf{cc}}(\mathsf{AND}\circ F^k)\big).$$

To derive Theorem 5.1, let F be the function in Theorem 5.2⋆, and let $F^* := \mathsf{AND}\circ F^k$ where $k := \mathsf{UP}^{\mathsf{cc}}(F)$. Then F^* witnesses Theorem 5.1: By Observation 5.5, $\mathsf{2UP}^{\mathsf{cc}}(F^*) \le O(k^2)$, and by Lemma 5.9, $\mathsf{BPP}^{\mathsf{cc}}(F^*) \ge \Omega\big(k\cdot(\mathsf{2WAPP}^{\mathsf{cc}}_{0.04}(F) - O(\log k))\big) \ge \Omega\big(k\cdot(\mathsf{coWAPP}^{\mathsf{cc}}_{0.04}(F) - O(\log k))\big) \ge \tilde{\Omega}(k^3)$.

Proof of Lemma 5.9. We start with the intuition for the proof of Lemma 5.6, which is a warmup for Lemma 5.9. For brevity let $f^* := \mathsf{AND}\circ f^k$. Given an input z for f, the basic idea is to plant z into a random coordinate of $f^*(z_1,\ldots,z_k)$, and plant random 1-inputs into the other coordinates, and then run the randomised decision tree for f^*. If q is the query complexity of f^*, the expected number of bits of z that are queried (over a random 1-input) will be at most q/k. Our new randomised decision tree will simulate this but abort after $8q/k$ queries to z have been made. If an answer is returned, we output the same value for $f(z)$, and if no answer is returned within this many queries, then we output 0. A simple analysis shows that we succeed with high probability in the average-case (which is equivalent to worst-case by the minimax theorem).

To prove Lemma 5.9, we would like to mimic this argument in the communication world, using the fact that internal information complexity is sandwiched between $\mathsf{BPP}^{\mathsf{cc}}$ and $\mathsf{2WAPP}^{\mathsf{cc}}$ [KLL+12] and satisfies a sort of AND-composition analogous to Lemma 5.6 using well-known properties (by planting the input into a random coordinate, and planting random 1-inputs into the other coordinates). However there is a significant barrier to this idea “just working”: the AND-composition property (direct sum lemma) requires a distribution over 1-inputs of F (one-sided), while the relation to $\mathsf{2WAPP}^{\mathsf{cc}}$ requires an arbitrary distribution over inputs to F

(two-sided). To bridge this divide, we prove a new property of information complexity: the one-sided version is essentially equivalent to the two-sided version. A key ingredient in showing the latter is the "information odometer" of [BW15a], which allows us to keep track of the amount of information that has been revealed, and abort the protocol once we have reached our limit, and argue that we can carry this out without revealing too much extra information. We note that this one-vs-two sided information complexity lemma is the only component of the proof of Theorem 5.1 that distinguishes between arbitrary rectangle partitions ($\mathsf{2UP}^{\mathsf{cc}}$) and rectangle partitions induced by protocols (P^{cc}).

Organization. The only ingredients that remain to be proved are Lemma 5.7 (which we prove in Section 5.4) and Lemma 5.6 and Lemma 5.9 (both of which we prove in Section 5.5).

5.4 Decision tree lower bound

In this section we prove Lemma 5.7, restated here for convenience.

Lemma 5.7 (**$\mathsf{coWAPP}^{\mathsf{dt}}$**-vs-**$\mathsf{UP}^{\mathsf{dt}}$**)**.** $\mathsf{coWAPP}^{\mathsf{dt}}_{0.05}(\mathsf{TL}) \geq \tilde{\Omega}(\mathsf{UP}^{\mathsf{dt}}(\mathsf{TL})^2)$.

Recall that $\mathsf{UP}^{\mathsf{dt}}(\mathsf{TL}) \leq O(k \log k)$ by definition. To prove Lemma 5.7 we show that there is no $o(k^2)$-degree conical junta $h = \sum w_C C$ that outputs values in $[0.95, 1]$ on inputs from $\mathsf{TL}^{-1}(0)$ and outputs values in $[0, 0.05]$ on inputs from $\mathsf{TL}^{-1}(1)$. A similar lower bound for the plain $k \times k$ *Tribes* function was proved by [JK10, Theorem 4] using LP duality; our argument is more direct.

To illustrate the basic style of argument, we start gently by proving an $\Omega(n)$ conical junta degree bound for approximating the NAND function—this lower bound will be used in the proof of Lemma 5.7, too.

5.4.1 Warm-up: Lower bound for NAND

Suppose for contradiction that $h = \sum w_C C$ is a conical junta of degree $o(n)$ computing the n-bit NAND function to within error $1/5$. We will argue that if h is correct on inputs of Hamming weights n and $n-1$, then it must mess up on inputs of Hamming weight $n-2$: h will output a value larger than 1, which is a contradiction. We now give the details.

To begin, we have $h(\vec{1}) \leq 1/5$ by the correctness of h (here $\vec{1}$ is the all-1 input). This means that the total weight (sum of w_C's) associated with conjunctions that read only 1's is at most $1/5$. Let $X \in \mathsf{NAND}^{-1}(1)$ be a uniformly random string of Hamming weight $n-1$. By correctness,

$$\mathbf{E}[h(X)] \;=\; \textstyle\sum w_C \, \mathbf{E}[C(X)] \;=\; \sum w_C \, \mathbf{Pr}[C(X) = 1] \;\geq\; 4/5.$$

In the above sum, there are two types of conjunctions that contribute with a positive acceptance probability: those that read only 1's, and those that read a single 0 and some $o(n)$ many 1's. Since the first type has total weight $\leq 1/5$ we must have $\sum_{C \in \mathscr{C}} w_C \, \mathbf{Pr}[C(X) = 1] \geq 3/5$

where $\mathscr{C}$ is the set of conjunctions of the second type. Consider the acceptance probability of any $C \in \mathscr{C}$ on a uniformly random string $Y \in \mathsf{NAND}^{-1}(1)$ of Hamming weight $n-2$: if the width of C is d, then $\mathbf{Pr}[C(Y)=1] = (n-d)/\binom{n}{2}$, which is $(2-o(1))/n$ for $d = o(n)$. Since $\mathbf{Pr}[C(X)=1] = 1/n$ we conclude that

$$\mathbf{Pr}[C(Y)=1] \;=\; (2-o(1)) \cdot \mathbf{Pr}[C(X)=1]. \tag{5.1}$$

We now arrive at the desired contradiction:

$$\mathbf{E}[h(Y)] \;\geq\; \textstyle\sum_{C\in\mathscr{C}} w_C \,\mathbf{Pr}[C(Y)=1] \;=\; (2-o(1)) \sum_{C\in\mathscr{C}} w_C \,\mathbf{Pr}[C(X)=1] \;\geq\; (2-o(1))\cdot 3/5 \;>\; 1.$$

5.4.2 Proof of Lemma 5.7

We prove a lower bound for $\mathsf{TL}\colon \Sigma^{k\times k} \to \{0,1\}$ by arguing that $\Omega(k^2)$ entries must be touched: We only charge one query for reading a whole matrix entry in $\Sigma = \{0,1\} \times ([k]^{k-1} \cup \{\bot\})$. That is, we assume each conjunction either reads nothing from an entry or reads one fully. The width of a conjunction is then understood as the number of entries it reads.

We study three types of *random* inputs to TL:

- $X \in \mathsf{TL}^{-1}(0)$ is defined so that the columns in X are independent, and in each column all entries are $(1,\bot)$ except we plant a single $(0,\bot)$ entry in a random row index. Hence there are altogether k many $(0,\bot)$ entries in X.
- $Y \in \mathsf{TL}^{-1}(0)$ is defined like X except we replace a random $(1,\bot)$ entry in X with a $(0,\bot)$ entry. Hence there are altogether $k+1$ many $(0,\bot)$ entries in Y, two of them sharing a column.
- $Z \in \mathsf{TL}^{-1}(1)$ is defined like X except we replace a random $(0,\bot)$ entry (k different choices) in X with a $(1,p)$ entry, where p is a list of pointers to all other positions of $(0,\bot)$ entries (making Z indeed a 1-input).

The crux of the argument is contained in the following claim.

Claim 5.10. *For every conjunction C of width $o(k^2)$, either $\mathbf{Pr}[C(Y)=1] \geq 1.4\cdot \mathbf{Pr}[C(X)=1]$ or $\mathbf{Pr}[C(Z)=1] \geq 0.5 \cdot \mathbf{Pr}[C(X)=1]$.*

Before proving Claim 5.10, let us see how to finish the proof of Lemma 5.7 assuming it. We have a similar claim for conical juntas:

Claim 5.11. *For every conical junta h of degree $o(k^2)$, either $\mathbf{E}[h(Y)] \geq 1.1 \cdot \mathbf{E}[h(X)]$ or $\mathbf{E}[h(Z)] \geq 0.1 \cdot \mathbf{E}[h(X)]$.*

Proof. Let $h = \sum w_C C$. By linearity, $\mathbf{E}[h(X)] = \sum w_C \,\mathbf{Pr}[C(X)=1]$ and similarly for Y and Z. By Claim 5.10, let $\mathscr{C}$ be a set of conjunctions such that for each $C \in \mathscr{C}$, $\mathbf{Pr}[C(Y) =$

$1] \geq 1.4 \cdot \mathbf{Pr}[C(X) = 1]$, and for each $C \notin \mathscr{C}$, $\mathbf{Pr}[C(Z) = 1] \geq 0.5 \cdot \mathbf{Pr}[C(X) = 1]$. Either $\sum_{C \in \mathscr{C}} w_C \, \mathbf{Pr}[C(X) = 1] \geq 0.8 \cdot \mathbf{E}[h(X)]$, in which case

$$\mathbf{E}[h(Y)] \;\geq\; \textstyle\sum_{C \in \mathscr{C}} w_C \, \mathbf{Pr}[C(Y) = 1] \;\geq\; \textstyle\sum_{C \in \mathscr{C}} w_C \cdot 1.4 \cdot \mathbf{Pr}[C(X) = 1] \;\geq\; 1.4 \cdot 0.8 \cdot \mathbf{E}[h(X)],$$

or $\sum_{C \notin \mathscr{C}} w_C \, \mathbf{Pr}[C(X) = 1] \geq 0.2 \cdot \mathbf{E}[h(X)]$, in which case

$$\mathbf{E}[h(Z)] \;\geq\; \textstyle\sum_{C \notin \mathscr{C}} w_C \, \mathbf{Pr}[C(Z) = 1] \;\geq\; \textstyle\sum_{C \notin \mathscr{C}} w_C \cdot 0.5 \cdot \mathbf{Pr}[C(X) = 1] \;\geq\; 0.5 \cdot 0.2 \cdot \mathbf{E}[h(X)]. \qquad \square$$

Now to prove Lemma 5.7, suppose for contradiction that h is a conical junta of degree $o(k^2)$ computing $\neg\mathsf{TL}$ to within error 0.05. That is, the value of h is in $[0.95, 1]$ on 0-inputs of TL and in $[0, 0.05]$ on 1-inputs of TL. In particular, $\mathbf{E}[h(X)] \in [0.95, 1]$, $\mathbf{E}[h(Y)] \in [0.95, 1]$, and $\mathbf{E}[h(Z)] \in [0, 0.05]$. This directly contradicts Claim 5.11.

Proof of Claim 5.10. We may assume that C accepts X with positive probability for otherwise the claim is trivial. Hence C reads at most a single $(0, \perp)$ entry from each column. We analyze two cases depending on how many $(0, \perp)$ entries C reads in total.

The first (easy) case is when C reads less than $k/2$ many $(0, \perp)$ entries. Here C cannot detect us replacing a random $(0, \perp)$ entry with a $(1, p)$ entry with probability better than 1/2. That is, $\mathbf{Pr}[C(Z) = 1] \geq 0.5 \cdot \mathbf{Pr}[C(X) = 1]$.

The second case is when C reads at least $k/2$ many $(0, \perp)$ entries. Because C has width $o(k^2)$ there is some $S_1 \subseteq [k]$ of size $|S_1| \geq (1 - o(1))k$ such that C reads $o(k)$ entries from each of the columns indexed by S_1. (More precisely, if C has width δk^2, then there is a set of $(1 - \sqrt{\delta})k$ columns from each of which C reads at most $\sqrt{\delta}k$ entries.) Let $S_2 \subseteq [k]$, $|S_2| \geq k/2$, be the set of columns where C reads a $(0, \perp)$. Let $i \in [k]$ denote the unique column where X and Y differ. Note that i is a uniform random variable; for example, $\mathbf{Pr}[i \in S_1] = 1 - o(1)$. In what follows, we take $\approx$ to mean *up to a* $(1 \pm o(1))$ *factor*. We calculate:

$$\begin{aligned}
\mathbf{Pr}[C(Y) = 1] \;&\geq\; \mathbf{Pr}[C(Y) = 1 \text{ and } i \in S_1] \\
&\approx\; \mathbf{Pr}[C(Y) = 1 \,|\, i \in S_1] \\
&=\; \mathbf{Pr}[C(Y) = 1 \text{ and } i \in S_2 \,|\, i \in S_1] \;+\; \mathbf{Pr}[C(Y) = 1 \text{ and } i \notin S_2 \,|\, i \in S_1] \\
&=\; \lambda \cdot \underbrace{\mathbf{Pr}[C(Y) = 1 \,|\, i \in S_1 \cap S_2]}_{\textbf{(I)}} \;+\; (1 - \lambda) \cdot \underbrace{\mathbf{Pr}[C(Y) = 1 \,|\, i \in S_1 \smallsetminus S_2]}_{\textbf{(II)}},
\end{aligned}$$

where $\lambda \coloneqq \mathbf{Pr}[i \in S_2 \,|\, i \in S_1] \geq 1/2 - o(1)$. In the first term, the condition $(i \in S_1 \cap S_2)$ means that C reads a single $(0, \perp)$ and $o(k)$ many $(1, \perp)$'s from the i-th column. Hence we are in a situation analogous to that in (5.1), and the same argument yields

$$\textbf{(I)} \;\geq\; (2 - o(1)) \cdot \mathbf{Pr}[C(X) = 1 \,|\, i \in S_1 \cap S_2] \;\approx\; 2 \cdot \mathbf{Pr}[C(X) = 1].$$

In the second term, the condition $(i \in S_1 \smallsetminus S_2)$ means that C reads $o(k)$ many $(1, \perp)$'s from the

i-th column. Hence C cannot detect our planting of an additional $(0,\perp)$ entry in that column with probability better than $o(1)$:

$$\textbf{(II)} \;\geq\; (1-o(1))\cdot\mathbf{Pr}[C(X)=1\,|\,i\in S_1\smallsetminus S_2] \;\approx\; \mathbf{Pr}[C(X)=1].$$

In summary, we get that for some $\lambda\geq 1/2-o(1)$,

$$\begin{aligned}\mathbf{Pr}[C(Y)=1] \;&\geq\; (2\lambda+(1-\lambda)-o(1))\cdot\mathbf{Pr}[C(X)=1]\\ &\geq\; (3/2-o(1))\cdot\mathbf{Pr}[C(X)=1]\\ &\geq\; 1.4\cdot\mathbf{Pr}[C(X)=1].\end{aligned}$$

□

5.5 AND-composition lemmas

In this section we prove Lemma 5.6 and Lemma 5.9, restated here for convenience.

Lemma 5.6. *For all f and k, we have* $\mathsf{BPP}^{\mathsf{dt}}(f)\leq O\big(\mathsf{BPP}^{\mathsf{dt}}(\mathsf{AND}\circ f^k)/k\big)$.

Lemma 5.9. *For all F, k, and constants $0<\epsilon<1/2$, we have*

$$\mathsf{2WAPP}^{\mathsf{cc}}_{\epsilon}(F) \;\leq\; O\big(\mathsf{BPP}^{\mathsf{cc}}(\mathsf{AND}\circ F^k)/k+\log\mathsf{BPP}^{\mathsf{cc}}(\mathsf{AND}\circ F^k)\big).$$

5.5.1 AND-composition for query complexity

We now prove Lemma 5.6. For brevity let $f^*\coloneqq\mathsf{AND}\circ f^k$. Let T^* be a height-q randomised decision tree for f^* with error $1/8$. We design a height-$8q/k$ randomised decision tree for f with error $1/4$.

Let D be an arbitrary distribution over $f^{-1}(1)$. Consider the following randomised decision tree T that takes $z\in\{0,1\}^n$ as input:

1. Pick $i\in[k]$ uniformly at random and let $z_i\coloneqq z$.
2. For $j\in[k]\smallsetminus\{i\}$ sample $z_j\sim D$ independently.
3. Run $T^*(z_1,\dots,z_k)$ until it has made $8q/k$ queries in the i-th component.
4. If T^* already produced an output in Step 3, output the same bit; otherwise output 0.

Note that with probability 1 we have $f^*(z_1,\dots,z_k)=f(z)$. Let R_T denote T's randomness and R_{T^*} denote T^*'s randomness. If $f(z)=0$ then

$$\mathbf{Pr}_{R_T}[T(z)=1] \;\leq\; \max_{(z_1,\dots,z_k)\in(f^*)^{-1}(0)}\mathbf{Pr}_{R_{T^*}}[T^*(z_1,\dots,z_k)=1] \;\leq\; 1/8 \;\leq\; 1/4.$$

Furthermore,

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{z\sim D,\,R_T}[T(z)=0] \;&=\; \mathop{\mathbf{Pr}}_{z_1,\dots,z_k\sim D,\,i\in[k],\,R_{T^*}}\begin{bmatrix}T^*(z_1,\dots,z_k)\text{ outputs 0 or makes more}\\ \text{than } 8q/k \text{ queries in the } i\text{-th component}\end{bmatrix}\\
&\le\; \max\nolimits_{(z_1,\dots,z_k)\in(f^*)^{-1}(1)}\left(\begin{array}{l}\mathbf{Pr}_{R_{T^*}}[T^*(z_1,\dots,z_k)=0]\,+\\ \max_{R_{T^*}}\mathbf{Pr}_{i\in[k]}\begin{bmatrix}T^*(z_1,\dots,z_k)\text{ makes more than}\\ 8q/k\text{ queries in the } i\text{-th component}\end{bmatrix}\end{array}\right)\\
&\le\; 1/8+1/8 \;=\; 1/4.
\end{aligned}$$

Now let D be an arbitrary distribution over $\{0,1\}^n$ and define T w.r.t. $(D\,|\,f^{-1}(1))$. We have

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{z\sim D,\,R_T}[T(z)\neq f(z)] \;&=\; \textstyle\sum_{b\in\{0,1\}}\mathbf{Pr}_{z\sim(D\,|\,f^{-1}(b)),\,R_T}[T(z)\neq b]\cdot\mathbf{Pr}_{z\sim D}[f(z)=b]\\
&\le\; \textstyle\sum_{b\in\{0,1\}}(1/4)\cdot\mathbf{Pr}_{z\sim D}[f(z)=b] \;=\; 1/4.
\end{aligned}$$

By the minimax theorem, there is a height-$8q/k$ randomised decision tree (a mixture of the T's) that on any input produces the wrong output with probability $\le 1/4$.

5.5.2 Definitions

We adopt the following conventions throughout the proof of Lemma 5.9. We denote random variables with upper-case letters, and we denote particular outcomes of the random variables with the corresponding lower-case letters. All communication protocols are randomised and mixed-coin, and we use (R, R_A, R_B) to denote the public randomness, Alice's private randomness, and Bob's private randomness, respectively. We say a protocol Π is ϵ-correct for F if for all (x,y), $\mathbf{Pr}_{R,R_A,R_B}[\Pi(x,y)=F(x,y)]\ge 1-\epsilon$. For a distribution D over inputs, we say Π is (ϵ,D)-correct for F if $\mathbf{Pr}_{(X,Y)\sim D,\,R,R_A,R_B}[\Pi(X,Y)=F(X,Y)]\ge 1-\epsilon$. The internal information cost of a protocol Π with respect to $(X,Y)\sim D$ is defined as $\mathrm{IC}_D(\Pi) := \mathbf{I}(R,M\,;X\,|\,Y)+\mathbf{I}(R,M\,;Y\,|\,X) = \mathbf{I}(M\,;X\,|\,Y,R)+\mathbf{I}(M\,;Y\,|\,X,R)$ where the random variable M is the concatenation of all messages. We also let $\mathrm{CC}(\Pi)$ denote the worst-case communication cost of Π.

It is convenient for us to work with a measure $\mathsf{2WAPP}^{\mathsf{cc}*}$ that is defined slightly differently from $\mathsf{2WAPP}^{\mathsf{cc}}$ but is equivalent in the sense that for all F and $0<\epsilon<1/2$, $\mathsf{2WAPP}^{\mathsf{cc}}_{\epsilon}(F)\le \mathsf{2WAPP}^{\mathsf{cc}*}_{\epsilon}(F)\le O(\mathsf{2WAPP}^{\mathsf{cc}}_{\epsilon/2}(F))$. We note that $\mathsf{2WAPP}^{\mathsf{cc}}$ directly expresses the two-sided smooth rectangle bound of [JK10], while $\mathsf{2WAPP}^{\mathsf{cc}*}$ directly expresses the relaxed partition bound of [KLL+12] and was the definition used in Chapter 2.

Definition 5.12. We define $\mathsf{2WAPP}^{\mathsf{cc}*}_{\epsilon}(F)$ as the minimum of $\mathrm{CC}(\Pi)+\log(1/\alpha)$ over all $\alpha>0$ and all protocols Π with output values $\{0,1,\perp\}$ such that for all (x,y), $\mathbf{Pr}[\Pi(x,y)\neq\perp]\le\alpha$ and $\mathbf{Pr}[\Pi(x,y)=F(x,y)]\ge(1-\epsilon)\alpha$ (i.e., Π is $(1-(1-\epsilon)\alpha)$-correct).

We also need the distributional version of $\mathsf{2WAPP}^{\mathsf{cc}*}$.

Definition 5.13. For an input distribution D, we define $\mathsf{2WAPP}^{\mathsf{cc}*}_{\epsilon,D}(F)$ as the minimum of $\mathrm{CC}(\Pi) + \log(1/\alpha)$ over all $\alpha > 0$ and all protocols Π with output values $\{0, 1, \bot\}$ such that $\mathbf{Pr}[\Pi(x, y) \neq \bot] \leq \alpha$ for all (x, y), and $\mathbf{Pr}[\Pi(X, Y) = F(X, Y)] \geq (1 - \epsilon)\alpha$ for $(X, Y) \sim D$ (i.e., Π is $(1 - (1 - \epsilon)\alpha, D)$-correct).

5.5.3 AND-composition for communication complexity

We now outline the proof of Lemma 5.9. Recall that the proof of Lemma 5.6 involved these steps:

(i) embedding the input into a random coordinate of a k-tuple and filling the other coordinates with random 1-inputs (to cut the cost on 1-inputs by a factor k),
(ii) aborting the execution if the cost became too high (to ensure low cost also on 0-inputs while maintaining average-case correctness on 1-inputs),
(iii) using the minimax theorem to go from average-case to worst-case correctness.

We start by noting that an analogue of (i) holds for information complexity (which lower bounds $\mathsf{BPP}^{\mathsf{cc}}$). Then as one of our main technical contributions we prove an analogue of (ii) for information complexity. Then inbetween (ii) and (iii) we insert a step applying the known result that information complexity upper bounds $\mathsf{2WAPP}^{\mathsf{cc}*}$ in the distributional setting. Finally we use the analogue of (iii) for $\mathsf{2WAPP}^{\mathsf{cc}*}$. Formally, Lemma 5.9 follows by stringing together the following lemmas.

Lemma 5.14. *Fix any F, k, $0 < \epsilon < 1/2$, and distribution D over $F^{-1}(1)$. For every ϵ-correct protocol Π for $\mathsf{AND} \circ F^k$ there is an ϵ-correct protocol Π' for F with $\mathrm{IC}_D(\Pi') \leq \mathrm{CC}(\Pi)/k$ and $\mathrm{CC}(\Pi') \leq \mathrm{CC}(\Pi)$.*

Lemma 5.15. *Fix any F, constants $0 < \epsilon < \delta < 1/2$, and input distribution D, and let $D^1 := (D \,|\, F^{-1}(1))$. For every (ϵ, D)-correct protocol Π there is a (δ, D)-correct protocol Π' with $\mathrm{IC}_D(\Pi') \leq O\big(\mathrm{IC}_{D^1}(\Pi) + \log(\mathrm{CC}(\Pi) + 2)\big)$.*

Lemma 5.16. *Fix any F, constants $0 < \epsilon < \delta < 1/2$, and input distribution D. For every (ϵ, D)-correct protocol Π we have $\mathsf{2WAPP}^{\mathsf{cc}*}_{\delta,D}(F) \leq O(\mathrm{IC}_D(\Pi) + 1)$.*

Lemma 5.17. *Fix any F and $0 < \epsilon < 1/2$. Then $\mathsf{2WAPP}^{\mathsf{cc}*}_{\epsilon}(F) \leq 2 + \max_D \mathsf{2WAPP}^{\mathsf{cc}*}_{\epsilon,D}(F)$.*

Lemma 5.14 is a standard application of the "direct sum" property of information cost; for completeness we sketch the argument in Section 5.6. Lemma 5.15 is proved in Section 5.5.4 and relies on [BW15a]. Lemma 5.16 is due to [KLL+12, Theorem 1.1 of the ECCC version]. Lemma 5.17 follows from an argument in [KLL+12, Appendix A of the ECCC version] that uses LP duality; for completeness, in Section 5.6 we give a more intuitive version of the argument phrased in terms of the minimax theorem.

The moral conclusion of Lemma 5.15 is that "one-sided information complexity" is essentially equivalent to "two-sided information complexity" for average-case protocols. Combining

Lemma 5.15 with [Bra12, Theorem 3.5 of the ECCC version] shows that a similar equivalence holds for worst-case protocols. More specifically, a distribution-independent definition of information complexity for bounded-error protocols can be obtained by maximizing over all input distributions; our corollary shows that this measure is essentially unchanged if we maximize only over distributions over 1-inputs (or symmetrically, 0-inputs). This is not needed for our results, but it has found other applications [ABB$^+$16b].

Corollary 5.18. *Fix any F, constants $0 < \epsilon < \delta < 1/2$, and $b \in \{0, 1\}$. Then*

$$\inf_{\substack{\delta\text{-correct}\\ \text{protocols } \Pi}} \max_{\substack{D \text{ over}\\ \text{all inputs}}} \mathrm{IC}_D(\Pi) \;\le\; \max_{\substack{D \text{ over}\\ b\text{-inputs}}} \inf_{\substack{\epsilon\text{-correct}\\ \text{protocols } \Pi}} O\big(\mathrm{IC}_D(\Pi) + \log(\mathrm{CC}(\Pi) + 2)\big).$$

5.5.4 One-sided information vs. two-sided information

Intuition for Lemma 5.15. Recall the following idea, which was implicit in the proof of Lemma 5.6. Suppose we have a randomised decision tree computing some function, and we have a bound b on the expected number of queries made over a random 1-input. Then to obtain a randomised decision tree with a worst-case query bound, we can keep track of the number of queries made during the execution and halt and output 0 if it exceeds, say, $8b$. Correctness on 0-inputs is maintained since we either run the original decision tree to completion and thus output 0 with high probability, or we abort and output 0 anyway. We get average-case correctness on 1-inputs since by Markov's inequality, with probability at least $7/8$ the original decision tree uses at most $8b$ queries, in which case we run it to completion and output 1 with high probability.

The high-level intuition is to mimic this idea for information complexity. We have a protocol with a bound on the information cost w.r.t. the distribution D^1 over 1-inputs. The "information odometer" of [BW15a] allows us to "keep track of" information cost, so we can halt and output 0 if it becomes too large. This will guarantee that the information cost is low w.r.t. the input distribution D, and correctness on 0-inputs is maintained. However, there is a complication with showing the average-case correctness on 1-inputs.

For each computation path specified by an input (x, y), an outcome of public randomness r, and a full sequence of messages m, there is a contribution $c_{x,y,r,m}$ such that the information cost w.r.t. D is the expectation of $c_{x,y,r,m}$ over a random computation path with $(x, y) \sim D$. Similarly, there is a contribution $c^1_{x,y,r,m}$ such that the information cost w.r.t. D^1 is the expectation of $c^1_{x,y,r,m}$ over a random computation path with $(x, y) \sim D^1$. These contributions play the role of "number of queries" along a computation path in the decision tree setting, but a crucial difference is that $c_{x,y,r,m} \ne c^1_{x,y,r,m}$ in general; i.e., the contribution to information cost depends on the input distribution (whereas number of queries did not). To show the average-case correctness on 1-inputs, we need a bound on the typical value of $c_{x,y,r,m}$, whereas the assumption that information cost w.r.t. D^1 is low gives us a bound on the typical value of $c^1_{x,y,r,m}$.

Thus the heart of the argument is to show that typically, $c_{x,y,r,m}$ is not much larger than

$c^1_{x,y,r,m}$. Intuitively, one might expect the difference to be at most 1, since the only additional information that can be revealed (beyond what is revealed under D^1) should be the fact that (x, y) is a 1-input (which is 1 bit of information). More precisely, we show that for given (x, y), the expected difference depends on how balanced F is on the x row and the y column. Then we just need to note that F is typically reasonably balanced for both the x row and the y column.

Formal proof of Lemma 5.15. Assume w.l.o.g. that every execution of Π communicates exactly the same number of bits, and that Alice always sends a bit in odd rounds and Bob always sends a bit in even rounds (by inserting dummy coin flip rounds if necessary). As shown in [BW15a], we can also assume that Π is "smooth" (i.e., in every step, the bit to be communicated is 1 with probability between 1/3 and 2/3)—this is needed in order to apply Lemma 5.19 below.

Consider a probability space with random variables X, Y, R, R_A, R_B, M, F where $(X, Y) \sim D$ is the input, (R, R_A, R_B) is Π's randomness, $M \coloneqq M_1, \ldots, M_{\mathrm{CC}(\Pi)}$ is the sequence of bits communicated by Π, and $F \coloneqq F(X, Y)$ is the function value. For convenience of notation, if we condition on "x", this is shorthand for conditioning on "$X = x$". Letting $t \in \{1, \ldots, \mathrm{CC}(\Pi)\}$ and letting $\mathbf{D}$ denote KL-divergence (relative entropy), if we define

$$\begin{aligned} d_{x,y,r,m_{<t}} &\coloneqq \mathbf{D}\left(\frac{M_t \mid x, y, r, m_{<t}}{M_t \mid y, r, m_{<t}}\right) + \mathbf{D}\left(\frac{M_t \mid x, y, r, m_{<t}}{M_t \mid x, r, m_{<t}}\right), \\ c_{x,y,r,m} &\coloneqq \textstyle\sum_t d_{x,y,r,m_{<t}}, \\ c_{x,y} &\coloneqq \mathbf{E}[c_{X,Y,R,M} \mid x, y], \end{aligned}$$

then it can be seen [BW15a, Appendix C of the ECCC version] that

$$\mathrm{IC}_D(\Pi) \;=\; \mathbf{E}[c_{X,Y,R,M}] \;=\; \mathbf{E}[c_{X,Y}]. \tag{5.2}$$

Note that if t is odd the second term of $d_{x,y,r,m_{<t}}$ is 0, and if t is even the first term is 0; hence we think of $d_{x,y,r,m_{<t}}$ as defined by a single term (depending on who communicates in round t).

Although the following lemma was not explicitly stated in this way in [BW15a], it follows immediately from the corresponding part of the argument for the "conditional abort theorem" in that paper [BW15b].

Lemma 5.19 (Odometer)**.** *For every smooth protocol Π, constant $\gamma > 0$, input distribution D, and $I > 0$, there is a protocol Π^* with $\mathrm{IC}_D(\Pi^*) \le O\big(I + \log(\mathrm{CC}(\Pi) + 2)\big)$ that simulates Π in the following sense: Π^* uses the same randomness (R, R_A, R_B) as Π and some additional, independent randomness Q. Consider any fixed outcome x, y, r, r_A, r_B, and let m be Π's messages. Then*

(i) for every q, Π^* outputs either $\perp$ or the same bit that Π does, and

(ii) if $c_{x,y,r,m} \le I$ then $\mathbf{Pr}_Q[\Pi^*$ outputs $\perp] \le \gamma$.

Define $\gamma := (\delta - \epsilon)/5$. To obtain Π' witnessing Lemma 5.15, we obtain Π^* from Lemma 5.19 with $I := (\mathrm{IC}_{D^1}(\Pi)/\gamma + 2\log(1/\gamma))/\gamma$ and replace the output $\perp$ with 0. Then we have $\mathrm{IC}_D(\Pi') = \mathrm{IC}_D(\Pi^*) \le O\big(\mathrm{IC}_{D^1}(\Pi) + \log(\mathrm{CC}(\Pi) + 2)\big)$, so we just need to verify that Π' is (δ, D)-correct. In the following, we use Π, Π^*, Π' to denote random variables (jointly distributed with $X, Y, R, R_A, R_B, M, F, Q$) representing the outputs of the protocols.

Claim 5.20. $\mathbf{Pr}[c_{X,Y,R,M} > I \text{ and } F = 1] \le 4\gamma$.

Assuming Claim 5.20, we have

$$\begin{aligned}
\mathbf{Pr}[\Pi' \neq \Pi = F] &= \mathbf{Pr}[\Pi^* = \perp \text{ and } \Pi = F = 1] \\
&\le \mathbf{Pr}[\Pi^* = \perp \text{ and } F = 1] \\
&\le \mathbf{Pr}\big[c_{X,Y,R,M} > I \text{ and } F = 1\big] + \mathbf{Pr}\big[\Pi^* = \perp \,\big|\, c_{X,Y,R,M} \le I \text{ and } F = 1\big] \\
&\le 4\gamma + \gamma \\
&= 5\gamma
\end{aligned}$$

where the first line follows by construction of Π' and part (i) of Lemma 5.19, and the fourth line follows by Claim 5.20 and part (ii) of Lemma 5.19. Finally,

$$\mathbf{Pr}[\Pi' \neq F] \;\le\; \mathbf{Pr}[\Pi \neq F] + \mathbf{Pr}[\Pi' \neq \Pi = F] \;\le\; \epsilon + 5\gamma \;=\; \delta$$

since Π is (ϵ, D)-correct. This finishes the proof of Lemma 5.15.

To prove Claim 5.20, we first need to state another claim. Analogously to the notation leading up to (5.2), if for $(x, y) \in F^{-1}(1)$ we define

$$\begin{aligned}
d^1_{x,y,r,m_{<t}} &:= \mathbf{D}\left(\frac{M_t \,|\, x, y, r, m_{<t}}{M_t \,|\, y, r, m_{<t}, F = 1}\right) + \mathbf{D}\left(\frac{M_t \,|\, x, y, r, m_{<t}}{M_t \,|\, x, r, m_{<t}, F = 1}\right), \\
c^1_{x,y,r,m} &:= \textstyle\sum_t d^1_{x,y,r,m_{<t}}, \\
c^1_{x,y} &:= \mathbf{E}[c^1_{X,Y,R,M} \,|\, x, y],
\end{aligned}$$

then we have

$$\mathrm{IC}_{D^1}(\Pi) \;=\; \mathbf{E}[c^1_{X,Y,R,M} \,|\, F = 1] \;=\; \mathbf{E}[c^1_{X,Y} \,|\, F = 1]. \tag{5.3}$$

Claim 5.21. *For $(x, y) \in F^{-1}(1)$, we have $c_{x,y} - c^1_{x,y} \le \log\big(1/\mathbf{Pr}[F = 1 \,|\, y]\big) + \log\big(1/\mathbf{Pr}[F = 1 \,|\, x]\big)$.*

Proof of Claim 5.20. For any (x, y), by Markov's inequality we have

$$\mathbf{Pr}\big[c_{X,Y,R,M} > c_{X,Y}/\gamma \,\big|\, x, y\big] \;\le\; \gamma. \tag{5.4}$$

Say y is bad if $\mathbf{Pr}[F = 1 \,|\, y] \le \gamma$, and x is bad if $\mathbf{Pr}[F = 1 \,|\, x] \le \gamma$. By Claim 5.21 and a union bound,

$$\begin{aligned}\mathbf{Pr}\big[c_{X,Y} > c^1_{X,Y} + 2\log(1/\gamma) \text{ and } F = 1\big] \;&\le\; \mathbf{Pr}\big[(Y \text{ is bad or } X \text{ is bad}) \text{ and } F = 1\big] \\ &\le\; \mathbf{Pr}[F = 1 \,|\, Y \text{ is bad}] + \mathbf{Pr}[F = 1 \,|\, X \text{ is bad}] \\ &\le\; 2\gamma. \qquad (5.5)\end{aligned}$$

By Markov's inequality and (5.3) we have

$$\mathbf{Pr}\big[c^1_{X,Y} > \mathrm{IC}_{D^1}(\Pi)/\gamma \text{ and } F = 1\big] \;\le\; \mathbf{Pr}\big[c^1_{X,Y} > \mathrm{IC}_{D^1}(\Pi)/\gamma \,\big|\, F = 1\big] \;\le\; \gamma. \qquad (5.6)$$

Claim 5.20 follows by combining (5.4), (5.5), and (5.6) using a union bound. □

Proof of Claim 5.21. Fix $(x, y) \in F^{-1}(1)$. Let $M_A \coloneqq M_1, M_3, \ldots$ be the bits sent by Alice, and let $M_B \coloneqq M_2, M_4, \ldots$ be the bits sent by Bob. Let $M_{A,<t} \coloneqq M_1, M_3, \ldots, M_k$ where k is the largest odd value $< t$, and let $M_{B,<t} \coloneqq M_2, M_4, \ldots, M_k$ where k is the largest even value $< t$.

For the moment, also consider any fixed r, r_B. Consider a separate probability space with random variables X^*, M^* distributed as $(X, M \,|\, y, r, r_B)$, and note that for even t, M^*_t is a deterministic function of $M^*_{A,<t}$. For the conditioning notation in the following, let $x^* \coloneqq x$. We have

$$\begin{aligned}\textstyle\sum_{\text{odd } t} \mathbf{E}\big[d_{X,Y,R,M_{<t}} \,\big|\, x, y, r, r_B\big] \;&=\; \textstyle\sum_{\text{odd } t} \displaystyle\mathop{\mathbf{E}}_{M^*_{A,<t}}\left[\mathbf{D}\left(\begin{array}{c} M^*_t \,|\, x^*, m^*_{A,<t} \\ \hline\hline M^*_t \,|\, m^*_{A,<t}\end{array}\right) \,\middle|\, x^*\right] \\ &=\; \mathbf{D}\left(\begin{array}{c} M^*_A \,|\, x^* \\ \hline\hline M^*_A\end{array}\right) \\ &=\; \mathbf{D}\left(\begin{array}{c} M_A \,|\, x, y, r, r_B \\ \hline\hline M_A \,|\, y, r, r_B\end{array}\right)\end{aligned}$$

where the middle equality is a direct application of the chain rule for $\mathbf{D}$. Similarly, for any fixed r, r_A, we have

$$\textstyle\sum_{\text{even } t} \mathbf{E}\big[d_{X,Y,R,M_{<t}} \,\big|\, x, y, r, r_A\big] \;=\; \mathbf{D}\left(\begin{array}{c} M_B \,|\, x, y, r, r_A \\ \hline\hline M_B \,|\, x, r, r_A\end{array}\right).$$

Then (no longer fixing any of r, r_A, r_B) we have

$$\begin{aligned} c_{x,y} \;&=\; \mathbf{E}\big[\textstyle\sum_t d_{X,Y,R,M_{<t}} \,\big|\, x, y\big] \\ &=\; \mathop{\mathbf{E}}_{R,R_B}\Big[\textstyle\sum_{\text{odd } t} \mathbf{E}\big[d_{X,Y,R,M_{<t}} \,\big|\, x, y, r, r_B\big]\Big] + \displaystyle\mathop{\mathbf{E}}_{R,R_A}\Big[\textstyle\sum_{\text{even } t} \mathbf{E}\big[d_{X,Y,R,M_{<t}} \,\big|\, x, y, r, r_A\big]\Big] \\ &=\; \mathop{\mathbf{E}}_{R,R_B}\left[\mathbf{D}\left(\begin{array}{c} M_A \,|\, x, y, r, r_B \\ \hline\hline M_A \,|\, y, r, r_B\end{array}\right)\right] + \mathop{\mathbf{E}}_{R,R_A}\left[\mathbf{D}\left(\begin{array}{c} M_B \,|\, x, y, r, r_A \\ \hline\hline M_B \,|\, x, r, r_A\end{array}\right)\right] \qquad (5.7)\end{aligned}$$

and similarly,

$$c^1_{x,y} \;=\; \mathop{\mathbf{E}}_{R,R_B}\left[\mathbf{D}\left(\frac{M_A \,|\, x,y,r,r_B}{M_A \,|\, y,r,r_B,F=1}\right)\right] + \mathop{\mathbf{E}}_{R,R_A}\left[\mathbf{D}\left(\frac{M_B \,|\, x,y,r,r_A}{M_B \,|\, x,r,r_A,F=1}\right)\right]. \tag{5.8}$$

Note that

$$\begin{aligned}
&\mathbf{D}\left(\frac{M_A \,|\, x,y,r,r_B}{M_A \,|\, y,r,r_B}\right) - \mathbf{D}\left(\frac{M_A \,|\, x,y,r,r_B}{M_A \,|\, y,r,r_B,F=1}\right)\\
=\;& \textstyle\sum_{m_A} \mathbf{Pr}[m_A \,|\, x,y,r,r_B] \cdot \log\left(\frac{\mathbf{Pr}[m_A \,|\, y,r,r_B,F=1]}{\mathbf{Pr}[m_A \,|\, y,r,r_B]}\right)\\
\le\;& \textstyle\sum_{m_A} \mathbf{Pr}[m_A \,|\, x,y,r,r_B] \cdot \log\big(1/\,\mathbf{Pr}[F=1 \,|\, y]\big)\\
=\;& \log\big(1/\,\mathbf{Pr}[F=1 \,|\, y]\big)
\end{aligned} \tag{5.9}$$

and similarly,

$$\mathbf{D}\left(\frac{M_B \,|\, x,y,r,r_A}{M_B \,|\, x,r,r_A}\right) - \mathbf{D}\left(\frac{M_B \,|\, x,y,r,r_A}{M_B \,|\, x,r,r_A,F=1}\right) \;\le\; \log\big(1/\,\mathbf{Pr}[F=1 \,|\, x]\big). \tag{5.10}$$

Claim 5.21 follows by combining (5.7), (5.8), (5.9), and (5.10) using linearity of expectation. □

5.6 Basic lemmas

5.6.1 Proof of Lemma 5.14

Write the input to $\mathsf{AND}\circ F^k$ as $\big((X_1,Y_1),\ldots,(X_k,Y_k)\big) \sim D^k$. Let (R,R_A,R_B) be Π's randomness and M be Π's messages. It is known (see [BR14, Lemma 3.14 of the ECCC Revision #1 version] and [BM13, Fact 2.3 of the ECCC Revision #1 version]) that

$$\mathrm{CC}(\Pi) \;\ge\; \mathrm{IC}_{D^k}(\Pi) \;\ge\; \sum_{i=1}^{k} \mathbf{I}\big(R,M\,;X_i \,\big|\, X_{1,\ldots,i-1},Y_i,Y_{i+1,\ldots,k}\big) + \mathbf{I}\big(R,M\,;Y_i \,\big|\, X_{1,\ldots,i-1},X_i,Y_{i+1,\ldots,k}\big).$$

Therefore there exists i and $x_{1,\ldots,i-1}, y_{i+1,\ldots,k}$ such that

$$\mathrm{CC}(\Pi)/k \;\ge\; \mathbf{I}\big(R,M\,;X_i \,\big|\, x_{1,\ldots,i-1},Y_i,y_{i+1,\ldots,k}\big) + \mathbf{I}\big(R,M\,;Y_i \,\big|\, x_{1,\ldots,i-1},X_i,y_{i+1,\ldots,k}\big)$$

which is exactly $\mathrm{IC}_D(\Pi')$ where Π' is the following protocol with input denoted (X_i,Y_i):

1. Sample the same public randomness R as Π.
2. Alice privately samples R_A and $X_{i+1,\ldots,k}$ according to D^{k-i} conditioned on $y_{i+1,\ldots,k}$.
3. Bob privately samples R_B and $Y_{1,\ldots,i-1}$ according to D^{i-1} conditioned on $x_{1,\ldots,i-1}$.
4. Run Π on $(x_{1,\ldots,i-1}, X_i, X_{i+1,\ldots,k})$, $(Y_{1,\ldots,i-1}, Y_i, y_{i+1,\ldots,k})$ with randomness (R, R_A, R_B).

Trivially, $\mathrm{CC}(\Pi') \le \mathrm{CC}(\Pi)$. The ϵ-correctness of Π' follows from the ϵ-correctness of Π since with probability 1, $F(x_j, Y_j) = 1$ for $j < i$ and $F(X_j, y_j) = 1$ for $j > i$ and thus

$$(\mathsf{AND} \circ F^k)\big((x_{1,\ldots,i-1}, X_i, X_{i+1,\ldots,k}),\, (Y_{1,\ldots,i-1}, Y_i, y_{i+1,\ldots,k})\big) \;=\; F(X_i, Y_i).$$

5.6.2 Proof of Lemma 5.17

Define α^* such that $\log(1/\alpha^*) = \max_D \mathsf{2WAPP}^{\mathsf{cc}*}_{\epsilon,D}(F)$. Consider the following two-player zero-sum game.

- Each pure row strategy is an input (x, y) to F.
- Each pure column strategy is a distribution μ over pairs (S, b), where S is a rectangle and $b \in \{0, 1, \perp\}$, such that $\mathbf{Pr}_{(S,b)\sim\mu}\big[(x, y) \in S \text{ and } b \neq \perp\big] \le \alpha^*$ holds for each (x, y).
- The payoff to the column player is $P((x, y), \mu) \coloneqq \mathbf{Pr}_{(S,b)\sim\mu}\big[(x, y) \in S \text{ and } b = F(x, y)\big]$.

We claim that for every mixed row strategy D there exists a pure column strategy μ such that $\mathbf{E}_{(x,y)\sim D}[P((x, y), \mu)] \ge (1 - \epsilon)\alpha^*$. By assumption, there exists a $\mathsf{2WAPP}^{\mathsf{cc}*}_{\epsilon,D}$ protocol Π with communication cost c and associated α satisfying $c + \log(1/\alpha) \le \log(1/\alpha^*)$. Assume Π only uses public randomness (by making any private randomness public). Consider the distribution μ over pairs (S, b) sampled as follows:

- with probability $1 - \alpha^* \cdot 2^c/\alpha$, let S be arbitrary and $b = \perp$;
- otherwise, sample the randomness of Π and a uniformly random transcript (of which we may assume there are exactly 2^c many) from the induced deterministic protocol, and let (S, b) be the rectangle and output of that transcript.

Then for each (x, y),

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{(S,b)\sim\mu}\big[(x, y) \in S \text{ and } b \neq \perp\big] \;&=\; (\alpha^* \cdot 2^c/\alpha) \cdot \mathop{\mathbf{Pr}}_{\Pi\text{'s randomness}}[\Pi(x, y) \neq \perp] \cdot \\
&\qquad \mathop{\mathbf{Pr}}_{\text{uniform transcript}}[\Pi(x, y) \text{ has that transcript}] \\
&\le\; (\alpha^* \cdot 2^c/\alpha) \cdot \alpha \cdot (1/2^c) \\
&=\; \alpha^*
\end{aligned}$$

so μ is a valid pure column strategy. Similarly, for each (x, y) we have $P((x, y), \mu) = (\alpha^*/\alpha) \cdot \mathbf{Pr}_{\Pi\text{'s randomness}}[\Pi(x, y) = F(x, y)]$, and thus

$$\mathop{\mathbf{E}}_{(x,y)\sim D}[P((x, y), \mu)] \;=\; (\alpha^*/\alpha) \cdot \mathop{\mathbf{Pr}}_{(x,y)\sim D,\, \Pi\text{'s randomness}}[\Pi(x, y) = F(x, y)] \;\ge\; (1 - \epsilon)\alpha^*.$$

Since the set of all pure column strategies μ forms a polytope, and since $P((x,y),\mu)$ is an affine function of μ for each (x,y), we may consider w.l.o.g. only the finitely-many pure column strategies that are vertices of the polytope. Thus we may employ the minimax theorem to find a mixed column strategy ν such that for every pure row strategy (x,y) we have $\mathbf{E}_{\mu\sim\nu}[P((x,y),\mu)] \geq (1-\epsilon)\alpha^*$. Consider a protocol Π that publicly samples $\mu \sim \nu$ and $(S,b) \sim \mu$, then checks whether $(x,y) \in S$ (with 2 bits of communication) and outputs b if so and $\perp$ if not. Then for each (x,y),

- $\mathbf{Pr}[\Pi(x,y) \neq \perp] = \mathbf{E}_{\mu\sim\nu}\big[\mathbf{Pr}_{(S,b)\sim\mu}\big[(x,y) \in S \text{ and } b \neq \perp\big]\big] \leq \mathbf{E}_{\mu\sim\nu}[\alpha^*] = \alpha^*$ by the definition of pure column strategies, and
- $\mathbf{Pr}[\Pi(x,y) = F(x,y)] = \mathbf{E}_{\mu\sim\nu}\big[\mathbf{Pr}_{(S,b)\sim\mu}\big[(x,y) \in S \text{ and } b = F(x,y)\big]\big] = \mathbf{E}_{\mu\sim\nu}[P((x,y),\mu)] \geq (1-\epsilon)\alpha^*$.

Thus Π witnesses that $\mathsf{2WAPP}^{\mathsf{cc}*}_{\epsilon}(F) \leq 2 + \log(1/\alpha^*)$.

Chapter 6

A Composition Theorem for Conical Juntas

Overview. In this chapter, we describe a general method of proving degree lower bounds for conical juntas that compute recursively defined boolean functions. Using the simulation theorem from Chapter 2, we give two applications: AND-OR *trees:* We show a near-optimal $\tilde{\Omega}(n^{0.753\ldots})$ randomised communication lower bound for the recursive NAND function. This answers an open question posed by Beame and Lawry [BL92, Law93]. *Majority trees:* We show an $\Omega(2.59^k)$ randomised communication lower bound for the 3-majority tree of height k. This improves over the state-of-the-art already in the context of randomised decision tree complexity. This chapter is based on the following publication:

[GJ16]: Mika Göös and T.S. Jayram. A composition theorem for conical juntas. In *Proceedings of the 31st Conference on Computational Complexity (CCC)*, pages 5:1–5:16. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.CCC.2016.5

6.1 Introduction

The purpose of this chapter is to prove lower bounds on the *degree* $\deg(h)$ (maximum width of a conjunction in h) of any conical junta h that computes—even approximately—a given boolean function $f\colon \{0,1\}^n \to \{0,1\}$. More precisely, we study the ϵ-*approximate* conical junta degree of f, denoted $\deg_\epsilon(f)$, that is defined as the minimum degree of a conical junta h satisfying

$$\forall x : \quad |h(x) - f(x)| \;\leq\; \epsilon.$$

Our main technical result is a *Composition Theorem* that makes it easy to prove conical junta degree lower bounds for functions that are defined from simpler functions via composition. Recall that if f and g are boolean functions on n and m bits, respectively, their *composition* $f \circ g^n$ is the function on nm bits that maps an input $x = (x_1, \ldots, x_n) \in (\{0,1\}^m)^n$ to the output

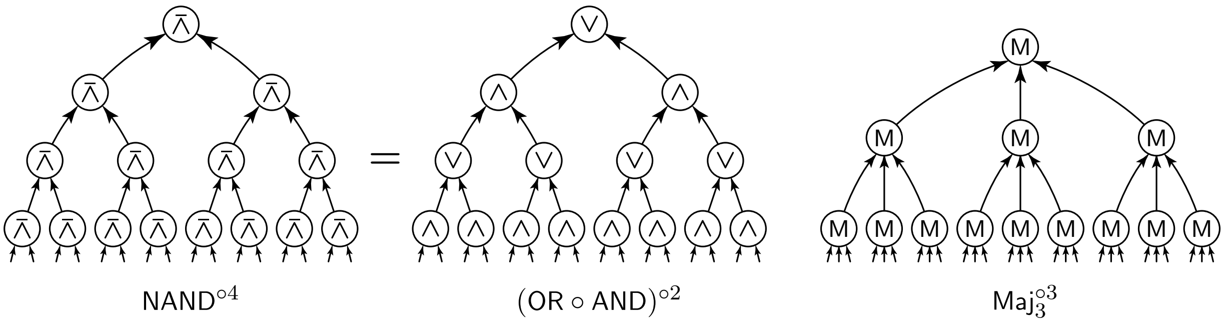


Figure 6.1: Examples of recursively defined boolean functions studied in this chapter.

$(f \circ g^n)(x) := f(g(x_1), \ldots, g(x_n))$. Define also $f^{\circ k} := f \circ (f^{\circ (k-1)})^n$ where $f^{\circ 1} := f$. The exact statement of the Composition Theorem is deferred to Section 6.3 as it is somewhat technical. It is phrased in terms of *dual solutions* (or *certificates*) to a linear program that captures a certain *average* version of conical junta degree (defined in Section 6.2). The theorem splits the task of proving lower bounds into two steps: we first need to find dual certificates for f and g (e.g., by solving an LP, either by inspection, or by using a computer), and then we can let the Composition Theorem construct a dual certificate for $f \circ g^n$ in a black-box fashion. We note that similar LP-based approaches have been extremely popular in analysing the degree of multivariate polynomials (see [She13a, She14a, BT15] for recent examples)—in short, this chapter develops such a framework for conical juntas, a nonnegative analogue of multivariate polynomials.

Setting these technical matters aside for a moment, let us illustrate the power the Composition Theorem by looking at some of its consequences.

6.1.1 Query complexity

We give applications for two well-studied recursively defined boolean functions; see Figure 6.1.

Theorem 6.1. $\deg_\epsilon(\mathsf{NAND}^{\circ k}) \geq \Omega(n^{0.753\ldots})$ *for all* $\epsilon \leq 1/n$ *where* $n := 2^k$.

Theorem 6.2. $\deg_\epsilon(\mathsf{Maj}_3^{\circ k}) \geq \Omega(2.59\ldots^k)$ *for all* $\epsilon \leq 1/n$ *where* $n := 3^k$.

Discussion of Theorem 6.1. The function $\mathsf{NAND}^{\circ k}$ is computed by a height-k binary tree consisting of NAND gates (a.k.a. AND-OR tree). A classical result [SW86, San95] states that any randomised decision tree needs to query $\Omega(n^{0.753\ldots})$ (here $0.753\ldots = \log(1+\sqrt{33})-2$) many input bits in order to compute $\mathsf{NAND}^{\circ k}$ with high probability. This matches an upper bound due to Snir [Sni85]. Our Theorem 6.1 shows that the same lower bound holds even for conical juntas that approximate $\mathsf{NAND}^{\circ k}$ sufficiently well. This is a qualitative strengthening of the classical results since conical juntas are relaxations of decision trees. Indeed, a randomised decision tree of depth d that computes a function f to within error $\epsilon > 0$ can be converted into a degree-d ϵ-approximate conical junta for f—the reason is the same as for multivariate polynomials [BdW02, Theorem 15]. Speaking of polynomials, Theorem 6.1 should be compared with the fact that the approximate polynomial degree of $\mathsf{NAND}^{\circ k}$ is only $O(\sqrt{n})$ (and this upper bound holds even for quantum algorithms [ACR+10]).

Note: A caveat with Theorems 6.1–6.2 is that we only know how to prove them for $\epsilon \leq 1/n$. By contrast, one usually takes $\epsilon = 1/3$ when studying decision trees, and this is well-known to be w.l.o.g., because the error can be reduced below any $\epsilon < 1/3$ with only a factor $O(\log(1/\epsilon))$ increase in query complexity. Interestingly, for conical juntas, we showed in Chapter 2 that ϵ cannot always be efficiently reduced: for any constants $\epsilon > \delta > 0$ there exists a *partial* function f with $\deg_\epsilon(f) = 1$ but $\deg_\delta(f) \geq \Omega(n)$. For *total* functions, it is still open whether efficient error reduction is possible (standard techniques [BdW02] at least show that $\deg_\epsilon(f)$ is polynomially related to $\deg_0(f)$). In any case, Theorems 6.1–6.2 do indeed imply lower bounds for randomised decision trees with error $\epsilon = 1/3$: we simply have to reduce the error below $1/n$ first and only then convert the decision tree into a conical junta. This incurs a factor $\Theta(\log n)$ loss in the value of the lower bound.

Discussion of Theorem 6.2. For the reasons discussed above, Theorem 6.2 implies a lower bound of $\tilde{\Omega}(2.59...^k) \geq \Omega(2.59^k)$ (here $2.59... = \sqrt[3]{35/2}$, and the $\tilde{\Omega}$-notation hides polylog(n) factors) for the randomised query complexity of the recursive majority function $\mathsf{Maj}_3^{\circ k}$. This slightly improves over the previous bound of $\Omega(2.57^k)$ that is the culmination of the line of work [JKS03, LNPV06, Leo13, MNS+15] wielding information theoretic tools. For comparison, a randomised zero-error decision tree of cost $O(2.65^k)$ is known [MNS+15]. Even though our quantitative improvement in Theorem 6.2 is modest, the theorem nevertheless suggests that our new techniques are rather powerful: they are already competitive with highly optimised prior work, especially [MNS+15].

6.1.2 Communication complexity

Using the machinery of Chapter 2 we can now translate Theorems 6.1–6.2 into analogous communication results. The translation incurs some polylog(n) factor loss in parameters, which is suppressed by the $\tilde{\Omega}$-notation used below. Here $\mathsf{BPP}^{\mathsf{cc}}(F)$ stands for the bounded-error communication complexity of F under a worst-case Alice–Bob bipartition of the input bits. For our functions, we may take the bipartition to be such that Alice gets the first bit of every bottom gate and Bob gets the rest.

Theorem 6.3. $\mathsf{BPP}^{\mathsf{cc}}(\mathsf{NAND}^{\circ k}) \geq \tilde{\Omega}(n^{0.753...})$.

Theorem 6.4. $\mathsf{BPP}^{\mathsf{cc}}(\mathsf{Maj}_3^{\circ k}) \geq \Omega(2.59^k)$.

Discussion of Theorem 6.3. The question of proving a lower bound for the randomised communication complexity of the balanced alternating AND-OR tree (with fan-in 2 gates next to the inputs) having n leaves was first posed by Beame and Lawry [BL92, Law93] to the best of our knowledge. They were interested in matching the randomised query complexity bound, towards separating randomised communication complexity from both nondeterministic and co-nondeterministic communication complexity. Two independent works [JKR09, LS10] (building on [JKS03]) arrived at a lower bound of $\Omega(n/2^{O(k)})$ (or slightly worse $\Omega(n/k^{O(k)})$ in [JKR09]) for the randomised communication complexity of any height-k unbounded fan-in alternating AND-OR tree (with fan-in 2 gates next to the inputs). While this lower bound is tight when $k = O(1)$, the bound becomes trivial in the setting of Theorem 6.3 where $k = \log n$. This shortcoming was partially addressed by [JKZ10] who showed, via a reduction from set-disjointness, a lower bound of $\Omega(\sqrt{n})$ for such AND-OR trees, independently of the height. Our Theorem 6.3 now gives an essentially optimal $\tilde{\Omega}(n^{0.753\ldots})$ bound for the particular case of $\mathsf{NAND}^{\circ k}$. It remains open whether this lower bound holds for *all* AND-OR trees (with the appropriate gates next to the inputs). For *query complexity*, Amano [Ama11a] has come close to settling this question, known as the Saks–Wigderson conjecture [SW86] for the class of read-once formulas (a more general version of the conjecture was recently disproved [ABB+16a]).

Discussion of Theorem 6.4. The function $\mathsf{Maj}_3^{\circ k}$ has not been studied in communication complexity previously—after all, even its randomised query complexity is not yet completely understood.

6.2 Definitions and examples

We write $h = \sum w_C C$ for a generic conical junta, where the sum ranges over different conjunctions of literals $C\colon \{0,1\}^n \to \{0,1\}$ and $w_C \geq 0$ for each C. Note that $h\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}$. Let $|C|$ denote the width of a conjunction C, i.e., the number of literals in C. The degree of h, denoted $\deg(h)$, is defined as the maximum width of a conjunction C with $w_C > 0$. Here, it is helpful to work with a more robust notion of degree that we call *average degree*. The *average degree* of h, denoted $\operatorname{adeg}(h)$, is defined as the maximum over all inputs x of

$$\operatorname{adeg}_x(h) \;\coloneqq\; \sum w_C |C| C(x) \;=\; \sum w_C \operatorname{adeg}_x(C).$$

In particular, $\operatorname{adeg}(h) \leq \deg(h)$ in the natural setting where $h(x) \leq 1$ for all x. Our definition of average degree is in perfect analogy to the usual definition of cost for randomised zero-error decision trees, namely, charging for the *expected* number of queries made on a given input. Indeed, it is not hard to see that any zero-error decision tree of cost d gives rise to a conical junta of average degree d computing exactly the same boolean function as the decision tree.

For a boolean function $f\colon \{0,1\}^n \to \{0,1\}$ we define

– *Degree:* $\deg(f)$ is the minimum $\deg(h)$ over all conical juntas h computing f.

– *Average degree:* $\operatorname{adeg}(f)$ is the minimum $\operatorname{adeg}(h)$ over all conical juntas h computing f.

- *Approximate degree:* $\deg_\epsilon(f)$ is the minimum $\deg(h)$ over all conical juntas h that compute f to within error ϵ, i.e., $h(x) \in f(x) \pm \epsilon$ for all x.

6.2.1 Tame examples

For our conical juntas h_1 and h_2 from (1.1), we have $\text{adeg}(h_1) = \text{adeg}_{10}(h_1) = 3/2 < 2 = \deg(h_1)$ and $\text{adeg}(h_2) = \text{adeg}_{110}(h_2) = 8/3 < 3 = \deg(h_2)$. In fact, h_1 and h_2 are optimal:

$$\text{adeg}(\mathsf{OR}) \;=\; 3/2 \qquad \text{and} \qquad \text{adeg}(\mathsf{Maj}_3) \;=\; 8/3.$$

This can be seen by solving an LP whose value is $\text{adeg}(f)$, as is discussed shortly. Note that our degree measures are inherently *one-sided*: f and its negation $\neg f$ need not have the same degree. For example, we have $\text{adeg}(\neg\mathsf{OR}) = 2$ (observe that $\bar{x}_1\bar{x}_2$ is the only conical junta for $\neg\mathsf{OR}$) even though $\text{adeg}(\mathsf{OR}) = 3/2$. (More dramatic gaps can be demonstrated using variations of a function introduced in Chapter 5.) By contrast, Maj_3 is *self-dual*, $\neg\mathsf{Maj}_3(x_1, x_2, x_3) = \mathsf{Maj}_3(\neg x_1, \neg x_2, \neg x_3)$, so we automatically have $\text{adeg}(\mathsf{Maj}_3) = \text{adeg}(\neg\mathsf{Maj}_3)$.

6.2.2 A wild example!

What is the average degree of $\mathsf{OR} \circ \mathsf{Maj}_3^2$? We can obtain a conical junta for this function starting with the optimal conical juntas $h_1(x)$, $h_2(y)$, $\bar{h}_2(y) := h_2(\bar{y}_1, \bar{y}_2, \bar{y}_3)$ computing OR, Maj_3, $\neg\mathsf{Maj}_3$, respectively, as follows: Let $z^1 = (z^1_1, z^1_2, z^1_3)$ and $z^2 = (z^2_1, z^2_2, z^2_3)$ be fresh variables. Start with $h_1(x)$ and replace every positive literal x_i by $h_2(z^i)$ and every negative literal $\bar{x}_i$ by $\bar{h}_2(z^i)$. This construction shows that

$$\text{adeg}(\mathsf{OR} \circ \mathsf{Maj}_3^2) \;\leq\; 3/2 \cdot 8/3 \;=\; 4.$$

It would be natural to conjecture that this is tight—*but this conjecture is false!* There is in fact a more effective conical junta of average degree only $47/12 \approx 3.92$. An analogous phenomenon is well-known in the context of zero-error decision trees: so-called *directional* decision trees need not be optimal for composed functions [SW86, Ver98, Ama11b].

What of it? This example shows that we cannot hope for a perfect composition theorem for average degree that would determine $\text{adeg}(f \circ g^n)$ solely in terms of $\text{adeg}(f)$, $\text{adeg}(g)$, and $\text{adeg}(\neg g)$, even assuming $\text{adeg}(g) = \text{adeg}(\neg g)$. Consequently, for our LP-based Composition Theorem, we will have to introduce *some* technical assumptions: to enable the construction of a dual certificate for $\text{adeg}(f \circ g^n)$, we assume we have dual certificates *of a special form* for $\text{adeg}(f)$, $\text{adeg}(g)$, $\text{adeg}(\neg g)$. The rest of this section develops our LP formalism for average degree.

6.2.3 Generalised input costs

Let us first generalise the definition of $\operatorname{adeg}(h)$ by allowing arbitrary costs $b_0, b_1 \geq 0$ to be assigned to reading the input bits. That is, for a conjunction C, we set $|C|_{b_0,b_1} := b_0 \cdot (\#\text{ of 0's read by } C) + b_1 \cdot (\#\text{ of 1's read by } C)$. In particular, $|C|_{1,1} = |C|$. Then $\operatorname{adeg}(h; b_0, b_1)$ is defined as the maximum over all inputs x of

$$\operatorname{adeg}_x(h; b_0, b_1) \;:=\; \sum w_C |C|_{b_0,b_1} C(x) \;=\; \sum w_C \operatorname{adeg}_x(C; b_0, b_1).$$

We also introduce some "distributional" notation: for a distribution D_1 over $f^{-1}(1)$ we let

$$\operatorname{adeg}_{D_1}(h; b_0, b_1) \;:=\; \mathop{\mathbf{E}}_{x \sim D_1} \big[\operatorname{adeg}_x(h; b_0, b_1)\big].$$

For a boolean function $f \colon \{0,1\}^n \to \{0,1\}$ we define

- $\operatorname{adeg}(f; b_0, b_1)$ is the minimum of $\operatorname{adeg}(h; b_0, b_1)$ over all conical juntas h computing f.
- $\operatorname{adeg}_{D_1}(f; b_0, b_1)$ is the minimum of $\operatorname{adeg}_{D_1}(h; b_0, b_1)$ over all conical juntas h computing f.

It is clear that $\operatorname{adeg}(f; b_0, b_1) \geq \operatorname{adeg}_{D_1}(f; b_0, b_1)$ for all distributions D_1. (In fact, it can be shown using the minimax theorem that this inequality can be turned into an equality if we maximise over D_1 on the right hand side—however, we do not use this fact.)

6.2.4 An LP for average degree

We formulate $\operatorname{adeg}_{D_1}(f; b_0, b_1)$ as the optimum value of an LP—here the data f, D_1, b_0, b_1, is thought of as fixed. We have a nonnegative variable $w_C \geq 0$ for each of the 3^n possible conjunctions $C \colon \{0,1\}^n \to \{0,1\}$. Here is the LP:

$$\begin{array}{rlll}
\min & \operatorname{adeg}_{D_1}\big(\sum w_C C; b_0, b_1\big) & & \\
\textit{subject to} & \sum w_C C(x) = f(x), & \forall x & \text{(Primal)} \\
& w_C \geq 0, & \forall C &
\end{array}$$

Here is the LP dual; the free variables are packaged into a function $\Psi \colon \{0,1\}^n \to \mathbb{R}$.

$$\begin{array}{rlll}
\max & \langle \Psi, f \rangle & & \\
\textit{subject to} & \langle \Psi, C \rangle \leq \operatorname{adeg}_{D_1}(C; b_0, b_1), & \forall C & \text{(Dual)} \\
& \Psi(x) \in \mathbb{R}, & \forall x &
\end{array}$$

Since we are interested in proving lower bounds on average degree, we are only going to need the "weak" form of LP duality: Suppose $h = \sum w_C C$ is an optimal solution to (Primal). Then any solution Ψ that is feasible for (Dual) witnesses a lower bound on $\operatorname{adeg}(f; b_0, b_1)$ like so:

$$\begin{aligned}
\operatorname{adeg}(f; b_0, b_1) &\geq \operatorname{adeg}_{D_1}(f; b_0, b_1) \\
&= \operatorname{adeg}_{D_1}(h; b_0, b_1) \\
&= \textstyle\sum w_C \operatorname{adeg}_{D_1}(C; b_0, b_1) \\
&\geq \textstyle\sum w_C \langle \Psi, C \rangle \\
&= \langle \Psi, \textstyle\sum w_C C \rangle \\
&= \langle \Psi, f \rangle.
\end{aligned} \tag{6.1}$$

6.3 Statement of the Composition Theorem

We start by defining an $(a_0, a_1; b_0, b_1)$-*certificate* for f as a special collection of certificates witnessing

$$\begin{aligned}
\operatorname{adeg}(f; b_0, b_1) &\geq a_1, \\
\operatorname{adeg}(\neg f; b_0, b_1) &\geq a_0.
\end{aligned} \tag{6.2}$$

Definition 6.5. Call a function $\Psi\colon \{0,1\}^n \to \mathbb{R}$ *balanced* if $\sum_x \Psi(x) = 0$, and also write $X_{\geq 0} \coloneqq \max\{X, 0\}$ for short. An $(a_0, a_1; b_0, b_1)$-*certificate* for a function $f\colon \{0,1\}^n \to \{0,1\}$ consists of four balanced functions $\{\Psi_v, \hat{\Psi}_v\}_{v=0,1}$ mapping $\{0,1\}^n \to \mathbb{R}$ such that the following hold.

- **Special form:** Functions Ψ_0 and Ψ_1 have the form

$$\Psi_v = a_v(D_v - D_{1-v}), \tag{6.3}$$

  where D_v is a distribution over $f^{-1}(v)$. Moreover, $\hat{\Psi}_v$ is supported on $f^{-1}(v)$.

- **Feasibility:** For all conjunctions C and $v \in \{0,1\}$,

$$\langle \Psi_v, C \rangle_{\geq 0} + \langle \hat{\Psi}_v, C \rangle \leq \operatorname{adeg}_{D_v}(C; b_0, b_1). \tag{6.4}$$

Theorem 6.6 (Composition Theorem)**.** *Suppose f admits an $(a_0, a_1; b_0, b_1)$-certificate and g admits a $(b_0, b_1; 1, 1)$-certificate. Then $f \circ g^n$ admits an $(a_0, a_1; 1, 1)$-certificate.*

Discussion. First, we note that (6.4) actually packs together two *linear* inequalities; it would be equivalent to require that both $\Psi_v + \hat{\Psi}_v$ and $\hat{\Psi}_v$ are feasible for (Dual), namely that

$$\begin{cases} \langle \Psi_v + \hat{\Psi}_v, C \rangle \leq \operatorname{adeg}_{D_v}(C; b_0, b_1), \\ \qquad\;\; \langle \hat{\Psi}_v, C \rangle \leq \operatorname{adeg}_{D_v}(C; b_0, b_1). \end{cases} \tag{6.4'}$$

Here $\Psi_1 + \hat{\Psi}_1$ is the main attraction: it witnesses a lower bound of $\langle \Psi_1 + \hat{\Psi}_1, f\rangle = \langle \Psi_1, f\rangle + \langle \hat{\Psi}_1, f\rangle = a_1 + 0 = a_1$ for $\operatorname{adeg}(f; b_0, b_1)$ as promised above (6.2); similarly, $\Psi_0 + \hat{\Psi}_0$ witnesses the complementary lower bound $\operatorname{adeg}(\neg f; b_0, b_1) \geq a_0$.

The requirement that $\Psi_1 + \hat{\Psi}_1$ must be balanced is perhaps our most critical assumption. We use it to manoeuvre around the counterexample of Section 6.2.2: we have $\operatorname{adeg}(\mathsf{Maj}_3) = 8/3$, while the best *balanced* solution to (Dual) only witnesses the lower bound $\operatorname{adeg}(\mathsf{Maj}_3) \geq 5/2$ (see also Figure 6.3). The requirement that $\hat{\Psi}_v$ is feasible for (Dual) is merely a technical assumption that helps us in the upcoming proof (akin to a "strengthened induction hypothesis"); we do not know whether the theorem is true without this condition. Another technical assumption is (6.3), which allows us to assume that Ψ_1 and Ψ_0 have opposite signs: $\Psi_1 = -a_1/a_0 \cdot \Psi_0$.

Some simple certificates are illustrated in Figures 6.2–6.3. Their feasibility can be checked by hand. For more involved functions, certificates can in principle be found via a computer search (using computers is not uncommon even in "lower bounds" research [Ama14a]). We will in fact use this approach for $\mathsf{Maj}_3^{\circ k}$ in Section 6.5.

6.4 Proof of the Composition Theorem

Let $\{\Psi_v, \hat{\Psi}_v\}_{v=0,1}$ and $\{\Phi_v, \hat{\Phi}_v\}_{v=0,1}$ be the certificates for f and g, respectively. Our goal is to construct a certificate $\{\Upsilon_v, \hat{\Upsilon}_v\}_{v=0,1}$ for $f \circ g^n$. We use the following notation:

$$\underbrace{\Psi_v \;\coloneqq\; a_v(F_v - F_{1-v})}_{\text{given}}, \qquad \underbrace{\Phi_v \;\coloneqq\; b_v(G_v - G_{1-v})}_{\text{given}}, \qquad \underbrace{\Upsilon_v \;\coloneqq\; a_v(D_v - D_{1-v})}_{\text{want to construct}}.$$

By assumption, the distribution F_v is supported on $f^{-1}(v)$ and G_v is supported on $g^{-1}(v)$. We will define D_v to be supported on $(f \circ g^n)^{-1}(v)$.

6.4.1 Construction

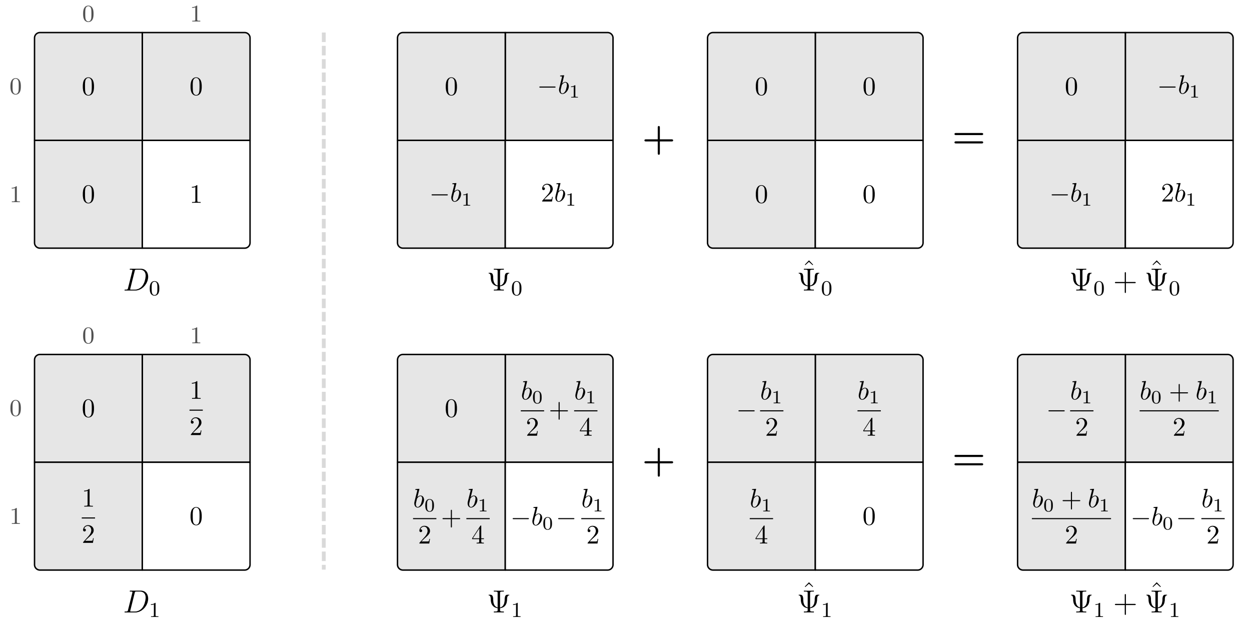
Lifts. Let $\Gamma\colon \{0,1\}^n \to \mathbb{R}$ and suppose that for each $y \in \{0,1\}^n$ we have a function $H_y\colon \{0,1\}^{mn} \to \mathbb{R}$ supported on $(g^n)^{-1}(y) = g^{-1}(y_1) \times \cdots \times g^{-1}(y_n)$. The *lift* of Γ by H is

$$\Gamma^H \;\coloneqq\; \textstyle\sum_{y\in\{0,1\}^n} \Gamma(y) \cdot H_y.$$

In particular, if Γ and the H_y's are probability distributions, so is Γ^H. Note also that if Γ is supported on $f^{-1}(v)$, then Γ^H is supported on $(f \circ g^n)^{-1}(v)$.

New certificate. Write $G_y \coloneqq G_{y_1} \times \cdots \times G_{y_n}$ for the canonical product distribution on $(g^n)^{-1}(y)$. We also need a modified version of G_y, denoted $(G \overset{i}{\leftarrow} \hat{\Phi})_y$ where $i \in [n]$, that has a copy of $\hat{\Phi}_{y_i}$ in place of G_{y_i}; more formally

$$(G \overset{i}{\leftarrow} \hat{\Phi})_y(x) \;\coloneqq\; \hat{\Phi}_{y_i}(x_i) \cdot \textstyle\prod_{j\neq i} G_{y_j}(x_j).$$



$$\underset{D_0}{\begin{array}{c|cc} & 0 & 1\\ \hline 0 & 0 & 0\\ 1 & 0 & 1\end{array}} \qquad \underset{\Psi_0}{\begin{pmatrix}0 & -b_1\\ -b_1 & 2b_1\end{pmatrix}} + \underset{\hat{\Psi}_0}{\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}} = \underset{\Psi_0+\hat{\Psi}_0}{\begin{pmatrix}0 & -b_1\\ -b_1 & 2b_1\end{pmatrix}}$$

$$\underset{D_1}{\begin{array}{c|cc} & 0 & 1\\ \hline 0 & 0 & \frac{1}{2}\\ 1 & \frac{1}{2} & 0\end{array}} \qquad \underset{\Psi_1}{\begin{pmatrix}0 & \frac{b_0}{2}+\frac{b_1}{4}\\ \frac{b_0}{2}+\frac{b_1}{4} & -b_0-\frac{b_1}{2}\end{pmatrix}} + \underset{\hat{\Psi}_1}{\begin{pmatrix}-\frac{b_1}{2} & \frac{b_1}{4}\\ \frac{b_1}{4} & 0\end{pmatrix}} = \underset{\Psi_1+\hat{\Psi}_1}{\begin{pmatrix}-\frac{b_1}{2} & \frac{b_0+b_1}{2}\\ \frac{b_0+b_1}{2} & -b_0-\frac{b_1}{2}\end{pmatrix}}$$

Figure 6.2: A $(2b_1, b_0 + \frac{1}{2}b_1; b_0, b_1)$-certificate for $\mathsf{NAND}\colon \{0,1\}^2 \to \{0,1\}$ that is valid for all $b_0, b_1 \geq 0$. The 1-inputs $\mathsf{NAND}^{-1}(1)$ are highlighted in gray. For feasibility, there are 6 equivalence classes (see Section 6.5.2) of conjunctions to check: $\{\ast\ast, \ast 0, \ast 1, 00, 10, 11\}$.

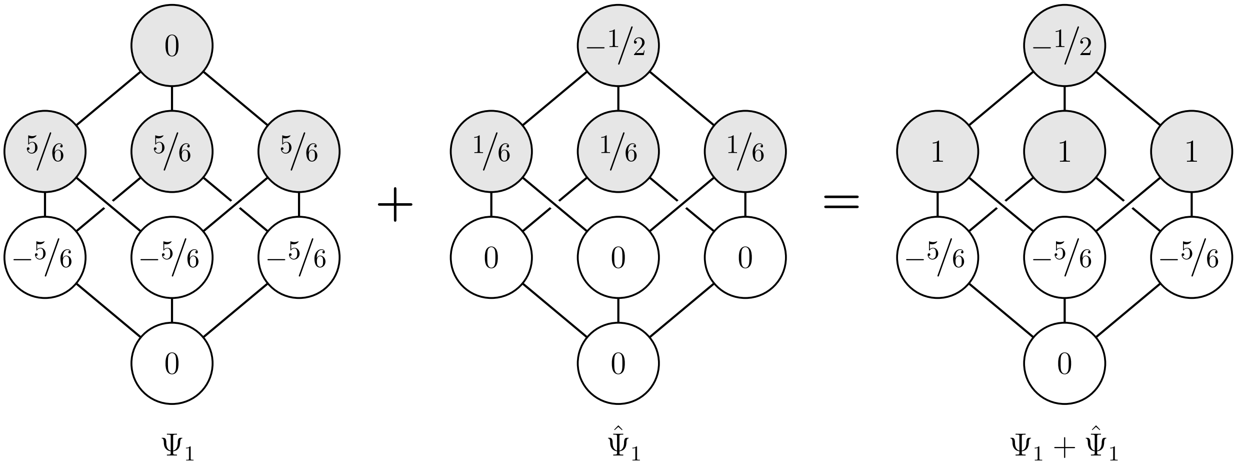


Figure 6.3: A $(\frac{5}{2}, \frac{5}{2}; 1, 1)$-certificate for $\mathsf{Maj}_3\colon \{0,1\}^3 \to \{0,1\}$. The 1-inputs $\mathsf{Maj}_3^{-1}(1)$ are highlighted in gray. Only $\Psi_1, \hat{\Psi}_1$ are shown as $\Psi_0, \hat{\Psi}_0$ are defined via self-duality. Here D_v is uniform on inputs of Hamming weight $v+1$. For feasibility, there are 10 equivalence classes of conjunctions to check: $\{\ast\ast\ast, \ast\ast 1, \ast\ast 0, \ast 00, \ast 10, \ast 11, 000, 100, 110, 111\}$. Note that for any $\alpha \geq 0$, we can obtain an $(\frac{5}{2}\alpha, \frac{5}{2}\alpha; \alpha, \alpha)$-certificate by simply scaling the functions $\Psi_v, \hat{\Psi}_v$ by α.

Note that $(G\overset{i}{\leftarrow}\hat{\Phi})_y$ is a balanced function supported on $(g^n)^{-1}(y)$. We now define $\{\Upsilon_v, \hat{\Upsilon}_v\}_{v=0,1}$ by

$$\begin{aligned} \Upsilon_v &\coloneqq \Psi^G_v, \\ \hat{\Upsilon}_v &\coloneqq \hat{\Psi}^G_v + \textstyle\sum_{i=1}^n F^{G\overset{i}{\leftarrow}\hat{\Phi}}_v. \end{aligned} \tag{6.5}$$

Since $\Psi^G_v = a_v(F^G_v - F^G_{1-v})$, we have $D_v = F^G_v$. It is also easy to check that $\hat{\Upsilon}_v$ is a balanced function supported on $(f \circ g^n)^{-1}(v)$. Hence $\{\Upsilon_v, \hat{\Upsilon}_v\}_{v=0,1}$ is of the special form required of an $(a_0, a_1; 1, 1)$-certificate for $f \circ g^n$. The interesting part is to verify the feasibility condition (6.4).

6.4.2 Feasibility

Fix a conjunction C in the domain of $f \circ g^n$. Our goal is to show

$$\langle \Psi^G_v, C\rangle_{\geq 0} + \langle \hat{\Psi}^G_v + \textstyle\sum_i F^{G\overset{i}{\leftarrow}\hat{\Phi}}_v, C\rangle \;\leq\; \mathrm{adeg}_{D_v}(C). \tag{6.6}$$

Extracting a conical junta from *C*. Our analysis will be centered around a conical junta $h(y)$, defined below, that computes the acceptance probability $\mathbf{Pr}_{x\sim G_y}[C(x) = 1] = \mathbf{E}_{x\sim G_y}[C(x)] = \langle G_y, C\rangle$. In a certain sense, h serves as a *projection* of C to the domain of f. Write $C(x) = \prod_{i=1}^n C_i(x_i)$ where C_i is a conjunction depending only on x_i. Since G_y is a product distribution,

$$\langle G_y, C\rangle \;=\; \textstyle\prod_i \langle G_{y_i}, C_i\rangle \;=:\; \prod_i p_{i,y_i},$$

where we wrote $p_{i,v} \coloneqq \langle G_v, C_i\rangle \in \mathbb{R}_{\geq 0}$ for short. Fix $y^* \in \{0,1\}^n$ such that $p_{i,y^*_i} \geq p_{i,1-y^*_i}$ for all i. We now define $h(y)$ that computes $\langle G_y, C\rangle$:

$$h(y) \;\coloneqq\; \textstyle\prod_{i=1}^n \big(p_{i,1-y^*_i} + \underbrace{(p_{i,y^*_i} - p_{i,1-y^*_i})}_{\geq 0} \cdot \ell_i\big) \qquad \text{where literal } \ell_i \text{ is } \begin{cases} y_i & \text{if } y^*_i = 1, \\ \bar{y}_i & \text{if } y^*_i = 0. \end{cases} \tag{6.7}$$

This product expression can be expanded into a conical combination of conjunctions, $h = \sum w_T T$, in the natural way, but the above "implicit" form is more concise.

Next, we record two properties of h that will suffice for the remaining analysis.

Lemma 6.7. $\mathrm{adeg}_y(h; b_0, b_1) = \sum_i \langle \Phi_{y_i}, C_i\rangle_{\geq 0} \prod_{j\neq i} \langle G_{y_j}, C_j\rangle$.

Proof. Write $h = \sum w_T T$. We compute the average degree by summing together the weights $\sum_{T\ni \ell_i} w_T T(y)$ contributed by each of the n literals ℓ_i, i.e.,

$$\mathrm{adeg}_y(h; b_0, b_1) \;=\; \textstyle\sum_i |\ell_i|_{b_0,b_1} \cdot \sum_{T\ni\ell_i} w_T T(y).$$

If i is such that $y_i \neq y^*_i$, we have $\ell_i(y) = 0$ and so $T(y) = 0$ for all $T \ni \ell_i$; hence ℓ_i contributes no weight in this case. Suppose then that i is such that $y_i = y^*_i$; here we can write

$$h(y) \;=\; p_{i,1-y_i} \textstyle\prod_{j\neq i} p_{j,y_j} \;+\; \ell_i \cdot (p_{i,y_i} - p_{i,1-y_i}) \prod_{j\neq i} p_{j,y_j}.$$

The conjunctions T underlying the first term do not involve ℓ_i, so they contribute no weight for ℓ_i. The conjunctions T underlying the second term all involve ℓ_i and contribute a total weight of $(p_{i,y_i} - p_{i,1-y_i})\prod_{j\neq i} p_{j,y_j}$. Altogether we get

$$\begin{aligned}
\operatorname{adeg}_y(h; b_0, b_1) &= \textstyle\sum_i |\ell_i|_{b_0,b_1} \cdot \sum_{T\ni\ell_i} w_T T(y) \\
&= \textstyle\sum_{i: y_i = y_i^*} b_{y_i} \cdot (p_{i,y_i} - p_{i,1-y_i}) \prod_{j\neq i} p_{j,y_j} \\
&= \textstyle\sum_i b_{y_i} (p_{i,y_i} - p_{i,1-y_i})_{\geq 0} \prod_{j\neq i} p_{j,y_j} \\
&= \textstyle\sum_i b_{y_i} (\langle G_{y_i}, C_i\rangle - \langle G_{1-y_i}, C_i\rangle)_{\geq 0} \prod_{j\neq i} \langle G_{y_j}, C_j\rangle \\
&= \textstyle\sum_i \langle b_{y_i}(G_{y_i} - G_{1-y_i}), C_i\rangle_{\geq 0} \prod_{j\neq i} \langle G_{y_j}, C_j\rangle \\
&= \textstyle\sum_i \langle \Phi_{y_i}, C_i\rangle_{\geq 0} \prod_{j\neq i} \langle G_{y_j}, C_j\rangle. \qquad \square
\end{aligned}$$

Lemma 6.8. $\langle \Gamma, h\rangle = \langle \Gamma^G, C\rangle$ *for all* $\Gamma\colon \{0,1\}^n \to \mathbb{R}$.

Proof. We calculate

$$\begin{aligned}
\langle \Gamma, h\rangle &= \textstyle\sum_y \Gamma(y) h(y) = \sum_y \Gamma(y)\langle G_y, C\rangle = \sum_y \Gamma(y)\big[\sum_x G_y(x) C(x)\big] \\
&= \textstyle\sum_x \big[\sum_y \Gamma(y) G_y(x)\big] C(x) = \sum_x \Gamma^G(x) C(x) = \langle \Gamma^G, C\rangle. \qquad \square
\end{aligned}$$

Analysis. Let us expand the right hand side of the desired inequality (6.6):

$$\begin{aligned}
\operatorname{adeg}_{D_v}(C) &= |C| \cdot \langle F_v^G, C\rangle \\
&= \mathbf{E}_{y\sim F_v} \big[|C| \cdot \langle G_y, C\rangle\big] \\
&= \mathbf{E}_{y\sim F_v} \big[\big(\textstyle\sum_i |C_i|\big) \cdot \prod_i \langle G_{y_i}, C_i\rangle\big] \\
&= \mathbf{E}_{y\sim F_v} \big[\textstyle\sum_i |C_i| \langle G_{y_i}, C_i\rangle \prod_{j\neq i} \langle G_{y_j}, C_j\rangle\big] \\
&= \mathbf{E}_{y\sim F_v} \big[\textstyle\sum_i \operatorname{adeg}_{G_{y_i}}(C_i) \prod_{j\neq i} \langle G_{y_j}, C_j\rangle\big].
\end{aligned}$$

Substituting our hypothesis $\operatorname{adeg}_{G_{y_i}}(C_i) \geq \langle \Phi_{y_i}, C_i\rangle_{\geq 0} + \langle \hat{\Phi}_{y_i}, C_i\rangle$ into the above, we obtain

$$\operatorname{adeg}_{D_v}(C) \;\geq\; \underbrace{\mathbf{E}_{y\sim F_v} \big[\textstyle\sum_i \langle \Phi_{y_i}, C_i\rangle_{\geq 0} \prod_{j\neq i} \langle G_{y_j}, C_j\rangle\big]}_{\textbf{(I)}} \;+\; \underbrace{\mathbf{E}_{y\sim F_v} \big[\textstyle\sum_i \langle \hat{\Phi}_{y_i}, C_i\rangle \prod_{j\neq i} \langle G_{y_j}, C_j\rangle\big]}_{\textbf{(II)}}.$$

For the first term,

$$\begin{aligned}
\textbf{(I)} &= \mathbf{E}_{y\sim F_v} \big[\operatorname{adeg}_y(h; b_0, b_1)\big] && \text{(Lemma 6.7)} \\
&= \operatorname{adeg}_{F_v}(h; b_0, b_1) \\
&\geq \langle \Psi_v, h\rangle_{\geq 0} + \langle \hat{\Psi}_v, h\rangle && \text{(Feasibility of } \{\Psi_v, \hat{\Psi}_v\} \text{ and (6.1))} \\
&= \langle \Psi_v^G, C\rangle_{\geq 0} + \langle \hat{\Psi}_v^G, C\rangle. && \text{(Lemma 6.8)}
\end{aligned}$$

For the second term,

$$\begin{aligned}\textbf{(II)} &= \mathbf{E}_{y\sim F_v}\left[\textstyle\sum_i \langle (G \xleftarrow{i} \hat{\Phi})_y, C\rangle\right] \\ &= \left\langle \textstyle\sum_i F_v^{G \xleftarrow{i} \hat{\Phi}}, C\right\rangle.\end{aligned}$$

Combining these yields (6.6). This concludes the proof of Theorem 6.6.

6.5 Approximate degree lower bounds

In this section we prove Theorems 6.1–6.2 using the Composition Theorem. We begin by observing that $(a_0, a_1; b_0, b_1)$-certificates $\{\Psi_v, \hat{\Psi}_v\}_{v=0,1}$ also yield lower bounds for approximate degree, if the 1-norm $\|\hat{\Psi}_1\|_1$ is not too large. We call $\{\Psi_v, \hat{\Psi}_v\}_{v=0,1}$ an $(a_0, a_1; b_0, b_1; c)$-*certificate* if $\max_v \|\hat{\Psi}_v\|_1 \le c$.

Lemma 6.9. *Suppose f admits an $(a_0, a_1; 1, 1; c)$-certificate. If $\epsilon \le 1/4$ and $c \cdot \epsilon \le a_1/4$, then*

$$\deg_\epsilon(f) \;\ge\; \Omega(a_1).$$

Proof. Fix a certificate $\{\Psi_v, \hat{\Psi}_v\}_{v=0,1}$ for f and suppose $\deg_\epsilon(f) = \deg(h)$ where h is a conical junta with $\|h - f\|_\infty \le \epsilon$. Since $h(x) \le 1 + \epsilon$ for all x, we have $\deg(h) \ge (1+\epsilon)^{-1}\operatorname{adeg}(h) \ge \Omega(\operatorname{adeg}(h))$. Now we calculate

$$\begin{aligned}\operatorname{adeg}(h) &\ge \langle \Psi_1 + \hat{\Psi}_1, h\rangle && \text{(as in (6.1))} \\ &= \langle \Psi_1 + \hat{\Psi}_1, f\rangle + \langle \Psi_1 + \hat{\Psi}_1, h - f\rangle \\ &\ge a_1 - |\langle \Psi_1 + \hat{\Psi}_1, h - f\rangle| \\ &\ge a_1 - \|\Psi_1 + \hat{\Psi}_1\|_1 \cdot \|h - f\|_\infty \\ &\ge a_1 - (\|\Psi_1\|_1 + \|\hat{\Psi}_1\|_1) \cdot \epsilon \\ &\ge a_1 - (2a_1 + c) \cdot \epsilon \\ &\ge a_1/4.\end{aligned}$$

□

We use the following version of the Composition Theorem where the bounds on 1-norms (following immediately from the definition (6.5)) are made explicit.

Theorem 6.10. *Suppose f admits an $(a_0, a_1; b_0, b_1; c)$-certificate and g admits a $(b_0, b_1; 1, 1; d)$-certificate. Then $f \circ g^n$ admits an $(a_0, a_1; 1, 1; c + nd)$-certificate.*

6.5.1 Proof of Theorem 6.1

We iteratively apply Theorem 6.10 as follows.

1. Assume we have an $(\alpha_k, \beta_k; 1, 1; \gamma_k)$-certificate for $\mathsf{NAND}^{\circ k}$ where $\gamma_k \ge \alpha_k, \beta_k$.

Function	Class representative	Class size	Ψ_1	$\hat{\Psi}_1$	$\Psi_1 + \hat{\Psi}_1$
$\mathsf{Maj}_3^{\circ 1}$	$(0,0,1)$	3	$-5/2$	0	$-5/2$
	$(0,1,1)$	3	$5/2$	$1/2$	3
	$(1,1,1)$	1	0	$-1/2$	$-1/2$
	All others		0	0	0
$\mathsf{Maj}_3^{\circ 2}$	$(001,001,011)$	81	$-20/3$	0	$-20/3$
	$(001,011,011)$	81	$20/3$	$7/3$	9
	$(000,011,011)$	27	0	$-1/3$	$-1/3$
	$(001,011,111)$	54	0	$-2/3$	$-2/3$
	$(011,011,011)$	27	0	$-4/3$	$-4/3$
	All others		0	0	0
$\mathsf{Maj}_3^{\circ 3}$	$(112,112,122)$	1594323	$-35/2$	0	-35/2
	$(112,122,122)$	1594323	$35/2$	$19/2$	27
	$(122,122,122)$	531441	0	$-7/2$	$-7/2$
	$(112,122,222)$	1062882	0	-2	-2
	$(112,122,123)$	2125764	0	$-4/3$	$-4/3$
	$(112,122,022)$	1062882	0	$-2/3$	$-2/3$
	$(111,122,122)$	531441	0	$-5/6$	$-5/6$
	$(113,122,122)$	531441	0	$-1/2$	$-1/2$
	$(012,122,122)$	1062882	0	$-2/3$	$-2/3$
	All others		0	0	0

Table 6.1: Certificates for $\mathsf{Maj}_3^{\circ\ell}$ for heights $\ell = 1,2,3$. The table lists $(\alpha_\ell, \alpha_\ell; 1, 1)$-certificates with values $\alpha_1 = 5/2$ (also illustrated in Figure 6.3), $\alpha_2 = 20/3$, and $\alpha_3 = 35/2$. Only $\Psi_1, \hat{\Psi}_1$ are shown as $\Psi_0, \hat{\Psi}_0$ are defined dually. We give the total weight for each equivalence class of inputs; the functions are uniform on each class. For height $\ell = 3$ we represent the inputs to the bottom-most Maj_3 gates by their Hamming weight, e.g., $001 \rightsquigarrow 1$, $011 \rightsquigarrow 2$, etc.

2. Obtain a $(2\beta_k, \alpha_k + \frac{1}{2}\beta_k; \alpha_k, \beta_k; \beta_k)$-certificate for NAND from Figure 6.2.
3. Compose the above to get an $(\alpha_{k+1}, \beta_{k+1}; 1, 1; \gamma_{k+1})$-certificate for $\mathsf{NAND}^{\circ(k+1)}$ where

$$\begin{aligned} \alpha_{k+1} &\coloneqq 2\beta_k, \\ \beta_{k+1} &\coloneqq \alpha_k + \beta_k/2, \\ \gamma_{k+1} &\coloneqq \beta_k + 2\gamma_k. \end{aligned}$$

Note that $\alpha_{k+1}, \beta_{k+1} \leq \gamma_{k+1} \leq 3\gamma_k$. Starting with $\alpha_0 = \beta_0 = \gamma_0 = 1$ these recurrences (famously [SW86]) evaluate to $\alpha_k, \beta_k = \Theta(n^{0.753\ldots})$ where $n \coloneqq 2^k$. In addition, $\gamma_k \leq 3^k \leq n^{1.6}$. Now take $\epsilon \leq 1/n$ in Lemma 6.9 to prove Theorem 6.1.

6.5.2 Computer search for certificates

Iteratively composing (scaled versions of) the $(5/2, 5/2; 1, 1)$-certificate given in Figure 6.3 would yield only an $\Omega(2.5^k)$ lower bound for $\mathsf{Maj}_3^{\circ k}$. This is the best possible for our approach if we were to just compose certificates for individual Maj_3 functions. To obtain a better lower bound, we can instead directly find a certificate for $\mathsf{Maj}_3^{\circ\ell}$ where ℓ is a small constant, and then compose that certificate. Table 6.1 gives certificates for $\mathsf{Maj}_3^{\circ\ell}$ for height up to $\ell = 3$. We used a computer to solve the dual LP (Dual), with the additional restriction that Ψ $(= \Psi_1 + \hat{\Psi}_1)$ should be balanced. The best balanced Ψ happened to satisfy the other conditions required by our Definition 6.5.

Notes on implementation. For computational efficiency, it is useful to prune the search space by eliminating symmetries. The symmetries of $\mathsf{Maj}_3^{\circ\ell}$ (permutations of input coordinates that do not change the value of the function) are the symmetries of the underlying height-ℓ ternary tree. These symmetries partition the set of inputs and the set of conjunctions into *equivalence classes*: two inputs/conjunctions are "equivalent" if one can be mapped to the other by a symmetry. The set of feasible solutions to the LP is also invariant under these symmetries. It follows that we may look w.l.o.g. for a Ψ that is invariant, i.e., uniform on each equivalence class. (Indeed, if Ψ is any feasible solution, we obtain an invariant solution by averaging Ψ over all the symmetries.) Thus we need only maintain one variable in the LP per equivalence class $\mathcal{X} \subseteq \{0,1\}^n$ recording the *total weight* $\sum_{x\in\mathcal{X}} \Psi(x)$ of that class. Also, for such invariant Ψ, we need only check the LP feasibility constraint $\langle \Psi, C\rangle \leq \mathrm{adeg}_{D_1}(C; b_0, b_1)$ for a single representative C from each class of conjunctions.

The optimal height-2 certificate happens to have the same *support* as the certificate produced by our Composition Theorem starting with two height-1 certificates. Inspired by this, in order to speed up the search for height 3, we only optimised over those Ψ whose support coincides with that coming from the Composition Theorem—this LP has only 9 variables (i.e., equivalence classes of inputs), but well over 100,000 constraints (i.e., equivalence classes of conjunctions).

It is open to analyse height 4. Is there an efficient separation oracle for (Dual)?

6.5.3 Proof of Theorem 6.2

Table 6.1 defines a certificate for $\mathsf{Maj}_3^{\circ 3}$ with parameters $(35/2, 35/2; 1, 1; 19)$ and we may scale the certificate by any scalar $\alpha \geq 0$ to obtain one with parameters $((35/2)\alpha, (35/2)\alpha; \alpha, \alpha; 19\alpha)$. Using Theorem 6.10 iteratively as in Section 6.5.1, we get a certificate for $\mathsf{Maj}_3^{\circ k}$ with parameters

$$((35/2)^{k/3}, (35/2)^{k/3}; 1, 1; 28^{k/3} \cdot 19).$$

Here $(35/2)^{k/3} \geq n^{0.8}$ and $28^{k/3} \cdot 19 \leq n^{1.1}$ where $n \coloneqq 3^k$. Hence we may apply Lemma 6.9 with $\epsilon \leq 1/n$ to conclude an ϵ-approximate degree lower bound of $\Omega((35/2)^{k/3}) = \Omega(2.59...^k)$.

6.6 Communication lower bounds

In this section we prove Theorems 6.3–6.4 by applying the main result of Chapter 2: a simulation of randomised communication protocols by conical juntas. To this end, let $\mathsf{IP}_b \colon \{0,1\}^b \times \{0,1\}^b \to$

$\{0,1\}$ be the two-party (Alice has x, Bob has y) inner-product function given by

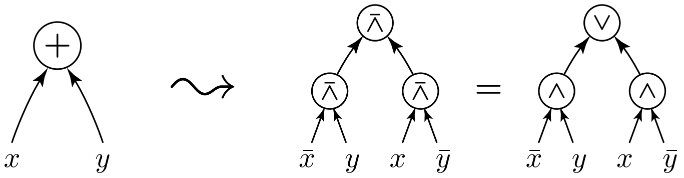
$$\mathsf{IP}_b(x,y) \;\coloneqq\; \langle x,y\rangle \bmod 2.$$

Let $\mathsf{BPP}^{\mathsf{cc}}_\epsilon(F)$ denote the randomised ϵ-error communication complexity of $F\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$. The following is a corollary of Theorem 2.35 (the original formulation there talks about $\mathsf{WAPP}^{\mathsf{dt}}_\epsilon(f)$ which is the same as $\deg_\epsilon(f)$; moreover, the result is stated for $\epsilon=\Theta(1)$, but the theorem is true more generally for $\epsilon = 2^{-\Theta(b)}$).

Theorem 6.11. *Let $\epsilon \coloneqq 1/n$ and $b\coloneqq\Theta(\log n)$ (with a large enough implicit constant). For any $f\colon\{0,1\}^n\to\{0,1\}$ we have*

$$\mathsf{BPP}^{\mathsf{cc}}_{\epsilon/2}(f\circ \mathsf{IP}_b^n) \;\geq\; \Omega(\deg_\epsilon(f)\cdot b).$$

Let us prove Theorem 6.3 (a similar argument works for Theorem 6.4). A key observation (also made in [JKZ10, §3]) is that $\mathsf{IP}_b = \mathsf{XOR}_b\circ\mathsf{AND}^b$ reduces to computing a binary NAND tree on $O(b^2)$ bits. To see this, think of the b-bit parity function XOR_b as a height-$(\log b)$ binary tree of XOR gates. Each such XOR gate can be rewritten as a height-2 NAND tree (with some negations on inputs):



In the binary XOR tree, replace the top XOR gate with this NAND tree (this involves making copies of some subtrees), push the negations to inputs, and repeat recursively. This gives us a height-$(2\log b)$ NAND tree. Moreover, the bottom layer of AND gates in IP_b is also easily simulated by NAND gates. Consequently, for some $N\coloneqq\Theta(nb^2)$, the communication matrix of $\mathsf{NAND}^{\circ\log n}\circ\mathsf{IP}_b^n$ appears as a submatrix of $\mathsf{NAND}^{\circ\log N}$ (relative to some bipartition of the input given by the reduction).

We can now derive Theorem 6.3—here ϵ and b are defined as in Theorem 6.11, and $\gtrsim$ means that we ignore polylog(N) factors.

$$\begin{aligned}
\mathsf{BPP}^{\mathsf{cc}}_{1/3}(\mathsf{NAND}^{\circ\log N}) \;&\gtrsim\; \mathsf{BPP}^{\mathsf{cc}}_{\epsilon/2}(\mathsf{NAND}^{\circ\log N}) && \text{(Error reduction)}\\
&\gtrsim\; \mathsf{BPP}^{\mathsf{cc}}_{\epsilon/2}(\mathsf{NAND}^{\circ\log n}\circ\mathsf{IP}_b^n) && \text{(Key observation)}\\
&\gtrsim\; \deg_\epsilon(\mathsf{NAND}^{\circ\log n}) && \text{(Theorem 6.11)}\\
&\gtrsim\; n^{0.753\ldots} && \text{(Theorem 6.1)}\\
&=\; \tilde\Theta(N^{0.753\ldots}).
\end{aligned}$$

Chapter 7

Lower Bounds via Critical Block Sensitivity

Overview. In this chapter, we use *critical block sensitivity*, a new complexity measure introduced by Huynh and Nordström [HN12], to study the communication complexity of search problems. To begin, we give a simple new proof of the following central result of Huynh and Nordström: if S is a search problem with critical block sensitivity b, then every randomised two-party protocol solving a the composed problem $S \circ g^n$ (where g is a certain constant-size gadget) requires $\Omega(b)$ bits of communication. Besides simplicity, our proof has the advantage of generalising to the multi-party setting. We obtain the following applications.

- *Monotone circuit depth:* We exhibit a monotone n-variable function in NP whose monotone circuits require depth $\Omega(n/\log n)$; previously, a bound of $\Omega(\sqrt{n})$ was known [RW92]. Moreover, we prove a $\Theta(\sqrt{n})$ monotone depth bound for a function in monotone P.
- *Proof complexity:* We prove new rank lower bounds as well as obtain the first length–space lower bounds for semi-algebraic proof systems, including Lovász–Schrijver and Lasserre (SOS) systems. In particular, these results extend and simplify the works of Beame et al. [BPS07] and Huynh and Nordström.

This chapter is based on the following publication:

[GP14]: Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the 46th Symposium on Theory of Computing (STOC)*, pages 847–856. ACM, 2014. doi:10.1145/2591796.2591838

7.1 Introduction

Apart from their intrinsic interest, communication lower bounds for *search problems* find applications in two major areas of complexity theory.

1. **Circuit complexity:** A famous theorem of Karchmer and Wigderson [KW88] states that for all boolean functions f, the minimum depth of a circuit computing f is equal to the communication complexity of a certain search problem, called the *Karchmer–Wigderson (KW) game* for f. While it still remains a major open problem to prove general depth lower bounds for explicit boolean functions, KW-games have permitted progress in *monotone* circuit complexity: there are monotone depth lower bounds for graph connectivity [KW88], clique functions [GH92, RW92], perfect matchings [RW92], and functions in monotone P [RM99]. See also Chapter 7 in Jukna's book [Juk12].
2. **Proof complexity:** Impagliazzo et al. [IPU94] (see also [Juk12, S19.3]) introduced an analogue of KW-games to proof complexity. They showed how small tree-like Cutting Planes refutations of an unsatisfiable CNF formula F can be converted into efficient *two-party* communication protocols for a certain canonical search problem associated with F. More recently, Beame et al. [BPS07] extended this connection by showing that suitable lower bounds for *multi-party* protocols imply degree/rank lower bounds for many well-studied semi-algebraic proof systems, including Lovász–Schrijver [LS91], Positivstellensatz [Gri01], Sherali–Adams [SA90], and Lasserre (SOS) [Las01] systems. In parallel to these developments, Huynh and Nordström [HN12] have also found a new kind of simulation of space-bounded proofs by communication protocols. They used this connection to prove length–space lower bounds in proof complexity.

In this chapter we obtain new randomised lower bounds for search problems in both two-party and multi-party settings. Our proofs are relatively simple reductions from the *set-disjointness* function, the canonical NP-complete problem in communication complexity. These results allow us to derive, almost for free, new lower bounds in the above two application domains.

1. **Monotone depth:** We introduce a certain monotone encoding of the *CSP satisfiability* problem and prove an $\Omega(n/\log n)$ monotone depth lower bound for it, where n is the number of input variables. Previously, the best bound for an explicit monotone function (perfect matchings) was $\Omega(\sqrt{n})$ due to Raz and Wigderson [RW92]. Moreover, we prove a $\Theta(\sqrt{n})$ monotone depth bound for a function in monotone P.
2. **Rank, length, and space:** We obtain new rank lower bounds for a family of semantic polynomial threshold proof systems called $\mathsf{T}^{\mathsf{cc}}(k)$, which includes many of the semi-algebraic proof systems mentioned above. This extends and simplifies the work of Beame et al [BPS07]. We also extend the length–space lower bound of Huynh and Nordström [HN12] to hold for $\mathsf{T}^{\mathsf{cc}}(k)$ systems of degree up to $k = (\log n)^{1-o(1)}$. In particular, this yields the first nontrivial length–space lower bounds for dynamic SOS proofs of this degree.

We state these results more precisely shortly, once we first formalise our basic communication complexity setup.

7.1.1 Starting point: Critical block sensitivity

We build on the techniques recently introduced by Huynh and Nordström [HN12]. They defined a new complexity measure for search problems called *critical block sensitivity*, which is a generalisation of the usual notion of block sensitivity for functions (see [BdW02] for a survey). They used this measure to give a general method of proving lower bounds for *composed* search problems in the two-party communication model. These notions will be so central to us that we proceed to define them immediately.

A *search problem* on n variables is a relation $S \subseteq \{0,1\}^n \times Q$ where Q is some set of possible solutions. On input $\alpha \in \{0,1\}^n$ the search problem is to find a solution $q \in Q$ that is *feasible for* α, that is, $(\alpha, q) \in S$. We assume that S is such that all inputs have at least one feasible solution. An input is called *critical* if it has a unique feasible solution.

Definition 7.1 (Critical block sensitivity [HN12])**.** Fix a search problem $S \subseteq \{0,1\}^n \times Q$. Let $f \subseteq S$ denote a total function that solves S, i.e., for each input $\alpha \in \{0,1\}^n$ the function picks out some feasible solution $f(\alpha)$ for α. We denote by $\mathrm{bs}(f,\alpha)$ the usual block sensitivity of f at α. That is, $\mathrm{bs}(f,\alpha)$ is the maximal number bs such that there are disjoint blocks of coordinates $B_1, \ldots, B_{\mathrm{bs}} \subseteq [n]$ satisfying $f(\alpha) \neq f(\alpha^{B_i})$ for all i; here, α^{B_i} is the same as α except the input bits in coordinates B_i are flipped. The *critical block sensitivity* of S is defined as

$$\mathrm{cbs}(S) \;:=\; \min_{f \subseteq S} \max_{\text{critical } \alpha} \mathrm{bs}(f,\alpha).$$

We note immediately that $\mathrm{cbs}(S)$ is a lower bound on the deterministic decision tree complexity of S. Indeed, a deterministic decision tree defines a total function $f \subseteq S$ and on each critical input α the tree must query at least one variable from each sensitive block of f at α (see [BdW02, Theorem 9]). It turns out that $\mathrm{cbs}(S)$ is also a lower bound on the *randomised* decision tree complexity (see Theorem 7.2 below).

7.1.2 Composed search problems

As is the theme of this thesis, we study composed variants $S \circ g^n$ of the query complexity problem; see Figure 7.1. In a composed problem, each of the n input bits of S are encoded using a small two-party gadget $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. As input to $S \circ g^n$ Alice gets an $x \in \mathcal{X}^n$ and Bob gets a $y \in \mathcal{Y}^n$. We think of the pair (x,y) as encoding the input

$$\alpha = g^n(x,y) = (\, g(x_1,y_1), \ldots, g(x_n,y_n)\,)$$

of the original problem S. The objective is to find a $q \in Q$ such that $(g^n(x,y), q) \in S$.

7.1.3 Our communication complexity results

We start by giving a simple new proof of the following central result of Huynh and Nordström [HN12]. (Strictly speaking, the statement of the original theorem [HN12] is slightly weaker

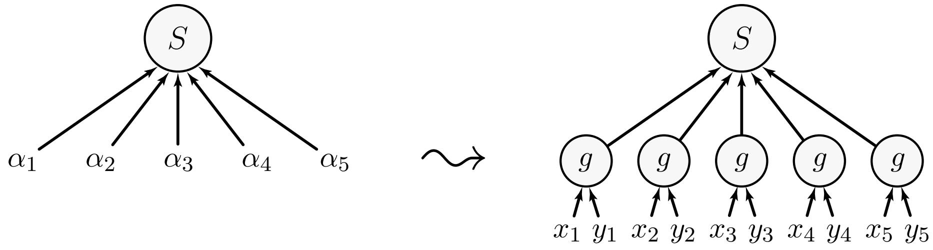


Figure 7.1: Composing a search problem S with a two-party gadget g.

in that it involves an additional "consistency" assumption, which we do not need.)

Theorem 7.2 (Two-party version)**.** *There is a two-party gadget* $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ *such that if* $S \subseteq \{0,1\}^n \times Q$ *is any search problem, then* $S \circ g^n$ *has randomised bounded-error communication complexity* $\Omega(\text{cbs}(S))$.

Huynh and Nordström proved Theorem 7.2 for the gadget $g = \mathsf{3IND}$, where $\mathsf{3IND}\colon [3] \times \{0,1\}^3 \to \{0,1\}$ is the indexing function that maps $(x, y) \mapsto y_x$. Their proof used the information complexity approach [CSWY01, BJKS04] and is quite intricate. By contrast, we prove Theorem 7.2 by a direct randomised reduction from the *set-disjointness* function

$$\mathsf{DISJ}_n(x,y) = (\mathsf{OR}_n \circ \mathsf{AND}^n)(x,y) = \textstyle\bigvee_{i\in[n]}(x_i \wedge y_i).$$

In the language of Babai et al. [BFS86] (see also [CP10]) the set-disjointness function is NP-complete in communication complexity: it is easy to certify that $\mathsf{DISJ}_n(x,y) = 1$, and conversely, every two-party function with low nondeterministic complexity reduces efficiently to DISJ_n. Our proof of Theorem 7.2 is inspired by a result of Zhang [Zha09] that essentially establishes Theorem 7.2 in case S is a function and cbs(S) is simply the standard block sensitivity. The key insight in our proof is to choose g to be *random-self-reducible* (see Section 7.2 for definitions). Random-self-reducibility is a notion often studied in cryptography and classical complexity theory, but less often in communication complexity. Most notably, random-self-reducibility was used implicitly in [RW92]. The definitions we adopt are similar to those introduced by Feige et al. [FKN94] in a cryptographic context.

Our proof has also the advantage of generalising naturally to the multi-party setting. This time we start with the k-party unique-disjointness function $\mathsf{UDISJ}_{k,n}$ and the proof involves the construction of k-party random-self-reducible functions g_k.

Theorem 7.3 (Multi-party version)**.** *There are k-party gadgets* $g_k\colon \mathcal{X}^k \to \{0,1\}$ *with domain size* $\log|\mathcal{X}| = k^{o(1)}$ *bits per player, such that if* $S \subseteq \{0,1\}^n \times Q$ *is any search problem, then* $S \circ g_k^n$ *has randomised bounded-error communication complexity at least that of* $\mathsf{UDISJ}_{k,\text{cbs}(S)}$ *(up to constants).*

Theorem 7.3 can be applied to the following multi-player communication models.

- **Number-in-hand:** The i-th player only sees the i-th part of the input. Here, set-disjointness has been studied under broadcast communication (e.g., [Gro09]) and under private channel communication [BEO$^+$13].
- **Number-on-forehead (NOF):** The i-th player sees all parts of the input except the i-th part [CFL83]. The current best randomised lower bound for $\mathsf{UDISJ}_{k,n}$ is $\Omega(\sqrt{n}/2^k k)$ by Sherstov [She13b]. We rely heavily on Sherstov's result in our proof complexity applications.

In the rest of this introduction we discuss the applications—the impatient reader who wants to see the proofs of Theorems 7.2 and 7.3 can immediately skip to Sections 7.2 and 7.3.

7.1.4 CSPs and their canonical search problems

To get the most out of Theorems 7.2 and 7.3 for the purposes of applications, we need to find search problems with high critical block sensitivity but low certificate complexity. Low-degree constraint satisfaction problems (CSPs) capture exactly the latter goal [LNNW95].

Definition 7.4 (d-CSPs)**.** A CSP F consists of a set of (boolean) variables $\mathrm{vars}(F)$ and a set of constraints $\mathrm{cons}(F)$. Each constraint $C \in \mathrm{cons}(F)$ is a function that maps a truth assignment $\alpha\colon \mathrm{vars}(F) \to \{0,1\}$ to either 0 or 1. If $C(\alpha) = 1$, we say that C is *satisfied* by α, otherwise C is *violated* by α. Let $\mathrm{vars}(C)$ denote the smallest subset of $\mathrm{vars}(F)$ such that C depends only on the truth values of the variables in $\mathrm{vars}(C)$. We say that F is of *degree* d, or F is a d-CSP, if $|\mathrm{vars}(C)| \le d$ for all C. Note that d-CNF formulas are a special case of d-CSPs, and conversely, each d-CSP can be written as an equivalent d-CNF with a factor 2^d blow-up in the number of constraints.

An *unsatisfiable* CSP F has no assignment that satisfies all the constraints. Each such F comes with an associated *canonical search problem* $S(F)$.

Definition 7.5 (Canonical search problems)**.** Let F be an unsatisfiable CSP. In the search problem $S(F)$ we are given an assignment $\alpha\colon \mathrm{vars}(F) \to \{0,1\}$ and the goal is to find a constraint $C \in \mathrm{cons}(F)$ that is violated by α.

We give new critical block sensitivity lower bounds for the canonical search problems associated with *Tseitin* and *Pebbling* formulas.

7.1.5 Sensitivity of Tseitin formulas

Tseitin formulas are well-studied examples of unsatisfiable CSPs that are hard to refute in many proof systems; for an overview, see Jukna [Juk12, §18.7].

Definition 7.6 (Tseitin formulas)**.** Let $G = (V, E, \ell)$ be a connected labelled graph of maximum degree d where the labelling $\ell\colon V \to \{0,1\}$ has odd Hamming weight. The *Tseitin formula* Tse_G

associated with G is the d-CSP that has the edges $e \in E$ as variables and for each node $v \in V$ there is a constraint C_v defined by

$$C_v(\alpha) = 1 \quad \iff \quad \sum_{e: v \in e} \alpha(e) \equiv \ell(v) \pmod 2.$$

It follows from a simple parity argument that Tse_G is unsatisfiable (see, e.g., Section 7.4.1).

Call G *κ-routable* if there is a set $T \subseteq V$ of size $|T| \geq 2\kappa$ such that for any set of κ disjoint pairs of nodes of T there are κ edge-disjoint paths in G that connect all the pairs. (Note: κ-routability is usually defined only for $T = V$, but we relax this condition.) The proof of the following theorem appears in Section 7.4.

Theorem 7.7 (Tseitin sensitivity)**.** *If G is κ-routable, then* $\mathrm{cbs}(S(\mathit{Tse}_G)) = \Omega(\kappa)$.

Theorem 7.7 can be applied to the following classes of bounded-degree graphs.

- **Grid graphs:** If G is a $\sqrt{n} \times \sqrt{n}$ grid graph, then we can take $\kappa = \Omega(\sqrt{n})$ by letting $T \subseteq V$ be any row (or column) of nodes. This is tight: the deterministic decision tree that solves $S(\mathit{Tse}_G)$ using binary search makes $O(\sqrt{n})$ queries.
- **Expanders:** If G is a sufficiently strong expander (e.g., a Ramanujan graph [LPS88]), then we can take $\kappa = \Omega(n/\log n)$ as shown by Frieze et al. [FZ00, Fri01].
- **Connectors:** A *κ-connector* is a bounded-degree graph with κ inputs $I \subseteq V$ and κ outputs $O \subseteq V$ such that for any one-to-one correspondence $\pi\colon I \to O$ there exist κ edge-disjoint paths that connect $i \in I$ to $\pi(i) \in O$. If we merge I and O in a 2κ-connector in some one-to-one manner and let $T = I = O$, we get a κ-routable graph. Conversely, if G is κ-routable, we can partition the set T as $I \cup O$ and get a κ-connector.

  It is known that simple κ-connectors with $\kappa = \Theta(n/\log n)$ exist and this bound is the best possible [Pip90]. Thus, the best lower bound provable using Theorem 7.7 is $\Theta(n/\log n)$.

It is well known that the *deterministic* decision tree complexity of $S(\mathit{Tse}_G)$ is $\Omega(n)$ when G is an expander [Urq87]. However, *randomised* lower bounds—which Theorem 7.7 provides—are more scarce. We are only aware of a single previous result in the direction of Theorem 7.7, namely, Lovász et al. [LNNW95, §3.2.1] announce a lower bound of $\Omega(n^{1/3})$ for the randomised decision tree complexity of $S(\mathit{Tse}_G)$ when G is an expander. Our Theorem 7.7 subsumes this.

7.1.6 Sensitivity of pebbling formulas

Pebble games have been studied extensively as means to understand time and space in computations; for an overview, see the survey by Nordström [Nor13]. In this chapter we restrict our attention to the simple *(black) pebble game* that is played on a directed acyclic graph G with a unique sink node t (i.e., having outdegree 0). In this game the goal is to place a pebble on the sink t using a sequence of *pebbling moves*. The allowed moves are:

(1) A pebble can be placed on a node if its in-neighbours have pebbles on them. In particular, we can always pebble a source node (i.e., having indegree 0).

(2) A pebble can be removed from any pebbled node (and reused later in the game).

The *(black) pebbling number* of G is the minimum number of pebbles that are needed to pebble the sink node in the pebble game on G.

The pebble game on G comes with an associated *pebbling formula*.

Definition 7.8 (Pebbling formulas. See [BSW01] and [Nor13, §2.3])**.** Let $G = (V, E, t)$ be a directed acyclic graph of maximum indegree d where t is a unique sink. The *pebbling formula* Peb_G associated with G is the $(d+1)$-CSP that has the nodes $v \in V$ as variables and the following constraints:

(1) The variable corresponding to the sink t is false.

(2) For all nodes v with in-neighbours $w_1, \dots, w_d$, we require that if all of $w_1, \dots, w_d$ are true, then v is true. In particular, each source node must be true.

It is not hard to see that Peb_G is unsatisfiable.

Classical complexity measures for $S(Peb_G)$ include the pebbling number of G (a measure of *space*) and the deterministic decision tree complexity (a measure of *parallel time*), which admits many equivalent characterisations [Cha13]. However, these complexity measures are fundamentally *deterministic* and do not seem to immediately translate into *randomised* lower bounds, which are needed in our applications. For this reason, Huyhn and Nordström [HN12] devised an elegant ad hoc proof method for their result that, for a *pyramid graph* G (see Figure 7.6), $\mathrm{cbs}(S(Peb_G)) = \Omega(n^{1/4})$. Annoyingly, this falls a little short of both the pebbling number $\Theta(\sqrt{n})$ of G and the decision tree complexity $\Theta(\sqrt{n})$ of $S(Peb_G)$. Here we close this gap by generalising their proof method: we get tight bounds for a different (but related) graph G. The proof appears in Section 7.4.

Theorem 7.9 (Pebbling sensitivity)**.** *There are n-node bounded-degree graphs G such that*

- *G has pebbling number $\Theta(\sqrt{n})$.*
- *$S(Peb_G)$ has deterministic decision tree complexity $\Theta(\sqrt{n})$.*
- *$S(Peb_G)$ has critical block sensitivity $\Theta(\sqrt{n})$.*

7.1.7 Applications: Monotone depth

Monotone depth from Tseitin. Let G be an $\Omega(n/\log n)$-routable graph of bounded degree $d = O(1)$. By Theorem 7.7 the lifted problem $S(Tse_G) \circ g^{O(n)}$ has two-party communication complexity $\Omega(n/\log n)$. By contrast, its nondeterministic communication complexity is just $\log n + O(1)$, since the players can guess a node $v \in V(G)$ and verify that it indeed induces a parity violation (which involves exchanging the inputs to $d = O(1)$ many copies of g associated to edges incident to v). It is known that *any* two-party search problem with nondeterministic communication complexity C reduces to a monotone KW-game for some monotone $f\colon \{0,1\}^N \to \{0,1\}$ on $N = 2^C$ variables; see Gál [Gál01, Lemma 2.3] for an exposition. In our case we get a monotone function on $N = O(n)$ variables whose monotone KW-game complexity—i.e., its monotone depth complexity—is $\Omega(N/\log N)$. Moreover, we make this general connection a bit more explicit in Section 7.5 by showing that our function can be taken to be a monotone variant of the usual *CSP satisfiability* function.

Corollary 7.10 (Monotone depth from Tseitin)**.** *There is an monotone function in* NP *on* N *input bits whose monotone depth complexity is* $\Omega(N/\log N)$*.*

Monotone depth from pebbling. We also get perhaps the simplest proof yet of a $n^{\Omega(1)}$ monotone depth bound for a function in monotone P. Indeed, we only need to apply a transformation of Raz and McKenzie, described in [RM99, S3], which translates our $\Omega(\sqrt{n})$ communication lower bound for $S(Peb_G) \circ g^{O(n)}$ (coming from Theorems 7.2 and 7.9) to a monotone depth lower bounds for a related "generation" function $\mathsf{GEN}_{G'}$ defined relative to a "lifted" version G' of G'. Raz and McKenzie originally studied the case when G is a pyramid graph, and they lifted $S(Peb_G)$ with some poly(n)-size gadget (making the number of input bits of $\mathsf{GEN}_{G'}$ a large polynomial in n). However, their techniques work for any graph G and any gadget g. In our case of constant-size gadgets, we get only constant factor blow-up in parameters; we refer to [RM99, S3] for the details of deriving the following.

Corollary 7.11 (Monotone depth from pebbling)**.** *There is an explicit function* f *on* N *input bits such that* f *admits polynomial size monotone circuits of depth* $\Theta(\sqrt{N})$ *and any monotone circuit for* f *requires depth* $\Theta(\sqrt{N})$*.* □

The original bounds of [RM99] went up to $\Omega(N^{\delta})$ for a small constant δ. This was recently improved by the works [CP12, FPRC13] that prove (among other things) monotone depth bounds of up to $\Omega(N^{1/6-o(1)})$ for GEN_G type functions. Our Corollary 7.11 achieves quantitatively the largest bound (to-date) for a function in monotone P.

7.1.8 Applications: Proof complexity

Over the last decade or so there have been a large number of results proving lower bounds on the rank required to refute (or approximately optimise over) systems of constraints in a wide variety of semi-algebraic (a.k.a. polynomial threshold) proof systems, including Lovász–Schrijver [LS91], Cutting Planes [Gom58, Chv73], Positivstellensatz [Gri01], Sherali–Adams [SA90], and Lasserre [Las01] proofs. Highlights of this work include recent linear rank lower bounds for many constraint optimisation problems [Sch08, Tul09, CMM09, STT07, GMPT10]. Nearly all of these results rely on delicate constructions of local distributions that are specific to both the problem and to the proof system.

A communication complexity approach for proving lower bounds for semi-algebraic proofs was developed by Beame et al. [BPS07]. They studied a semantic proof system called $\mathsf{T}^{\mathsf{cc}}(k)$

whose proofs consist of lines that are computed by a low-cost (i.e., polylog communication) k-party NOF protocols (see Section 7.6 for definitions). They prove that if a CNF formula F has a small tree-like $\mathsf{T}^{\mathsf{cc}}(k)$ refutation, then $S(F)$ has an efficient k-party NOF protocol. Thus, lower bounds for the tree-size of $\mathsf{T}^{\mathsf{cc}}(k)$ proofs follow from NOF lower bounds for $S(F)$.

Rank lower bounds. Using this relationship we can now prove the following result[1] for $\mathsf{T}^{\mathsf{cc}}(k)$ proof systems, where k can be almost logarithmic in the size of the formula. We state the theorem only for rank, with the understanding that a bound of $\Omega(R)$ on rank also implies a bound of $\exp(\Omega(R))$ on tree-size. The proof appears in Section 7.6.

Theorem 7.12 (Rank lower bounds)**.** *There are explicit CNF formulas F of size s and width $O(\log s)$ such that all $\mathsf{T}^{\mathsf{cc}}(k)$ refutations of F require rank at least*

$$R_k(s) = \begin{cases} s^{1-o(1)}, & \text{for } k = 2, \\ s^{1/2-o(1)}, & \text{for } 3 \le k \le (\log s)^{1-o(1)}. \end{cases}$$

Theorem 7.12 simplifies the proof of a similar theorem from [BPS07], which held only for a specific family of formulas obtained from non-constant degree graphs, and only for $k < \log\log s$.

We note already here that the quadratic gap between $R_2(s)$ and $R_3(s)$ will be an artefact of us switching from two-party communication to multi-party communication. More specifically, while the two-party communication complexity of set-disjointness DISJ_n is $\Omega(n)$, the corresponding lower bound for three parties is only $\Omega(\sqrt{n})$ [She13b]. Whether the multi-party bound can be improved to $\Omega(n)$ is an open problem.

Length–space lower bounds. Continuing in similar spirit, [HN12] showed how to prove length–space lower bounds for $\mathsf{T}^{\mathsf{cc}}(2)$ systems from lower bounds on the communication complexity of $S(F)$. Using this relationship together with our new multi-party lower bounds, we can extend this result to $\mathsf{T}^{\mathsf{cc}}(k)$ systems of degree $k > 2$.

Theorem 7.13 (Length–space lower bounds)**.** *There are CNF formulas F of size s such that*

- *F admits a Resolution refutation of length $L = s^{1+o(1)}$ and space $Sp = s^{1/2+o(1)}$.*
- *Any length L and space Sp refutation of F in $\mathsf{T}^{\mathsf{cc}}(k)$ must satisfy*

$$Sp \cdot \log L \;\geq\; \begin{cases} s^{1/2-o(1)}, & \text{for } k = 2, \\ s^{1/4-o(1)}, & \text{for } 3 \le k \le (\log s)^{1-o(1)}. \end{cases} \tag{7.1}$$

We hesitate to call Theorem 7.13 a *tradeoff* result since our only upper bound is a refutation requiring space $Sp = s^{1/2+o(1)}$ and we do not know how to decrease this space usage by trading it for length; this is the same situation as in [HN12]. Surprisingly, in a subsequent work, Galesi

[1] Similar claims were made in [BHP10]. Unfortunately, as pointed out by [HN12], Lemma 3.5 in [BHP10] is incorrect and this renders many of the theorems in the paper incorrect.

et al. [GPT15] have shown that *any* unsatisfiable CNF formula admits an exponentially long Cutting Planes refutation in *constant* space, which gives a second data point in the length–space parameter space for which an upper bound exists. We also mention that while the CNF formulas F in Theorem 7.13 are lifted versions of pebbling formulas, we could have formulated similar length–space lower bounds for lifted Tseitin formulas (where, e.g., $Sp \cdot \log L \geq s^{1-o(1)}$ for $k = 2$). But for Tseitin formulas we do not have close-to-matching upper bounds.

In any case, Theorem 7.13 gives, in particular, the first length–space lower bounds for dynamic SOS proofs of degree k. In addition, even in the special case of $k = 2$, Theorem 7.13 simplifies and improves on [HN12]. However, for Polynomial Calculus Resolution (a $\mathsf{T}^{\mathsf{cc}}(2)$ system), the best known length–space tradeoff results are currently proved in the recent work of Beck et al. [BNT13]. For Resolution (maybe the simplest $\mathsf{T}^{\mathsf{cc}}(2)$ system), even stronger tradeoff results have been known since [BSN11]; see also Beame et al. [BBI12] for nontrivial length lower bounds in the superlinear space regime. For Cutting Planes (a $\mathsf{T}^{\mathsf{cc}}(2)$ system) Theorem 7.13 remains the state-of-the-art to our knowledge.

7.1.9 Models of communication complexity

We work in the standard models of two-party and multi-party communication complexity; see [KN97, Juk12] for definitions. Here we only recall some conventions about randomised protocols. A protocol Π solves a search problem S with *error* ϵ iff on any input x the probability that $(x, \Pi(x)) \in S$ is at least $1 - \epsilon$ over the random coins of the protocol. Note that $\Pi(x)$ need not be the same feasible solution; it can depend on the outcomes of the random coins. The protocol is of *bounded-error* if $\epsilon \leq 1/4$. The constant $1/4$ here can often be replaced with any other constant less than $1/2$ without affecting the definitions too much. In the case of computing boolean functions this follows from standard boosting techniques [KN97, Exercise 3.4]. While these boosting techniques may fail for general search problems, we do not encounter any such problems in this chapter.

7.2 Versatile gadgets

In this section we introduce *versatile* two-party and multi-party functions. Our proofs of Theorems 7.2 and 7.3 will work whenever we choose g or g_k to be a versatile gadget. We start by introducing the terminology in the two-party case; the multi-party case will be analogous.

7.2.1 Self-reductions and versatility

The simplest reductions between communication problems are those that can be computed without communication. Let $f_i \colon \mathcal{X}_i \times \mathcal{Y}_i \to \{0,1\}$ for $i = 1, 2$, be two-party functions. We say that f_1 *reduces to* f_2, written $f_1 \leq f_2$, if the communication matrix of f_1 appears as a submatrix of the communication matrix of f_2. Equivalently, $f_1 \leq f_2$ iff there exist one-to-one mappings π_A

and π_B such that

$$f_1(x,y) = f_2(\pi_A(x), \pi_B(y)) \qquad \text{for all} \quad (x,y) \in \mathcal{X}_1 \times \mathcal{Y}_1.$$

Our restriction to one-to-one reductions above is merely a technical convenience (cf. Babai et al. [BFS86] allow reductions to be many-to-one).

Example 1. Let $\mathsf{3EQ}\colon [3] \times [3] \to \{0,1\}$ be the equality function with inputs from $[3]$. Then AND reduces to $\mathsf{3EQ}$ since $\mathsf{AND}(x,y) = \mathsf{3EQ}(1+x, 3-y)$.

We will be interested in special kinds of reductions that reduce a function to *itself*. Our first flavour of self-reducibility relates a function f and its negation $\neg f$:

Flippability. A function f is called *flippable* if $\neg f \leq f$. Note that since the associated reduction maps z-inputs to $(1-z)$-inputs in a one-to-one fashion, a flippable function must be *balanced*: exactly half of the inputs satisfy $f(x,y) = 1$.

Example 2. The XOR function is flippable via $\neg\mathsf{XOR}(x,y) = \mathsf{XOR}(1-x, y)$. By contrast, AND and $\mathsf{3EQ}$ are not balanced and hence not flippable.

We will also consider randomised reductions where the two parties are allowed to *synchronise* their computations using public randomness. More precisely, even though the two parties are still not communicating, we can let the mappings π_A and π_B depend on a public random string $\boldsymbol{r} \in \{0,1\}^*$, whose distribution the two parties can freely choose. This way, a random reduction computes $(x,y) \mapsto (\pi_A(x,\boldsymbol{r}), \pi_B(y,\boldsymbol{r}))$. The following definition is similar to the *perfectly secure* functions of Feige et al. [FKN94].

Random self-reducibility. A function f is called *random-self-reducible* if there are mappings π_A and π_B together with a random variable $\boldsymbol{r}$ such that for every z-input $(x,y) \in f^{-1}(z)$ the random pair $(\pi_A(x,\boldsymbol{r}), \pi_B(y,\boldsymbol{r}))$ is uniformly distributed among all the z-inputs of f.

Example 3. The equality function $\mathsf{EQ}\colon [n] \times [n] \to \{0,1\}$ is random-self-reducible: we can use the public randomness to sample a permutation $\boldsymbol{\pi}\colon [n] \to [n]$ uniformly at random and let the two parties compute $(x,y) \mapsto (\boldsymbol{\pi}(x), \boldsymbol{\pi}(y))$. (In fact, to further save on the number of random bits used, it would suffice to choose $\boldsymbol{\pi}$ from any group that acts 2-transitively on $[n]$.)

A notable example of a function that is *not* random-self-reducible is AND; it has only one 1-input, which forces any self-reduction to be the identity map. This is particularly inconvenient since AND is featured in the set-disjointness function $\mathsf{DISJ}_n = \mathsf{OR}_n \circ \mathsf{AND}^n$, which will be the starting point for our reductions. To compensate for the shortcomings of AND we work with a slightly larger function $g \geq \mathsf{AND}$ instead.

Definition 7.14 (Versatility)**.** A two-party function g is called *versatile* if (1) $g \geq \mathsf{AND}$, (2) g is flippable, and (3) g is random-self-reducible.

	0	**1**	**2**	**3**
0	0	0	1	1
1	0	1	1	0
2	1	1	0	0
3	1	0	0	1

Figure 7.2: Function VER.

1	0	0	0	1	1
0	1	0	1	0	1
0	0	1	1	1	0

Figure 7.3: Function HN.

7.2.2 Two-party example

Consider the function VER: $\mathbb{Z}_4 \times \mathbb{Z}_4 \to \{0,1\}$ defined by

$$\mathsf{VER}(x,y) = 1 \quad \Longleftrightarrow \quad x+y \in \{2,3\}, \qquad \text{for all } x,y \in \mathbb{Z}_4, \tag{7.2}$$

where the arithmetic is that of $\mathbb{Z}_4$; see Figure 7.2.

Lemma 7.15. VER *is versatile.*

Proof. The reduction from AND is simply given by $\mathsf{AND}(x,y) = \mathsf{VER}(x,y)$. Moreover, VER is flippable because $\neg\mathsf{VER}(x,y) = \mathsf{VER}(x+2,y)$. To see that VER is random-self-reducible, start with (x,y) and compute as follows. First, choose $(\boldsymbol{x},\boldsymbol{y})$ uniformly at random from the set $\{(x,y),(1-x,-y)\}$ so that $\boldsymbol{x}+\boldsymbol{y}$ is uniformly distributed either in the set $\{0,1\}$ if (x,y) was a 0-input, or in the set $\{2,3\}$ if (x,y) was a 1-input. Finally, choose a random $\boldsymbol{a} \in \mathbb{Z}_4$ and output $(\boldsymbol{x}+\boldsymbol{a}, \boldsymbol{y}-\boldsymbol{a})$. □

It is not hard to show that VER is in fact a minimum-size example of a versatile function: if $g\colon [a] \times [b] \to \{0,1\}$ is versatile then $a, b \geq 4$. Indeed, VER is the smallest two-party function for which our proof of Theorem 7.2 applies. By comparison, the original proof of Theorem 7.2 [HN12] uses a certain subfunction HN $\leq$ 3IND whose communication matrix is illustrated in Figure 7.3. Thus, somewhat interestingly, our proof yields a result that is incomparable to [HN12] since we have neither VER $\leq$ HN nor HN $\leq$ VER.

Coincidentally, VER makes an appearance in Sherstov's pattern matrix method [She11a, S12], too. There, the focus is on exploiting the *matrix-analytic* properties of the communication matrix of VER. By contrast, in this chapter, we celebrate its *self-reducibility* properties.

7.2.3 Multi-party examples

In the multi-party setting we restrict our attention to k-party reductions $f_1 \leq f_2$ for k-party functions $f_i\colon \mathcal{X}_i^k \to \{0,1\}$ that are determined by one-to-one mappings $\pi_1, \ldots, \pi_k$ satisfying

$$f_1(x_1,\ldots,x_k) = f_2(\pi_1(x_1),\ldots,\pi_k(x_k)) \qquad \text{for all} \quad (x_1,\ldots,x_k) \in \mathcal{X}_1^k.$$

This way any player that sees an input x_i can evaluate $\pi_i(x_i)$ without communication. As before, a randomised reduction can also depend on public coins.

Versatile k-party functions $g_k\colon \mathcal{X}^k \to \{0,1\}$ are defined analogously to the two-party case: we require that the k-party k-bit AND_k function reduces to g_k, and that g_k is both flippable and random-self-reducible—all under k-party reductions.

It is known that every k-party function is a subfunction of some, perhaps exponentially large random-self-reducible function [FKN94]. However, in the following, we are interested in finding examples of *small* versatile k-party functions in order to optimise our constructions. We proceed to give two examples of well-studied k-party functions and prove them versatile.

First example: Quadratic character. Denote by $\chi\colon \mathbb{Z}_p^\times \to \{0,1\}$ the indicator function for quadratic residuosity modulo p, i.e., $\chi(x) = 1$ iff x is a square in $\mathbb{Z}_p$. The pseudo-random qualities of χ have often made it an object of study in communication complexity [BNS92, BGKL03, ACFN12]. Moreover, the self-reducibility properties of χ are famously useful in cryptography, starting with [GM84].

For our purposes we let p to be an $O(k)$-bit prime. Following [BNS92, S2.5] the k-party quadratic character function $\mathsf{QCS}_k\colon \mathbb{Z}_p^k \to \{0,1\}$ is defined as

$$\mathsf{QCS}_k(x_1,\ldots,x_k) := \chi\big(\textstyle\sum_i x_i\big). \tag{7.3}$$

We leave $\mathsf{QCS}_k(x_1,\ldots,x_k)$ undefined for inputs with $\sum_i x_i = 0$, i.e., we consider QCS_k to be a promise problem. Our three items of versatility fall out of the well-known properties of χ.

Lemma 7.16. QCS_k *is versatile.*

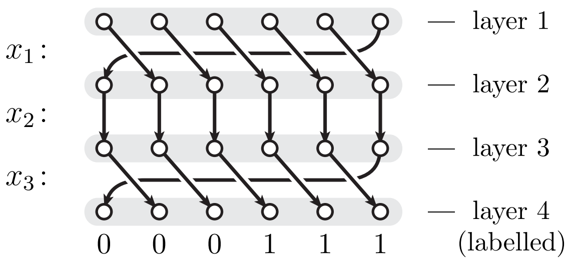
Proof. Reduction from AND_k*:* We need the following nonelementary fact (see, e.g., Lemma 6.13 in [BGKL03] or the recent work [Wri13]): if p is a large enough $O(k)$-bit prime then there are $k+1$ consecutive integers $\{a, a+1, \ldots, a+k\} \subseteq \mathbb{Z}_k^\times$ realising the pattern

$$\chi(a) = \chi(a+1) = \cdots = \chi(a+k-1) = 0 \qquad \text{and} \qquad \chi(a+k) = 1.$$

This immediately facilitates the reduction: an input $(y_1,\ldots,y_k)$ of AND_k is mapped to an input $(a+y_1, y_2, \ldots, y_k)$ of QCS_k. *Flippability:* Map $x_i \mapsto s \cdot x_i$ for all i, where $s \neq 0$ is a fixed quadratic nonresidue. *Random-self-reducibility:* Choose a random quadratic residue $\boldsymbol{r} \in \mathbb{Z}_p$ and numbers $\boldsymbol{a}_1,\ldots,\boldsymbol{a}_k \in \mathbb{Z}_p$ satisfying $\boldsymbol{a}_1 + \cdots + \boldsymbol{a}_k = 0$. The random self-reduction maps $x_i \mapsto \boldsymbol{r} \cdot x_i + \boldsymbol{a}_i$ for all i. □

Second example: Pointer jumping. Next, we observe that certain variants of the k-party pointer jumping function are versatile. To explain this idea, we begin by describing a simple construction where each of the k inputs requires $\Theta(k \log k)$ bits to represent. After this we improve on the construction by using known results on branching programs; we note that similar ideas have been used in the context of secure multi-party computations [CFIK03].

Figure 7.4: Example of $\mathsf{AND}_3 \leq \mathsf{Jump}_3$. The input (x_1, x_2, x_3) of Jump_3 is the result of applying the reduction to the input $(1, 0, 1)$ of AND_3.



Define the k-party *pointer jumping* function $\mathsf{Jump}_k \colon \mathcal{X}^k \to \{0,1\}$ as follows. The inputs are permutations $x_i \colon [2k] \to [2k]$, $i \in [k]$, and the function value is given by

$$\mathsf{Jump}_k(x_1, \ldots, x_k) = 0 \qquad \Longleftrightarrow \qquad (x_k \circ x_{k-1} \circ \cdots \circ x_1)(1) \in [k]. \tag{7.4}$$

A useful way to view the input $(x_1, \ldots, x_k)$ is as a layered digraph: there are $k+1$ layers, each containing $2k$ nodes; the input x_i defines a perfect matching between layers i and $i+1$; and the nodes on the last layer are labelled in a *balanced* way with k zeroes and k ones. The value of the function is the label of the sink that is reachable from the 1st node of the 1st layer.

Lemma 7.17. Jump_k *is versatile.*

Proof. Reduction from AND_k*:* Given an input $(y_1, \ldots, y_k)$ of AND_k we reduce it to an input $(x_1, \ldots, x_k)$ of Jump_k as follows (see Figure 7.4). If $y_i = 0$ then x_i is defined to be the identity permutation on $[2k]$, otherwise x_i is the cyclic permutation that maps $j \mapsto j+1$ for $j \in [2k-1]$ and $2k \mapsto 1$. *Flippability:* Replace the input x_k with $\pi \circ x_k$, where $\pi \colon [2k] \to [2k]$ is some fixed permutation that swaps the sets $[k]$ and $[k+1, 2k]$, i.e., $\pi([k]) = [k+1, 2k]$. *Random-self-reducibility:* The random self-reduction is best visualised as acting on the layered graph associated with an input $(x_1, \ldots, x_k)$. First, sample $k+1$ permutations $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_{k+1} \colon [2k] \to [2k]$ uniformly and independently at random under the restrictions that $\boldsymbol{\pi}_1$ fixes the element 1 and $\boldsymbol{\pi}_{k+1}$ fixes the set $[k]$. Then use $\boldsymbol{\pi}_i$ to relabel the nodes on the i-th layer. Formally this means that the input x_i is mapped to $\boldsymbol{\pi}_{i+1} \circ x_i \circ \boldsymbol{\pi}_i^{-1}$. □

The reduction $\mathsf{AND}_k \leq \mathsf{Jump}_k$ above was implicitly using a simple read-once permutation branching program for AND_k; see Figure 7.4. We will now optimise this construction by using more efficient branching programs.

Definition 7.18 (PBPs)**.** A *permutation branching program* (PBP) of width w and length ℓ is defined by a sequence of instructions (i_l, π_l, τ_l), $l \in [\ell]$, where $\pi_l, \tau_l \colon [w] \to [w]$ are permutations and each $i_l \in [n]$ indexes one of the n input variables $x_1, \ldots, x_n$. Let an input $x \in \{0,1\}^n$ be given. We say that an instruction (i, π, τ) *evaluates to* π if $x_i = 0$; otherwise the instruction *evaluates to* τ. The PBP *evaluates* to the composition of the permutations evaluated at the instructions. Finally, if $\gamma \colon [w] \to [w]$ is a permutation, we say that the PBP γ*-computes* a function $f \colon \{0,1\}^n \to \{0,1\}$ if it evaluates to the identity permutation $e \colon [w] \to [w]$ on each 0-input in $f^{-1}(0)$ and to the permutation $\gamma \neq e$ on each 1-input in $f^{-1}(1)$.

Lemma 7.19. *Suppose there exists a width-w length-ℓ PBP that γ-computes the AND_k function. Then there exists a versatile k-party function on $O(\ell w \log w)$ input bits.*

Proof. Fix a width-w PBP (i_l, π_l, τ_l), $l \in [\ell]$, that γ-computes AND_k. By modifying the PBP if necessary, we may assume that w is even and $\gamma(1) \in [w/2+1, w]$. The versatile function corresponding to the given PBP is the pointer jumping function $\mathsf{Jump}_k^\ell(x_1, \ldots, x_\ell)$ defined similarly to (7.4):

$$\mathsf{Jump}_k^\ell(x_1, \ldots, x_\ell) = 0 \qquad \Longleftrightarrow \qquad (x_\ell \circ x_{\ell-1} \circ \cdots \circ x_1)(1) \in [w/2].$$

To define the input partition, let $L_i := \{l \in [\ell] : i_l = i\}$ be the set of layers where the PBP reads the i-th input. We let the i-th player hold (on its forehead) the inputs x_l for $l \in L_i$.

Reduction from AND_k: The reduction $\mathsf{AND}_k \le \mathsf{Jump}_k^\ell$ is naturally determined by the PBP: given an input $(y_1, \ldots, y_k)$ of AND_k, we define x_l to be the permutation that the instruction (i_l, π_l, τ_l) evaluates to under $(y_1, \ldots, y_k)$. Because of our input partition, it is possible to compute x_l without communication.

Flippability and random-self-reducibility: Same as in the proof of Lemma 7.17. □

Barrington's celebrated theorem [Bar89] gives a PBP implementation of AND_k with parameters $w = 5$ and $\ell = O(k^2)$. This corresponds to having $O(k)$ input bits per player, matching the quadratic character example above. Cleve [Cle91] has improved this to a tradeoff result where for any $\epsilon > 0$ one can take $\ell = k^{1+\epsilon}$ provided that $w = w(\epsilon)$ is a large enough constant. Cleve's construction also has the property that every input variable of AND_k is read equally many times (i.e., the L_i in the above proof have the same size). Thus, letting w grow sufficiently slowly, we get a versatile k-party gadget on $O(\ell w \log w) = k^{1+o(1)}$ bits, which is $k^{o(1)}$ bits per player.

Corollary 7.20. *There are versatile k-party gadgets $g_k \colon \mathcal{X}^k \to \{0,1\}$ where $\log|\mathcal{X}| = k^{o(1)}$.* □

7.3 Communication lower bound

In this section we prove the communication lower bound for two parties (Theorem 7.2) assuming that g is a versatile gadget. The generalisation to multiple parties (Theorem 7.3) follows by the same argument—one only needs to replace g with a versatile k-party gadget g_k.

Our proof builds on a result of Zhang [Zha09] that lower bounds the two-party communication complexity of a composed function $f \circ g^n$ in terms of the block sensitivity of f. We start by outlining Zhang's approach.

7.3.1 Functions: Zhang's approach

Zhang [Zha09] proved the following theorem by a reduction from the *unique-disjointness* function UDISJ_n. Here, $\mathsf{UDISJ}_n = \mathsf{OR}_n \circ \mathsf{AND}^n$ is the usual set-disjointness function together with the promise that if $\mathsf{UDISJ}_n(a, b) = 1$, then there is a unique coordinate $i \in [n]$ such that

$a_i = b_i = 1$. The randomised communication complexity of UDISJ_n is well-known to be $\Theta(n)$ [KS92, Raz92, BJKS04]. Zhang's proof works for any gadget g with $\mathsf{AND}, \mathsf{OR} \leq g$.

Theorem 7.21 (Zhang)**.** *There is a two-party gadget $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ such that if $f\colon \{0,1\}^n \to Q$ is a function, then $f \circ g^n$ has communication complexity $\Omega(\mathrm{bs}(f))$.*

The proof runs roughly as follows. Fix an input $\alpha \in \{0,1\}^n$ for f that witnesses the block sensitivity $\mathrm{bs}(f, \alpha) = \mathrm{bs}(f)$. Also, let $B_1, \ldots, B_{\mathrm{bs}} \subseteq [n]$ be the sensitive blocks of f at α. Given an input (a, b) to $\mathsf{UDISJ}_{\mathrm{bs}}$ the goal in the reduction is for the two parties to compute, without communication, an input (x, y) for $f \circ g^n$ such that

(T1) *0-inputs:* If $\mathsf{UDISJ}_{\mathrm{bs}}(a, b) = 0$, then $g^n(x, y) = \alpha$.
(T2) *1-inputs:* If $\mathsf{UDISJ}_{\mathrm{bs}}(a, b) = 1$ with $a_i = b_i = 1$, then $g^n(x, y) = \alpha^{B_i}$.

Clearly, if we had a reduction $(a, b) \mapsto (x, y)$ satisfying (T1–T2), then the output of $\mathsf{UDISJ}_{\mathrm{bs}}(a, b)$ could be recovered from $(f \circ g^n)(x, y)$. Thus, an ϵ-error protocol for $f \circ g^n$ would imply an ϵ-error protocol for $\mathsf{UDISJ}_{\mathrm{bs}}$ with the same communication cost.

7.3.2 Search problems: Our approach

We are going to prove Theorem 7.2 (restated below) in close analogy to the proof template (T1–T2) above. However, as discussed below, noncritical inputs to search problems introduce new technical difficulties.

Theorem 7.2 (Two-party version)**.** *There is a two-party gadget $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ such that if $S \subseteq \{0,1\}^n \times Q$ is any search problem, then $S \circ g^n$ has randomised bounded-error communication complexity $\Omega(\mathrm{cbs}(S))$.*

Setup. Fix any versatile gadget $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. Let Π be a randomised ϵ-error protocol for a composed search problem $S \circ g^n$. Recall that an input (x, y) for the problem $S \circ g^n$ is *critical* if there is exactly one solution q with $((x, y), q) \in S \circ g^n$. In particular, if $g^n(x, y)$ is critical for S, then (x, y) is critical for $S \circ g^n$. The behaviour of the protocol Π on a critical input (x, y) is predictable: the protocol's output $\Pi(x, y)$ is the unique solution with probability at least $1 - \epsilon$.

However, noncritical inputs (x, y) are much trickier: not only can the distribution of the output $\Pi(x, y)$ be complex, but the distributions of $\Pi(x, y)$ and $\Pi(x', y')$ can differ even if (x, y) and (x', y') encode the same input $g^n(x, y) = g^n(x', y')$ of S. The latter difficulty is the main technical challenge, and we address it by using random-self-reducible gadgets.

Defining a function $f \subseteq S$. We start by following very closely the initial analysis in the proof of Huynh and Nordström [HN12]. First, we record for each $\alpha \in \{0,1\}^n$ the *most likely* *feasible output* of Π on inputs (x, y) that encode α. More formally, for each α we define μ_α to be the uniform distribution on the set of preimages of α, i.e.,

$$\mu_\alpha \text{ is uniform on } \{(x, y) : g^n(x, y) = \alpha\}. \tag{7.5}$$

Alternatively, this can be viewed as a product distribution

$$\mu_\alpha = \mu_{\alpha_1} \times \mu_{\alpha_2} \times \cdots \times \mu_{\alpha_n}, \tag{7.6}$$

where μ_z, $z \in \{0, 1\}$, is the uniform distribution on $g^{-1}(z)$.

The most likely feasible solution output by Π on inputs $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu_\alpha$ is now captured by a total function $f \subseteq S$ defined by

$$f(\alpha) \;:=\; \operatorname*{arg\,max}_{q:(\alpha,q)\in S} \mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu_\alpha}[\,\Pi(\boldsymbol{x}, \boldsymbol{y}) = q\,]. \tag{7.7}$$

Here, ties are broken arbitrarily and the randomness is taken over both $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu_\alpha$ and the random coins of the protocol Π. (Note that, in general, the most likely output of $\Pi(\boldsymbol{x}, \boldsymbol{y})$ may not be feasible. However, above, we explicitly pick out the most likely *feasible* solution. Thus, f is indeed a subfunction of S.)

The sensitive critical input. We can now use the critical block sensitivity of S: there is a critical input α such that $\mathrm{bs}(f, \alpha) \geq \mathrm{cbs}(S)$. Let $B_1, \ldots, B_{\mathrm{bs}} \subseteq [n]$ be the sensitive blocks with $f(\alpha^{B_i}) \neq f(\alpha)$.

Lemma 7.22. *The protocol Π can distinguish between μ_α and $\mu_{\alpha^{B_i}}$ in the sense that*

$$(\boldsymbol{x}, \boldsymbol{y}) \sim \mu_\alpha \quad\Longrightarrow\quad \mathbf{Pr}[\,\Pi(\boldsymbol{x}, \boldsymbol{y}) = f(\alpha)\,] \geq 1 - \epsilon, \tag{7.8}$$

$$(\boldsymbol{x}, \boldsymbol{y}) \sim \mu_{\alpha^{B_i}} \quad\Longrightarrow\quad \mathbf{Pr}[\,\Pi(\boldsymbol{x}, \boldsymbol{y}) = f(\alpha)\,] \leq 1/2. \tag{7.9}$$

Proof. The consequent in the first property (7.8) is true even for each individual (x, y) in the support of μ_α since α is critical. To see that the second property (7.9) is true, suppose for a contradiction that we had $\mathbf{Pr}[\,\Pi(\boldsymbol{x}, \boldsymbol{y}) = f(\alpha)\,] > 1/2$ for $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu_{\alpha^{B_i}}$. By averaging, there is a fixed input (x, y) in the support of $\mu_{\alpha^{B_i}}$ such that $\mathbf{Pr}[\,\Pi(x, y) = f(\alpha)\,] > 1/2$. By the correctness of Π (i.e., $1 - \epsilon > 1/2$) this implies that $f(\alpha)$ is feasible for α^{B_i}. Thus, $f(\alpha)$ is the most likely feasible solution output by $\Pi(\boldsymbol{x}, \boldsymbol{y})$, that is, $f(\alpha^{B_i}) = f(\alpha)$ by the definition (7.7). But this contradicts the fact that f is sensitive to B_i at α. □

The reduction. Lemma 7.22 suggests a reduction strategy analogous to the template (T1–T2) of Section 7.3.1. Given an input (a, b) for $\mathsf{UDISJ}_{\mathrm{bs}}$ our goal is to describe a randomised reduction $(a, b) \mapsto (\boldsymbol{x}, \boldsymbol{y})$ such that

(P1) *0-inputs:* If $\mathsf{UDISJ}_{\mathrm{bs}}(a, b) = 0$, then $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu_\alpha$.

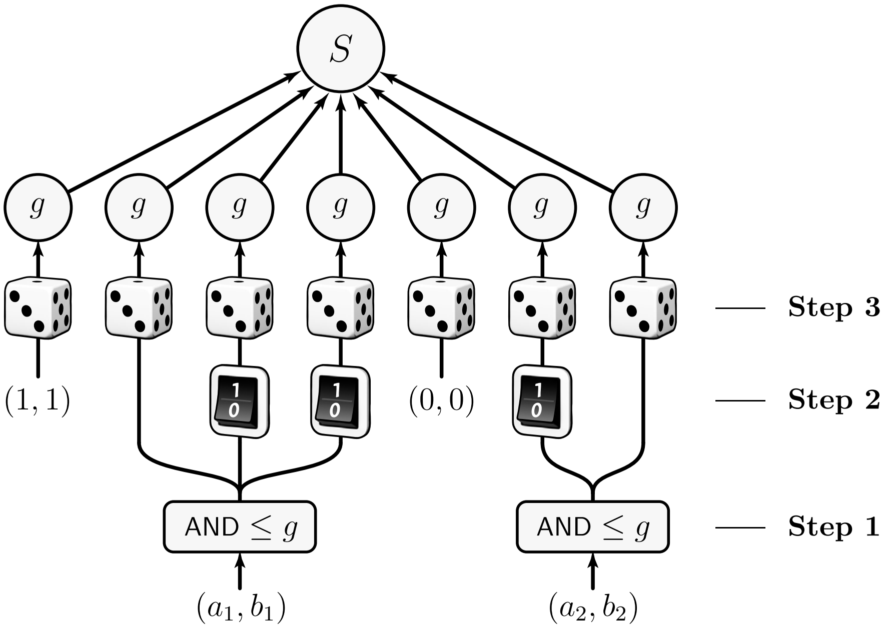


Figure 7.5: The reduction $(a,b) \mapsto (\boldsymbol{x},\boldsymbol{y})$. In this example $\mathrm{bs} = 2$ and $n = 7$. The critical input is $\alpha = 1011010$ and the two sensitive blocks are $B_1 = \{2,3,4\}$ and $B_2 = \{6,7\}$. The input pair (a_i, b_i), $i = 1,2$, is plugged in for the block B_i.

(P2) *1-inputs:* If $\mathsf{UDISJ}_{\mathrm{bs}}(a,b) = 1$ with $a_i = b_i = 1$, then $(\boldsymbol{x},\boldsymbol{y}) \sim \mu_{\alpha^{B_i}}$.

Suppose for a moment that we had a reduction with properties (P1–P2). Let Π' be the protocol that on input (a,b) first applies the reduction $(a,b) \mapsto (\boldsymbol{x},\boldsymbol{y})$ with properties (P1–P2), then runs Π on $(\boldsymbol{x},\boldsymbol{y})$, and finally outputs 0 if $\Pi(\boldsymbol{x},\boldsymbol{y}) = f(\alpha)$ and 1 otherwise. Lemma 7.22 tells us that

- If $\mathsf{UDISJ}_{\mathrm{bs}}(a,b) = 0$, then $\Pi'(a,b) = 0$ with probability at least $1-\epsilon$.
- If $\mathsf{UDISJ}_{\mathrm{bs}}(a,b) = 1$, then $\Pi'(a,b) = 1$ with probability at least $1/2$.

The error probability of Π' can be bounded away from $1/2$ by repeating Π' twice and outputting 0 iff both runs of Π' output 0. (Here we are assuming that ϵ is small enough, say at most $1/4$. If not, we can use some other standard success probability boosting tricks.) This gives a randomised protocol for $\mathsf{UDISJ}_{\mathrm{bs}}$ with the same communication cost (up to constants) as that of Π. Theorem 7.2 follows.

Indeed, it remains to implement a reduction $(a,b) \mapsto (\boldsymbol{x},\boldsymbol{y})$ satisfying (P1–P2). We do it in three steps; see Figure 7.5.

Step 1. On input $(a,b) = (a_1 \dots a_{\mathrm{bs}}, b_1 \dots b_{\mathrm{bs}})$ to $\mathsf{UDISJ}_{\mathrm{bs}}$ we first take each pair (a_i, b_i) through the reduction $\mathsf{AND} \leq g$ to obtain instances $(a'_1, b'_1), \dots, (a'_{\mathrm{bs}}, b'_{\mathrm{bs}})$ of g. Note that

- if $\mathsf{UDISJ}_{\mathrm{bs}}(a,b)=0$, then $g(a'_i,b'_i)=0$ for all i;
- if $\mathsf{UDISJ}_{\mathrm{bs}}(a,b)=1$, then there is a unique i with $g(a'_i,b'_i)=1$.

Step 2. Next, the instances (a'_i,b'_i) are used to populate a vector $(x,y)=(x_1\dots x_n, y_1\dots y_n)$ carrying n instances of g, as follows. The instance (a'_i,b'_i) is plugged in for the coordinates $j\in B_i$ with the copies corresponding to $\alpha_j=1$ flipped. That is, we define for $j\in B_i$:

- if $\alpha_j=0$, then $(x_j,y_j):=(a'_i,b'_i)$;
- if $\alpha_j=1$, then $(x_j,y_j):=(\pi_A(a'_i),\pi_B(b'_i))$, where (π_A,π_B) is the reduction $\neg g\le g$.

For $j\notin\cup_i B_i$ we simply fix an arbitrary $(x_j,y_j)\in g^{-1}(\alpha_j)$. We now have that

- if $\mathsf{UDISJ}_{\mathrm{bs}}(a,b)=0$, then $g^n(x,y)=\alpha$;
- if $\mathsf{UDISJ}_{\mathrm{bs}}(a,b)=1$ with $a_i=b_i=1$, then $g^n(x,y)=\alpha^{B_i}$.

Step 3. Finally, we apply a random-self-reduction independently for each component (x_i,y_i) of (x,y): this maps a z-input (x_i,y_i) to a uniformly random z-input $(\boldsymbol{x}_i,\boldsymbol{y}_i)\sim\mu_z$. The result is a random vector $(\boldsymbol{x},\boldsymbol{y})$ that has a distribution of the form (7.6) and matches our requirements (P1–P2), as desired.

This concludes the proof of Theorem 7.2. The proof of the multi-party version (Theorem 7.3) is exactly the same, except with g and $\mathsf{UDISJ}_{\mathrm{bs}}$ replaced by a versatile g_k and $\mathsf{UDISJ}_{k,\mathrm{bs}}$. Here, in particular, $\mathsf{UDISJ}_{k,n}$ is the usual k-party disjointness function $\mathsf{DISJ}_{k,n}=\mathsf{OR}_n\circ\mathsf{AND}_k^n$ together with the promise that at most one of the AND_k's evaluates to 1.

7.4 Critical block sensitivity lower bounds

In this section we prove our new critical block sensitivity bounds, Theorems 7.7 and 7.9.

7.4.1 Tseitin sensitivity

Let $G=(V,E,\ell)$ be a connected graph with an odd-weight labelling $\ell\colon V\to\{0,1\}$. Recall that in the problem $S(\mathit{Tse}_G)$ the input is an assignment $\alpha\colon E\to\{0,1\}$ and the goal is to find a parity violation, that is, a node in $\mathrm{Viol}(\alpha):=\{v\in V: C_v(\alpha)=0\}$.

For the readers' convenience, we recall some basic facts about Tse_G. Since each edge $e\in E$ participates in two constraints, the sum $\sum_v\sum_{e:v\in e}\alpha(e)$ will be even. By contrast, the sum $\sum_v\ell(v)$ is odd. It follows that $|\mathrm{Viol}(\alpha)|$ must be odd, and, in particular, non-empty. Conversely, for every odd-size set $U\subseteq V$, there is an α with $\mathrm{Viol}(\alpha)=U$. To see this, start with any assignment $E\to\{0,1\}$ and let p be a simple path in G. If we flip the truth values of the edges in p, we end up flipping whether or not the constraints at the endpoints of p are satisfied. Depending on whether the endpoints of p were satisfied to begin with, this results in one of the following scenarios: (1) we create a pair of violations; (2) we remove a pair of violations; or (3)

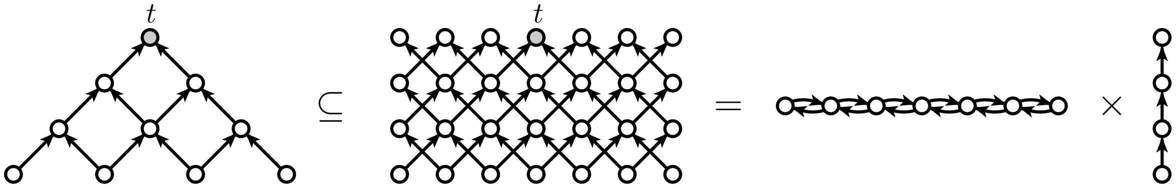


Figure 7.6: Pyramid graph viewed as a subgraph of a tensor product of paths.

we move a violation from one endpoint of p to the other. It is not hard to see that by using (1)–(3) repeatedly, we can design an assignment α such that $\mathrm{Viol}(\alpha) = U$.

We are now ready to prove Theorem 7.7.

Theorem 7.7 (Tseitin sensitivity)**.** *If G is κ-routable, then* $\mathrm{cbs}(S(\mathit{Tse}_G)) = \Omega(\kappa)$*.*

Proof. Let $G = (V, E, \ell)$ be $(\kappa + 1)$-routable. Fix a set $T \subseteq V$ of size $|T| = 2\kappa + 1$ such that whenever M is a set of κ disjoint pairs of nodes from T, there are κ edge-disjoint paths connecting each pair in M. We denote by $\mathrm{Paths}(M)$ some canonical set of such paths.

Consider the following bipartite auxiliary graph on *left* and *right* vertices:

- **Left vertices** are pairs (α, M), where $\alpha \colon E \to \{0,1\}$ has a *unique* violation that is in T (i.e., $|\mathrm{Viol}(\alpha)| = 1$ and $\mathrm{Viol}(\alpha) \subseteq T$), and M is a partition of the set $T \smallsetminus \mathrm{Viol}(\alpha)$ into κ pairs of nodes.
- **Right vertices** are pairs (α', M'), where $\alpha' \colon E \to \{0,1\}$ has *three* violations that are all in T (i.e., $|\mathrm{Viol}(\alpha')| = 3$ and $\mathrm{Viol}(\alpha') \subseteq T$), and M' is a partition of the set $T \smallsetminus \mathrm{Viol}(\alpha')$ into $\kappa - 1$ pairs of nodes.
- **Edges** are defined as follows. A left vertex (α, M) is connected to a right vertex (α', M') if $M' \subseteq M$ and α' is obtained from α by flipping the values along the path in $\mathrm{Paths}(M)$ that connects the pair $\mathrm{Viol}(\alpha') \smallsetminus \mathrm{Viol}(\alpha)$.

The key fact, which is easy to verify, is that the auxiliary graph is *biregular*: its left-degree is κ and its right-degree is 3.

To prove the block sensitivity bound, let f be a function solving $S(\mathit{Tse}_G)$. We say that an edge from (α, M) to (α', M') in the auxiliary graph is *sensitive* if $f(\alpha) \neq f(\alpha')$. Clearly, for each right vertex exactly two (out of three) of its incident edges are sensitive. Thus, by averaging, we may find a left vertex (α, M) such that at least a fraction 2/3 of its incident edges are sensitive. But this means that α is a critical input with block sensitivity at least $2\kappa/3$; the blocks are given by a subset of $\mathrm{Paths}(M)$. □

7.4.2 Pebbling sensitivity

Theorem 7.9 (Pebbling sensitivity)**.** *There are n-node bounded-degree graphs G such that*

- *G has pebbling number $\Theta(\sqrt{n})$.*

- *$S(\mathit{Peb}_G)$ has deterministic decision tree complexity $\Theta(\sqrt{n})$.*
- *$S(\mathit{Peb}_G)$ has critical block sensitivity $\Theta(\sqrt{n})$.*

Overview. Our proof of Theorem 7.9 generalises the original proof from [HN12] that held for pyramid graphs. The key idea is natural: In a pyramid graph, each horizontal layer can be interpreted as a path—this is made precise by viewing the pyramid graph as a subgraph of a tensor product of paths as in Figure 7.6. The analysis in the original proof suffered from the fact that random walks do not mix well on paths. So, we replace the paths by graphs with better mixing properties! (Perhaps surprisingly, we do not need to rely on expanders here.)

Definition of *G*. Let H be the 3-dimensional grid graph on $m = r^3$ nodes where r is odd. For convenience, we think of H as a directed Cayley graph on $\mathbb{Z}_r^3$ generated by the 6 elements

$$\mathcal{B} = \{\pm(1,0,0), \pm(0,1,0), \pm(0,0,1)\}.$$

That is, there is an edge $(v,u) \in E(H)$ iff $u = v + b$ for some $b \in \mathcal{B}$. The key property of H (which is not satisfied by d-dimensional grid graphs for $d < 3$) is the following.

Lemma 7.23 (Partial cover time)**.** *Starting from any node of H the expected number of steps it takes for a random walk to visit at least half of the nodes of H is $\mathrm{pct}(H) = O(m)$.*

Proof. This follows from Lemma 2.8 in [Lov93] and the fact that the maximum hitting time of H is $O(m)$ (e.g., [CRR$^+$96]). □

Let $\ell := 2 \cdot \mathrm{pct}(H) + 1 = \Theta(m)$ so that by Markov's inequality a random walk of length $\ell - 1$ in H will cover a at least a fraction $1/2$ of H with probability at least $1/2$. Let P be the directed path on $[\ell]$ with edges $(i, i+1)$, $i \in [\ell - 1]$. We construct the tensor product graph

$$G := H \times P$$

that is defined by $V(G) = \mathbb{Z}_r^3 \times [\ell]$ and there is a directed edge from (v,i) to (u,j) iff $j = i + 1$ and $u = v + b$ for some $b \in \mathcal{B}$.

The $n = m\ell$ nodes of G are naturally partitioned into ℓ *layers* (or *steps*). In order to turn G into a pebbling formula, we need to fix some *sink* node t in the ℓ-th layer and delete all nodes from which t is not reachable. We do not let this clean-up operation affect our notations, though. For example, we continue to think of the resulting graph as $G = H \times P$. The nodes $\mathbb{Z}_r^3 \times \{1\}$ of indegree 0 will be the *sources*.

Note that each source–sink path p in G contains exactly one node from each layer. We view the projection of p onto H as a walk of length $\ell - 1$ in H; we can describe the walk uniquely by a sequence of $\ell - 1$ generators from $\mathcal{B}$. We denote by $\pi(p) \subseteq V(H)$ the set of nodes visited by the projected walk.

We can now study the search problem $S(\mathit{Peb}_G)$ associated with the pebbling formula Peb_G.

Pebbling number. The pebbling strategy for G that uses $O(\sqrt{n}) = O(m)$ pebbles proceeds as follows. We first pebble the 1st layer (the sources), then the 2nd layer, then remove pebbles from the 1st layer, then pebble the 3rd layer, then remove pebbles from the 2nd layer, etc.

The matching lower bound follows from the fact that G contains a pyramid graph on $\Omega(n)$ nodes as a subgraph, and the pebbling number of pyramid graphs is $\Theta(\sqrt{n})$ [Coo74].

Decision tree complexity. The deterministic decision tree that uses $O(\sqrt{n}) = O(m)$ queries proceeds as follows. We start our search for a violated clause at the sink t. If the sink variable is false, we query its children to find a child v whose associated variable is false. The search continues at v in the same manner. In at most $\ell - 1 = O(m)$ steps we find a false node v whose children are all true (perhaps v is a source node).

The matching lower bound follows from the critical block sensitivity lower bound proved below, and the fact that critical block sensitivity is a lower bound on the decision tree complexity.

Critical block sensitivity. It remains to prove that $\mathrm{cbs}(S(Peb_G)) = \Omega(m)$. The following proof is a straightforward generalisation of the original proof from (the full version of) [HN12].

All *paths* that we consider in the following are source–sink paths in G. We associate with each path p a critical input $\alpha_p \colon V(G) \to \{0,1\}$ that assigns to each node on p the value 0 and elsewhere the value 1. This creates a unique clause violation at the source where p starts.

If p and q are two paths, we say that p and q are *paired at* $i \geq 2$ if the following hold.

- *Agreement:* p and q do not meet before layer i, but they agree on all layers $i, \ldots, \ell$.
- *Mirroring:* if the first $i-1$ steps of p are described by $(b_1, b_2, \ldots, b_{i-1}) \in \mathcal{B}^{i-1}$, then the first $i-1$ steps of q are described by $(-b_1, -b_2, \ldots, -b_{i-1}) \in \mathcal{B}^{i-1}$.

Each path can be paired with at most $\ell - 1$ other paths—often, there are plenty such:

Lemma 7.24. *Each path p is paired with at least $|\pi(p)| - 1$ other paths.*

Proof. For each node $v \in \pi(p)$, except the starting point of p, we construct a pair q for p. To this end, let $i \geq 2$ be the first step at which the projection of p visits v. Since the mirroring property uniquely determines q given p and i, we only need to show that this q satisfies the agreement property. Thus, suppose for a contradiction that p and q meet at some node (u, j) where $j < i$. We have, in $\mathbb{Z}_r^3$ arithmetic,

$$\begin{aligned} v &= u + b_j + b_{j+1} + \cdots + b_{i-1} && \text{(according to } p\text{)}, \\ v &= u - b_j - b_{j+1} - \cdots - b_{i-1} && \text{(according to } q\text{)}. \end{aligned}$$

This implies $2v = 2u$, but since r is odd, we get $v = u$. This contradicts our choice of i. □

If p and q are paired, we can consider the assignment $\alpha_{p\cup q}$ that is the node-wise logical AND of the assignments α_p and α_q. In $\alpha_{p\cup q}$ we have *two* clause violations associated with the two starting points of the paths.

To prove the critical block sensitivity bound $\Omega(m)$, let f be a function solving $S(Peb_G)$. Consider the following auxiliary graph.

- The **vertices** are the source–sink paths.
- There is a **directed edge** from p to q iff p and q are paired and $f(\alpha_{p\cup q})$ is the starting point of q. Thus, each two paired paths are connected by an edge one way or the other.

Recall that if we start a random walk of length $\ell - 1$ on H at any fixed node, the walk covers a fraction $\geq 1/2$ of H with probability $\geq 1/2$. If we view a source-sink path p in G in the *reverse order* (starting at the sink and going towards the source), this translates into saying that $|\pi(p)| \geq m/2$ for a fraction $\geq 1/2$ of all paths p. Applying Lemma 7.24 for such paths we conclude that the auxiliary graph has average outdegree at least $d = m/8 - 1$. By averaging, we can now find a path p with out-neighbours $q_1, \ldots, q_d$. Define $q_i' \coloneqq q_i \smallsetminus p$. Clearly the critical assignment α_p is sensitive to each q_i'. To see that the q_i' are pairwise disjoint, we note that they take steps in the same direction in $\mathcal{B}$ at each layer (i.e., opposite to that of p), and the q_i meet p for the first time at distinct layers. This concludes the proof of Theorem 7.9.

7.5 Monotone CSP-SAT

In this section we introduce a monotone variant of the CSP satisfiability problem and show how lifted search problems $S(F) \circ g^n$ reduce to its monotone Karchmer–Wigderson game. In particular, in Corollary 7.10 we can take the explicit function to be a CSP satisfiability function. We also note that our function has been further studied by Oliveira [Oli15, Chapter 3].

Definition of monotone CSP-SAT. The function is defined relative to some finite alphabet Σ and a fixed constraint topology given by a bipartite graph G with left vertices V (*variable nodes*) and right vertices U (*constraint nodes*). We think of each $v \in V$ as a variable taking on values from Σ; an edge $(v, u) \in E(G)$ indicates that variable v is involved in constraint node u. Let d be the maximum degree of a node in U. We define $\mathsf{SAT} = \mathsf{SAT}_{G,\Sigma}\colon \{0,1\}^N \to \{0,1\}$ on $N \leq |U| \cdot |\Sigma|^d$ bits as follows. An input $\alpha \in \{0,1\}^N$ describes a CSP instance by specifying, for each constraint node $u \in U$, its *truth table*: a list of at most $|\Sigma|^d$ bits that record which assignments to the variables involved in u satisfy u. Then $\mathsf{SAT}(\alpha) \coloneqq 1$ iff the CSP instance described by α is satisfiable. This encoding of CSP satisfiability is indeed monotone: if we flip any 0 in a truth table of a constraint into a 1, we are only making the constraint easier to satisfy.

7.5.1 Reduction to CSP-SAT

Recall the characterisation of monotone depth due to Karchmer and Wigderson [KW88]: if $f\colon \{0,1\}^N \to \{0,1\}$ is a monotone function, then its monotone depth complexity is equal to the (deterministic) communication complexity of the following search problem.

> **Monotone KW-game for f:** Alice holds a $a \in f^{-1}(1)$ and Bob holds a $b \in f^{-1}(0)$. The goal is to find a coordinate $i \in [N]$ such that $a_i = 1$ and $b_i = 0$.

The next lemma shows that for any search problem of the form $S(F) \circ g^n$ there is a some monotone CSP-SAT function whose monotone KW-game embeds $S(F) \circ g^n$. (The reduction can be seen as a generalisation of Lemma 3.5 in [RM99].)

We define the *constraint topology* of F naturally as the bipartite graph G with left vertices $\mathrm{vars}(F)$ and right vertices $\mathrm{cons}(F)$. For a constraint $C \in \mathrm{cons}(F)$ we use the lower case c to denote the corresponding node in G (forgetting that C is actually a function).

Lemma 7.25. *Let $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a two-party gadget and let F be an unsatisfiable d-CSP on n variables and m constraints. Let G be the constraint topology of F. Then the monotone depth complexity of $\mathsf{SAT}_{G,\mathcal{X}}\colon \{0,1\}^N \to \{0,1\}$, $N \le m|\mathcal{X}|^d$, is lower bounded by the (deterministic) communication complexity of $S(F) \circ g^n$.*

Proof. We reduce the search problem $S(F) \circ g^n$ to the monotone KW-game for $\mathsf{SAT}_{G,\mathcal{X}}$. To this end, let (x, y) be an input to the search problem $S(F) \circ g^n$ and compute as follows.

- Alice maps $x \in \mathcal{X}^{\mathrm{vars}(F)}$ to the CSP whose sole satisfying assignment is x. That is, the truth table for a constraint node c is set to all-0 except for the entry indexed by $x \restriction \mathrm{vars}(C)$ (restriction of x to the variables in C).
- Bob maps $y \in \mathcal{Y}^{\mathrm{vars}(F)}$ to an unsatisfiable CSP as follows. The truth table for a constraint node c is such that the bit indexed by $\ell \in \mathcal{X}^{\mathrm{vars}(C)}$ is set to 1 iff C is satisfied under the partial assignment $v \mapsto g(\ell(v), y(v))$ where $v \in \mathrm{vars}(C)$.

Alice clearly constructs a 1-input of $\mathsf{SAT}_{G,\mathcal{X}}$. To see that Bob constructs a 0-input of $\mathsf{SAT}_{G,\mathcal{X}}$, suppose for a contradiction that there is a global assignment $\ell\colon \mathrm{vars}(F) \to \mathcal{X}$ so that the truth table of each c has a 1 in position indexed by $\ell \restriction \mathrm{vars}(C)$. This would mean that the truth assignment $v \mapsto g(\ell(v), y(v))$ satisfies all the constraints of F. But this contradicts the unsatisfiability of F.

Assume then that Alice and Bob run a protocol for the monotone KW-game on the CSP instances constructed above. The output of the protocol is a some entry $\ell \in \mathcal{X}^{\mathrm{vars}(C)}$ in the truth table of some constraint node c where Alice has a 1 and Bob has a 0. Because Alice's CSP was constructed so that for each constraint node c exactly one entry is 1, we must have that $\ell = x \restriction \mathrm{vars}(C)$. On the other hand, Bob's construction ensures that C is not satisfied under the assignment $v \mapsto g(\ell(v), y(v)) = g(x(v), y(v))$. Thus, we have found a violated constraint C for the canonical search problem for F. □

7.5.2 Proof of Corollary 7.10

Theorems 7.2 and 7.7 yield a search problem $S(Tse_G) \circ g^m$ of communication complexity $\Omega(n/\log n)$ where G is an n-node m-edge bound-degree graph ($d = O(1)$, $m = O(n)$) and g is a constant-size gadget ($|\mathcal{X}| = O(1)$). Using Lemma 7.25 we can then construct a CSP-SAT function on $N = O(n)$ bits having monotone depth $\Omega(n/\log n) = \Omega(N/\log N)$. This proves Corollary 7.10.

7.6 Applications: Proof complexity

In this section we prove our new proof complexity lower bounds as stated in Section 7.1.8. We start by reviewing some standard proof complexity terminology.

7.6.1 Background

In this chapter we focus on proof systems that refute unsatisfiable CNF formulas. Given a proof system, a *refutation* (or a *proof*) P of an unsatisfiable CNF formula F in the system is expressed as a sequence of *lines*, denoted $\mathrm{Lines}(P)$, each of which is either (a translation of) a clause of F or follows from some previous lines via some sound *inference rule*. The refutation ends with some trivially false line.

For each proof P we can associate a directed acyclic graph $G_P = (V, E)$ where $V = \mathrm{Lines}(P)$ and there is an edge $(u, v) \in E$ if v is derived via some inference rule using line u.

Complexity measures. For the purposes of this chapter, we define the *size* of a proof P simply as the number of lines $|\mathrm{Lines}(P)|$. The *rank* of P is the length of the longest path in G_P. The *size complexity* and *rank complexity* of F in a proof system are the minimum size and minimum rank, respectively, of all refutations of F in that system.

We consider G_P to be a tree if every internal node has fan-out one, that is, the clauses of F, which are not internal nodes, can be repeated. If G_P is a tree, we say that P is *tree-like*. The *tree-like size complexity* of F is the minimum size of a tree-like refutation of F. Note that restricting a refutation to be tree-like does not increase the rank because each line can be re-derived multiple times without affecting the rank. Tree-like size, however, can be much larger than general size.

Examples of proof systems. We mention some of the most well-studied proof systems. In each of these systems, there is a set of derivation rules (which can be thought of as inference schemas) of the form $F_1, F_2, \ldots, F_t \vdash F_{t+1}$ and each inference in a proof must be an instantiation of one of these rules.

A basic system is *Resolution* whose lines are clauses. Its only rule is the *resolution rule*: the clause $(A \vee B)$ can be derived from $(A \vee x)$ and $(B \vee \neg x)$, where A and B are arbitrary disjunctions of literals and x is a variable. A Resolution refutation of an unsatisfiable CNF formula f is a sequence of clauses, ending with the empty clause, such that each clause in the sequence is either a clause of f, or follows from two previously derived clauses via the resolution rule.

Another proof system is the *Cutting Planes* (CP) proof system that manipulates integer linear inequalities. A CP refutation is a sequence of inequalities, ending with $0 \geq 1$, such that all inequalities are either translations of clauses of F, or follow from two previously derived inequalities via one of the two CP rules, addition and division with rounding. There is a natural extension of CP, denoted $\mathsf{CP}(k)$, in which the above CP proof rules may also be applied when the lines are allowed to be degree k multivariate polynomials.

Other important well-studied proof systems are the Lovász–Schrijver proof systems (LS_0, LS, LS_+, and $\mathsf{LS}_{+,\star}$) which are dynamic proof systems that manipulate polynomial inequalities of degree at most 2; the Sherali–Adams and Lasserre (SOS) systems that are static proof systems allowing polynomial inequalities of higher degree; and the dynamic Lasserre (dynamic SOS), and $\mathsf{LS}^k_{+,\star}$ systems, which generalize the Lovász–Schrijver systems to higher degree. We refer the reader to [OZ13] for formal definitions and a thorough history for these and related proof sytems.

Semantic proof systems. Each of the above proof systems has a specific set of inference rule schemas, which allows them to have polynomial-time verifiers. In this chapter we consider more powerful *semantic* proof systems that restrict the form of the lines and the fan-in of the inferences but dispense with the requirement of a polynomial-time verifier and allow any semantically sound inference rule with a given fan-in. The fan-in must be restricted because the semantic rules are so strong. The following system was introduced in [BPS07].

Definition 7.26 (Degree k threshold proofs)**.** We denote by $\mathsf{Th}(k)$ the semantic proof system whose proofs have fan-in 2 and each line in a refutation of a formula F is a polynomial inequality of degree at most k in the variables of F. In particular, each clause of F enters the system as translated into a linear inequality (similarly to the CP system discussed above).

The following lemma follows from Caratheodory's Theorem.

Lemma 7.27. CP *and* LS *proofs can be efficiently converted into* $\mathsf{Th}(k)$ *proofs:*

- *Any* CP *proof of size (tree-like size) s and rank r can be converted to a* $\mathsf{Th}(1)$ *proof of size (tree-like size) $O(s)$ and rank $O(r \log s)$.*
- *Any* LS_0*,* LS*, or* LS_+ *proof of size (tree-like size) s and rank r can be converted to a* $\mathsf{Th}(2)$ *proof of size (tree-like size) $O(s)$ and rank $O(r \log s)$.*

Moreover, it is not hard to show that one can extend the above simulations by $\mathsf{Th}(k)$ proofs to $\mathsf{CP}(k)$, $\mathsf{LS}^k_{+,\star}$, and degree k (dynamic) Lasserre proofs.

In this paper we consider semantic proof systems that are even more general than $\mathsf{Th}(k)$, namely those for which the fan-in is bounded and the truth value of each line can be computed by an efficient multi-party NOF communication protocol.

Definition 7.28 (Proofs with k-party verifiers)**.** We denote by $\mathsf{T}^{\mathsf{cc}}(k, c)$ the semantic proof system of fan-in 2 in which each proof line is a boolean function whose value, for every k-partition of the input variables, can be computed by a c-bit randomised k-party NOF protocol of error at most 1/4. Both $k = k(s)$ and $c = c(s)$ may be functions of s, the size of the input formula. In keeping with the usual notions of what constitutes efficient communication, we use $\mathsf{T}^{\mathsf{cc}}(k)$ to denote $\mathsf{T}^{\mathsf{cc}}(k, \operatorname{polylog} s)$.

Note that via standard boosting, we can replace the error 1/4 in the above definition by ϵ at the cost of increasing c by an $O(\log 1/\epsilon)$ factor. Therefore, without loss of generality, in the definition of $\mathsf{T}^{\mathsf{cc}}(k)$ we can assume that the error is at most $2^{-\operatorname{polylog} s}$.

For polylogarithmic k, the following lemma shows that $\mathsf{Th}(k)$ is a subclass of $\mathsf{T}^{\mathsf{cc}}(k+1)$.

Lemma 7.29. *Every $\mathsf{Th}(k)$ refutation of an n-variable CNF formula is a $\mathsf{T}^{\mathsf{cc}}(k+1, O(k^3 \log^2 n))$ refutation.*

Proof. By the well-known result of Muroga [Mur71], linear threshold functions on n boolean variables only require coefficients of $O(n \log n)$ bits. Since a degree k threshold polynomial is a linear function on at most n^k monomials, it is equivalent to a degree k threshold polynomial with coefficients of $O(kn^k \log n)$ bits. As shown in [BPS07], over any input partition there is a randomized $(k+1)$-party communication protocol of cost $O(k \log^2 b)$ and error $\leq 1/b^{\Omega(1)}$ to verify a degree k polynomial inequality with b-bit coefficients. □

The following lemma, which is implicit in [BPS07], gives the key relationships between $\mathsf{T}^{\mathsf{cc}}(k)$ and randomised communication protocols for $S(F)$.

Lemma 7.30. *If a CNF formula F has a $\mathsf{T}^{\mathsf{cc}}(k, c)$ refutation of rank r then, over any k-partition of the variables, there is a randomised bounded-error k-party NOF protocol for $S(F)$ with communication cost $O(c \cdot r \log r)$.*

7.6.2 Lifting CNF formulas

In order to import our communication lower bounds to proof complexity, we need to encode composed search problems $S \circ g_k^n$ as CNF formulas. We describe a natural way of doing this in case $S = S(F)$ is the search problem associated with some CNF formula F.

Fix a d-CNF formula F on n variables and m clauses. Also, fix a k-party gadget $g_k \colon \mathcal{X}^k \to \{0, 1\}$ where each player holds $l \coloneqq \log |\mathcal{X}|$ bits as input. We construct a new D-CNF formula $F \circ g_k^n$ on N variables and M clauses, where

$$D = d \cdot kl, \qquad N = n \cdot kl, \qquad \text{and} \qquad M \leq m \cdot 2^{dkl}. \tag{7.10}$$

Variables of $F \circ g_k^n$. For each variable x of F we create a matrix of variables

$$X = \{\, X_{ij} : i \in [k],\ j \in [l] \,\}.$$

The idea is that truth assignments $\alpha_X \colon X \to \{0, 1\}$ are in a natural one-to-one correspondence with the set $\mathcal{X}^k$, the domain of g_k. Namely, the value of the j-th bit of the i-th player is encoded by X_{ij}. We take the variable set of $F \circ g_k^n$ to be the union $X \cup Y \cup \ldots$, where $x, y, \ldots$ are the original variables of F.

Clauses of $\boldsymbol{F \circ g_k^n}$. Let C be a clause of F; suppose first that $C = (x \vee \neg y)$ for simplicity. We will replace C with a set of clauses $\mathcal{C}$ on the variables $X \cup Y$ such that all clauses of $\mathcal{C}$ are satisfied under an assignment $\alpha\colon X \cup Y \to \{0,1\}$ if and only if $g_k(\alpha_X) = 1$ or $g_k(\alpha_Y) = 0$; here α_X and α_Y are elements of $\mathcal{X}^k$ associated with the restrictions of α to X and Y. Indeed, let $X_{ij}^\alpha = X_{ij}$ if $\alpha(X_{ij}) = 1$, and $X_{ij}^\alpha = \neg X_{ij}$ if $\alpha(X_{ij}) = 0$, and similarly for Y_{ij}^α. Define a clause

$$C_\alpha = \Big(\neg \bigwedge_{i,j} X_{ij}^\alpha\Big) \vee \Big(\neg \bigwedge_{i,j} Y_{ij}^\alpha\Big),$$

and let $\mathcal{C}$ consist of all the clauses C_α where α is such that $g_k(\alpha_X) = 0$ and $g_k(\alpha_Y) = 1$.

More generally, if we had started with a clause on d variables, each clause C_α would involve dkl variables and so we would have $|\mathcal{C}| \le 2^{dkl}$. This completes the description of $F \circ g_k^n$.

The formula $F \circ g_k^n$ comes with a natural partition of the variables into k parts as determined by the k-party gadget. Thus, we can consider the canonical search problem $S(F \circ g_k^n)$.

Lemma 7.31. *The two problems $S(F \circ g_k^n)$ and $S(F) \circ g_k^n$ have the same k-party communication complexity up to an additive dkl term.*

Proof. As discussed above, the *inputs* to the two problems are in a natural one-to-one correspondence. How about translating *solutions* between the problems? Given a violated clause C_α in the problem $S(F \circ g_k^n)$, it is easy to reconstruct C from C_α without communication. Moreover, given a violated clause C of F in the problem $S(F) \circ g_k^n$, we can construct a violated C_α by first finding out what encoding α was used for each of the d variables of C. This can be done by communicating dkl bits (even in the number-in-hand model). □

7.6.3 Rank lower bounds

We are now ready to prove Theorem 7.12, restated here for convenience.

Theorem 7.12 (Rank lower bounds)**.** *There are explicit CNF formulas F of size s and width $O(\log s)$ such that all $\mathsf{T}^{\mathsf{cc}}(k)$ refutations of F require rank at least*

$$R_k(s) = \begin{cases} s^{1-o(1)}, & \textit{for } k = 2, \\ s^{1/2-o(1)}, & \textit{for } 3 \le k \le (\log s)^{1-o(1)}. \end{cases}$$

Proof. We start with a Tseitin formula F with n variables, $O(n)$ clauses, and width $O(1)$ that is associated with a $\Omega(n/\log n)$-routable bounded-degree graph. Let $k = k(n)$ be a parameter. We construct the formula $F \circ g_k^n$ where $g_k^n\colon \mathcal{X}^k \to \{0,1\}$ is the gadget of Corollary 7.20. Recall that $\log|\mathcal{X}| = k^\epsilon$ where $\epsilon = \epsilon(k) \to 0$ as $k \to \infty$. Using (7.10), we observe

- $F \circ g_k^n$ has size $s = O(n) \cdot \exp(O(k^{1+\epsilon}))$,
- $F \circ g_k^n$ has width $O(k^{1+\epsilon})$,
- $S(F \circ g_k^n)$ has k-party NOF communication complexity $\mathsf{CC} = \Omega(\sqrt{n/\log n}/2^k k)$; this follows from Lemma 7.31, Theorems 7.3 and 7.7, and Sherstov's lower bound [She13b]. (Alternatively, the complexity is $\Omega(n/\log n)$ in case $k = 2$.)

Fix $\delta > 0$ and choose $k = (\log n)^{1-\delta}$. For large n, the above bounds translate into:

$$s = n^{1+o(1)}, \qquad \text{width} \le \log n, \qquad \text{and} \qquad \mathsf{CC} \ge n^{1/2-o(1)}.$$

Therefore, by Lemma 7.30, there are no $\mathsf{T}^{\mathsf{cc}}(k)$ refutations of $F \circ g_k^n$ with rank at most $n^{1/2-o(1)}/\operatorname{polylog} n = n^{1/2-o(1)}$. The result follows by letting $\delta \to 0$ sufficiently slowly. □

7.6.4 Length–space lower bounds

In order to study the space that is required by a refutation, we need to switch to a more appropriate *space-oriented* view of proofs.

Definition 7.32 (Space-oriented proofs. E.g., [Nor13, S2.2])**.** A refutation of a CNF formula F in *length* L and *space* Sp is a sequence of *configurations* $\mathbb{D}_0, \ldots, \mathbb{D}_L$ where each $\mathbb{D}_i$ is a set of lines (of the underlying proof system) satisfying $|\mathbb{D}_i| \le Sp$ and such that $\mathbb{D}_0 = \emptyset$, $\mathbb{D}_L$ contains a trivially false line, and $\mathbb{D}_i$ is obtained from $\mathbb{D}_{i-1}$ via one of the following derivation steps:

- **Clause download:** $\mathbb{D}_i = \mathbb{D}_{i-1} \cup \{v_C\}$ where v_C is a translation of some clause C of F.
- **Inference:** $\mathbb{D}_i = \mathbb{D}_{i-1} \cup \{v\}$ where v follows from some number of lines of $\mathbb{D}_{i-1}$ by an inference rule of the system.
- **Erasure:** $\mathbb{D}_i = \mathbb{D}_{i-1} \smallsetminus \{v\}$ for some $v \in \mathbb{D}_{i-1}$.

Huynh and Nordström [HN12] proved that if F has a $\mathsf{T}^{\mathsf{cc}}(2)$ refutation of short length and small space, then there is a low-cost randomised two-party protocol for $S(F)$. It is straightforward to show that this result holds more generally for $\mathsf{T}^{\mathsf{cc}}(k)$ proofs and k-party protocols. The high level idea is that the players can use the refutation of F to do a binary search for a violated clause.

Lemma 7.33 (Simulation of space-bounded proofs)**.** *Fix a CNF formula F of size s and some k-partition of its variables. If F has a $\mathsf{T}^{\mathsf{cc}}(k)$ refutation of length L and space Sp, then there is a k-party randomised bounded-error protocol for $S(F)$ of communication cost*

$$Sp \cdot \log L \cdot \operatorname{polylog} s.$$

Proof. Let $\alpha\colon \operatorname{vars}(F) \to \{0,1\}$ be an input to the search problem $S(F)$. Fix a length-L space-Sp refutation of F with configurations $\mathbb{D}_0, \ldots, \mathbb{D}_L$.

We will describe a k-party protocol to find a clause of F that is violated under α. The k players first consider the configuration $\mathbb{D}_{L/2}$ in the refutation and communicate in order to evaluate the truth value of all lines in $\mathbb{D}_{L/2}$ under α. If all lines of $\mathbb{D}_{L/2}$ are true, they continue their search on the subderivation $\mathbb{D}_{L/2}, \ldots, \mathbb{D}_L$, and otherwise the search continues on the subderivation $\mathbb{D}_0, \ldots, \mathbb{D}_{L/2}$. In this way, we do a binary search, always maintaining the invariant that the first configuration in the subderivation evaluates to true, but some line in the last configuration evaluates to false. After $\log L$ steps, the players will find an $i \in [L]$ such that all of $\mathbb{D}_{i-1}$ evaluates to true but some line in $\mathbb{D}_i$ is false under α. By the soundness of the proof system, the false line in $\mathbb{D}_i$ must have been a download of a some clause of F and this clause solves the search problem.

Let us analyse the communication complexity of the protocol. The cost of evaluating any particular configuration with error at most $(4 \log L)^{-1} \leq (4s)^{-1}$ is $\mathit{Sp} \cdot \operatorname{polylog} s$. Thus the overall cost is $\mathit{Sp} \cdot \log L \cdot \operatorname{polylog} s$ and the total error is at most $1/4$. □

Huynh and Nordström proceeded to construct formulas Peb_G of size s such that they admit Resolution refutations of size $O(s)$, but for which any $\mathsf{T}^{\mathsf{cc}}(2)$ refutation in space Sp and length L must satisfy $\mathit{Sp} \cdot \log L = s^{1/4-o(1)}$. Using our multi-party lower bounds, we can now generalise this tradeoff result to $\mathsf{T}^{\mathsf{cc}}(k)$ proof systems. Namely, we prove the following result, which was stated in the introduction.

Theorem 7.13 (Length–space lower bounds). *There are CNF formulas F of size s such that*

- *F admits a Resolution refutation of length $L = s^{1+o(1)}$ and space $\mathit{Sp} = s^{1/2+o(1)}$.*
- *Any length L and space Sp refutation of F in $\mathsf{T}^{\mathsf{cc}}(k)$ must satisfy*

$$\mathit{Sp} \cdot \log L \;\geq\; \begin{cases} s^{1/2-o(1)}, & \text{for } k = 2, \\ s^{1/4-o(1)}, & \text{for } 3 \leq k \leq (\log s)^{1-o(1)}. \end{cases} \tag{7.1}$$

Proof. The formula family, parameterised by $n \in \mathbb{N}$, is

$$\mathit{Peb}_G \circ g_k^n,$$

where G is the graph from Theorem 7.9 with n nodes and maximum degree $d = O(1)$, and where $k = k(n)$ is a parameter, and where $g_k \colon \mathcal{X}^k \to \{0,1\}$ is again our gadget from Corollary 7.20. In particular, letting $l = \log|\mathcal{X}|$, these formulas have size

$$s \leq \Theta(n) \cdot 2^{dkl}.$$

Lower bound. Using $\operatorname{cbs}(S(\mathit{Peb}_G)) = \Omega(n^{1/2})$ and an argument similar to the proof of Theorem 7.12, we conclude that $S(\mathit{Peb}_G \circ g_k^n)$ has k-party randomised communication complexity $\Omega(n^{1/4-o(1)})$ when we choose $k = (\log n)^{1-o(1)}$ appropriately. (Alternatively, the complexity is $\Omega(n^{1/2-o(1)})$ for $k = 2$.) Recall also that with this choice of k, we have $s = n^{1+o(1)}$. This proves the lower bound (7.1) in view of Lemma 7.33.

Upper bound (sketch). To see that the lifted formula $Peb_G \circ g_k^n$ has a Resolution refutation of length $s^{1+o(1)}$ and space $s^{1/2+o(1)}$, we will mimic the usual length-$O(n)$ space-$O(n^{1/2})$ refutation of the original formula Peb_G. This refutation follows the pebbling of G: whenever a node v, with in-neighbours $w_1, \dots, w_d$, is pebbled, we derive the clause (v) from previously derived clauses $(w_1), \dots, (w_d)$ and the clause $(\neg w_1 \vee \cdots \vee \neg w_d \vee v)$ of Peb_G.

For the lifted version $Peb_G \circ g_k^n$ we want to do the same thing, deriving the lifted clauses associated with (v) from the lifted clauses associated with $(w_1), \dots, (w_d)$ and $(\neg w_1 \vee \cdots \vee \neg w_d \vee v)$. The number of lifted variables that underlie each pebbling step is dkl, and since there is always a Resolution refutation of size exponential in the number of variables, it follows that each resolution step in the original refutation of Peb_G can be simulated by $O(2^{dkl}) = s^{o(1)}$ steps in the lifted proof. Thus the total length of the lifted refutation is $O(n) \cdot s^{o(1)} = s^{1+o(1)}$. Similarly, the space used is $s^{1/2+o(1)}$. □

Chapter 8

Extension Complexity of Independent Set Polytopes

Overview. In this chapter, we exhibit an n-node graph whose independent set polytope requires extended formulations of size exponential in $\Omega(n/\log n)$. Previously, no explicit examples of n-dimensional 0/1-polytopes were known with extension complexity larger than exponential in $\Theta(\sqrt{n})$. Our construction is inspired by a relatively little-known connection between extended formulations and (monotone) circuit depth. This chapter is based on the following publication:

[GJW16]: Mika Göös, Rahul Jain, and Thomas Watson. Extension complexity of independent set polytopes. In *Proceedings of the 57th Symposium on Foundations of Computer Science (FOCS)*, 2016. To appear. URL: http://eccc.hpi-web.de/report/2016/070/

8.1 Introduction

A polytope $P \subseteq \mathbb{R}^n$ with many facets can sometimes admit a concise description as the projection of a higher dimensional polytope $E \subseteq \mathbb{R}^e$ with few facets. This phenomenon is studied in the theory of "extended formulations". The *extension complexity* $\mathrm{xc}(P)$ of a polytope P is defined as the minimum number of facets in any E (called an *extended formulation* for P) such that

$$P \;=\; \{x \in \mathbb{R}^n : (x, y) \in E \text{ for some } y\}.$$

Extended formulations are useful for solving combinatorial optimization problems: instead of optimizing a linear function over P, we can optimize it over E—this may be more efficient since the runtime of LP solvers often depends on the number of facets.

Fiorini et al. [FMP$^+$15] were the first to show (using methods from communication complexity [KN97, Juk12]) exponential extension complexity lower bounds for many explicit polytopes of relevance to combinatorial optimization, thereby solving an old challenge set by Yannakakis [Yan91]. For example, their results include a $2^{\Omega(m)}$ lower bound for the $\binom{m}{2}$-dimensional *correlation/cut polytope*. In another breakthrough, Rothvoß [Rot14] proved a much-conjectured $2^{\Omega(m)}$ lower bound for the $\binom{m}{2}$-dimensional *matching polytope*. By now, many accessible introductions to extended formulations are available; e.g., Roughgarden [Rou15, §5], Kaibel [Kai11], Conforty et al. [CCZ10] or their textbook [CCZ14, §4.10].

$\sqrt{n}$-frontier. Both of the results quoted above—while optimal for their respective polytopes—seem to get "stuck" at being exponential in the square root of their dimension. In fact, no explicit n-dimensional 0/1-polytope (convex hull of a subset of $\{0,1\}^n$) was known with extension complexity asymptotically larger than $2^{\Theta(\sqrt{n})}$. In comparison, Rothvoß [Rot12] showed via a counting argument that most n-dimensional 0/1-polytopes have extension complexity $2^{\Omega(n)}$.

8.1.1 Our result

Our main result is to construct an explicit 0/1-polytope of near-maximal extension complexity $2^{\Omega(n/\log n)}$. Moreover, the polytope can be taken to be the *independent set polytope* P_G of an n-node graph G, i.e., the convex hull of (the indicator vectors of) the independent sets of G. Previously, a lower bound of $2^{\Omega(\sqrt{n})}$ was known for independent set polytopes [FMP⁺15].

Theorem 8.1. *There is an (explicit) family of n-node graphs G with* $\mathrm{xc}(P_G) \geq 2^{\Omega(n/\log n)}$.

In fact, our graph family has bounded degree. Hence, using known reductions, we get as a corollary quantitative improvements—from $2^{\Omega(\sqrt{n})}$ to $2^{\Omega(n/\log n)}$—for the extension complexity of, for instance, *3SAT* and *knapsack polytopes*; see [AT14, PV13] for details.

We strongly conjecture that our graph family actually satisfies $\mathrm{xc}(P_G) \geq 2^{\Omega(n)}$, i.e., that the $\log n$ factor in the exponent is an artefact of our proof technique. We give concrete evidence for this by proving an optimal bound for a certain *query complexity* analogue of Theorem 8.1. In particular, the conjectured bound $\mathrm{xc}(P_G) \geq 2^{\Omega(n)}$ would follow from quantitative improvements to the known communication-to-query simulation theorems (Chapter 2 in particular). Incidentally, this also answers a question of Lovász, Naor, Newman, and Wigderson [LNNW95]: we obtain a maximal $\Omega(n)$ lower bound on the randomised query complexity of a search problem with constant certificate complexity.

8.1.2 Our approach

Curiously enough, an analogous $\sqrt{n}$-frontier existed in the seemingly unrelated field of *monotone circuits*: Raz and Wigderson [RW92] proved an $\Omega(m)$ lower bound for the depth of any monotone circuit computing the *matching function* on $\binom{m}{2}$ input bits. This remained the largest monotone depth bound for an explicit function until our work of Chapter 7 appeared, where we exhibit a function with monotone depth $\Omega(n/\log n)$. In short, our idea is to prove an extension complexity analogue of this latter result.

The conceptual inspiration for our construction is a relatively little-known connection between Karchmer–Wigderson games [KW88] (which characterize circuit depth) and extended formulations. This "KW/EF connection" (see Section 8.2 for details) was pointed out by Hrubeš [Hru12] as a nonnegative analogue of a classic rank-based method of Razborov [Raz90]. In this chapter, we focus only on the monotone setting. For any monotone $f\colon \{0,1\}^n \to \{0,1\}$ we can study the convex hull of its 1-inputs, namely, the polytope

$$F \;\coloneqq\; \operatorname{conv} f^{-1}(1).$$

The upshot of the KW/EF connection is that extension complexity lower bounds for F follow from a certain type of *strengthening* of monotone depth lower bounds for f. For example, using this connection, it turns out that Rothvoß's result [Rot14] implies the result of Raz and Wigderson [RW92] in a simple black-box fashion (Section 8.2.3).

Our main technical result is to strengthen the existing monotone depth lower bound from Chapter 7 into a lower bound for the associated polytope (though we employ substantially different techniques than were used in that paper). The key communication search problem studied in Chapter 7 is a communication version of the well-known *Tseitin* problem, which has especially deep roots in proof complexity (e.g., [Juk12, §18.7]) and has also been studied in query complexity [LNNW95]. We use information complexity techniques to prove the required $\Omega(n/\log n)$ communication lower bound for the relevant variant of the Tseitin problem; information theoretic tools have been used in extension complexity several times [BM13, BP13, BP15]. One relevant work is Huynh and Nordström [HN12] (predecessor to Chapter 7), whose information complexity arguments we extend in this chapter.

(Instead of using information complexity, an alternative seemingly promising approach would be to "lift" a strong enough query complexity lower bound for Tseitin into communication complexity. Unfortunately, this approach runs into problems due to limitations in existing communication-to-query simulation theorems; we discuss this in Section 8.7.)

Theorem 8.1 follows by reductions from the result for Tseitin (Section 8.4). Indeed, it was known that the Tseitin problem reduces to the monotone KW game associated with an $f\colon \{0,1\}^{O(n)} \to \{0,1\}$ that encodes (in a monotone fashion) a certain CSP satisfiability problem. This gives us an extension complexity lower bound for the (explicit) polytope $F \coloneqq \operatorname{conv} f^{-1}(1)$. As a final step, we give a reduction from F to an independent set polytope.

8.1.3 Background

Let M be a nonnegative matrix. The *nonnegative rank* of M, denoted $\operatorname{rk}^+(M)$, is the minimum r such that M can be decomposed as a sum $\sum_{i\in[r]} R_i$ where each R_i is a rank-1 nonnegative matrix.

Randomized protocols. Faenza et al. [FFGT14] observed that a nonnegative rank decomposition can be naturally interpreted as a type of randomised protocol that computes the matrix M "in expectation". We phrase this connection precisely as follows: $\log \operatorname{rk}^+(M) + \Theta(1)$ is the minimum communication cost of a private-coin protocol Π whose acceptance probability on each input (x, y) satisfies $\mathbf{Pr}[\Pi(x, y) \text{ accepts}] = \alpha \cdot M_{x,y}$ where $\alpha > 0$ is an absolute constant of proportionality (depending on Π but not on x, y). All communication protocols in this paper are private-coin.

Slack matrices. The extension complexity of a polytope $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ can be characterized in terms of the nonnegative rank of the *slack matrix* $M = M(P)$ associated with P. The entries of M are indexed by (v, i) where $v \in P$ is a vertex of P and i refers to the i-th facet-defining inequality $A_i x \geq b_i$ for P. We define $M_{v,i} := A_i v - b_i \geq 0$ as the distance (*slack*) of the i-th inequality from being tight for v. Yannakakis [Yan91] showed that $\mathrm{xc}(P) = \mathrm{rk}^+(M(P))$.

A convenient fact for proving lower bounds on $\mathrm{rk}^+(M)$ is that the nonnegative rank is unaffected by the addition of columns to M that each record the slack between vertices of P and some valid (but not necessarily facet-defining) inequality for P. For notation, let $P \subseteq Q$ be two nested polytopes (in fact, Q can be an unbounded polyhedron). We define $M(P; Q)$ as the slack matrix whose rows correspond to vertices of P and columns correspond to the facets of Q (hence $M(P; P) = M(P)$). We have $\mathrm{rk}^+(M(P)) \geq \mathrm{rk}^+(M(P) \cup M(P; Q)) - 1 \geq \mathrm{rk}^+(M(P; Q)) - 1$ where "$\cup$" denotes concatenation of columns.[1] We summarize all the above in the following.

Fact 8.2. *For all polytopes $P \subseteq Q$, we have* $\mathrm{xc}(P) = \mathrm{rk}^+(M(P)) \geq \mathrm{rk}^+(M(P; Q)) - 1$.

8.2 KW/EF connection

We now describe the connection showing that EF lower bounds follow from a certain type of strengthening of lower bounds for monotone KW games (and similarly, lower bounds for monotone KW games follow from certain strong enough EF lower bounds). This is not directly used in the proof of Theorem 8.1, but it serves as inspiration by suggesting the approach we use in the proof.

8.2.1 Definitions

Let $f \colon \{0, 1\}^n \to \{0, 1\}$ be a monotone function. We define $\mathrm{KW}^+(f)$ as the deterministic communication complexity of the following *monotone KW game* associated with f.

KW$^+$-game

Input: Alice gets $x \in f^{-1}(1)$, and Bob gets $y \in f^{-1}(0)$.

Output: An index $i \in [n]$ such that $x_i = 1$ and $y_i = 0$.

[1] Specifically, Farkas's Lemma implies that the slack of any valid inequality for P can be written as a nonnegative linear combination of the slacks of the facet-defining inequalities for P, plus a nonnegative constant [Zie95, Proposition 1.9]. Thus if we take $M(P) \cup M(P; Q)$ and subtract off (possibly different) nonnegative constants from each of the "new" columns $M(P; Q)$, we get a matrix each of whose columns is a nonnegative linear combination of the "original" columns $M(P)$ and hence has the same nonnegative rank as $M(P)$. Since we subtracted off a nonnegative rank-1 matrix, we find that $\mathrm{rk}^+(M(P) \cup M(P; Q)) \leq \mathrm{rk}^+(M(P)) + 1$.

We often think of x and y as subsets of $[n]$. In this language, a feasible solution for the KW$^+$-game is an $i \in x \cap \bar{y}$ where $\bar{y} \coloneqq [n] \smallsetminus y$. Given a monotone f, we denote by $F \coloneqq \operatorname{conv} f^{-1}(1)$ the associated polytope. We can express the fact that any pair $(x, y) \in f^{-1}(1) \times f^{-1}(0)$ admits at least one witness $i \in x \cap \bar{y}$ via the following linear inequality:

$$\sum_{i\,:\,y_i=0} x_i \;\geq\; 1. \tag{8.1}$$

Since (8.1) is valid for all the vertices $x \in F$, it is valid for the whole polytope F. Define $F_{\text{KW}} \supseteq F$ as the polyhedron whose facets are determined by the inequalities (8.1), as indexed by 0-inputs y. The (x, y)-th entry in the slack matrix $M(F; F_{\text{KW}})$ is then $\sum_{i\,:\,y_i=0} x_i - 1$. In words, this quantity counts the number of witnesses in the KW$^+$-game on input (x, y) minus one.

More generally, let $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Q}$ be *any* communication search problem (not necessarily a KW$^+$-game, even though any S can be reformulated as such [Gál01, Lemma 2.3]). Here $\mathcal{Q}$ is some set of solutions/witnesses, and letting $S(x, y) \coloneqq \{q \in \mathcal{Q} : (x, y, q) \in S\}$ denote the set of feasible solutions for input (x, y), we assume that $S(x, y) \neq \emptyset$ for all (x, y). We associate with S the following natural *"number of witnesses minus one"* communication game.

$(\#\exists-1)$-game

Input: Alice gets $x \in \mathcal{X}$, and Bob gets $y \in \mathcal{Y}$.

Output: Accept with probability proportional to $|S(x, y)| - 1$.

The communication complexity of this game is simply $\log \operatorname{rk}^+(M^S) + \Theta(1)$ where $M^S_{x,y} \coloneqq |S(x, y)| - 1$.

8.2.2 The connection

What Hrubeš [Hru12, Proposition 4] observed was that an efficient protocol for a search problem S implies an efficient protocol for the associated $(\#\exists-1)$-game. In particular, for KW$^+$-games,

$$\log \operatorname{rk}^+(M(F; F_{\text{KW}})) \;\leq\; O(\text{KW}^+(f)). \tag{KW/EF}$$

The private-coin protocol for $M(F; F_{\text{KW}})$ computes as follows. On input $(x, y) \in f^{-1}(1) \times f^{-1}(0)$ we first run the optimal deterministic protocol for the KW$^+$-game for f to find a particular $i \in [n]$ witnessing $x_i = 1$ and $y_i = 0$. Then, Alice uses her private coins to sample a $j \in [n] \smallsetminus \{i\}$ uniformly at random, and sends this j to Bob. Finally, the two players check whether $x_j = 1$ and $y_j = 0$ accepting iff this is the case. The acceptance probability of this protocol is proportional to the number of witnesses minus one, and the protocol has cost $\text{KW}^+(f) + \log n + O(1) \leq O(\text{KW}^+(f))$ (where we assume w.l.o.g. that f depends on all of its input bits so that $\text{KW}^+(f) \geq \log n$).

8.2.3 Example: Matchings

Rothvoß vs. Raz–Wigderson. Consider the monotone function $f\colon \{0,1\}^{\binom{m}{2}} \to \{0,1\}$ that outputs 1 iff the input, interpreted as a graph on m nodes (m even), contains a perfect matching. Then $F \coloneqq \operatorname{conv} f^{-1}(1)$ is the perfect matching polytope. The inequalities (8.1) for f happen to include the so-called "odd set" inequalities, which were exploited by Rothvoß [Rot14] in showing that $\log \operatorname{rk}^+(M(F; F_{\text{KW}})) \geq \Omega(m)$. Applying the (KW/EF) connection to Rothvoß's lower bound implies in a black-box fashion that $\text{KW}^+(f) \geq \Omega(m)$, which is the result of Raz and Wigderson [RW92].

Converse to (KW/EF)*?* It is interesting to compare the above with the case of *bipartite* perfect matchings. Consider a monotone $f\colon \{0,1\}^{m\times m} \to \{0,1\}$ that takes a bipartite graph as input and outputs 1 iff the graph contains a perfect matching. It is well-known that $F \coloneqq \operatorname{conv} f^{-1}(1)$ admits a polynomial-size extended formulation [Sch03, Theorem 18.1]. By contrast, the lower bound $\text{KW}^+(f) \geq \Omega(m)$ from [RW92] continues to hold even in the bipartite case. This example shows that the converse inequality to (KW/EF) does not hold in general. Hence, a lower bound for the $(\#\exists-1)$-game can be a strictly stronger result than a similar lower bound for the KW^+-game.

8.2.4 Minterms and maxterms

A *minterm* $x \in f^{-1}(1)$ is a minimal 1-input in the sense that flipping any 1-entry of x into a 0 will result in a 0-input. Analogously, a *maxterm* $y \in f^{-1}(0)$ is a maximal 0-input. It is a basic fact that solving the KW^+-game for minterms/maxterms is enough to solve the search problem on any input: Say that Alice's input x is not a minterm. Then Alice can replace x with any minterm $x' \subseteq x$ and run the protocol on x'. A witness $i \in [n]$ for (x', y) works also for (x, y). A similar fact holds for the $(\#\exists-1)$-game: we claim that the nonnegative rank does not change by much when restricted to minterms/maxterms. Say that Alice's input x is not a minterm. Then Alice can write $x = x' \cup x''$ (disjoint union) where x' is a minterm. Then $|x \cap \bar{y}| - 1 = (|x' \cap \bar{y}| - 1) + |x'' \cap \bar{y}|$ where the first term is the $(\#\exists-1)$-game for (x', y) and the second term has nonnegative rank at most n. (A similar argument works if Bob does not have a maxterm.)

8.3 Tseitin problem

8.3.1 Query version

Fix a connected node-labeled graph $G = (V, E, \ell)$ where $\ell \in \mathbb{Z}_2^V$ has *odd weight*, i.e., $\sum_{v\in V} \ell(v) = 1$ where the addition is modulo 2. For any edge-labeling $z \in \mathbb{Z}_2^E$ and a node $v \in V$ we write concisely $z(v) \coloneqq \sum_{e\ni v} z(e)$ for the mod-2 sum of the edge-labels adjacent to v.

Tseitin problem: Tse_G

Input: Labeling $z \in \mathbb{Z}_2^E$ of the edges.

Output: A node $v \in V$ containing a *parity violation* $z(v) \neq \ell(v)$.

As a sanity check, we note that on each input z there must exist at least one node with a parity violation. This follows from the fact that, since each edge has two endpoints, the sum $\sum_v z(v)$ is even, whereas we assumed that the sum $\sum_v \ell(v)$ is odd.

Basic properties. The above argument implies more generally that the set of violations $\mathrm{viol}(z) \coloneqq \{v \in V : z(v) \neq \ell(v)\}$ is always of odd size. Conversely, for any odd-size set $S \subseteq V$ we can design an input z such that $\mathrm{viol}(z) = S$. To see this, it is useful to understand what happens when we *flip a path* in an input z. Formally, suppose $p \in \mathbb{Z}_2^E$ is (an indicator vector of) a path. Define z^p as z with bits on the path p flipped (note that $z^p = z + p \in \mathbb{Z}_2^E$; however, the notation z^p will be more convenient later). Flipping p has the effect of flipping whether each endpoint of p is a violation. More precisely, the violated nodes in z^p are related to those in z as follows: (i) if both endpoints of p are violated in z then the flip causes that pair of violations to disappear; (ii) if neither endpoint of p is violated in z, then the flip introduces a pair of new violations; (iii) if precisely one endpoint of p was violated in z, then the flip moves a violation from one endpoint of p to the other. By applying (i)–(iii) repeatedly in a connected graph G, we can design an input z where $\mathrm{viol}(z)$ equals any prescribed odd-size set S.

If z and z' have the same set of violations, $\mathrm{viol}(z) = \mathrm{viol}(z')$, then their difference $q \coloneqq z - z' \in \mathbb{Z}_2^E$ satisfies $q(v) = 0$ for all $v \in V$. That is, q is an *eulerian* subgraph of G. On the other hand, for any eulerian graph q, the inputs z and z^q have the same violations. Consequently, to generate a random input with the same set of violations as some fixed z, we need only pick a random eulerian graph q and output z^q. (Eulerian graphs form a subspace of $\mathbb{Z}_2^E$, sometimes called the *cycle space* of G.)

8.3.2 Communication version

The communication version of the Tseitin problem is obtained by composing (or *lifting*) Tse_G with a constant-size two-party gadget $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. In the lifted problem $\mathsf{Tse}_G \circ g^n$, where $n \coloneqq |E|$, Alice gets $x \in \mathcal{X}^n$ as input, Bob gets $y \in \mathcal{Y}^n$ as input, and their goal is to find a node $v \in V$ that is violated for

$$z \coloneqq g^n(x,y) = (g(x_1,y_1),\ldots,g(x_n,y_n)).$$

We define our gadget precisely in Section 8.5. For now—in particular, for the reductions presented in the next section—the only important property of our gadget is that $|\mathcal{X}|, |\mathcal{Y}| \leq O(1)$.

8.3.3 Statement of result

We prove that there is a family of bounded-degree graphs G such that the $(\#\exists-1)$-game associated with $\mathsf{Tse}_G \circ g^n$ requires $\Omega(n/\log n)$ bits of communication. We prove our lower bound assuming only that $G = (V, E)$ is well-connected enough as captured by the following definition (also used in Chapter 7). A graph G is *k-routable* iff there is a set of $2k+1$ nodes $T \subseteq V$ called *terminals* such that for any *pairing* $\mathcal{P} := \{\{s_i, t_i\} : i \in [\kappa]\}$ (set of pairwise disjoint pairs) of 2κ terminals ($\kappa \leq k$), there exist κ edge-disjoint paths (called *canonical* paths for $\mathcal{P}$) such that the i-th path connects s_i to t_i. Furthermore, we tacitly equip G with an arbitrary odd-weight node-labeling.

Theorem 8.3. *There is a constant-size g such that for every k-routable graph G with n edges, the $(\#\exists-1)$-game for $\mathsf{Tse}_G \circ g^n$ requires $\Omega(k)$ bits of communication.*

If we choose G to be a sufficiently strong expander graph, we may take $k = \Theta(n/\log n)$ as shown by Frieze et al. [FZ00, Fri01]. Alternative constructions with $k = \Theta(n/\log n)$ exist based on bounded-degree "butterfly" graphs; see [Nor15, §5] for an exposition.

Corollary 8.4. *There is a constant-size g and an explicit bounded-degree graph G with n edges such that the $(\#\exists-1)$-game for $\mathsf{Tse}_G \circ g^n$ requires $\Omega(n/\log n)$ bits of communication.*

As a bonus, we also prove that the *query complexity* of the $(\#\exists-1)$-game for Tse_G is $\Omega(n)$ on any expander G (see Section 8.7).

8.4 Reductions

The goal of this section is to show, via reductions, that a lower bound on the $(\#\exists-1)$-game for $\mathsf{Tse}_G \circ g^n$ (where $G = (V, E)$ is of bounded degree and $n := |E|$) translates directly into a lower bound on the extension complexity of P_K for an $O(n)$-node bounded-degree graph K.

8.4.1 Definition: Monotone CSP-SAT

We start by describing a way of representing constraint satisfaction problems (CSP) as a monotone function; this was introduced in Chapter 7 and further studied by Oliveira [Oli15, Chapter 3]. The function is defined relative to some finite alphabet Σ and a fixed constraint topology determined by a bipartite graph $H := (L \cup R, E)$. The left nodes L are thought of as *variables* (taking values in Σ) and the right nodes R correspond to *constraints*. For a constraint $c \in R$, let $\mathrm{var}(c) \subseteq L$ denote the variables involved in c. Let d denote the maximum degree of a node in R. The function $\mathsf{SAT} = \mathsf{SAT}_{\Sigma,H} \colon \{0,1\}^m \to \{0,1\}$, where $m \leq |R| \cdot |\Sigma|^d$, is now defined as follows. An input $x \in \{0,1\}^m$ defines a CSP instance by specifying, for each $c \in R$, a truth table $\Sigma^{\mathrm{var}(c)} \to \{0,1\}$ that records which assignments to the variables $\mathrm{var}(c)$ satisfy c. Then $\mathsf{SAT}(x) := 1$ iff there is some global assignment $L \to \Sigma$ that satisfies all the constraints as specified by x. This is monotone: if we flip any 0 into a 1 in the truth table of a constraint, we are only making the constraint easier to satisfy.

8.4.2 From Tseitin to CSP-SAT

For the readers convenience, we recount the reduction (from Section 7.5.1) from the search problem $\mathsf{Tse}_G \circ g^n$ to the KW$^+$-game for $\mathsf{SAT} = \mathsf{SAT}_{\mathcal{X},H} \colon \{0,1\}^m \to \{0,1\}$. Here the alphabet is $\mathcal{X}$ and the bipartite graph H is defined on $E(G) \cup V(G)$ such that there is an edge $(e,v) \in E(H)$ iff $v \in e$. Note that $m \leq O(n)$ provided that $|\mathcal{X}| \leq O(1)$ and that G is of bounded degree.

On input (x,y) to $\mathsf{Tse}_G \circ g^n$ the two players proceed as follows:

- Alice maps her $x \in \mathcal{X}^{E(G)}$ into a CSP whose sole satisfying assignment is x. Namely, for each constraint $v \in V(G)$, the truth table $\mathcal{X}^{\mathrm{var}(v)} \to \{0,1\}$ is all-0 except for a unique 1 in position $x|_{\mathrm{var}(v)}$ (restriction of x to coordinates in $\mathrm{var}(v)$).
- Bob maps his $y \in \mathcal{Y}^{E(G)}$ into an unsatisfiable CSP. Namely, for each constraint $v \in V(G)$, the truth table $t_v \colon \mathcal{X}^{\mathrm{var}(v)} \to \{0,1\}$ is given by $t_v(\hat{x}) \coloneqq 1$ iff $(g(\hat{x}_e, y_e))_{e \in \mathrm{var}(v)} \in \{0,1\}^{\mathrm{var}(v)}$ is a partial edge-labeling of G that does *not* create a parity violation on v.

Let us explain why Bob really produces a 0-input of SAT. Suppose for contradiction that there is an $\hat{x} \in \mathcal{X}^{E(G)}$ that satisfies all of Bob's constraints: $t_v(\hat{x}|_{\mathrm{var}(v)}) = 1$ for all v. By definition, this means that $z \coloneqq g^n(\hat{x}, y)$ is an input to Tse_G without any violated nodes—a contradiction.

This reduction is parsimonious: it maps witnesses to witnesses in 1-to-1 fashion. Indeed, a node v is violated for $\mathsf{Tse}_G \circ g^n$ if and only if Alice's truth table for v has its unique 1 in a coordinate where Bob has a 0. In conclusion, the $(\#\exists{-}1)$-game associated with (the KW$^+$-game for) SAT is at least as hard as the $(\#\exists{-}1)$-game for $\mathsf{Tse}_G \circ g^n$.

8.4.3 From CSP-SAT to independent sets

As a final step, we start with $\mathsf{SAT} = \mathsf{SAT}_{\Sigma,H} \colon \{0,1\}^m \to \{0,1\}$ and construct an m-node graph K such that a slack matrix of the independent set polytope P_K embeds the $(\#\exists{-}1)$-game for SAT (restricted to minterms). Let $H \coloneqq (L \cup R, E)$ (as above) and define $n \coloneqq |R|$ (above we had $n = |L|$, but in our case $|L| = \Theta(|R|)$ anyway).

The m-node graph K is defined as follows (this is reminiscent of a reduction from [FGL$^+$96]).

- The nodes of K are in 1-to-1 correspondence with the input bits of SAT. That is, for each constraint $c \in R$ we have $|\Sigma^{\mathrm{var}(c)}|$ many nodes in K labeled with assignments $\mathrm{var}(c) \to \Sigma$.
- There is an edge between any two nodes whose assignments are *inconsistent* with one another. (Here $\phi_i \colon \mathrm{var}(c_i) \to \Sigma$, $i \in \{1,2\}$, are inconsistent iff there is some $e \in \mathrm{var}(c_1) \cap \mathrm{var}(c_2)$ such that $\phi_1(e) \neq \phi_2(e)$.) In particular, the truth table of each constraint becomes a clique.

(It can be seen that K has bounded degree if H has bounded left- and right-degree, which it does after our reduction from Tseitin for a bounded-degree G.)

The key property of this construction is the following:

The minterms of SAT *are precisely the (indicator vectors of) maximal independent sets of* K*.*

Indeed, the minterms $x \in \mathsf{SAT}^{-1}(1)$ correspond to CSPs with a unique satisfying assignment $\phi\colon L \to \Sigma$; there is a single 1-entry in each of the n truth tables (so that $|x| = n$) consistent with ϕ. Such an x, interpreted as a subset of nodes, is independent in K as it only contains nodes whose labels are consistent with ϕ. Conversely, because every independent set $x \subseteq V(K)$ can only contain pairwise consistently labeled nodes, x naturally defines a partial assignment $L' \to \Sigma$ for some $L' \subseteq L$. A maximal independent set x corresponds to picking a node from each of the n constraint cliques consistent with some total assignment $\phi\colon L \to \Sigma$. Hence x is a 1-input to SAT with unique satisfying assignment ϕ.

Our goal is now to exhibit a set of valid inequalities for the independent set polytope P_K whose associated slack matrix embeds the $(\#\exists-1)$-game for SAT. Let $x \subseteq V(K)$ be an independent set and $y \in \mathsf{SAT}^{-1}(0)$. We claim that the following inequalities (indexed by y) are valid:

$$|x \cap y| \;=\; \sum_{i\,:\,y_i=1} x_i \;\leq\; n-1. \tag{8.2}$$

Clearly (8.2) holds whenever $|x| \leq n-1$. Since it is impossible to have $|x| \geq n+1$, assume that x is maximal: $|x| = n$. As argued above, x is a minterm of SAT. Hence (x, y) is a valid pair of inputs to the KW$^+$-game, and so they admit a witness: $|x \cap \bar{y}| \geq 1$. Therefore $|x \cap y| = n - |x \cap \bar{y}| \leq n-1$. This shows that (8.2) is valid. The slack matrix associated with inequalities (8.2) has entries

$$n - 1 - |x \cap y| \;=\; |x \cap \bar{y}| - 1,$$

for any minterm x and any $y \in \mathsf{SAT}^{-1}(0)$. But this is just the $(\#\exists–1)$-game for SAT with Alice's input restricted to minterms.

8.4.4 Proof of Theorem 8.1

Here we simply string the above reductions together. By Corollary 8.4 there is a constant-size g and a bounded-degree G with n edges such that the $(\#\exists-1)$-game for $\mathsf{Tse}_G \circ g^n$ requires $\Omega(n/\log n)$ bits of communication. By the reduction of Section 8.4.2 this implies an $\Omega(n/\log n)$ lower bound for the $(\#\exists-1)$-game associated with (the KW$^+$-game for) a monotone function $\mathsf{SAT}\colon \{0,1\}^{O(n)} \to \{0,1\}$. As discussed in Section 8.2.4, the complexity of the $(\#\exists-1)$-game for SAT is affected only by $\pm \log n$ when restricted to minterms. Thus the minterm-restricted $(\#\exists–1)$-game for SAT still has complexity $\Omega(n/\log n)$. (Alternatively, one can note that the reduction from Tseitin to CSP-SAT produced only minterms.) Hence the nonnegative rank of the matrix for that game is $2^{\Omega(n/\log n)}$. By the reduction of Section 8.4.3 there is a bounded-degree $O(n)$-node graph K and a system of valid inequalities (8.2) for the independent set polytope P_K such that the slack matrix $M(P_K; Q)$, where Q is the polyhedron with facets determined by (8.2), embeds the matrix for the minterm-restricted $(\#\exists–1)$-game for SAT. Thus $\log \mathrm{rk}^+(M(P_K; Q)) \geq \Omega(n/\log n)$. By Fact 8.2 we have $\log \mathrm{xc}(P_K) = \log \mathrm{rk}^+(M(P_K)) \geq \log\big(\mathrm{rk}^+(M(P_K; Q)) - 1\big) \geq \Omega(n/\log n)$.

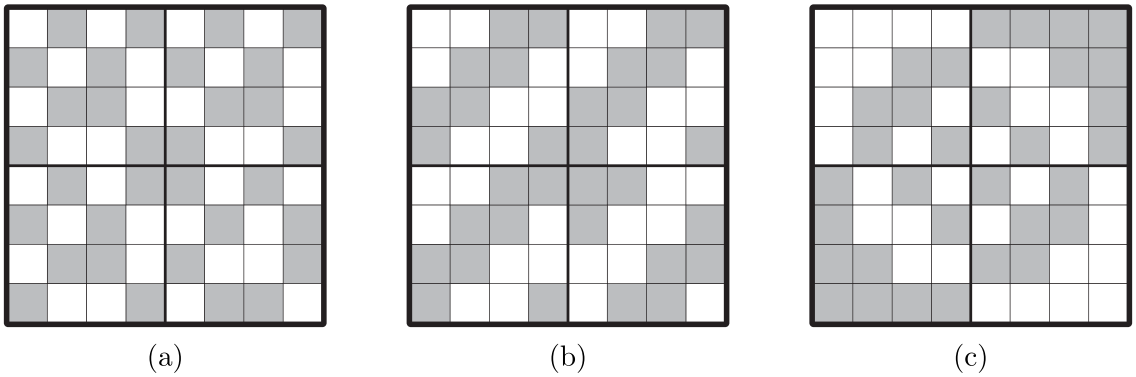


Figure 8.1: Three ways to view our gadget $g\colon \mathcal{X}\times\mathcal{Y}\to\{0,1\}$ by permuting rows and columns. The white and gray cells represent 0- and 1-inputs, respectively.

8.5 Our gadget

We define our two-party gadget $g\colon \{0,1\}^3\times\{0,1\}^3\to\{0,1\}$ as follows; see Figure 8.1:

$$g(x,y) \;\coloneqq\; x_1+y_1+x_2y_2+x_3y_3 \pmod 2.$$

We note that the smaller gadget $x_1+y_1+x_2y_2 \pmod 2$ was considered in Chapter 7.

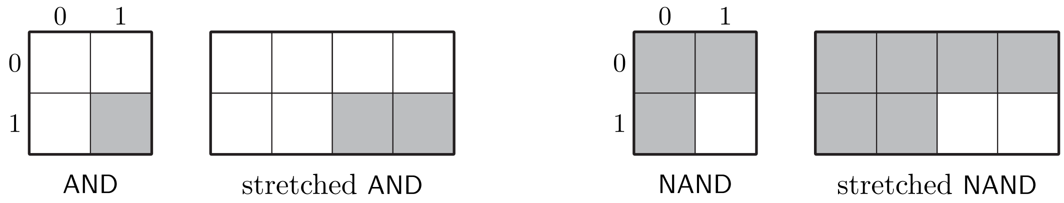
8.5.1 Flips and windows

The most basic property of g is that it admits *Alice/Bob-flips*:

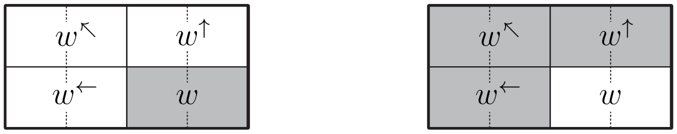
(1) *Alice-flips:* There is a row permutation $\pi_{\mathrm{A}}\colon \mathcal{X}\to\mathcal{X}$ that flips the output of the gadget: $g(\pi_{\mathrm{A}}(x),y)=\neg g(x,y)$ for all x,y. Namely, Alice just flips the value of x_1.

(2) *Bob-flips:* There is a column permutation $\pi_{\mathrm{B}}\colon \mathcal{Y}\to\mathcal{Y}$ that flips the output of the gadget: $g(x,\pi_{\mathrm{B}}(y))=\neg g(x,y)$ for all x,y. Namely, Bob just flips the value of y_1.

A more interesting feature of our gadget (which $x_1+y_1+x_2y_2$ does not possess) is that g embeds—in an especially uniform manner—certain 2×4 and 4×2 submatrices which we call "stretched AND" and "stretched NAND". For terminology, we define a *z-window* where $z\in\{0,1\}$ as a z-monochromatic rectangle of size 2 in the domain of g, i.e., an all-z submatrix of either horizontal shape 1×2 or vertical shape 2×1. Here is an illustration of *horizontally* stretched AND/NAND, which are composed of four horizontally shaped windows (for *vertical* stretch, the illustration should be transposed):



The key property is that each z-window w is embedded as the stretched $(1,1)$-input to a *unique* embedding of stretched AND (if $z=1$) or NAND (if $z=0$) inside g. That is, for each w we can find the following unique submatrix (illustrated again for horizontal shapes), where we denote by $w^{\leftarrow}$, $w^{\nwarrow}$, and $w^{\uparrow}$ the $(1-z)$-windows corresponding to the stretched $(1,0)$-, $(0,0)$-, and $(0,1)$-inputs to the stretched AND/NAND.



if w is a 1-window

if w is a 0-window

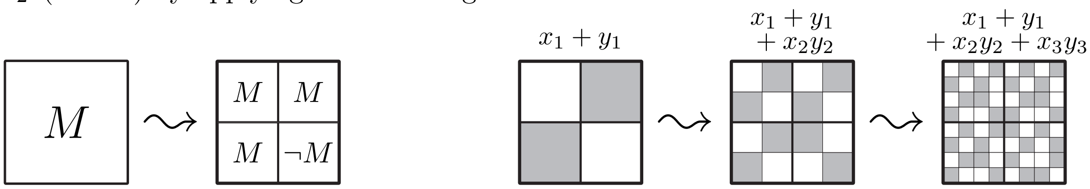
This defines three maps ("*directed flips*") $w \mapsto w^{\leftarrow}$, $w \mapsto w^{\nwarrow}$, $w \mapsto w^{\uparrow}$, which turn out to be shape-maintaining *bijections* between the set of z-windows and the set of $(1-z)$-windows. In particular, if w is a uniformly random z-window of g, then each of $w^{\leftarrow}$, $w^{\nwarrow}$, $w^{\uparrow}$ is a uniformly random $(1-z)$-window.

8.5.2 Checking the existence of flips

The properties of g claimed above can be verified by directly inspecting the gadget (by hand). Luckily, this task can be eased by exploiting symmetries.

(3) *Transitive symmetry*: The gadget admits a group of symmetries (permutations of its rows and columns leaving g invariant) which splits the domain of g into two orbits, $g^{-1}(1)$ and $g^{-1}(0)$. Specifically, there is a group $\mathcal{S} \subseteq \mathfrak{S}_8 \times \mathfrak{S}_8$ (here $\mathfrak{S}_8$ is the symmetric group on 8 elements) such that when $(\pi_1,\pi_2) \in \mathcal{S}$ acts on g, the output remains invariant: $g(\pi_1(x),\pi_2(y)) = g(x,y)$ for all x, y; and moreover, $\mathcal{S}$ is transitive in the sense that for any two 1-inputs $(x,y),(x',y') \in g^{-1}(1)$ (or 0-inputs) there is a symmetry $(\pi_1,\pi_2) \in \mathcal{S}$ such that $(\pi_1(x),\pi_2(y)) = (x',y')$.

To see that g really does have property (3), we visualize g as constructed from $\mathsf{XOR}(x_1,x_2) \coloneqq x_1 + x_2 \pmod 2$ by applying the following "$\rightsquigarrow$" transformation twice:



It is easy to see that XOR has the properties (1)–(3). We argue that if M is a boolean matrix with the properties (1)–(3) and $M \rightsquigarrow M'$, then M' has the properties (1)–(3). Suppose the entries of M are indexed by (x,y); we use (xa,yb) to index the entries of M' where $a,b \in \{0,1\}$

are bits. If π_{A}, π_{B} are the Alice/Bob-flips for M, then Alice/Bob-flips for M' are

$$\begin{aligned} xa &\mapsto \pi_{\mathrm{A}}(x)a, \\ yb &\mapsto \pi_{\mathrm{B}}(y)b. \end{aligned}$$

Suppose $\mathcal{S}$ is the transitive symmetry group for M. Then the transitive symmetry group for M' is generated by the following symmetries (here $\pi_{\mathrm{A}}^0(x) := x$ and $\pi_{\mathrm{A}}^1(x) := \pi_{\mathrm{A}}(x)$ and similarly for π_{B}^b):

$$\begin{aligned} \forall(\pi_1,\pi_2)\in\mathcal{S}: \quad (xa,yb) &\mapsto (\pi_1(x)a,\pi_2(y)b), \\ (xa,yb) &\mapsto (\pi_{\mathrm{A}}^a(x)a, y(1-b)), \\ (xa,yb) &\mapsto (x(1-a),\pi_{\mathrm{B}}^b(y)b). \end{aligned}$$

The first family of symmetries makes each quadrant of M' transitive, whereas the last two symmetries map entries between quadrants. In the second-to-last symmetry, Bob swaps the left and right halves while Alice applies her flip to the bottom half. In the last symmetry, Alice swaps the top and bottom halves while Bob applies his flip to the right half. This shows that g satisfies (1)–(3).

Rather than checking that each z-window w appears as the stretched $(1,1)$-input to a unique embedding of stretched AND/NAND and that the directed flips are bijections, it is equivalent to check that for all $\ell\in\{(0,0),(0,1),(1,0),(1,1)\}$ each w appears as the stretched ℓ-input to a unique embedding of stretched AND/NAND in g. Let us check this assuming w is a 0-window of shape 1×2 (the other possibilities can be checked similarly). By transitive symmetry, we may assume that w is picked among the four 0's of the first row of Figure 8.1(c) (so $\binom{4}{2}$ choices for w). The key observation is that the four columns corresponding to these 0's define a submatrix of g (left half of (c)) that contains each even Hamming weight row once, and that the other four columns (right half of (c)) also contain each even Hamming weight row once. We consider the four cases for ℓ.

$\ell=(0,0)$: To see that w is the stretched $(0,0)$-input to a unique embedding of stretched AND, find the unique other row that has 0's in the same columns as w. The other two columns in the left half of (c) have 0's in the top row and 1's in the other row.

$\ell=(0,1)$: To see that w is the stretched $(0,1)$-input to a unique embedding of stretched AND, find the unique other row that has 1's in the same columns as w and 0's in the other two columns of the left half of (c). These other two columns have 0's in the top row.

$\ell=(1,0)$: To see that w is the stretched $(1,0)$-input to a unique embedding of stretched AND, find the unique other row that has 0's in the same columns as w, then find the unique pair of columns in the right half of (c) that has 0's in that other row. This pair of columns has 1's in the first row.

$\ell=(1,1)$: To see that w is the stretched $(1,1)$-input to a unique embedding of stretched NAND, find the unique other row that has 1's in the same columns as w and 0's in the other two columns of the left half of (c), then find the unique pair of columns in the right half of (c) that has 1's in that other row. This pair of columns has 1's in the first row.

8.6 Communication lower bound

In this section we prove Theorem 8.3, where g is the gadget from Section 8.5.

8.6.1 High-level intuition

The high-level reason for why the $(\#\exists-1)$-game for Tseitin (or really for any sufficiently unstructured search problem) is hard is the same as for the $(\#\exists-1)$-game for matching [Rot14]: A correct protocol Π dare not accept its input before it has found at least two witnesses, lest it risk accepting with positive probability an input with a unique witness (which would contradict correctness). However, in an input with i witnesses, there are $\binom{i}{2}$ pairs of witnesses for the protocol to find. Hence one expects the acceptance probability of Π (that communicates too few bits and never errs when $i = 1$) to grow at least *quadratically* with i rather than linearly as required by the $(\#\exists-1)$-game.

Formalizing this quadratic increase in acceptance probability for protocols takes some technical work given the current tools available in communication complexity. However, the quadratic increase phenomenon for Tseitin is easier to formalize in the query complexity setting, which we do in Section 8.7. The reader may want to have a look at that simpler proof first, even though the query proof is somewhat incomparable to our approach for protocols (which revolves around k-routability).

8.6.2 Preliminaries

Probability and information theory. We use standard notions from information theory: $\mathbf{H}(X)$ is Shannon entropy; $\mathbf{H}(X \mid Y) \coloneqq \mathbf{E}_{y\sim Y}\, \mathbf{H}(X \mid Y = y)$ is conditional entropy; $\mathbf{I}(X\,; Y) \coloneqq \mathbf{H}(X) - \mathbf{H}(X \mid Y) = \mathbf{H}(Y) - \mathbf{H}(Y \mid X)$ is mutual information; $\Delta(X, Y)$ is statistical (total variation) distance. We use upper-case letters for random variables and corresponding lower-case letters for particular outcomes. Throughout the whole proof, all random choices are assumed to be uniform in their respective domains unless otherwise stated.

Inputs and transcripts. Let XY be random inputs to a private-coin protocol Π. We denote by $\Pi = \Pi(X, Y)$ the transcript of the protocol on input XY, and we let $|\Pi|$ be the maximum length of a transcript (i.e., the communication cost of Π). Note that the transcript Π depends on both XY and the private coins of the players. We let $\Pi^{\text{acc}} \coloneqq (\Pi \mid \Pi \text{ accepts})$ denote the transcript conditioned on the protocol accepting. For each input $z \in \mathbb{Z}_2^n$ to the query problem Tse_G we can associate in a natural way a pair of random inputs XY to the communication problem $\mathsf{Tse}_G \circ g^n$ that are *consistent with* z in the sense that $g^n(X, Y) = z$; namely, we let XY be uniformly distributed on

$$(g^n)^{-1}(z) \ = \ g^{-1}(z_1) \times \cdots \times g^{-1}(z_n).$$

We write $\Pi|z$ as a shorthand for $\Pi(X, Y)$ where XY are drawn at random from the above set.

Windows. As is often the case with information complexity arguments, we need to introduce a conditioning variable W whose purpose is to make X and Y conditionally independent. To this end, we employ windows (Section 8.5.1): we call a rectangle $w \coloneqq w_1 \times \cdots \times w_n \subseteq (g^n)^{-1}(z)$ a (multi-gadget) *window of* z iff each w_i is a z_i-window in g (so $w_i \subseteq g^{-1}(z_i)$). Now, to generate XY as above, we first pick W uniformly at random among all the windows of z, and then, conditioned on an outcome $W = w$, we pick $XY \in w$ uniformly at random. In conclusion, XY is uniform on $(g^n)^{-1}(z)$ (since each row and column of g is balanced) and X and Y are conditionally independent given W. We write $\Pi|w \coloneqq (\Pi(X, Y) \,|\, W = w)$ for short.

Alice-flips. Let (x, y) be an input consistent with $z \coloneqq g^n(x, y)$ and let $B \subseteq [n]$ be any subset of coordinates of z. (B stands for "block" by analogy with the concept of block sensitivity from query complexity.) We denote by (x^B, y) the input obtained from (x, y) by letting Alice flip the outputs of all gadgets corresponding to coordinates in B, i.e., for every $i \in B$ Alice replaces her input x_i with $\pi_{\mathrm{A}}(x_i)$ where π_{A} is the row permutation from Section 8.5.1. Hence (x^B, y) is an input consistent with z^B. We can also have Alice flip whole windows: $w^B \coloneqq \{(x^B, y) : (x, y) \in w\}$. We henceforth refer to such Alice-flips as just "flips". (We could equally well have Bob be the flipper throughout the whole proof, but we needed to make an arbitrary choice between the players.)

Smooth protocols. Recall that if z is an input to Tse_G and $B \subseteq E(G)$ is an eulerian graph, then z and z^B have the same set of violations. Consequently, any protocol Π for the ($\#\exists$–1)-game must accept inputs (x, y) and (x^B, y) with the same probability. We note that we may assume w.l.o.g. that the transcript distribution of Π is not sensitive to flipping eulerian graphs: if w is a window and B an eulerian graph, then $\Pi|w$ and $\Pi|w^B$ have the same distribution. Indeed, if Π does not satisfy this, then we may replace it by a new "smoothed" protocol Π' that computes as follows on input (x, y): Alice uses her private coins to choose a uniformly random eulerian graph B and then the players run Π on input (x^B, y). The fact that we may assume Π is smooth is critically used later in the proof.

8.6.3 Proof outline

Let us assume for the sake of contradiction that Π is a private-coin protocol of cost $|\Pi| \leq o(k)$ that accepts each input (x, y) with probability $\alpha \cdot (|\mathrm{viol}(z)| - 1)$ where $\alpha > 0$ is a constant (independent of (x, y)) and $z \coloneqq g^n(x, y)$. We call an input z (and any (x, y) consistent with z)

an *i-violation input* if $|\mathrm{viol}(z)| = i$ and all violations occur at the terminals T. We analyze the behavior of Π on i-violation inputs with $i \in \{1, 3, 7\}$ and show a contradiction via the following implication:

($*$) *If protocol* Π *accepts all 1-violation (resp. 3-violation) inputs with probability* 0 *(resp.* 2α*), then* Π *must mess up by accepting some 7-violation input with probability* $> 6\alpha$*.*

Henceforth, we use $o(1)$ to denote anonymous quantities that tend to 0 as $|\Pi|/k$ tends to 0.

The implication ($*$) can be derived cleanly from two types of limitations of our too-good-to-be-true Π. The first limitation concerns the situation where we start with a 1-violation input z, and consider 3-violation inputs z^{B_1} and z^{B_2} that are obtained from z by flipping either a typical path B_1 or another typical path B_2 that is edge-disjoint from B_1 (the endpoints of B_i are terminals). The protocol should accept both z^{B_1} and z^{B_2} (more precisely, any (x, y) consistent with them) with probability 2α, but it better not accept both inputs while generating the same transcript—otherwise we could cut-and-paste z^{B_1} and z^{B_2} together and fool Π into accepting z (which would contradict correctness). What we actually get is that the accepting transcripts for z^{B_1} and z^{B_2} should be near-disjoint:

1-vs-3 Lemma. *Let* z *be any 1-violation input and let* $\mathcal{P}$ *be any pairing of the non-violated terminals with canonical edge-disjoint paths* $B_1, \ldots, B_k$*. Let* w *be a random window of* z*, and choose distinct* $i, j \in [k]$ *at random. Then, with probability* $\geq 1 - o(1)$*,*

$$\Delta\big(\Pi^{\mathrm{acc}}|w^{B_i}, \Pi^{\mathrm{acc}}|w^{B_j}\big) \;\geq\; 1 - o(1).$$

The second limitation concerns the situation where we start with a 3-violation input z and flip a typical path B to obtain a 5-violation input z^B. Consider a typical accepting transcript τ in $\Pi|z$. It is unlikely that the execution τ catches us making the tiny local change $z \mapsto z^B$ in the input, and one expects that τ continues to appear in $\Pi|z^B$. (This is the usual *corruption* property of large rectangles.) Formally, for windows w_1 and w_2, we say

$$\Pi|w_1 \;\; \textit{overflows onto} \;\; \Pi|w_2 \qquad \text{iff} \qquad \textstyle\sum_\tau \max(p^1_\tau - p^2_\tau, 0) \;\leq\; o(\alpha), \tag{8.3}$$

where[2] $p^i_\tau \coloneqq \mathbf{Pr}[\Pi|w_i = \tau]$ and the sum is over accepting transcripts τ. (The definition of overflow makes sense for any distributions over transcripts; we will also apply it to $\Pi|z$.) For technical reasons (which will become apparent shortly), we shall flip two paths instead of one in order to pass from 3-violation inputs to 7-violation inputs.

3-vs-7 Lemma. *Let* z *be any 3-violation input and let* $\mathcal{P}$ *be any pairing of the non-violated terminals with canonical edge-disjoint paths* $B_1, \ldots, B_{k-1}$*. Let* w *be a random window of* z*, and choose distinct* $i, j \in [k-1]$ *at random. Then, with probability* $\geq 1 - o(1)$*,*

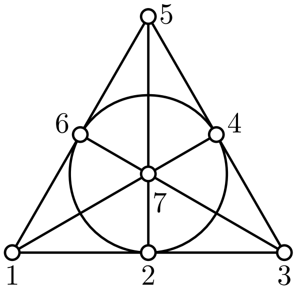
$$\Pi|w \;\; \textit{overflows onto} \;\; \Pi|w^{B_i \cup B_j}.$$

[2] Note that the event in $\mathbf{Pr}[\Pi|w_i = \tau]$ is to be parsed as "a sample from the distribution $(\Pi|w_i)$ yields τ".

8.6.4 Deriving the contradiction

We now prove $(*)$ by applying the 1-vs-3 Lemma and the 3-vs-7 Lemma in a black-box fashion to find some 7-violation input that Π accepts with too high a probability $> 6\alpha$.

Define $F := ([7], E)$ as the *Fano plane* hypergraph on 7 nodes. See the figure on the right. This hypergraph has 7 hyperedges, each of which is incident to 3 nodes, and the hyperedges are pairwise uniquely intersecting. For each hyperedge $e \in E$ choose some arbitrary but fixed pairing $\mathcal{P}^e$ of the remaining nodes in $[7] \smallsetminus e$.



$F = ([7], E)$

Probability space. Choose the following at random:

1. An injection of $[7]$ into T. Denote the result by $v_1, \ldots, v_7 \in T$.
2. A pairing $\mathcal{P}$ of the remaining terminals $T \smallsetminus \{v_1, \ldots, v_7\}$.
3. A 7-violation input z_7 with $\text{viol}(z_7) = \{v_1, \ldots, v_7\}$.
4. A window w_7 of z_7.

We do not make a distinction between the nodes of F and their embedding $\{v_1, \ldots, v_7\}$ in T. In particular, we think of the hyperedges $e \in E$ as triples of terminals, and the $\mathcal{P}^e$ as pairings of terminals. Associated with the pairing $\mathcal{P}^e \cup \mathcal{P}$ there is a canonical collection of edge-disjoint paths; let $\{B_1^e, B_2^e\}$ denote the two paths that connect $\mathcal{P}^e$ in this collection.

Based on the above, we define seven 3-violation windows, indexed by $e \in E$:

$$\text{window } \; w_e := w_7^{B_1^e \cup B_2^e} \;\text{ of }\; z_e := z_7^{B_1^e \cup B_2^e} \qquad \textit{(note: } \text{viol}(z_e) = e\textit{)}.$$

The following claim (proved at the end of this subsection) follows directly from the 1-vs-3 Lemma and the 3-vs-7 Lemma as soon as we view our probability space from the right perspective.

Claim 8.5. *In the following list of 28 events, each occurs with probability $\geq 1 - o(1)$:*

- *Overflow for $e \in E$: $\Pi|w_e$ overflows onto $\Pi|w_7$.*
- *Near-disjointness for $\{e, e'\} \subseteq E$: $\Delta\big(\Pi^{\text{acc}}|w_e, \Pi^{\text{acc}}|w_{e'}\big) \geq 1 - o(1)$.*

By a union bound over all the 28 events in the above list, we can fix our random choices 1–4 to obtain a fixed 7-violation window w_7 and fixed 3-violation windows w_e such that

$$\textit{Overflow:} \quad \forall e \in E: \qquad \textstyle\sum_\tau \max(p_\tau^e - p_\tau^7, 0) \;\leq\; o(\alpha), \tag{8.4}$$

$$\textit{Near-disjointness:} \quad \forall \{e, e'\} \subseteq E: \qquad \textstyle\sum_\tau \min(p_\tau^e, p_\tau^{e'}) \;\leq\; o(\alpha). \tag{8.5}$$

Here $p_\tau^7 := \mathbf{Pr}[\Pi|w_7 = \tau]$, $p_\tau^e := \mathbf{Pr}[\Pi|w_e = \tau]$, and the sums are over accepting transcripts; we have also rephrased the near-disjointness property using the fact that $\mathbf{Pr}[\Pi|w_e \text{ accepts}] = 2\alpha$.

These two properties state that typical accepting transcripts for $\Pi|w_e$ contribute to the acceptance probability of $\Pi|w_7$, and these contributions are pairwise near-disjoint. Hence, roughly speaking, one expects $\mathbf{Pr}[\Pi|w_7 \text{ accepts}]$ to be at least $\sum_{e\in E}\mathbf{Pr}[\Pi|w_e \text{ accepts}] = 7\cdot 2\alpha = 14\alpha > 6\alpha$. But then some 7-violation input in w_7 would be accepted with probability $> 6\alpha$, which completes the proof of $(*)$ (and hence Theorem 8.3). Indeed, we perform this calculation carefully as follows. We first partition the set of accepting transcripts as $\bigcup_{e\in E} S_e$ where S_e consists of those τ's for which $p^e_\tau = \max_{e'} p^{e'}_\tau$ (breaking ties arbitrarily). Then

$$\begin{aligned}
\mathbf{Pr}[\Pi|w_7 \text{ accepts}] &= \textstyle\sum_\tau p^7_\tau \\
&\geq \textstyle\sum_{e\in E,\,\tau\in S_e} \min(p^7_\tau, p^e_\tau) \\
&= \textstyle\sum_{e\in E,\,\tau\in S_e} \big(p^e_\tau - \max(p^e_\tau - p^7_\tau, 0)\big) \\
&\geq \textstyle\sum_{e\in E,\,\tau\in S_e} p^e_\tau - \sum_{e\in E,\,\tau} \max(p^e_\tau - p^7_\tau, 0) \\
&\geq \textstyle\sum_{e\in E,\,\tau\in S_e} p^e_\tau - 7\cdot o(\alpha) && \text{(via (8.4))} \\
&= \textstyle\sum_{e\in E,\,\tau} p^e_\tau - \sum_{e\in E,\,e'\in E\smallsetminus\{e\},\,\tau\in S_{e'}} p^e_\tau - o(\alpha) \\
&= \textstyle\sum_{e\in E,\,\tau} p^e_\tau - \sum_{e\in E,\,e'\in E\smallsetminus\{e\},\,\tau\in S_{e'}} \min(p^e_\tau, p^{e'}_\tau) - o(\alpha) \\
&\geq \textstyle\sum_{e\in E,\,\tau} p^e_\tau - \sum_{e\in E,\,e'\in E\smallsetminus\{e\},\,\tau} \min(p^e_\tau, p^{e'}_\tau) - o(\alpha) \\
&\geq \textstyle\sum_{e\in E,\,\tau} p^e_\tau - 7\cdot 6\cdot o(\alpha) - o(\alpha) && \text{(via (8.5))} \\
&= \textstyle\sum_{e\in E} \mathbf{Pr}[\Pi|w_e \text{ accepts}] - o(\alpha) \\
&= 7\cdot 2\alpha - o(\alpha) \\
&= (14 - o(1))\cdot\alpha \\
&> 6\alpha.
\end{aligned}$$

Proof of Claim 8.5. Overflow. For notational convenience, suppose $e = \{v_1, v_2, v_3\}$ and $\mathcal{P}^e = \{\{v_4, v_7\}, \{v_5, v_6\}\}$. An alternative way to generate a sample from our probability space is (in steps 1 and 6, we are really picking random injections):

1. Random $\{v_1, v_2, v_3\} \subseteq T$.
2. Random 3-violation input z_e subject to $\text{viol}(z_e) = \{v_1, v_2, v_3\}$.
3. Random pairing $\mathcal{P}' = \{P_1, \ldots, P_{k-1}\}$ of $T \smallsetminus \{v_1, v_2, v_3\}$ with canonical paths $B_1, \ldots, B_{k-1}$.
4. Random window w_e of z_e.
5. Random distinct $i, j \in [k-1]$.
6. Random $\{v_4, v_7\} = P_i$ and $\{v_5, v_6\} = P_j$.
7. Deterministically, define $z_7 := z_e^{B_i\cup B_j}$ and $w_7 := w_e^{B_i\cup B_j}$ and $\mathcal{P} := \mathcal{P}' \smallsetminus \{P_i, P_j\}$.

The choices made in steps 1–3 match the data that is quantified universally in the 3-vs-7 Lemma, whereas steps 4 and 5 make random choices as in the 3-vs-7 Lemma; hence the lemma applies.

Near-disjointness. For notational convenience, suppose $e = \{v_1, v_2, v_3\}$, $e' = \{v_3, v_4, v_5\}$, $\mathcal{P}^e = \{\{v_4, v_7\}, \{v_5, v_6\}\}$, and $\mathcal{P}^{e'} = \{\{v_1, v_7\}, \{v_2, v_6\}\}$ (it does not matter for the proof how

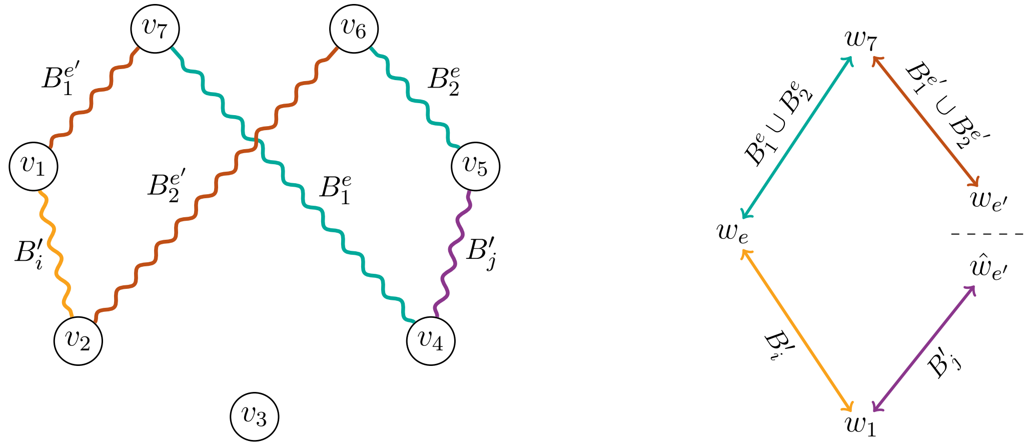


Figure 8.2: Illustration for the proof of Claim 8.5. *Left:* Paths flipped between terminals. *Right:* Relationships between windows.

$\mathcal{P}^e$ and $\mathcal{P}^{e'}$ were chosen). An alternative way to generate a sample from our probability space is (see Figure 8.2):

1. Random $v_3 \in T$.
2. Random 1-violation input z_1 subject to $\text{viol}(z_1) = \{v_3\}$.
3. Random pairing $\mathcal{P}' = \{P_1', \dots, P_k'\}$ of $T \smallsetminus \{v_3\}$ with canonical paths $B_1', \dots, B_k'$.
4. Random window w_1 of z_1.
5. Random distinct $i, j, l \in [k]$.
6. Random $\{v_1, v_2\} = P_i'$ and $\{v_4, v_5\} = P_j'$ and $\{v_6, v_7\} = P_l'$.
7. Deterministically, define
   - $z_e \coloneqq z_1^{B_i'}$ and $w_e \coloneqq w_1^{B_i'}$,
   - $\hat{z}_{e'} \coloneqq z_1^{B_j'}$ and $\hat{w}_{e'} \coloneqq w_1^{B_j'}$,
   - $\mathcal{P} \coloneqq \mathcal{P}' \smallsetminus \{P_i, P_j, P_l\}$,
   - $\{B_1^e, B_2^e\}$ according to the canonical paths for $\mathcal{P}^e \cup \mathcal{P}$,
   - $\{B_1^{e'}, B_2^{e'}\}$ according to the canonical paths for $\mathcal{P}^{e'} \cup \mathcal{P}$,
   - $z_7 \coloneqq z_e^{B_1^e \cup B_2^e}$ and $w_7 \coloneqq w_e^{B_1^e \cup B_2^e}$,
   - $z_{e'} \coloneqq z_7^{B_1^{e'} \cup B_2^{e'}}$ and $w_{e'} \coloneqq w_7^{B_1^{e'} \cup B_2^{e'}}$.

The choices made in steps 1–3 match the data that is quantified universally in the 1-vs-3 Lemma, whereas steps 4 and 5 (excluding variable l) make random choices as in the 1-vs-3 Lemma. Hence that lemma applies and shows that $\Pi^{\text{acc}}|w_e$ and $\Pi^{\text{acc}}|\hat{w}_{e'}$ are near-disjoint with high probability. Finally, we note that $\hat{w}_{e'}$ and $w_{e'}$ differ by the flipping of an eulerian graph, namely $B_j' \oplus B_i' \oplus B_1^e \oplus B_2^e \oplus B_1^{e'} \oplus B_2^{e'}$ (where $\oplus$ means symmetric difference), so $\Pi|w_{e'}$ and $\Pi|\hat{w}_{e'}$ have the same distribution assuming w.l.o.g. that Π is smooth (as discussed in Section 8.6.2). Thus $\Pi^{\text{acc}}|w_e$ and $\Pi^{\text{acc}}|w_{e'}$ are also near-disjoint with high probability. $\square$

8.6.5 Roadmap for the rest of the proof

We prove the 1-vs-3 Lemma in Section 8.6.6 and the 3-vs-7 Lemma in Section 8.6.7. Both proofs rely on another technical lemma, the Homogeneity Lemma (stated below, proved in Section 8.6.8), which generalizes a lemma from (the full version of) [HN12, §5]. In fact, we prove the Homogeneity Lemma for any gadget g that is *regular* (as defined in Section 8.6.8), which our gadget is.

Homogeneity Lemma. *Fix an arbitrary $z \in \{0,1\}^m$ for some m. Let W be a random window of z in g^m, let XY be a random input in W, and let R be an arbitrary random variable that is conditionally independent of W given XY. If $\mathbf{I}(R\,;XY\,|\,W) \leq o(1)$ then at least a $1-o(1)$ fraction of windows w of z are such that $\Delta(R|w, R|z) \leq o(1)$.*

In the statement, $R|w$ is shorthand for $R|(W = w)$, and $R|z$ denotes the marginal distribution of R in the whole probability space, which is over uniformly random $XY \in (g^m)^{-1}(z)$. Furthermore, we mention that our proof shows that at least a $1-o(1)$ fraction of $xy \in (g^m)^{-1}(z)$ are such that $\Delta(R|xy, R|z) \leq o(1)$, but for the 1-vs-3 Lemma and the 3-vs-7 Lemma we only require the property for windows.

In Section 8.5 we defined the directed flips $w^{\leftarrow}, w^{\nwarrow}, w^{\uparrow}$ for a single-gadget window. We now also define directed flips for multi-gadget windows w: if B is a subset of coordinates then $w^{\leftarrow B}, w^{\nwarrow B}, w^{\uparrow B}$ are defined by applying the corresponding directed flips to the coordinates in B. Then we have the following key property of our gadget.

Fact 8.6. *If w is a uniformly random window of z, then each of $w^{\leftarrow B}, w^{\nwarrow B}, w^{\uparrow B}$ is marginally a uniformly random window of z^B.*

This concept is used in the proofs of the 1-vs-3 Lemma and the 3-vs-7 Lemma. It turns out that the 3-vs-7 Lemma can be proved (with a small modification to our proof) even for the simpler gadget that was used in Chapter 7 (as can the Homogeneity Lemma since that gadget is regular), but our proof of the 1-vs-3 Lemma crucially uses Fact 8.6, which does not hold for that simpler gadget.

8.6.6 Proof of the 1-vs-3 Lemma

Consider a probability space with the following random variables: $I \in [k]$, $J \in [k] \smallsetminus \{I\}$, W is a random window of z^{B_I}, XY is a random input in W, and Π^{acc} is the random transcript of Π on input XY conditioned on acceptance. For convenience, denote $B \coloneqq B_1 \cup \cdots \cup B_k$ and $B_{-i} \coloneqq B \smallsetminus B_i$. We have

$$\mathbf{I}\big(\Pi^{\text{acc}}\,; (XY)_{B_{-I}}\,\big|\, IW\big) \;\leq\; \mathbf{H}(\Pi^{\text{acc}}\,|\, IW) \;\leq\; |\Pi| \;\leq\; o(k)$$

so by the standard direct sum property [BJKS04],

$$\begin{aligned}\mathbf{I}\big(\Pi^{\mathrm{acc}}\,;(XY)_{B_J}\,\big|\,IJW\big) &= \tfrac{1}{k-1}\cdot\mathbf{E}_{i\sim I}\textstyle\sum_{j\in[k]\smallsetminus\{i\}}\mathbf{I}\big(\Pi^{\mathrm{acc}}\,;(XY)_{B_j}\,\big|\,W,I=i\big)\\ &\le \tfrac{1}{k-1}\cdot\mathbf{I}\big(\Pi^{\mathrm{acc}}\,;(XY)_{B_{-I}}\,\big|\,IW\big)\\ &\le o(1).\end{aligned}$$

Define $H \coloneqq \{I,J\}$, and abbreviate $B_I \cup B_J$ as B_H and $W_{[n]\smallsetminus(B_I\cup B_J)}$ as W_{-B_H}. By Markov's inequality, with probability $\ge 1-o(1)$ over $h \sim H$ and $w_{-B_h} \sim W_{-B_h}$, we have

$$\mathbf{I}\big(\Pi^{\mathrm{acc}}\,;(XY)_{B_J}\,\big|\,IJW_{B_h},H=h,W_{-B_h}=w_{-B_h}\big)\;\le\;o(1).$$

Fixing such h and w_{-B_h} (henceforth), say $h=\{1,2\}$, it suffices to show that with probability $\ge 1-o(1)$ over a random window w_{B_h} of z_{B_h}, we have $\Delta\big(\Pi^{\mathrm{acc}}|w^{B_1},\Pi^{\mathrm{acc}}|w^{B_2}\big)\ge 1-o(1)$ (where w is the combination of w_{B_h} and w_{-B_h}).

We rephrase the situation as follows. Consider a protocol Π_* that interprets its input as $(xy)_{B_h}$, uses private coins to sample random $(xy)_{-B_h}$ from w_{-B_h}, and runs Π on the input xy (the combination of $(xy)_{B_h}$ and $(xy)_{-B_h}$). Henceforth recycling notation by letting $z\in\{0,1\}^{|B_h|}$ refer to z_{B_h}, and letting (I,J) be random in $\{(1,2),(2,1)\}$, W be a random window of (the new) z^{B_I}, and XY be a random input to Π_* in W, the situation is:

Assumption: $\mathbf{I}\big(\Pi_*^{\mathrm{acc}}\,;(XY)_{B_J}\,\big|\,IJW\big)\le o(1)$.
Want to show: For $\ge 1-o(1)$ fraction of windows w of z, $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{B_1},\Pi_*^{\mathrm{acc}}|w^{B_2}\big)\ge 1-o(1)$.

The assumption holds (with factor 2 loss in the $o(1)$) conditioned on either outcome of (I,J); let us tacitly condition on the outcome $(1,2)$. Then $\mathbf{I}\big(\Pi_*^{\mathrm{acc}}\,;(XY)_{B_2}\,\big|\,W\big)\le o(1)$ where W is a random window of z^{B_1}. By Markov's inequality, with probability $\ge 1-o(1)$ over $w_{B_1}\sim W_{B_1}$ we have $\mathbf{I}\big(\Pi_*^{\mathrm{acc}}\,;(XY)_{B_2}\,\big|\,W_{B_2},W_{B_1}=w_{B_1}\big)\le o(1)$; call such a w_{B_1} *good.* Hence for a good w_{B_1}, we can apply the Homogeneity Lemma with $m\coloneqq|B_2|$ and $R\coloneqq\Pi_*^{\mathrm{acc}}|(W_{B_1}=w_{B_1})$ (note that $R|(xy)_{B_2}$ is the distribution of Π_*^{acc} on input $(XY)_{B_1}(xy)_{B_2}$ where $(XY)_{B_1}$ is random in w_{B_1}). This tells us that for a good w_{B_1}, with probability $\ge 1-o(1)$ over $w_{B_2}\sim W_{B_2}$ we have $\Delta\big(\Pi_*^{\mathrm{acc}}|w_{B_1}w_{B_2},\Pi_*^{\mathrm{acc}}|w_{B_1}z_{B_2}\big)\le o(1)$, where the distribution $\Pi_*^{\mathrm{acc}}|w_{B_1}z_{B_2}$ is over random $(XY)_{B_1}\in w_{B_1}$ and $(XY)_{B_2}\in(g^m)^{-1}(z_{B_2})$. We summarize the above with the following claim.

Claim 8.7. *For $\ge 1-o(1)$ fraction of windows w of z^{B_1}, we have $\Delta\big(\Pi_*^{\mathrm{acc}}|w,\Pi_*^{\mathrm{acc}}|w_{B_1}z_{B_2}\big)\le o(1)$.*

Conditioning on the other outcome $(I,J)=(2,1)$ yields the symmetric property.

Claim 8.8. *For $\ge 1-o(1)$ fraction of windows w of z^{B_2}, we have $\Delta\big(\Pi_*^{\mathrm{acc}}|w,\Pi_*^{\mathrm{acc}}|z_{B_1}w_{B_2}\big)\le o(1)$.*

Now pick a random window w of z^{B_h}. Using Fact 8.6, w^{B_2} and $w^{\nwarrow B_2}$ are both uniformly random (albeit correlated) windows of z^{B_1}, and w^{B_1} and $w^{\nwarrow B_1}$ are both uniformly random (albeit correlated) windows of z^{B_2}. Hence by Claim 8.7, Claim 8.8, and a union bound, with probability $\geq 1 - o(1)$ over the choice of w, the following four distances are simultaneously $\leq o(1)$: $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{B_2}, \Pi_*^{\mathrm{acc}}|w_{B_1}z_{B_2}\big)$, $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{\nwarrow B_2}, \Pi_*^{\mathrm{acc}}|w_{B_1}z_{B_2}\big)$, $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{B_1}, \Pi_*^{\mathrm{acc}}|z_{B_1}w_{B_2}\big)$, $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{\nwarrow B_1}, \Pi_*^{\mathrm{acc}}|z_{B_1}w_{B_2}\big)$.

We argue shortly that $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{\nwarrow B_1}, \Pi_*^{\mathrm{acc}}|w^{\nwarrow B_2}\big) = 1$ with probability 1; putting everything together then shows that $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{B_1}, \Pi_*^{\mathrm{acc}}|w^{B_2}\big) \geq 1 - o(1)$, as illustrated below. (This is equivalent to what we want to show, since sampling a window w of z^{B_h} and taking w^{B_1}, w^{B_2} is equivalent to sampling a window w of z and taking w^{B_2}, w^{B_1}.)

$$\begin{tikzcd}
\Pi_*^{\mathrm{acc}}|w^{B_1} \arrow[r, leftrightarrow, "\Delta \leq o(1)"] \arrow[d, leftrightarrow, dashed, "\Delta \geq 1 - o(1)"'] & \Pi_*^{\mathrm{acc}}|z_{B_1}w_{B_2} \arrow[r, leftrightarrow, "\Delta \leq o(1)"] & \Pi_*^{\mathrm{acc}}|w^{\nwarrow B_1} \arrow[d, leftrightarrow, "\Delta = 1"] \\
\Pi_*^{\mathrm{acc}}|w^{B_2} \arrow[r, leftrightarrow, "\Delta \leq o(1)"'] & \Pi_*^{\mathrm{acc}}|w_{B_1}z_{B_2} \arrow[r, leftrightarrow, "\Delta \leq o(1)"'] & \Pi_*^{\mathrm{acc}}|w^{\nwarrow B_2}
\end{tikzcd}$$

To finish the proof, suppose for contradiction that some accepting transcript has positive probability under both $\Pi_*^{\mathrm{acc}}|xy$ and $\Pi_*^{\mathrm{acc}}|x'y'$ for some $xy \in w^{\nwarrow B_1}$ and $x'y' \in w^{\nwarrow B_2}$. Then Π_* would also accept xy' with positive probability. We claim that $g^{|B_h|}(xy') = z$. To see this, consider any coordinate c of z; suppose $c \in B_1$ (the case $c \in B_2$ is similar). There is an embedding of stretched AND (if $z_c = 0$) or NAND (if $z_c = 1$) such that $w_c^{\nwarrow B_1}$ is the image of $(0, 0)$ (hence is z_c-monochromatic) and $w_c^{\nwarrow B_2} = w_c$ is the image of $(1, 1)$ (hence is $(1 - z_c)$-monochromatic). Since $(xy)_c \in w_c^{\nwarrow B_1}$ and $(x'y')_c \in w_c$, it follows that $(xy')_c$ is in the image of $(0, 1)$, which is z_c-monochromatic. So $g((xy')_c) = z_c$ and the claim is proved.

Since Π_* accepts some input in $(g^{|B_h|})^{-1}(z)$ with positive probability (for the new z), it follows that Π accepts some input in $(g^n)^{-1}(z)$ with positive probability, for the original z, which is a contradiction since the original z has only one violation.

8.6.7 Proof of the 3-vs-7 Lemma

Assume for convenience that $k-1$ is even. Note that sampling distinct $i, j \in [k-1]$ is equivalent to sampling a permutation σ of $[k-1]$ and an $h \in [\frac{k-1}{2}]$ and setting $i = \sigma(2h-1)$, $j = \sigma(2h)$.

Thus we have a probability space with random variables Σ, H, I, J corresponding to the above, as well as the following: W is a random window of z, XY is a random input in W, and Π^{acc} is the random transcript of Π on input XY conditioned on acceptance. For convenience, denote $B \coloneqq B_1 \cup \cdots \cup B_{k-1}$ and $B_{ij} \coloneqq B_i \cup B_j$. We have

$$\mathbf{I}\big(\Pi^{\mathrm{acc}}\,; (XY)_B \,\big|\, W\big) \;\leq\; \mathbf{H}(\Pi^{\mathrm{acc}} \,|\, W) \;\leq\; |\Pi| \;\leq\; o(k)$$

so by the standard direct sum property [BJKS04],

$$\begin{aligned}
\mathbf{I}\big(\Pi^{\text{acc}}\,;(XY)_{B_{IJ}}\,\big|\,WIJ\big) &= \mathbf{I}\big(\Pi^{\text{acc}}\,;(XY)_{B_{IJ}}\,\big|\,W\Sigma H\big)\\
&= \tfrac{2}{k-1}\cdot\textstyle\sum_{h\in[(k-1)/2]}\mathbf{I}\big(\Pi^{\text{acc}}\,;(XY)_{B_{IJ}}\,\big|\,W\Sigma, H=h\big)\\
&\leq \tfrac{2}{k-1}\cdot\mathbf{I}\big(\Pi^{\text{acc}}\,;(XY)_{B}\,\big|\,W\Sigma\big)\\
&= \tfrac{2}{k-1}\cdot\mathbf{I}\big(\Pi^{\text{acc}}\,;(XY)_{B}\,\big|\,W\big)\\
&\leq o(1).
\end{aligned}$$

Abbreviate $W_{[n]\smallsetminus B_{ij}}$ as $W_{-B_{ij}}$. By Markov's inequality, with probability $\geq 1-o(1)$ over $ij\sim IJ$ and $w_{-B_{ij}}\sim W_{-B_{ij}}$, we have $\mathbf{I}\big(\Pi^{\text{acc}}\,;(XY)_{B_{ij}}\,\big|\,W_{B_{ij}}, W_{-B_{ij}}=w_{-B_{ij}}\big)\leq o(1)$. Fixing such ij and $w_{-B_{ij}}$ (henceforth), it suffices to show that with probability $\geq 1-o(1)$ over $w_{B_{ij}}\sim W_{B_{ij}}$, $\Pi|w$ overflows onto $\Pi|w^{B_{ij}}$ (where w is the combination of $w_{B_{ij}}$ and $w_{-B_{ij}}$).

We rephrase the situation as follows. Consider a protocol Π_* that interprets its input as $(xy)_{B_{ij}}$, uses private coins to sample random $(xy)_{-B_{ij}}$ from $w_{-B_{ij}}$, and runs Π on the input xy (the combination of $(xy)_{B_{ij}}$ and $(xy)_{-B_{ij}}$). Henceforth recycling notation by letting $z\in\{0,1\}^{|B_{ij}|}$ refer to $z_{B_{ij}}$, letting B refer to B_{ij}, and letting W be a random window of (the new) z and XY be a random input to Π_* in W, the situation is:

Assumption: $\mathbf{I}\big(\Pi_*^{\text{acc}}\,;XY\,\big|\,W\big)\leq o(1)$.
Want to show: For $\geq 1-o(1)$ fraction of windows w of z, $\Pi_*|w$ overflows onto $\Pi_*|w^B$.

Claim 8.9. *For $\geq 1-o(1)$ fraction of windows w of z^B, $\Pi_*|z$ overflows onto $\Pi_*|w$.*

We prove Claim 8.9 shortly, but first we finish the proof of the 3-vs-7 Lemma assuming it. By the Homogeneity Lemma (with $m:=|B|$ and $R:=\Pi_*^{\text{acc}}$), Claim 8.9, and a union bound, at least a $1-o(1)$ fraction of windows w of z are such that both $\Delta\big(\Pi_*^{\text{acc}}|w,\Pi_*^{\text{acc}}|z\big)\leq o(1)$ and $\Pi_*|z$ overflows onto $\Pi_*|w^B$ (since w^B is a uniform window of z^B if w is a uniform window of z). We show that this implies that $\Pi_*|w$ overflows onto $\Pi_*|w^B$ as follows (letting p_τ^z, p_τ^w, $p_\tau^{w^B}$ denote the probability of a transcript τ under the distributions $\Pi_*|z$, $\Pi_*|w$, $\Pi_*|w^B$ respectively, and summing only over accepting τ's):

$$\textstyle\sum_\tau \max(p_\tau^w-p_\tau^{w^B},0)\;\leq\;\sum_\tau\max(p_\tau^z-p_\tau^{w^B},0)+\sum_\tau|p_\tau^w-p_\tau^z|\;\leq\;o(\alpha)+o(\alpha)\;=\;o(\alpha).$$

Proof of Claim 8.9. By Fact 8.6, if w is a random window of z^B, then $w^{\leftarrow B}$, $w^{\nwarrow B}$, $w^{\uparrow B}$ are each marginally uniformly random windows of z. Thus by the Homogeneity Lemma (with $m:=|B|$ and $R:=\Pi_*^{\text{acc}}$) and a union bound, with probability $\geq 1-o(1)$ over the choice of w, the following three distances are simultaneously $\leq o(1)$: $\Delta\big(\Pi_*^{\text{acc}}|w^{\leftarrow B},\Pi_*^{\text{acc}}|z\big)$, $\Delta\big(\Pi_*^{\text{acc}}|w^{\nwarrow B},\Pi_*^{\text{acc}}|z\big)$, $\Delta\big(\Pi_*^{\text{acc}}|w^{\uparrow B},\Pi_*^{\text{acc}}|z\big)$. Now assuming this good event occurs for some particular w, we just need to show that $\Pi_*|z$ overflows onto $\Pi_*|w$.

(See Figure 8.3 for a proof-by-picture.) Let p_τ, p_τ^{11}, p_τ^{10}, p_τ^{00}, p_τ^{01} denote the probabilities of a transcript τ under $\Pi_*|z$, $\Pi_*|w$, $\Pi_*|w^{\leftarrow B}$, $\Pi_*|w^{\nwarrow B}$, $\Pi_*|w^{\uparrow B}$ respectively. Let $\gamma_\tau^{00}:=|p_\tau-p_\tau^{00}|$,

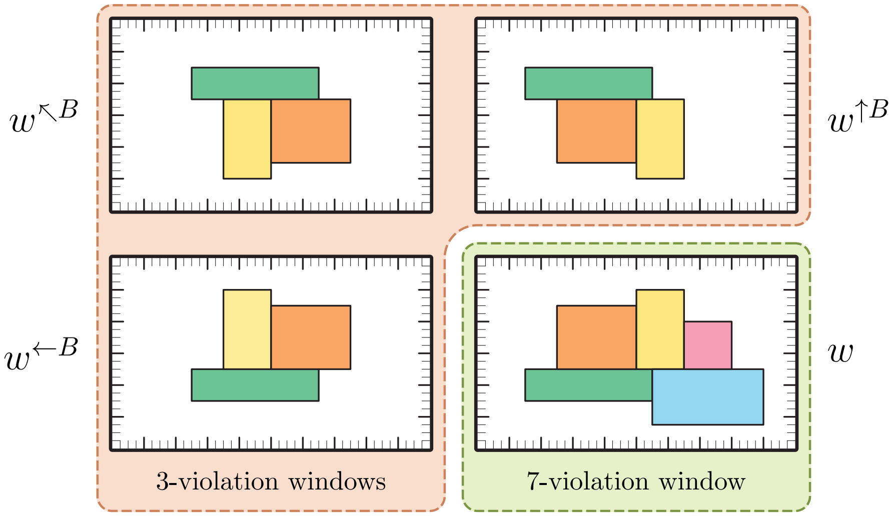


Figure 8.3: Proof of Claim 8.9 illustrated. The four windows w, $w^{\leftarrow B}$, $w^{\nwarrow B}$, $w^{\uparrow B}$ are rectangles of (x, y)'s. Each (x, y) can be further subdivided according to the private coins $(r_\mathrm{A}, r_\mathrm{B})$ of the players. The protocol Π_* partitions the extended input space of $(xr_\mathrm{A}, yr_\mathrm{B})$'s into *transcript rectangles*—above, we have only drawn *accepting* transcript rectangles (in various colors). For a window w', the probability $\mathbf{Pr}[\Pi_*|w' = \tau]$ is simply the *area* (appropriately scaled) of the transcript rectangle of τ inside w'. In the proof of Claim 8.9, the relevant case is when all of $\Pi_*^{\mathrm{acc}}|w^{\leftarrow B}$, $\Pi_*^{\mathrm{acc}}|w^{\nwarrow B}$, $\Pi_*^{\mathrm{acc}}|w^{\uparrow B}$ have roughly the same distribution, say, D (in fact, $D := \Pi_*^{\mathrm{acc}}|z$). By the rectangular property of transcripts, this forces $\Pi_*|z$ to *overflow onto* $\Pi_*|w$. (Note that $\Pi_*^{\mathrm{acc}}|w$ may contain additional transcripts to those in D, since the acceptance probability is higher.)

and for $ab \in \{01, 10\}$ let $\gamma_\tau^{ab} := |p_\tau^{00} - p_\tau^{ab}|$. We claim that for all τ, $p_\tau - p_\tau^{11} \leq \gamma_\tau^{00} + \gamma_\tau^{01} + \gamma_\tau^{10}$; this will finish the proof since then (summing only over accepting τ's)

$$\textstyle\sum_\tau \max(p_\tau - p_\tau^{11}, 0) \;\leq\; \sum_\tau(\gamma_\tau^{00} + \gamma_\tau^{01} + \gamma_\tau^{10}) \;\leq\; o(\alpha) + o(\alpha) + o(\alpha) \;=\; o(\alpha)$$

where the second inequality is because $\sum_\tau \gamma_\tau^{00}$, $\sum_\tau \gamma_\tau^{01}$, $\sum_\tau \gamma_\tau^{10} \leq o(\alpha)$ follow from (respectively) $\Delta\big(\Pi_*^{\mathrm{acc}}|z, \Pi_*^{\mathrm{acc}}|w^{\nwarrow B}\big)$, $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{\nwarrow B}, \Pi_*^{\mathrm{acc}}|w^{\uparrow B}\big)$, $\Delta\big(\Pi_*^{\mathrm{acc}}|w^{\nwarrow B}, \Pi_*^{\mathrm{acc}}|w^{\leftarrow B}\big) \leq o(1)$.

To verify the subclaim, it suffices to show that

$$p_\tau^{01} \cdot p_\tau^{10} \;\geq\; (p_\tau^{00})^2 - p_\tau^{00}\gamma_\tau^{01} - p_\tau^{00}\gamma_\tau^{10} \tag{8.6}$$

since by the rectangular nature of transcripts, we have $p_\tau^{00} \cdot p_\tau^{11} = p_\tau^{01} \cdot p_\tau^{10}$, and thus if $p_\tau^{00} > 0$ then

$$p_\tau^{11} \;=\; \frac{p_\tau^{01} \cdot p_\tau^{10}}{p_\tau^{00}} \;\geq\; p_\tau^{00} - \gamma_\tau^{01} - \gamma_\tau^{10} \;\geq\; p_\tau - \gamma_\tau^{00} - \gamma_\tau^{01} - \gamma_\tau^{10}$$

and if $p_\tau^{00} = 0$ then of course $p_\tau^{11} \geq p_\tau^{00} = p_\tau - \gamma_\tau^{00}$. To see (8.6), note that for some signs $\sigma_\tau^{01}, \sigma_\tau^{10} \in \{1, -1\}$, the left side of (8.6) equals $\big(p_\tau^{00} + \sigma_\tau^{01}\gamma_\tau^{01}\big) \cdot \big(p_\tau^{00} + \sigma_\tau^{10}\gamma_\tau^{10}\big)$, which expands to

$$(p_\tau^{00})^2 + \sigma_\tau^{01} p_\tau^{00}\gamma_\tau^{01} + \sigma_\tau^{10} p_\tau^{00}\gamma_\tau^{10} + \sigma_\tau^{01}\sigma_\tau^{10}\gamma_\tau^{01}\gamma_\tau^{10}. \tag{8.7}$$

If $\sigma_\tau^{01} = \sigma_\tau^{10}$ then (8.7) is at least the right side of (8.6) since the last term of (8.7) is nonnegative. If $\sigma_\tau^{01} \neq \sigma_\tau^{10}$, say $\sigma_\tau^{01} = -1$ and $\sigma_\tau^{10} = 1$, then (8.7) is at least the right side of (8.6) since the sum of the last two terms in (8.7) is $p_\tau^{00}\gamma_\tau^{10} - \gamma_\tau^{01}\gamma_\tau^{10} = p_\tau^{01}\gamma_\tau^{10} \geq 0$. □

8.6.8 Proof of the Homogeneity Lemma

Definition 8.10. For a gadget $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and $b \in \{0,1\}$, define the digraph $\mathcal{G}^b$ as follows: the nodes are the b-inputs of g, and there is an edge from xy to $x'y'$ iff $x = x'$ or $y = y'$. (That is, each node has a self-loop, and all b-inputs in a given row or column have all possible edges between them.)

Definition 8.11. We say a gadget $g\colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is *regular* iff (i) $|\mathcal{X}| = |\mathcal{Y}|$ is even, (ii) each row and each column is balanced (half 0's and half 1's), and (iii) $\mathcal{G}^0$ and $\mathcal{G}^1$ are both strongly connected.

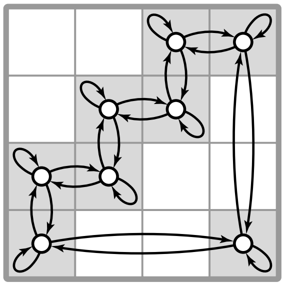
Our gadget g is indeed regular, but we proceed to prove the lemma for any regular g.

The first part of the proof is inspired by a similar approach that was used in [HN12]. We augment the probability space with the following random variables: let $X'Y'$ be a random input in W that is conditionally independent of XY given W, and let $E \in ((g^m)^{-1}(z))^2$ be chosen randomly from $\{(XY, X'Y'), (X'Y', XY)\}$. We have $\mathbf{H}(R \,|\, E) = \mathbf{H}(R \,|\, WE) \leq \mathbf{H}(R \,|\, W)$ since R is conditionally independent of W given E, and conditioning decreases entropy. We also have $\mathbf{H}(R \,|\, XYE) = \mathbf{H}(R \,|\, XY) = \mathbf{H}(R \,|\, XYW)$ since R is conditionally independent of WE given XY. Putting these together, we get

$$\mathbf{I}(R\,;XY \,|\, E) \;=\; \mathbf{H}(R \,|\, E) - \mathbf{H}(R \,|\, XYE) \;\leq\; \mathbf{H}(R \,|\, W) - \mathbf{H}(R \,|\, XYW) \;=\; \mathbf{I}(R\,;XY \,|\, W) \;\leq\; o(1).$$

By Markov's inequality, with probability $\geq 1 - o(1)$ over $e \sim E$, we have $\mathbf{I}(R\,;XY \,|\, E = e) \leq o(1)$, in which case if $e = (x^{(0)}y^{(0)}, x^{(1)}y^{(1)})$ then by Pinsker's inequality[3], $\Delta\big(R|x^{(0)}y^{(0)}, R|x^{(1)}y^{(1)}\big) \leq o(1)$; let us use $\epsilon > 0$ for the latter $o(1)$ quantity. We describe what the above means in graph theoretic terms.

Define the digraph $\mathcal{G}^z$ as follows: the nodes are the inputs in $(g^m)^{-1}(z)$, and there is an edge from one input to another iff there exists a window of z containing both inputs; this includes a self-loop at each node. Note that $\mathcal{G}^z$ is the tensor product $\mathcal{G}^{z_1} \otimes \cdots \otimes \mathcal{G}^{z_m}$, i.e., each node of $\mathcal{G}^z$ corresponds to an m-tuple of nodes from those digraphs, and each edge of $\mathcal{G}^z$ corresponds to an m-tuple of edges. For convenience, we make the dependence of the random variable E on z explicit using the notation E^z; thus E^z is distributed over



Example of $\mathcal{G}^1$ for $x_1 + y_1 + x_2y_2$

[3] Specifically, if RB are jointly distributed random variables where $B \in \{0,1\}$ is a uniformly random bit, and R_b denotes the distribution of $R|(B = b)$, then $\mathbf{I}(R\,;B) = \mathbb{D}(R_0 \,\|\, R)/2 + \mathbb{D}(R_1 \,\|\, R)/2 \geq 2 \cdot (\Delta(R_0, R)^2/2 + \Delta(R_1, R)^2/2) \geq 2 \cdot (\Delta(R_0, R)/2 + \Delta(R_1, R)/2)^2 \geq \Delta(R_0, R_1)^2/2$, where $\mathbb{D}$ denotes KL-divergence, and the first inequality is Pinsker's, the second is by convexity of the square function, and the third is by the triangle inequality.

the edges of $\mathcal{G}^z$. By regularity, for $b \in \{0,1\}$ the distribution of E^b over the edges of $\mathcal{G}^b$ puts half its mass uniformly over the self-loops, and half its mass uniformly over the non-self-loops. Note that the distribution of E^z is the product of the distributions of $E^{z_1}, \ldots, E^{z_m}$, i.e., E^z can be sampled by taking samples $(x^{(0,i)}y^{(0,i)}, x^{(1,i)}y^{(1,i)})$ from E^{z_i} (independent over $i \in [m]$) and forming the edge $\big(x^{(0,1)}y^{(0,1)} \cdots x^{(0,m)}y^{(0,m)}, x^{(1,1)}y^{(1,1)} \cdots x^{(1,m)}y^{(1,m)}\big)$ in $\mathcal{G}^z$.

We say an edge $(x^{(0)}y^{(0)}, x^{(1)}y^{(1)})$ of $\mathcal{G}^z$ is *great* iff $\Delta\big(R|x^{(0)}y^{(0)}, R|x^{(1)}y^{(1)}\big) \le \epsilon$. Thus the great edges have at least $1 - o(1)$ probability mass under E^z.

Let L be the number of non-self-loop edges in $\mathcal{G}^b$ (which is the same for $b = 0$ and $b = 1$).

Claim 8.12. *There exists a distribution over length-$2L$ walks on $\mathcal{G}^z$ such that (i) the first and last nodes are independent and each marginally uniform, and (ii) each of the $2L$ edges on the walk is marginally distributed according to E^z.*

Proof. By the product structure of $\mathcal{G}^z$ and E^z, it suffices to prove this claim for a bit b instead of z (as the claim for z follows by sampling m independent such walks on the $\mathcal{G}^{z_i}$'s and running them "in parallel"). By regularity, if we ignore the self-loops, there exists an eulerian tour in $\mathcal{G}^b$ that uses all the non-self-loop edges exactly once, and pays an equal number of visits to each node. Let $v_0, v_1, \ldots, v_{L-1}, v_0$ denote the sequence of nodes visited (with repeats) on a fixed such tour. We explicitly describe the distribution of walks $v_{i_0}, \ldots, v_{i_{2L}}$ on $\mathcal{G}^b$, using mod-L arithmetic:

1. Independently sample i_0 and ℓ uniformly from $\{0, \ldots, L-1\}$.
2. For $j = 1, \ldots, \ell$, execute one of the following with probability $1/2$ each:
   2a. Use the self-loop then move forward (i.e., $i_{2j-1} = i_{2j-2}$ and $i_{2j} = i_{2j-1} + 1$).
   2b. Move forward then use the self-loop (i.e., $i_{2j-1} = i_{2j-2} + 1$ and $i_{2j} = i_{2j-1}$).
3. For $j = \ell + 1, \ldots, L$, execute one of the following with probability $1/2$ each:
   3a. Use the self-loop twice (i.e., $i_{2j} = i_{2j-1} = i_{2j-2}$).
   3b. Move forward then backward (i.e., $i_{2j-1} = i_{2j-2} + 1$ and $i_{2j} = i_{2j-1} - 1$).

This procedure has L phases, each taking 2 steps of the walk. Each of the first ℓ phases has the effect of moving forward one node on the tour, and each of the last $L - \ell$ phases has the effect of ending up at the same node the phase started at. Thus $i_{2L} = i_0 + \ell$ and is hence independent of i_0 and uniform over $\{0, \ldots, L-1\}$ (since ℓ is independent of i_0 and uniform); hence also v_{i_0} and $v_{i_{2L}}$ are independent and uniform (since the tour visits each node equally often) and so (i) is verified. Property (ii) holds even conditioned on any ℓ, and can be verified by a little case analysis; e.g., if $\ell > 1$ then the first edge is (v_{i_0}, v_{i_0}) with probability $1/2$, and is (v_{i_0}, v_{i_0+1}) with probability $1/2$ (this is a sample from E^b since v_{i_0} is a uniform node and (v_{i_0}, v_{i_0+1}) is a uniform non-self-loop edge). □

If we sample a walk $x^{(0)}y^{(0)}, \ldots, x^{(2L)}y^{(2L)}$ in $\mathcal{G}^z$ as in Claim 8.12, then by property (ii) and a union bound, with probability $\ge 1 - 2L \cdot o(1) = 1 - o(1)$, each of the edges on the walk is great, in which case by the triangle inequality, $\Delta\big(R|x^{(0)}y^{(0)}, R|x^{(2L)}y^{(2L)}\big) \le 2L\epsilon$. In summary, by property (i), a $1-o(1)$ fraction of pairs of inputs in $(g^m)^{-1}(z)$ are *good* in the sense that their conditional distributions of R are within statistical distance $2L\epsilon = o(1)$. Thus a $1-o(1)$ fraction of inputs $xy \in (g^m)^{-1}(z)$ are such that $(xy, \overline{xy})$ is good for a $1-o(1)$ fraction of $\overline{xy} \in (g^m)^{-1}(z)$, in which case (letting $\overline{xy}$ be random in $(g^m)^{-1}(z)$ in the following)

$$\begin{aligned}
\Delta(R|xy, R) &= \Delta\big(R|xy, \mathbf{E}_{\overline{xy}}\, R|\overline{xy}\big) \\
&\leq \mathbf{E}_{\overline{xy}}\, \Delta\big(R|xy, R|\overline{xy}\big) \\
&\leq \mathbf{Pr}_{\overline{xy}}[(xy, \overline{xy}) \text{ is good}] \cdot o(1) + \mathbf{Pr}_{\overline{xy}}[(xy, \overline{xy}) \text{ is not good}] \cdot 1 \\
&\leq 1 \cdot o(1) + o(1) \cdot 1 \\
&= o(1)
\end{aligned}$$

where the second line is a basic general fact about statistical distance. Say xy is *typical* if $\Delta(R|xy, R) \leq o(1)$ as above. Note that in the original probability space, XY is marginally uniform over $(g^m)^{-1}(z)$ and thus with probability at least $1-o(1)$ over sampling $w \sim W$ and $xy \sim XY \in w$, xy is typical. It follows that for at least $1-o(1)$ fraction of w, at least $1-o(1)$ fraction of $xy \in w$ are typical, in which case

$$\begin{aligned}
\Delta(R|w, R) &= \Delta\big(\mathbf{E}_{xy\in w}\, R|xy, R\big) \\
&\leq \mathbf{E}_{xy\in w}\, \Delta(R|xy, R) \\
&\leq \mathbf{Pr}_{xy\in w}[xy \text{ is typical}] \cdot o(1) + \mathbf{Pr}_{xy\in w}[xy \text{ is not typical}] \cdot 1 \\
&\leq 1 \cdot o(1) + o(1) \cdot 1 \\
&= o(1).
\end{aligned}$$

8.7 Query lower bound

An alternative approach for proving a lower bound for the $(\#\exists{-}1)$-game for $\mathsf{Tse}_G \circ g^n$ is to (Step 1): use a communication-to-query simulation theorem (like Chapter 2) and then (Step 2): prove an appropriate query complexity lower bound for Tse_G. In this section, we carry out the second step by proving an optimal $\Omega(n)$ lower bound (which in particular answers a question from [LNNW95])—this proof is a lot simpler than our proof for the $\Omega(n/\log n)$ communication lower bound in Section 8.6. Unfortunately, as we discuss below, it is not known how to perform the first step for constant-size gadgets g.

The result of this section can be interpreted as evidence that the right bound in Theorem 8.1 is $2^{\Omega(n)}$ and the right bound in Corollary 8.4 is $\Omega(n)$, and also as motivation for further work to improve parameters for simulation theorems.

8.7.1 Query vs. communication

The query complexity analogue of nonnegative rank decompositions (nonnegative combinations of nonnegative rank-1 matrices) are *conical juntas*: nonnegative combinations of conjunctions of literals (input bits or their negations). We write a conical junta as $h = \sum_C w_C C$ where $w_C \geq 0$ and C ranges over all conjunctions $C\colon \{0,1\}^n \to \{0,1\}$. The *degree* of h is the maximum number of literals in a conjunction C with $w_C > 0$. Each conical junta naturally computes a nonnegative function $h\colon \{0,1\}^n \to \mathbb{R}_{\geq 0}$. Hence we may study (#∃–1)-games in query complexity. In particular, the query complexity of the (#∃−1)-game for Tse_G is the least degree of a conical junta h that on input z outputs $h(z) = |\mathrm{viol}(z)| - 1$.

The main result of Chapter 2 is a simulation of randomised protocols (or nonnegative rank decompositions) by conical juntas: a cost-d protocol for a lifted problem $F \circ g^n$ can be simulated by a degree-$O(d)$ conical junta (approximately) computing F. While F here is arbitrary, the result unfortunately assumes that $g \coloneqq \mathsf{IP}_b$ is a logarithmic-size, $b \coloneqq \Theta(\log n)$, inner-product function $\mathsf{IP}_b\colon \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$ given by $\mathsf{IP}_b(x,y) \coloneqq \langle x, y\rangle \bmod 2$.

Plugging b-bit gadgets into the reductions of Section 8.4 would blow up the number of input bits of CSP-SAT exponentially in b. This is not only an artefact of our particular reduction! Consider more generally any reduction from a communication search problem $S \circ g^n$ to a KW$^+$-game for a monotone $f\colon \{0,1\}^m \to \{0,1\}$. Since the KW$^+$-game has *nondeterministic* communication complexity $\log m$ (number of bits the players must nondeterministically guess to find a witness), the reduction would imply $c \leq \log m$ where c is the nondeterministic communication complexity of $S \circ g^n$. If merely computing g requires b bits of nondeterministic communication, then clearly $c \geq b$ so that $m \geq 2^b$.

8.7.2 A linear lower bound

Theorem 8.13. *There is a family of n-node bounded-degree graphs G such that the (#∃–1)-game for Tse_G requires query complexity $\Omega(n)$.*

Relation to [LNNW95]. An analogue of the (KW/EF) connection holds for query complexity: if there is a deterministic decision tree of height d that solves the search problem Tse_G, we can convert this into a degree-$(d+O(1))$ conical junta for the associated (#∃−1)-game. Moreover, if we only have a *randomised* ϵ-error decision tree for the search problem, then the connection gives us a conical junta h that *approximately* solves the (#∃−1)-game: $h(z) \in (|\mathrm{viol}(z)| - 1) \cdot (1 \pm \epsilon)$ for all z.

Our proof below is robust enough that the $\Omega(n)$ bound holds even for conical juntas that merely approximately solve the (#∃−1)-game. Hence we get a randomised $\Omega(n)$ lower bound for Tse_G, which was conjectured by [LNNW95, p. 125]; note however that in Chapter 7 we already got a near-optimal $\Omega(n/\log n)$ bound. In any case, to our knowledge, this is the first $O(1)$-vs-$\Omega(n)$ separation between certificate complexity and randomised query complexity for search problems.

The proof. Fix an n-node bounded-degree expander $G = (V, E)$. That is, for any subset $U \subseteq V$ of size $|U| \leq n/2$, the number of edges leaving U is $\Theta(|U|)$. We tacitly equip G with an arbitrary odd-weight node-labeling. Assume for the sake of contradiction that there is a conical junta $h = \sum w_C C$ of degree $o(n)$ for the $(\#\exists{-}1)$-game for Tse_G. Let C be a conjunction with $w_C > 0$. Denote by $S \subseteq E$ the set of edges that C reads; hence $|S| \leq o(n)$. Below, we write $G \setminus S$ for the graph induced on the edges $E \setminus S$ (deleting nodes that become isolated).

Claim 8.14. *We may assume w.l.o.g. that $G \setminus S$ is connected.*

Proof. If $G \setminus S$ is not connected, we may replace C with a conjunction (actually, a sum of them) that reads more input variables; namely, we let C read a larger set of edges $S' \supseteq S$ including all edges from connected components of $G \setminus S$ of "small" size $\leq n/2$. When adding some small component $K \subseteq E$ to S' we note that, because G is expanding, the size of K is big-O of the size of the edge boundary of K (which is contained in S). On the other hand, every edge in S lies on the boundary of at most two components. It follows that $|S'| = O(|S|)$, i.e., we increased the degree of h only by a constant factor. Now in $G \setminus S'$ we have only components of size $> n/2$, but there can only be one such component. □

Claim 8.15. *We may assume w.l.o.g. that C witnesses at least two fixed nodes with a parity violation (i.e., C reads all the edge labels incident to the two nodes).*

Proof. Suppose for contradiction that C witnesses at most one violation. Then we may fool C into accepting an input (and hence h into outputting a positive value on that input) where the number of violations is 1, which is a contradiction to the definition of the $(\#\exists{-}1)$-game. Indeed, let z be some input accepted by C. Then we may modify z freely on the connected graph $G \setminus S$ (by Claim 8.14) without affecting C's acceptance: we may eliminate pairs of violations from z by flipping paths (as in Section 8.3) until only one remains. (This is possible since by definition, all the non-witnessed violations of z remain in $G \setminus S$.) □

Let μ_i (i odd) denote the distribution on inputs that have i violations at a random set of i nodes, and are otherwise random with this property. We may generate an input from μ_i as follows:

1. Choose an i-set $T_i \subseteq V$ of nodes at random.
2. Let $z \in \mathbb{Z}_2^E$ be any fixed input with $\mathrm{viol}(z) = T_i$.
3. Let $q \in \mathbb{Z}_2^E$ be a random eulerian graph.
4. Output $z + q$.

Theorem 8.13 follows from the following lemma. Here we identify C with the set (subcube) of inputs it accepts.

Lemma 8.16. $\mu_5(C) \geq (10/3 - o(1)) \cdot \mu_3(C)$.

Indeed, consider the expected output value $\mathbf{E}_{z_i\sim\mu_i}[h(z_i)]$. This should be 2 for $i=3$, and 4 for $i=5$, i.e., a factor 2 increase. However, the above lemma implies that the output value gets multiplied by more than a factor 3, which is the final contradiction.

Proof of Lemma 8.16. By Claim 8.15 let $\{v_1,v_2\}$ be a pair of nodes where C witnesses two violations. For $i=3,5$, let $z_i\sim\mu_i$ and denote by T_i the i-set of its violations. Then

$$\begin{aligned}\mu_3(C) &= \mathbf{Pr}[C(z_3)=1]\\ &= \mathbf{Pr}[C(z_3)=1 \text{ and } T_3\supseteq\{v_1,v_2\}]\\ &= \tbinom{n-2}{1}/\tbinom{n}{3}\cdot\mathbf{Pr}[C(y_3)=1], && (\text{for } y_3 \coloneqq (z_3\,|\,T_3\supseteq\{v_1,v_2\}))\end{aligned}$$

$$\begin{aligned}\mu_5(C) &= \mathbf{Pr}[C(z_5)=1]\\ &= \mathbf{Pr}[C(z_5)=1 \text{ and } T_5\supseteq\{v_1,v_2\}]\\ &= \tbinom{n-2}{3}/\tbinom{n}{5}\cdot\mathbf{Pr}[C(y_5)=1]. && (\text{for } y_5 \coloneqq (z_5\,|\,T_5\supseteq\{v_1,v_2\}))\end{aligned}$$

So their ratio is

$$\frac{\mu_5(C)}{\mu_3(C)} = \frac{10}{3}\cdot\frac{\mathbf{Pr}[C(y_5)=1]}{\mathbf{Pr}[C(y_3)=1]}.$$

Hence the following claim concludes the proof of Lemma 8.16. □

Claim 8.17. $\mathbf{Pr}[C(y_5)=1]/\,\mathbf{Pr}[C(y_3)=1]\geq 1-o(1)$.

Proof. We can generate y_3 and y_5 jointly as follows:

- $\boldsymbol{y_3}$**:** Choose $v_3\in V\setminus\{v_1,v_2\}$ uniformly random and let x_3 be some input with $\mathrm{viol}(x_3)=\{v_1,v_2,v_3\}$. Output $y_3\coloneqq x_3+q$ where q is a random eulerian graph.
- $\boldsymbol{y_5}$**:** Continuing from the above, choose $\{v_4,v_5\}\subseteq V\setminus\{v_1,v_2,v_3\}$ at random. If possible, let p be a path in $G\setminus S$ joining $\{v_4,v_5\}$ (a "good" event), otherwise let p be any path joining $\{v_4,v_5\}$. Output $y_5\coloneqq x_3+p+q$.

It suffices to prove the claim conditioned on any particular v_3 (and hence also on x_3). By Claim 8.14 we have $\mathbf{Pr}[\text{“good”}\,|\,v_3]=\mathbf{Pr}\big[v_4,v_5\in G\setminus S\,\big|\,v_3\big]\geq 1-o(1)$ since $|S|\leq o(n)$. If the "good" event occurs, then C cannot distinguish between $y_3=x_3+q$ and $y_5=x_3+p+q$ so that $\mathbf{Pr}[C(y_3)=1\,|\,v_3]=\mathbf{Pr}\big[C(y_5)=1\,\big|\,\text{“good”},v_3\big]$. The claim follows as

$$\begin{aligned}\mathbf{Pr}[C(y_5)=1\,|\,v_3] &\geq \mathbf{Pr}\big[C(y_5)=1 \text{ and “good”}\,\big|\,v_3\big]\\ &= \mathbf{Pr}\big[C(y_5)=1\,\big|\,\text{“good”},v_3\big]\cdot\mathbf{Pr}[\text{“good”}\,|\,v_3]\\ &= \mathbf{Pr}[C(y_3)=1\,|\,v_3]\cdot\mathbf{Pr}[\text{“good”}\,|\,v_3]\\ &\geq \mathbf{Pr}[C(y_3)=1\,|\,v_3]\cdot(1-o(1)).\end{aligned}$$

□

Chapter 9

Open problems

We have encountered many open problems throughout this thesis. In this final chapter, we highlight some of the more important ones. Perhaps the most famous open problem in communication complexity remains the log-rank conjecture [LS88].

Open problem 9.1. *Does* $\mathsf{P}^{\mathsf{cc}}(F) \leq \log^{O(1)} \operatorname{rank}(F)$ *hold for all* F*?*

Our results in Chapter 4 imply that one has to allow the constant in the exponent to be at least 2. Lovett [Lov14] has a survey on some recent progress on proving upper bounds.

Another outstanding open problem is to prove explicit lower bounds against the communication analogue of the polynomial hierarchy $\mathsf{PH}^{\mathsf{cc}}$ [BFS86]. This is a necessary step towards constructing explicit *rigid matrices* [Raz89, Lok01, Lok09, Wun12] with connections to circuit lower bounds [Val77]. In fact, the frontier of our understanding currently lies with Arthur–Merlin communication protocols $\mathsf{AM}^{\mathsf{cc}}$, which are situated below the second level of $\mathsf{PH}^{\mathsf{cc}}$.

Open problem 9.2. *Prove explicit lower bounds against* $\mathsf{PH}^{\mathsf{cc}}$ *(or even* $\mathsf{AM}^{\mathsf{cc}}$*).*

While we have not studied Arthur–Merlin communication in this thesis (except for a brief mention in Section 2.1.3), understanding it remains one of my favourite open problems. In [GPW16b] we investigated whether information complexity arguments (which *have* featured heavily in this thesis) can be used to prove lower bounds against $\mathsf{AM}^{\mathsf{cc}}$.

We have gotten a lot of mileage out of our junta-based simulation theorem from Chapter 2. Therefore it seems important to determine the best possible parameters for which such a simulation theorem can be proved.

Open problem 9.3. *Does (some version of) Theorem 2.1 hold for gadget size* $b = O(1)$*?*

Such a simulation theorem for constant-size gadgets would render several of our results (e.g., in Chapters 7 and 8) optimal. It would also give a unified, less ad-hoc way of proving many other known results in communication complexity, e.g., randomised lower bounds for set-disjointness (and higher depth AND-OR trees) and gap-hamming.

Can we prove simulation theorems for other models besides those considered in this thesis? Can the proof techniques of Chapter 2 and Chapter 4 be combined to yield a simulation theorem for randomised bounded-error (BPP) computations?

Open problem 9.4. *Prove a communication-to-query simulation theorem for* BPP.

Such a simulation theorem would give an alternative, less ad-hoc proof for the results of Chapter 5 and also those of [ABB$^+$16b].

Lastly, we mention a well-known (e.g., [KLTT15, §7]) open problem in extension complexity, which we tried our hand at solving—but hit a barrier. For a matroid M, its associated *matroid polytope* P_M is the convex hull of (the indicator vectors of) the independent sets of M.

Open problem 9.5. *Exhibit a matroid polytope with superpolynomial extension complexity.*

Coming up with a candidate hard matroid is already challenging. It follows from Rothvoß's counting argument [Rot12] that there exist matroids M on the ground set $[n]$ such that $\mathrm{xc}(P_M) \geq 2^{\Omega(n)}$. The counting argument exploits the fact that there exist doubly-exponential-in-n many *paving* matroids. However, we note that there is a technical barrier to using paving matroids to answer the above open problem. Indeed, from the perspective of the KW/EF connection (Section 8.2), the monotone function $f\colon \{0,1\}^n \to \{0,1\}$ associated with a paving matroid is a *slice function*, i.e., non-constant on only a single Hamming slice. A theorem of Berkowitz (see [Juk12, §10.1.1]) says that for slice functions f, any (non-monotone) circuit computing f can be efficiently simulated with a monotone one. Hence proving extension complexity lower bounds using (KW/EF) is hopeless: any explicit lower bound for the ($\#\exists{-}1$)-game associated with the KW$^+$-game of a slice function would imply explicit non-monotone depth lower bounds! Moreover, for so-called "sparse" paving matroids M, it can be checked that (the nontrivial part of) the slack matrix of P_M *coincides* with the ($\#\exists{-}1$)-game. In summary, we have the following barrier to proving lower bounds against explicit sparse paving matroids M: *any lower bound on* $\mathrm{xc}(P_M)$ *would imply explicit circuit depth lower bounds.*

Bibliography

[Aar05] Scott Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society A*, 461(2063):3473–3482, 2005. doi:10.1098/rspa.2005.1546.

[ABB$^+$16a] Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in query complexity based on pointer functions. In *Proceedings of the 48th Symposium on Theory of Computing (STOC)*, pages 800–813. ACM, 2016. doi:10.1145/2897518.2897524.

[ABB$^+$16b] Anurag Anshu, Aleksandrs Belovs, Shalev Ben-David, Mika Göös, Rahul Jain, Robin Kothari, Troy Lee, and Miklos Santha. Separations in communication complexity using cheat sheets and information complexity. In *Proceedings of the 57th Symposium on Foundations of Computer Science (FOCS)*, 2016. To appear. URL: http://eccc.hpi-web.de/report/2016/072/.

[ABK16] Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the 48th Symposium on Theory of Computing (STOC)*, pages 863–876. ACM, 2016. doi:10.1145/2897518.2897644.

[AC08] Anil Ada and Arkadev Chattopadhyay. Multiparty communication complexity of disjointness. Technical Report TR08-002, Electronic Colloquium on Computational Complexity (ECCC), 2008. URL: http://eccc.hpi-web.de/report/2008/002/.

[ACFN12] Anil Ada, Arkadev Chattopadhyay, Omar Fawzi, and Phuong Nguyen. The NOF multiparty communication complexity of composed functions. In *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 7391 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2012. doi:10.1007/978-3-642-31594-7_2.

[ACR$^+$10] Andris Ambainis, Andrew Childs, Ben Reichardt, Robert Špalek, and Shengyu Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM Journal on Computing*, 39(6):2513–2530, 2010. doi:10.1137/080712167.

[AKK15] Andris Ambainis, Martins Kokainis, and Robin Kothari. Nearly optimal separations between communication (or query) complexity and partitions. In *Proceedings of the 31st Conference on Computational Complexity (CCC)*, pages 4:1–4:14. Schloss Dagstuhl, 2015. doi:10.4230/LIPIcs.CCC.2016.4.

[Alo03] Noga Alon. Problems and results in extremal combinatorics–I. *Discrete Mathematics*, 273(1–3):31–53, 2003. doi:10.1016/S0012-365X(03)00227-9.

[Ama11a] Kazuyuki Amano. Bounding the randomized decision tree complexity of read-once boolean functions. In *Proceedings of the 22nd Symposium on Discrete Algorithms (SODA)*, pages 1729–1744. SIAM, 2011.

[Ama11b] Kazuyuki Amano. On directional vs. general randomized decision tree complexity for read-once formulas. *Chicago Journal of Theoretical Computer Science*, 2011(3), 2011. doi:10.4086/cjtcs.2011.003.

[Ama14a] Kazuyuki Amano. Researching the complexity of boolean functions with computers. *Bulletin of EATCS*, 101:64–91, 2014. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/181.

[Ama14b] Kazuyuki Amano. Some improved bounds on communication complexity via new decomposition of cliques. *Discrete Applied Mathematics*, 166(0):249–254, 2014. doi:10.1016/j.dam.2013.09.015.

[AT14] David Avis and Hans Raj Tiwary. On the extension complexity of combinatorial polytopes. *Mathematical Programming*, 153(1):95–115, 2014. doi:10.1007/s10107-014-0764-2.

[AUY83] Alfred Aho, Jeffrey Ullman, and Mihalis Yannakakis. On notions of information transfer in VLSI circuits. In *Proceedings of the 15th Symposium on Theory of Computing (STOC)*, pages 133–139. ACM, 1983. doi:10.1145/800061.808742.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Transactions on Computation Theory*, 1(1), 2009. doi:10.1145/1490270.1490272.

[Bar89] David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989. doi:10.1016/0022-0000(89)90037-8.

[BBI12] Paul Beame, Christopher Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 213–232, New York, NY, USA, 2012. ACM. doi:10.1145/2213977.2213999.

[BdW02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

[BEGJ00] Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computing*, 30(5):1462–1484, 2000. doi:10.1137/S0097539799352474.

[Bel06] Aleksandrs Belovs. Non-intersecting complexity. In *Proceedings of the 32nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pages 158–165. Springer, 2006. doi:10.1007/11611257_13.

[BEO$^+$13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *Proceedings of the 54th Symposium on Foundations of Computer Science (FOCS)*, pages 668–677. IEEE, 2013. doi:10.1109/FOCS.2013.77.

[BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *Proceedings of the 27th Symposium on Foundations of Computer Science (FOCS)*, pages 337–347. IEEE, 1986. doi:10.1109/SFCS.1986.15.

[BGKL03] László Babai, Anna Gál, Peter G. Kimmel, and Satyanarayana V. Lokam. Communication complexity of simultaneous messages. *SIAM Journal on Computing*, 33(1):137–166, 2003. doi:10.1137/S0097539700375944.

[BGM06] Elmar Böhler, Christian Glaßer, and Daniel Meister. Error-bounded probabilistic computations between MA and AM. *Journal of Computer and System Sciences*, 72(6):1043–1076, 2006. doi:10.1016/j.jcss.2006.05.001.

[BHP10] Paul Beame, Trinh Huynh, and Toniann Pitassi. Hardness amplification in proof complexity. In *Proceedings of the 42nd Symposium on Theory of Computing (STOC)*, pages 87–96. ACM, 2010. doi:10.1145/1806689.1806703.

[BJKS04] Ziv Bar-Yossef, T.S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004. doi:10.1016/j.jcss.2003.11.006.

[BL92] Paul Beame and Joan Lawry. Randomized versus nondeterministic communication complexity. In *Proceedings of the 24th Symposium on Theory of Computing (STOC)*, pages 188–199. ACM, 1992. doi:10.1145/129712.129732.

[BLT14] Nicolas Bousquet, Aurélie Lagoutte, and Stéphan Thomassé. Clique versus independent set. *European Journal of Combinatorics*, 40(0):73–92, 2014. doi:10.1016/j.ejc.2014.02.003.

[BM13] Mark Braverman and Ankur Moitra. An information complexity approach to extended formulations. In *Proceedings of the 45th Symposium on Theory of Computing (STOC)*, pages 161–170. ACM, 2013. doi:10.1145/2488608.2488629.

[BNS92] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time–space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992. doi:10.1016/0022-0000(92)90047-M.

[BNT13] Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus (extended abstract). In *Proceedings of the 45th Symposium on Theory of Computing (STOC)*, pages 813–822. ACM, 2013. doi:10.1145/2488608.2488711.

[BP13] Gábor Braun and Sebastian Pokutta. Common information and unique disjointness. In *Proceedings of the 54th Symposium on Foundations of Computer Science (FOCS)*, pages 688–697. IEEE, 2013. doi:10.1109/FOCS.2013.79.

[BP15] Gábor Braun and Sebastian Pokutta. The matching polytope does not admit fully-polynomial size relaxation schemes. In *Proceedings of the 26th Symposium on Discrete Algorithms (SODA)*, pages 837–846. ACM–SIAM, 2015. doi:10.1137/1.9781611973730.57.

[BPS07] Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for Lovász–Schrijver systems and beyond follow from multiparty communication complexity. *SIAM Journal on Computing*, 37(3):845–869, 2007. doi:10.1137/060654645.

[BPSW06] Paul Beame, Toniann Pitassi, Nathan Segerlind, and Avi Wigderson. A strong direct product theorem for corruption and the multiparty communication complexity of disjointness. *Computational Complexity*, 15(4):391–432, 2006. doi:10.1007/s00037-007-0220-2.

[BR14] Mark Braverman and Anup Rao. Information equals amortized communication. *IEEE Transactions on Information Theory*, 60(10):6058–6069, 2014. doi:10.1109/TIT.2014.2347282.

[Bra12] Mark Braverman. Interactive information complexity. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 505–524. ACM, 2012. doi:10.1145/2213977.2214025.

[BSN11] Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions (extended abstract). In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS)*, pages 401–416. Tsinghua University Press, 2011. arXiv:1008.1789.

[BSW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001. doi:10.1145/375827.375835.

[BT15] Mark Bun and Justin Thaler. Dual lower bounds for approximate degree and Markov–Bernstein inequalities. *Information and Computation*, 243:2–25, 2015. doi:http://dx.doi.org/10.1016/j.ic.2014.12.003.

[BVdW07] Harry Buhrman, Nikolai Vereshchagin, and Ronald de Wolf. On computation and communication with small bias. In *Proceedings of the 22nd Conference on Computational Complexity (CCC)*, pages 24–32. IEEE, 2007. doi:10.1109/CCC.2007.18.

[BW15a] Mark Braverman and Omri Weinstein. An interactive information odometer and applications. In *Proceedings of the 47th Symposium on Theory of Computing (STOC)*, pages 341–350. ACM, 2015. doi:10.1145/2746539.2746548.

[BW15b] Mark Braverman and Omri Weinstein. Personal communication, 2015.

[CCZ10] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Extended formulations in combinatorial optimization. *4OR*, 8(1):1–48, 2010. doi:10.1007/s10288-010-0122-z.

[CCZ14] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming*. Springer, 2014. doi:10.1007/978-3-319-11008-0.

[CFIK03] Ronald Cramer, Serge Fehr, Yuval Ishai, and Eyal Kushilevitz. Efficient multi-party computation over rings. In *Proceedings of the 22nd International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, volume 2656 of *Lecture Notes in Computer Science*, pages 596–613. Springer, 2003. doi:10.1007/3-540-39200-9_37.

[CFL83] Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multi-party protocols. In *Proceedings of the 15th Symposium on Theory of Computing (STOC)*, pages 94–99. ACM, 1983. doi:10.1145/800061.808737.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988. doi:10.1137/0217015.

[Cha08] Arkadev Chattopadhyay. *Circuits, Communication and Polynomials*. PhD thesis, McGill University, 2008.

[Cha13] Siu Man Chan. Just a pebble game. In *Proceedings of the 28th Conference on Computational Complexity (CCC)*, pages 133–143, 2013. doi:10.1109/CCC.2013.22.

[Chv73] Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973. doi:10.1016/0012-365X(73)90167-2.

[CKW12] Amit Chakrabarti, Ranganath Kondapally, and Zhenghui Wang. Information complexity versus corruption and applications to orthogonality and gap-hamming. In *Proceedings of the 16th International Workshop on Randomization and Computation (RANDOM)*, pages 483–494. Springer, 2012. doi:10.1007/978-3-642-32512-0_41.

[Cle91] Richard Cleve. Towards optimal simulations of formulas by bounded-width programs. *Computational Complexity*, 1(1):91–105, 1991. doi:10.1007/BF01200059.

[CLRS13] Siu On Chan, James Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. In *Proceedings of the 54th Symposium on Foundations of Computer Science (FOCS)*, pages 350–359. IEEE, 2013. Latest version: arXiv:1309.0563v3.

[CMM09] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Integrality gaps for Sherali–Adams relaxations. In *Proceedings of the 41st Symposium on Theory of Computing (STOC)*, pages 283–292. ACM, 2009. doi:10.1145/1536414.1536455.

[Coo74] Stephen A. Cook. An observation on time-storage trade off. *Journal of Computer and System Sciences*, 9(3):308–316, 1974. doi:10.1016/S0022-0000(74)80046-2.

[CP10] Arkadev Chattopadhyay and Toniann Pitassi. The story of set disjointness. *SIGACT News*, 41(3):59–85, 2010. doi:10.1145/1855118.1855133.

[CP12] Siu Man Chan and Aaron Potechin. Tight bounds for monotone switching networks via Fourier analysis. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 495–504. ACM, 2012. doi:10.1145/2213977.2214024.

[CR12] Amit Chakrabarti and Oded Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. *SIAM Journal on Computing*, 41(5):1299–1317, 2012. doi:10.1137/120861072.

[CRR$^+$96] Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, and Prasoon Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, 1996. doi:10.1007/BF01270385.

[CSWY01] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proceedings of the 42nd Symposium on Foundations of Computer Science (FOCS)*, pages 270–278. IEEE, 2001. doi:10.1109/SFCS.2001.959901.

[Die10] Reinhard Diestel. *Graph Theory*. Springer, 4th edition, 2010. URL: http://diestel-graph-theory.com/.

[Dru09] Andrew Drucker. Multitask efficiencies in the decision tree model. In *Proceedings of the 24th Conference on Computational Complexity (CCC)*, pages 286–297. IEEE, 2009. doi:10.1109/CCC.2009.33.

[EIRS01] Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jiří Sgall. Communication complexity towards lower bounds on circuit depth. *Computational Complexity*, 10(3):210–246, 2001. doi:10.1007/s00037-001-8195-x.

[Fen03] Stephen Fenner. PP-lowness and a simple definition of AWPP. *Theory of Computing Systems*, 36(2):199–212, 2003. doi:10.1007/s00224-002-1089-8.

[FFGT14] Yuri Faenza, Samuel Fiorini, Roland Grappe, and Hans Raj Tiwary. Extended formulations, nonnegative factorizations, and randomized communication protocols. *Mathematical Programming*, 153(1):75–94, 2014. doi:10.1007/s10107-014-0755-3.

[FGJ$^+$16] Magnus Find, Mika Göös, Matti Järvisalo, Petteri Kaski, Mikko Koivisto, and Janne H. Korhonen. Separating OR, SUM, and XOR circuits. *Journal of Computer and System Sciences*, 82(5):793–801, 2016. doi:10.1016/j.jcss.2016.01.001.

[FGK$^+$14] Pierre Fraigniaud, Mika Göös, Amos Korman, Merav Parter, and David Peleg. Randomized distributed decision. *Distributed Computing*, 27(6):419–434, 2014. doi:10.1007/s00446-014-0211-x.

[FGKS13] Pierre Fraigniaud, Mika Göös, Amos Korman, and Jukka Suomela. What can be decided locally without identifiers? In *Proceedings of the 32nd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 157–165. ACM, 2013. doi:10.1145/2484239.2484264.

[FGL$^+$96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996. doi:10.1145/226643.226652.

[FKN94] Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation. In *Proceedings of the 26th Symposium on Theory of Computing (STOC)*, pages 554–563. ACM, 1994. doi:10.1145/195058.195408.

[FMP$^+$15] Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *Journal of the ACM*, 62(2):17:1–17:23, 2015. doi:10.1145/2716307.

[FPRC13] Yuval Filmus, Toniann Pitassi, Robert Robere, and Stephen A. Cook. Average case lower bounds for monotone switching networks. In *Proceedings of the 54th Symposium on Foundations of Computer Science (FOCS)*, pages 598–607. ACM, 2013. doi:10.1109/FOCS.2013.70.

[Fri01] Alan Frieze. Edge-disjoint paths in expander graphs. *SIAM Journal on Computing*, 30(6):1790–1801, 2001. doi:10.1137/S0097539700366103.

[FZ00] Alan Frieze and Lei Zhao. Optimal construction of edge-disjoint paths in random regular graphs. *Combinatorics, Probability, & Computing*, 9:241–263, 4 2000. doi:10.1017/S0963548300004284.

[Gál01] Anna Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10(4):277–296, 2001. doi:10.1007/s000370100001.

[GH92] Mikael Goldmann and Johan Håstad. A simple lower bound for monotone clique using a communication game. *Information Processing Letters*, 41(4):221–226, 1992. doi:10.1016/0020-0190(92)90184-W.

[GHL$^+$16] Mika Göös, Juho Hirvonen, Reut Levi, Moti Medina, and Jukka Suomela. Non-local probes do not help with many graph problems. In *Proceedings of 30th Symposium on Distributed Computing (DISC)*, 2016. To appear. arXiv:1512.05411.

[GHS14a] Mika Göös, Juho Hirvonen, and Jukka Suomela. Linear-in-Δ lower bounds in the LOCAL model. In *Proceedings of the 33rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 86–95. ACM, 2014. doi:10.1145/2611462.2611467.

[GHS14b] Mika Göös, Juho Hirvonen, and Jukka Suomela. Lower bounds for local approximation. *Journal of the ACM*, 60(5):175–184, 2014. doi:10.1145/2528405.

[GJ16] Mika Göös and T.S. Jayram. A composition theorem for conical juntas. In *Proceedings of the 31st Conference on Computational Complexity (CCC)*, pages 5:1–5:16. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.CCC.2016.5.

[GJPW15] Mika Göös, T.S. Jayram, Toniann Pitassi, and Thomas Watson. Randomized communication vs. partition number. Technical Report TR15-169, Electronic Colloquium on Computational Complexity (ECCC), 2015. URL: http://eccc.hpi-web.de/report/2015/169/.

[GJW16] Mika Göös, Rahul Jain, and Thomas Watson. Extension complexity of independent set polytopes. In *Proceedings of the 57th Symposium on Foundations of Computer Science (FOCS)*, 2016. To appear. URL: http://eccc.hpi-web.de/report/2016/070/.

[GL14] Dmitry Gavinsky and Shachar Lovett. En route to the log-rank conjecture: New reductions and equivalent formulations. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 514–524. Springer, 2014. doi:10.1007/978-3-662-43948-7_43.

[GLM$^+$15] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. In *Proceedings of the 47th Symposium on Theory of Computing (STOC)*, pages 257–266. ACM, 2015. doi:10.1145/2746539.2746596.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. doi:10.1016/0022-0000(84)90070-9.

[GMPT10] Konstantinos Georgiou, Avner Magen, Toniann Pitassi, and Iannis Tourlakis. Integrality gaps of $2 - o(1)$ for vertex cover SDPs in the Lovász–Schrijver hierarchy. *SIAM Journal on Computing*, 39(8):3553–3570, 2010. doi:10.1137/080721479.

[Gom58] Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of AMS*, 64:275–278, 1958.

[Göö15] Mika Göös. Lower bounds for clique vs. independent set. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, pages 1066–1076. IEEE, 2015. doi:10.1109/FOCS.2015.69.

[GP14] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the 46th Symposium on Theory of Computing (STOC)*, pages 847–856. ACM, 2014. doi:10.1145/2591796.2591838.

[GPT15] Nicola Galesi, Pavel Pudlák, and Neil Thapen. The space complexity of cutting planes refutations. In *Proceedings of the 30th Conference on Computational Complexity (CCC)*, pages 433–447. Schloss Dagstuhl, 2015. doi:10.4230/LIPIcs.CCC.2015.433.

[GPW15] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS)*, pages 1077–1088. IEEE, 2015. doi:10.1109/FOCS.2015.70.

[GPW16a] Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 86:1–86:15. Schloss Dagstuhl, 2016. doi:10.4230/LIPIcs.ICALP.2016.86.

[GPW16b] Mika Göös, Toniann Pitassi, and Thomas Watson. Zero-information protocols and unambiguity in Arthur–Merlin communication. *Algorithmica*, 2016. Online First. doi:10.1007/s00453-015-0104-9.

[GR15] Tom Gur and Ran Raz. Arthur–Merlin streaming complexity. *Information and Computation*, 243:145–165, 2015. doi:10.1016/j.ic.2014.12.011.

[Gri01] Dima Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1–2):613–622, 2001. doi:10.1016/S0304-3975(00)00157-2.

[Gro09] André Gronemeier. Asymptotically optimal lower bounds on the NIH-multi-party information complexity of the AND-function and disjointness. In *Proceedings of* *the 26th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 505–516. Schloss Dagstuhl, 2009. doi:10.4230/LIPIcs.STACS.2009.1846.

[GS10] Dmitry Gavinsky and Alexander Sherstov. A separation of NP and coNP in multiparty communication complexity. *Theory of Computing*, 6(1):227–245, 2010. doi:10.4086/toc.2010.v006a010.

[GS13] Mika Göös and Jukka Suomela. No sublogarithmic-time approximation scheme for bipartite vertex cover. *Distributed Computing*, 27(6):435–443, 2013. doi:10.1007/s00446-013-0194-z.

[GS16] Mika Göös and Jukka Suomela. Locally checkable proofs. *Theory of Computing*, 2016. To appear. Preliminary conference version in PODC'11. doi:10.1145/1993806.1993829.

[GW14] Mika Göös and Thomas Watson. Communication complexity of set-disjointness for all probabilities. In *Proceedings of the 18th International Workshop on Randomization and Computation (RANDOM)*, pages 721–736. Schloss Dagstuhl, 2014. doi:10.4230/LIPIcs.APPROX-RANDOM.2014.721.

[HHT97] Yenjo Han, Lane Hemaspaandra, and Thomas Thierauf. Threshold computation and cryptographic security. *SIAM Journal on Computing*, 26(1):59–78, 1997. doi:10.1137/S0097539792240467.

[HJ13] Prahladh Harsha and Rahul Jain. A strong direct product theorem for the tribes function via the smooth-rectangle bound. In *Proceedings of the 33rd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 141–152. Schloss Dagstuhl, 2013. doi:10.4230/LIPIcs.FSTTCS.2013.141.

[HN12] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time–space trade-offs in proof complexity. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 233–248. ACM, 2012. doi:10.1145/2213977.2214000.

[Hru12] Pavel Hrubeš. On the nonnegative rank of distance matrices. *Information Processing Letters*, 112(11):457–461, 2012. doi:10.1016/j.ipl.2012.02.009.

[HS12] Hao Huang and Benny Sudakov. A counterexample to the Alon–Saks–Seymour conjecture and related problems. *Combinatorica*, 32(2):205–219, 2012. doi:10.1007/s00493-012-2746-4.

[IPU94] Russell Impagliazzo, Toniann Pitassi, and Alasdair Urquhart. Upper and lower bounds for tree-like cutting planes proofs. In *Proceedings of the 9th Symposium on Logic in Computer Science (LICS)*, pages 220–228. IEEE, 1994. doi:10.1109/LICS.1994.316069.

[JK10] Rahul Jain and Hartmut Klauck. The partition bound for classical communication complexity and query complexity. In *Proceedings of the 25th Conference on Computational Complexity (CCC)*, pages 247–258. IEEE, 2010. doi:10.1109/CCC.2010.31.

[JKR09] T.S. Jayram, Swastik Kopparty, and Prasad Raghavendra. On the communication complexity of read-once AC^0 formulae. In *Proceedings of the 24th Conference on Computational Complexity (CCC)*, pages 329–340, 2009. doi:10.1109/CCC.2009.39.

[JKS03] T.S. Jayram, Ravi Kumar, and D. Sivakumar. Two applications of information complexity. In *Proceedings of the 35th Symposium on Theory of Computing (STOC)*, pages 673–682. ACM, 2003. doi:10.1145/780542.780640.

[JKS10] Rahul Jain, Hartmut Klauck, and Miklos Santha. Optimal direct sum results for deterministic and randomized decision tree complexity. *Information Processing Letters*, 110(20):893–897, 2010. doi:10.1016/j.ipl.2010.07.020.

[JKZ10] Rahul Jain, Hartmut Klauck, and Shengyu Zhang. Depth-independent lower bounds on the communication complexity of read-once boolean formulas. In *Proceedings of the 16th International Computing and Combinatorics Conference (COCOON)*, pages 54–59. Springer, 2010. doi:10.1007/978-3-642-14031-0_8.

[JLV14] Rahul Jain, Troy Lee, and Nisheeth Vishnoi. A quadratically tight partition bound for classical communication complexity and query complexity. Technical report, arXiv, 2014. arXiv:1401.4512.

[Joh01] Jan Johannsen. Depth lower bounds for monotone semi-unbounded fan-in circuits. *RAIRO - Theoretical Informatics and Applications*, 35:277–286, 2001. doi:10.1051/ita:2001120.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012.

[JY12] Rahul Jain and Penghui Yao. A strong direct product theorem in terms of the smooth rectangle bound. Technical report, arXiv, 2012. arXiv:1209.0263.

[Kai11] Volker Kaibel. Extended formulations in combinatorial optimization. Technical report, arXiv, 2011. arXiv:1104.1023.

[Kla03] Hartmut Klauck. Rectangle size bounds and threshold covers in communication complexity. In *Proceedings of the 18th Conference on Computational Complexity (CCC)*, pages 118–134. IEEE, 2003. doi:10.1109/CCC.2003.1214415.

[Kla07] Hartmut Klauck. Lower bounds for quantum communication complexity. *SIAM Journal on Computing*, 37(1):20–46, 2007. doi:10.1137/S0097539702405620.

[Kla10] Hartmut Klauck. A strong direct product theorem for disjointness. In *Proceedings of the 42nd Symposium on Theory of Computing (STOC)*, pages 77–86. ACM, 2010. doi:10.1145/1806689.1806702.

[Kla11] Hartmut Klauck. On Arthur Merlin games in communication complexity. In *Proceedings of the 26th Conference on Computational Complexity (CCC)*, pages 189–199. IEEE, 2011. doi:10.1109/CCC.2011.33.

[KLdW15] Jedrzej Kaniewski, Troy Lee, and Ronald de Wolf. Query complexity in expectation. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 761–772. Springer, 2015. doi:10.1007/978-3-662-47672-7_62.

[KLL$^+$12] Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jérémie Roland, and David Xiao. Lower bounds on information complexity via zero-communication protocols and applications. In *Proceedings of the 53rd Symposium on Foundations of Computer Science (FOCS)*, pages 500–509. IEEE, 2012. doi:10.1109/FOCS.2012.68.

[KLL$^+$15] Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jérémie Roland, and David Xiao. Lower bounds on information complexity via zero-communication protocols and applications. *SIAM Journal on Computing*, 44(5):1550–1572, 2015. doi:10.1137/130928273.

[KLO99] Eyal Kushilevitz, Nathan Linial, and Rafail Ostrovsky. The linear-array conjecture in communication complexity is false. *Combinatorica*, 19(2):241–254, 1999. doi:10.1007/s004930050054.

[KLTT15] Hartmut Klauck, Troy Lee, Dirk Oliver Theis, and Rekha Thomas. Limitations of convex programming: Lower bounds on extended formulations and factorization ranks. Dagstuhl Reports 15082, 2015. doi:10.4230/DagRep.5.2.109.

[KMSY14] Gillat Kol, Shay Moran, Amir Shpilka, and Amir Yehudayoff. Approximate nonnegative rank is equivalent to the smooth rectangle bound. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 701–712. Springer, 2014. doi:10.1007/978-3-662-43948-7_58.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[KRS15] Robin Kothari, David Racicot-Desloges, and Miklos Santha. Separating decision tree complexity from subcube partition complexity. In *Proceedings of the 19th International Workshop on Randomization and Computation (RANDOM)*, pages 915–930. Schloss Dagstuhl, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.915.

[KS92] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992. doi:10.1137/0405044.

[Kus94] Eyal Kushilevitz. Unpublished. Cited in [NW95], 1994.

[KW88] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the 20th Symposium on Theory of Computing (STOC)*, pages 539–550. ACM, 1988. doi:10.1145/62212.62265.

[KW09a] Eyal Kushilevitz and Enav Weinreb. The communication complexity of set-disjointness with small sets and 0-1 intersection. In *Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS)*, pages 63–72, 2009. doi:10.1109/FOCS.2009.15.

[KW09b] Eyal Kushilevitz and Enav Weinreb. On the complexity of communication complexity. In *Proceedings of the 41st Symposium on Theory of Computing (STOC)*, pages 465–474, 2009. doi:10.1145/1536414.1536479.

[KW14] Volker Kaibel and Stefan Weltge. A short proof that the extension complexity of the correlation polytope grows exponentially. *Discrete & Computational Geometry*, 53(2):397–401, 2014. doi:10.1007/s00454-014-9655-9.

[Las01] Jean B. Lasserre. An explicit SDP relaxation for nonlinear 0-1 programs. In *Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 2081 of *Lecture Notes in Computer Science*, pages 293–303. Springer, 2001. doi:10.1007/3-540-45535-3_23.

[Law93] Joan Lawry. *Communication complexity: Iterative techniques for lower bounds.* PhD thesis, University of Washington, 1993. UW-CSE-93-3-04.

[Leo13] Nikos Leonardos. An improved lower bound for the randomized decision tree complexity of recursive majority. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 696–708. Springer, 2013. doi:10.1007/978-3-642-39206-1_59.

[LNNW95] László Lovász, Moni Naor, Ilan Newman, and Avi Wigderson. Search problems in the decision tree model. *SIAM Journal on Discrete Mathematics*, 8(1):119–132, 1995. doi:10.1137/S0895480192233867.

[LNPV06] Itamar Landau, Asaf Nachmias, Yuval Peres, and Sithparran Vanniasegaram. The lower bound for evaluating a recursive ternary majority function: an entropy-free proof. Technical report, U.C. Berkeley, 2006.

[Lok01] Satyanarayana Lokam. Spectral methods for matrix rigidity with applications to size-depth trade-offs and communication complexity. *Journal of Computer and System Sciences*, 63(3):449–473, 2001. doi:10.1006/jcss.2001.1786.

[Lok09] Satyanarayana Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1–2):1–155, 2009. doi:10.1561/0400000011.

[Lov93] László Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdős is eighty*, 2:1–46, 1993.

[Lov14] Shachar Lovett. Recent advances on the log-rank conjecture in communication complexity. *Bulletin of EATCS*, 1(112), 2014. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/260/245.

[LPS88] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988. doi:10.1007/BF02126799.

[LRS15] James Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the 47th Symposium on Theory of Computing (STOC)*, pages 567–576. ACM, 2015. doi:10.1145/2746539.2746599.

[LS88] László Lovász and Michael Saks. Lattices, Möbius functions and communication complexity. In *Proceedings of the 29th Symposium on Foundations of Computer Science (FOCS)*, pages 81–90. IEEE, 1988. doi:10.1109/SFCS.1988.21924.

[LS91] László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0–1 optimization. *SIAM Journal on Optimization*, 1(2):166–190, 1991. doi:10.1137/0801013.

[LS07] Troy Lee and Adi Shraibman. *Lower Bounds in Communication Complexity*, volume 3. Now Publishers, 2007. doi:10.1561/0400000040.

[LS09a] Troy Lee and Adi Shraibman. An approximation algorithm for approximation rank. In *Proceedings of the 24th Conference on Computational Complexity (CCC)*, pages 351–357. IEEE, 2009. doi:10.1109/CCC.2009.25.

[LS09b] Troy Lee and Adi Shraibman. Disjointness is hard in the multiparty number-on-the-forehead model. *Computational Complexity*, 18(2):309–336, 2009. doi:10.1007/s00037-009-0276-2.

[LS10] Nikos Leonardos and Michael Saks. Lower bounds on the randomized communication complexity of read-once functions. *Computational Complexity*, 19(2):153–181, 2010. doi:10.1007/s00037-010-0292-2.

[LSŠ08] Troy Lee, Adi Shraibman, and Robert Špalek. A direct product theorem for discrepancy. In *Proceedings of the 23rd Conference on Computational Complexity (CCC)*, pages 71–80. IEEE, 2008. doi:10.1109/CCC.2008.25.

[LZ10] Troy Lee and Shengyu Zhang. Composition theorems in communication complexity. In *Proceedings of the 37th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 475–489. Springer, 2010. doi:10.1007/978-3-642-14165-2_41.

[Mid04] Gatis Midrijānis. Exact quantum query complexity for total boolean functions. Technical report, arXiv, 2004. arXiv:quant-ph/0403168.

[MNS$^+$15] Frédéric Magniez, Ashwin Nayak, Miklos Santha, Jonah Sherman, Gábor Tardos, and David Xiao. Improved bounds for the randomized decision tree complexity of recursive majority. *Random Structures & Algorithms*, 2015. In press. doi:10.1002/rsa.20598.

[MS15] Sagnik Mukhopadhyay and Swagato Sanyal. Towards better separation between deterministic and randomized query complexity. In *Proceedings of the 35th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 206–220. Schloss Dagstuhl, 2015. doi:10.4230/LIPIcs.FSTTCS.2015.206.

[Mur71] Saburo Muroga. *Threshold logic and its applications*. John Wiley & Sons, 1971.

[New91] Ilan Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67–71, 1991. doi:10.1016/0020-0190(91)90157-D.

[Nor13] Jakob Nordström. Pebble games, proof complexity, and time-space trade-offs. *Logical Methods in Computer Science*, 9(3:15):1–63, 2013. doi:10.2168/LMCS-9(3:15)2013.

[Nor15] Jakob Nordström. New wine into old wineskins: A survey of some pebbling classics with supplemental results. Technical report, KTH Royal Institute of Technology, 2015.

[NS94] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994. doi:10.1007/BF01263419.

[NW95] Noam Nisan and Avi Wigderson. On rank vs. communication complexity. *Combinatorica*, 15(4):557–565, 1995. doi:10.1007/BF01192527.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. URL: http://www.analysisofbooleanfunctions.org.

[Oli15] Igor Oliveira. *Unconditional Lower Bounds in Complexity Theory*. PhD thesis, Columbia University, 2015. doi:10.7916/D8ZP45KT.

[OZ13] Ryan O'Donnell and Yuan Zhou. Approximability and proof complexity. In *Proceedings of the 24th Symposium on Discrete Algorithms (SODA)*, pages 1537–1556. SIAM, 2013.

[Pip90] Nicholas Pippenger. Communication networks. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 15, pages 805–833. Elsevier, 1990.

[PS86] Ramamohan Paturi and Janos Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986. doi:10.1016/0022-0000(86)90046-2.

[PV13] Sebastian Pokutta and Mathieu Van Vyve. A note on the extension complexity of the knapsack polytope. *Operations Research Letters*, 41(4):347–350, 2013. doi:10.1016/j.orl.2013.03.010.

[Raz89] Alexander Razborov. On rigid matrices. Technical report, Steklov Mathematical Institute, 1989. In Russian.

[Raz90] Alexander Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1):81–93, 1990. doi:10.1007/BF02122698.

[Raz92] Alexander Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992. doi:10.1016/0304-3975(92)90260-M.

[Raz03] Alexander Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya: Mathematics*, 67(1):145–159, 2003. doi:10.1070/IM2003v067n01ABEH000422.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. doi:10.1007/s004930050062.

[Rot12] Thomas Rothvoß. Some 0/1 polytopes need exponential size extended formulations. *Mathematical Programming*, 142(1):255–268, 2012. doi:10.1007/s10107-012-0574-3.

[Rot14] Thomas Rothvoß. The matching polytope has exponential extension complexity. In *Proceedings of the 46th Symposium on Theory of Computing (STOC)*, pages 263–272. ACM, 2014. doi:10.1145/2591796.2591834.

[Rou15] Tim Roughgarden. Communication complexity (for algorithm designers). Technical report, arXiv, 2015. arXiv:1509.06257.

[RS04] Ran Raz and Amir Shpilka. On the power of quantum proofs. In *Proceedings of the 19th Conference on Computational Complexity (CCC)*, pages 260–274. IEEE, 2004. doi:10.1109/CCC.2004.1313849.

[RS10] Alexander Razborov and Alexander Sherstov. The sign-rank of AC^0. *SIAM Journal on Computing*, 39(5):1833–1855, 2010. doi:10.1137/080744037.

[RW92] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM*, 39(3):736–744, 1992. doi:10.1145/146637.146684.

[RY15] Anup Rao and Amir Yehudayoff. Simplified lower bounds on the multiparty communication complexity of disjointness. In *Proceedings of the 30th Conference on Computational Complexity (CCC)*, pages 88–101. Schloss Dagstuhl, 2015. doi:10.4230/LIPIcs.CCC.2015.88.

[RY16] Anup Rao and Amir Yehudayoff. Communication complexity, 2016. In preparation.

[SA90] Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990. doi:10.1137/0403036.

[SA15] Manami Shigeta and Kazuyuki Amano. Ordered biclique partitions and communication complexity problems. *Discrete Applied Mathematics*, 184:248–252, 2015. doi:10.1016/j.dam.2014.10.029.

[San95] Miklos Santha. On the Monte Carlo boolean decision tree complexity of read-once formulae. *Random Structures & Algorithms*, 6(1):75–87, 1995. doi:10.1002/rsa.3240060108.

[Sav02] Petr Savický. On determinism versus unambiquous nondeterminism for decision trees. Technical Report TR02-009, Electronic Colloquium on Computational Complexity (ECCC), 2002. URL: http://eccc.hpi-web.de/report/2002/009/.

[Sch03] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[Sch08] Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th Symposium on Foundations of Computer Science (FOCS)*, pages 593–602. IEEE, 2008. doi:10.1109/FOCS.2008.74.

[Sha03] Ronen Shaltiel. Towards proving strong direct product theorems. *Computational Complexity*, 12(12):1–22, 2003. doi:10.1007/s00037-003-0175-x.

[She08] Alexander Sherstov. Communication lower bounds using dual polynomials. *Bulletin of EATCS*, 95:5993, 2008.

[She09] Alexander Sherstov. Separating AC^0 from depth-2 majority circuits. *SIAM Journal on Computing*, 38(6):2113–2129, 2009. doi:10.1137/08071421X.

[She11a] Alexander Sherstov. The pattern matrix method. *SIAM Journal on Computing*, 40(6):1969–2000, 2011. doi:10.1137/080733644.

[She11b] Alexander Sherstov. The unbounded-error communication complexity of symmetric functions. *Combinatorica*, 31(5):583–614, 2011. doi:10.1007/s00493-011-2580-0.

[She12a] Alexander Sherstov. The communication complexity of gap hamming distance. *Theory of Computing*, 8(1):197–208, 2012. doi:10.4086/toc.2012.v008a008.

[She12b] Alexander Sherstov. The multiparty communication complexity of set disjointness. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, pages 525–548. ACM, 2012. doi:10.1145/2213977.2214026.

[She13a] Alexander Sherstov. Approximating the AND-OR tree. *Theory of Computing*, 9(20):653–663, 2013. doi:10.4086/toc.2013.v009a020.

[She13b] Alexander A. Sherstov. Communication lower bounds using directional derivatives. In *Proceedings of the 45th Symposium on Theory of Computing (STOC)*, pages 921–930. ACM, 2013. doi:10.1145/2488608.2488725.

[She14a] Alexander Sherstov. Breaking the Minsky-Papert barrier for constant-depth circuits. In *Proceedings of the 46th Symposium on Theory of Computing (STOC)*, pages 223–232. ACM, 2014. doi:10.1145/2591796.2591871.

[She14b] Alexander Sherstov. Communication lower bounds using directional derivatives. *Journal of the ACM*, 61(6):1–71, 2014. doi:10.1145/2629334.

[Sni85] Marc Snir. Lower bounds on probabilistic linear decision trees. *Theoretical Computer Science*, 38:69–82, 1985. doi:10.1016/0304-3975(85)90210-5.

[STT07] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. A linear round lower bound for Lovász–Schrijver SDP relaxations of vertex cover. In *Proceedings of the 22nd Conference on Computational Complexity (CCC)*, pages 205–216. IEEE, 2007. doi:10.1109/CCC.2007.2.

[SW86] Michael Saks and Avi Wigderson. Probabilistic boolean decision trees and the complexity of evaluating game trees. In *Proceedings of the 27th Symposium on Foundations of Computer Science (FOCS)*, pages 29–38. IEEE, 1986. doi:10.1109/SFCS.1986.44.

[SZ09] Yaoyun Shi and Yufan Zhu. Quantum communication complexity of block-composed functions. *Quantum Information and Computation*, 9(5–6):444–460, 2009.

[TTV09] Luca Trevisan, Madhur Tulsiani, and Salil Vadhan. Regularity, boosting, and efficiently simulating every high-entropy distribution. In *Proceedings of the 24th*

Conference on Computational Complexity (CCC), pages 126–136. IEEE, 2009. doi:10.1109/CCC.2009.41.

[Tul09] Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *Proceedings of the 41st Symposium on Theory of Computing (STOC)*, pages 303–312. ACM, 2009. doi:10.1145/1536414.1536457.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, 1987. doi:10.1145/7531.8928.

[Vad12] Salil Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1–3):1–336, 2012. doi:10.1561/0400000010.

[Val77] Leslie Valiant. Graph-theoretic arguments in low-level complexity. In *Proceedings of the 6th Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 162–176. Springer, 1977. doi:10.1007/3-540-08353-7_135.

[Vaz86] Umesh Vazirani. *Randomness, Adversaries and Computation.* PhD thesis, University of California, Berkeley, 1986.

[Ver98] Nikolai Vereshchagin. Randomized boolean decision trees: Several remarks. *Theoretical Computer Science*, 207(2):329–342, 1998. doi:10.1016/S0304-3975(98)00071-1.

[Ver99] Nikolai Vereshchagin. Relativizability in complexity theory. In *Provability, Complexity, Grammars*, volume 192 of *AMS Translations, Series 2*, pages 87–172. American Mathematical Society, 1999.

[Vid13] Thomas Vidick. A concentration inequality for the overlap of a vector on a large set, with application to the communication complexity of the gap-hamming-distance problem. *Chicago Journal of Theoretical Computer Science*, 2013(1):1–12, 2013. doi:10.4086/cjtcs.2012.001.

[Vio15] Emanuele Viola. The communication complexity of addition. *Combinatorica*, 35(6):703–747, 2015. doi:10.1007/s00493-014-3078-3.

[Wri13] Steve Wright. Quadratic residues and non-residues in arithmetic progression. *Journal of Number Theory*, 133(7):2398–2430, 2013. doi:10.1016/j.jnt.2013.01.004.

[Wun12] Henning Wunderlich. On a theorem of Razborov. *Computational Complexity*, 21(3):431–477, 2012. doi:10.1007/s00037-011-0021-5.

[Yan91] Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991. doi:10.1016/0022-0000(91)90024-Y.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing. In *Proceedings of the 11th Symposium on Theory of Computing (STOC)*, pages 209–213. ACM, 1979. doi:10.1145/800135.804414.

[Yao83] Andrew Yao. Lower bounds by probabilistic arguments. In *Proceedings of the 24th Symposium on Foundations of Computer Science (FOCS)*, pages 420–428. IEEE, 1983. doi:10.1109/SFCS.1983.30.

[Zha09] Shengyu Zhang. On the tightness of the Buhrman–Cleve–Wigderson simulation. In *Proceedings of the 20th International Symposium on Algorithms and Computation (ISAAC)*, volume 5878 of *Lecture Notes in Computer Science*, pages 434–440. Springer, 2009. doi:10.1007/978-3-642-10631-6_45.

[Zie95] Günter Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer, 1995.