Communication Complexity

Anil Ada

School of Computer Science
McGill University, Montréal

July, 2013

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Ph.D. Science.

Abstract

Communication complexity studies how many bits a certain number of parties need to communicate with each other in order to compute a function whose input is distributed among those parties. Although it is a natural area of investigation based on practical considerations, the main motivation comes from the myriad of applications in theoretical computer science.

This thesis has three main parts, studying three different aspects of communication complexity.

- The first part is concerned with the k-party communication complexity of functions $F : (\{0,1\}^n)^k \to \{0,1\}$ in the 'number on the forehead' (NOF) model. This is a fundamental model in communication complexity with applications in circuit complexity, proof complexity, branching programs and Ramsey theory. In this model we study composed functions $f \circ g$. These functions include most of the well-known and studied functions in communication complexity literature. A major goal is to understand which combinations of f and g lead to hard communication functions. In particular, due to important circuit applications, it is of great interest to understand how powerful the NOF model becomes when the number of parties is $\log n$ or more. Motivated by these goals, we show that there is an efficient $O(\log^3 n)$ cost *simultaneous* protocol for SYM $\circ\, g$ when the number of players is more than $1 + \log n$, SYM is any symmetric function and g is *any function.* This class of functions includes some functions that were previously conjectured to be hard and our result rules this class out for possible very important circuit complexity applications. We also give Ramsey theoretic applications of our efficient protocol.

In the setting of less than $\log n$ many players, we study more closely functions of the form MAJORITY $\circ\, g$, MOD$_m \circ g$, and NOR $\circ\, g$, where the latter two are generalizations of the well-known and studied functions Generalized Inner Product and Disjointness respectively. We characterize the communication complexity of these functions with respect to the choice of g. As the main application of our results, we answer a question posed by Babai et al. (*SIAM Journal on Computing, 33:137–166, 2004*) and determine the communication complexity of MAJORITY $\circ$ QCSB, where QCSB is the "quadratic character of the sum of the bits" function.

- The second part is about Fourier analysis of symmetric boolean functions and its applications in communication complexity and other areas. The spectral norm of a boolean function $f : \{0,1\}^n \to \{0,1\}$ is the sum of the absolute values of its Fourier coefficients. This quantity provides useful upper and lower bounds on the complexity of a function in areas such as communication complexity, learning theory and circuit complexity. We give a combinatorial characterization for the spectral norm of symmetric functions. We show that the logarithm of the spectral norm is of the same order of magnitude as $r(f)\log(n/r(f))$

where $r(f) = \max\{r_0, r_1\}$, and r_0 and r_1 are the smallest integers less than $n/2$ such that $f(x)$ or $f(x) \cdot \textsc{parity}(x)$ is constant for all x with $\sum x_i \in [r_0, n - r_1]$. We present some applications to the decision tree and communication complexity of symmetric functions.

- The third part studies *privacy* in the context of communication complexity: how much information do the players reveal about their input when following a communication protocol? The unattainability of perfect privacy for many functions motivates the study of *approximate* privacy. Feigenbaum et al. (*Proceedings of the 11th Conference on Electronic Commerce, 167–178, 2010*) defined notions of worst-case as well as average-case approximate privacy, and presented several interesting upper bounds, and some open problems for further study. In this thesis, we obtain asymptotically tight bounds on the trade-offs between both the worst-case and average-case approximate privacy of protocols and their communication cost for Vickrey Auction, which is the canonical example of a *truthful* auction. We also prove exponential lower bounds on the approximate privacy of protocols computing the Intersection function, independent of its communication cost. This proves a conjecture of Feigenbaum et al.

Résumé

La complexité de communication étudie combien de bits un groupe de joueurs donné doivent échanger entre eux pour calculer une function dont l'input est distribué parmi les joueurs. Bien que ce soit un domaine de recherche naturel basé sur des considérations pratiques, la motivation principale vient des nombreuses applications théoriques.

Cette thèse comporte trois parties principales, étudiant trois aspects de la complexité de communication.

- La première partie discute le modèle "number on the forehead" (NOF) dans la complexité de communication à plusieurs joueurs. Il s'agit d'un modèle fondamental en complexité de communication, avec des applications à la complexité des circuits, la complexité des preuves, les programmes de branchement et la théorie de Ramsey. Dans ce modèle, nous étudions les fonctions composeés $f \circ g$. Ces fonctions comprennent la plupart des fonctions bien connues qui sont étudiées dans la littérature de la complexité de communication. Un objectif majeur est de comprendre quelles combinaisons de f et g produisent des compo-

sitions qui sont difficiles du point de vue de la communication. En particulier, à cause de l'importance des applications aux circuits, il est intéressant de comprendre la puissance du modèle NOF quand le nombre de joueurs atteint ou dépasse $\log n$. Motivé par ces objectifs nous montrons l'existence d'un protocole *simultané* efficace à k joueurs de coût $O(\log^3 n)$ pour $\textsc{sym} \circ g$ lorsque $k > 1 + \log n$, $\textsc{sym}$ est une function symmétrique quelconque et g est une fonction *arbitraire*. Cette classe de fonctions inclut certaines fonctions qui étaient jusqu'ici présumées être difficiles et notre résultat élimine la possibilité d'utiliser cette classe pour des applications importantes aux circuits. Nous donnons aussi des applications de notre protocole efficace à la théorie de Ramsey.

Dans le contexte où $k \leq \log n$, nous étudions de plus près des fonctions de la forme $\textsc{majority} \circ g$, $\textsc{mod}_m \circ g$ et $\textsc{nor} \circ g$, où les deux derniers cas sont des généralisations des fonctions bien connues et très étudiées Generalized Inner Product et Disjointness respectivement. Nous caractérisons la complexité de communication de ces fonctions par rapport au choix de g. Comme application principale de nos résultats, nous répondons à une question posée par Babai et al (*SIAM Journal on Computing, 33:137–166, 2004*) et nous déterminons la complexité de communication de $\textsc{majority} \circ \textsc{qcsb}$ où $\textsc{qcsb}$ est la function "caractère quadratique de la somme des bits".

- La deuxième partie considère les applications de l'analyse de Fourier des fonctions symmétriques à la complexité de communication et autres domaines. La norme spectrale d'une function booléenne $f : \{0,1\}^n \rightarrow \{0,1\}$ est la somme des valeurs absolues de ses coefficients de Fourier. Ce paramètre procure des bornes supérieures et inférieures utiles pour la complexité de fonctions dans des domaines comme la communication, l'apprentissage ou les circuits. Nous donnons une caractérisation combinatoire pour la norme spectrale des fonctions symmétriques. Nous montrons que le logarithme de la norme spectrale est du même ordre de grandeur que $r(f)\log(n/r(f))$, avec $r(f) = \max\{r_0, r_1\}$ où r_0 et r_1 sont les entiers minimaux plus petits que $n/2$ pour lesquels $f(x)$ ou $f(x) \cdot \textsc{parity}(x)$ est constant pour tout x tel que $\sum x_i \in [r_0, n - r_1]$. Nous présentons quelques applications aux arbres de décision et à la complexité de communication des fonctions symmétriques.

- La troisième partie étudie la *confidentialité* dans le contexte de la complexité de communication: quelle quantité d'information est-ce que les joueurs révèlent sur leur input en suivant un protocole donné? L'inatteignabilité de la confidentialité parfaite pour plusieurs fonctions motivent l'étude de la confidentialité *approximative*. Feigenbaum et al. (*Proceedings of the 11th Conference on Electronic Commerce, 167–178, 2010*) ont défini des notions de confidentialité approximative dans le pire cas et dans le cas moyen, et ont présenté plusieurs bornes supérieures intéressantes ainsi que quelques questions ouvertes. Dans cette thèse, nous obtenons des bornes asymptotiques précises, pour le pire cas aussi bien que pour le cas moyen, sur l'échange entre la confidentialité approximative de protocoles et le coût de communication pour les enchères Vickrey Auction, qui constituent l'exemple canonique d'une enchère *honnête*. Nous démontrons aussi des bornes inférieures exponentielles sur la confidentialité approximative de protocoles calculant la function Intersection, indépendamment du coût de communication. Ceci résout une conjecture de Feigenbaum et al.

Acknowledgements

This thesis came to life with the support of many individuals that I was fortunate enough to meet and interact with. It is my pleasure to acknowledge them here in the opening pages of my thesis.

First and foremost I would like to thank my advisor and friend Denis Thérien. I started my journey of scientific research many years ago with Denis' trust and support. His passion for mathematics was highly contagious and his wisdom very valuable. He has supported me financially over many years. One of the best things about being Denis' student was the chance to attend the annual workshops in Barbados that he organizes. These workshops are without a doubt the best workshops on theoretical computer science.

Second, I would like to thank my co-advisor Hamed Hatami. I have learned a lot from him in the last couple of years of my Ph.D., especially about Fourier analysis and additive combinatorics. I have benefited greatly from the depth of his knowledge and insights.

A very special thanks goes to Arkadev Chattopadhyay. When I first started working in Denis' group, Arkadev was an experienced and very ap-

proachable Ph.D. student. Ever since, Arkadev has been like a co-advisor to me. I have learned an enormous amount from him and my interests in mathematics and theoretical computer science was heavily influenced by him. I can't thank him enough for all the support and encouragement he has given me.

Next, I would like to thank my office mate, partner in crime, László Egri. He is my academic brother and a dear friend. We have had countless discussions on computational complexity theory and mathematics in general, among other things. These discussions and his friendship have taught me a lot and I am grateful for all of it.

My other academic brother is Omar Fawzi. In the second half of my Ph.D. Omar and I started working closely with each other and this has produced two papers presented in this thesis. It is rare to meet someone with very similar scientific interests and therefore I feel extremely fortunate to have met Omar. It is a great joy to work with him and learn about probability, quantum theory, and other interesting things.

During my Ph.D., Phuong Nguyen spent two years at McGill as a postdoc. Since his first day at McGill, we have engaged in countless hours of scientific and non-scientific discussions that helped me grow as a researcher and as a person. I thank him for being such a valuable collaborator and friend.

In the 2010-2011 academic year, I had the opportunity to spend 6 months in University of Toronto and meet the wonderful theory group there. I would like to especially thank Toniann Pitassi for hosting me and for many hours of stimulating discussions. A big part of the work presented in Chapter 6 was done during my stay in Toronto.

I would like to thank all my co-authors: Arkadev Chattopadhyay, Stephen Cook, Omar Fawzi, Lila Fontes, Hamed Hatami, Michal Koucký, Phuong Nguyen and Toniann Pitassi. I learned a lot from the discussions I had with each one of them.

During the first couple of years of my graduate studies, I was lucky to interact with Denis' former students. Even though those years seem like ancient history, their contributions were invaluable. I have already thanked Arkadev. Pascal Tesson was my co-advisor for my M.Sc. thesis; I learned a lot about algebraic automata theory and communication complexity from him. I also thank Mark Mercer who has always been very helpful and eager to share his knowledge.

I would like to thank all the professors in the School of Computer Science, who have either been my teacher, a part of one of my Ph.D. exam committees or organized great seminars that I attended to. I would like single out Luc Devroye who is a fantastic teacher and a great mentor.

School of Computer Science systems and administrative staff have always been very kind and helpful. I thank all of them for their invaluable support, especially Diti Anastasopoulos, Tricia Bernier, Vanessa Bettencourt-Laberge, Andrew Bogecho, Heather Holowathy, Sheryl Morrissey, Kailesh Mussai and Ron Simpson.

This thesis would not have existed without the great times I had with my close friends in Montréal. I will not list their names here but I want them to know that they had a big part in my Ph.D. life and therefore this thesis as well.

Lastly and most importantly, biggest thanks go to my parents Emel Ada and Mesut Ada. My achievements in life are a direct product of their love and support. There are no words to describe how lucky I feel to have them as my anne and baba. This thesis is dedicated to them.

Dedicated to my parents, Emel Ada and Mesut Ada.

Contribution of the thesis

This thesis is based on three papers. The first one is joint work with Arkadev Chattopadhyay, Omar Fawzi, and Phuong Nguyen [ACFN12], and covers Chapters 3 and 4 of this thesis. The second paper is joint work with Omar Fawzi and Hamed Hatami [AFH12], and covers Chapter 5 of this thesis. The third paper is joint work with Arkadev Chattopadhyay, Stephen Cook, Lila Fontes, Michal Koucký, and Toniann Pitassi [ACC$^+$12], and covers Chapter 6 of the thesis. The author of this thesis is the main author of [ACFN12] and [AFH12]. Lila Fontes is the main author of [ACC$^+$12].

Contents

CHAPTER 1

Introduction

Suppose there are two computers and each contains a file. Let's represent the files as bit strings and assume both have length n. How many bits do the computers have to communicate with each other in order to determine if the two files are the same or not? Intuitively one expects that the computers have to compare each bit one by one and therefore they need to exchange n bits. This intuition is essentially correct but how can we rigorously prove it? If we change the model slightly and assume the computers can make randomized decisions, how many bits do they need to communicate in order to determine whether the files are the same or not with 0.0000000000001% probability of error? One might be tempted to think that the best strategy is to sample indices until we find an index where the two files differ or be confident that they are the same. This leads to $\Omega(n)$ bits of communication. However, a more clever protocol requires only $O(\log n)$ bits (we describe this protocol in Section 2.2.2).

The rigorous study of the *communication complexity* of such distributed tasks is an important area of theoretical computer science. A bit more formally, in the two party setting of communication complexity, there are two players (computers) called Alice and Bob who wish to determine the output of a known fixed function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Z}$ on a given input $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Alice gets x and Bob gets y, and since neither of them sees the whole input, they need to communicate with each other in order to compute $F(x, y)$. They do so according to a protocol (algorithm) that they have agreed upon beforehand. The protocol determines whose turn it is to speak and what a player should send. The cost of a protocol is the maximum number of bits communicated, where the maximum is over all possible inputs. The communication complexity of F is the cost of the most efficient protocol that computes F. Here the word *compute* may have different meanings depending on the particular model we are interested in. We might require the protocol to be deterministic and give the correct answer on all inputs. This is called the *deterministic* model. We might allow our protocols to be randomized and err with a small constant probability. This is the *randomized* model. We might allow the protocol to be non-deterministic, where players receive a proof string and engage in a verification process. This is the *non-deterministic* model. We might allow the players to exchange quantum bits and exploit the properties of quantum mechanics. This leads to the quantum analogs of the just mentioned classical models. Suffices to say there are other natural and well studied models.

Unless stated otherwise, we will be dealing with the standard setting of $\mathcal{X} = \mathcal{Y} = \{0, 1\}^n$ and $\mathcal{Z} = \{0, 1\}$. It is easy to see that every function can be computed with $n + 1$ bits of communication: Alice sends her input x to Bob, Bob computes $F(x, y)$ and sends the result to Alice. By convention, we think of functions with at most poly-log(n) communication complexity as being efficiently computable while other functions are viewed as hard functions. In the opening paragraph, the function we considered was the *equality* function, $\mathrm{EQ}(x, y) = 1$ iff $x = y$. While the *equality* function requires $n + 1$ bits of communication in the deterministic model (we will see a proof of this in Chapter 2), there are many functions which can be computed with fewer number of bits. For example, if we consider the *parity* function, $\mathrm{PAR}(x, y) \overset{\text{def}}{=} x_1 \oplus \cdots \oplus x_n \oplus y_1 \oplus \cdots \oplus y_n$, Alice and Bob can compute it with only 2 bits.

Although there is plenty of motivation to study communication complexity for its own sake, over the years communication complexity has placed itself at the core of complexity theory. The general theme behind communication complexity's relation to diverse areas of theoretical computer science can be mildly described as follows. Suppose we are in some setting where we are trying to solve some task while minimizing the use of some "resource" (e.g., we might want to design a small chip or come up with a short proof of a theorem). In many situations, one can show that if we can solve our task using few resources, then we have an efficient communication protocol that computes a certain function. Thus, a lower bound on the communication complexity of the function would lead to a lower bound on the amount of resources needed to solve the task. In other words, many optimization problems contain an implicit *communication bottleneck*, which can be exploited to prove lower bounds.

The above idea may seem too vague at first so let us try to demonstrate it by giving a relatively simple example from an area called *data streaming*. In the streaming model, an algorithm sees a stream $S \in [k]^m$, one symbol at a time, and its goal is to compute some function $f(S)$ (usually k is about $O(\log m)$). Imagine that m is very large and that we cannot afford to store all the symbols we see. Given a small amount of memory, say $O(\log m)$, is there a good approximation algorithm that can approximate $f(S)$ with high probability? The study of such problems is well motivated for instance in the context of IP network traffic analysis, and processing massive data sets in general.

In breakthrough work, Alon, Matias and Szegedy [AMS99] proved lower bounds for the space requirements of streaming algorithms that compute important statistical functions f_k called frequency moments. Let's just consider one of these functions, denoted f_∞, that outputs the number of occurrences of the most frequent element in S. If there is a streaming algorithm for f_∞ that uses small amount of memory, we can devise an efficient protocol for the well known communication problem *disjointness*. Given two n-bit strings x and y, we let $\mathrm{DISJ}(x, y) = 1$ iff there is some i such that $x_i = y_i = 1$. The protocol for disjointness is as follows. Given x, Alice converts her input to a stream $S_x = \{i \mid x_i = 1\}$. Similarly, given y, Bob converts his input to a stream $S_y = \{i \mid y_i = 1\}$. Alice simulates the streaming algorithm on S_x. After all the symbols in S_x are read, she sends the memory contents of the algorithm to Bob.[1] Bob continues to simulate the streaming algorithm with the stream S_y. Observe that $f_\infty(S_x \cdot S_y) = 1$ iff $\mathrm{DISJ}(x, y) = 0$ and $f_\infty(S_x \cdot S_y) = 2$ iff $\mathrm{DISJ}(x, y) = 1$. Note that the communication complexity of the protocol coincides with the memory usage of the streaming algorithm. The known randomized communication complexity lower bound for DISJ now implies a lower bound on the memory requirements of any randomized streaming algorithm computing f_∞.

A natural question that immediately arises is: have we gained anything by reducing the original lower bound problem to a lower bound problem in communication complexity? The answer is an emphatic yes. Communication complexity provides a beautiful mathematical framework to tackle these tough lower bound questions. Given a communication problem $F : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, consider the matrix M_F whose rows are indexed by the elements of $\mathcal{X}$ and columns are indexed by the elements of $\mathcal{Y}$. The (x, y)th entry contains

[1] Alice also sends Bob which state the algorithm is in, but this is only constant number of bits.

$F(x, y)$. This is the matrix representation of the function F. The communication complexity of F can often be characterized or lower bounded by a natural combinatorial measure of M_F such as the partition number, the covering number, discrepancy, etc. These quantities can be studied combinatorially or they can be bounded by well studied algebraic or analytic notions such as the rank, sign rank, approximate rank, a norm, approximate norm, etc. This point of view, for instance, puts very powerful algebraic and analytic tools at our disposal. In the last decade, information theoretic tools have also played a major role in advancing the field.

The above illustrative reduction in the setting of data streaming is by no means an exception. Communication complexity has connections and applications to circuit complexity, time/space trade-offs for Turing Machines, VLSI chips, machine learning, game theory, data structures, proof complexity, pseudorandom generators, pseudorandomness, branching programs, lower bounds for polytopes representing NP-complete problems, quantum computation, etc. It is fair to describe communication complexity as the Swiss Army knife of computational complexity theory.

To see some of the main motivations driving the research in communication complexity, let us give a list of the most famous and important open problems in the area:[2]

1. **The log-rank conjecture:** It is well known that $\log \operatorname{rank}(M_F)$ lower bounds the deterministic communication complexity of F. Is it true that $\log^c \operatorname{rank}(M_F)$, for some absolute constant c, upper bounds the deterministic communication complexity?

2. **Quantum-Classical equivalence:** Is the quantum communication complexity of a function polynomially related to its classical commu-

[2] One might extend this list perhaps by a couple more open problems, depending on their personal preference.

nication complexity?

3. **The $\log n$ barrier:** Find an explicit function that is hard in the multiparty *number on the forehead* model when the number of players is $\log n$ (we describe this model shortly).

4. **Direct sum:** Can we solve two instances of a communication problem more efficiently than solving each instance separately?

The second and third open problems above, and the multiparty 'number on the forehead' (NOF) model in general will be an important part of this thesis. In the multiparty NOF model, there are k players who wish to evaluate a function $F : \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_k \to \{0, 1\}$ on a given input $(x_1, x_2, \ldots, x_k)$, where the input is distributed in a way that Player i sees all x_j with $j \neq i$. This scenario is visualized as x_i being written on the forehead of Player i. The main motivation for this model comes from its important theoretical applications rather than any practical considerations. Note that when $k = 2$, this model is the same as the 2 player model introduced earlier. For $k > 2$, there is a significant overlap of information among the players and this makes the NOF model quite powerful. Despite intense effort, even the 3 player model is far from being well understood and many important problems that have been solved in the 2 player setting remain open for the 3 player setting. All of the challenges of the NOF model, however, are rewarded by the rich set of important applications, for example in branching program lower bounds, boolean circuit complexity, proof complexity and pseudorandom generators.

Another pillar of this thesis is on Fourier analysis of boolean functions and its applications in communication complexity and other areas. Fourier analysis of boolean functions has a myriad of applications in theoretical computer science and it is one of the main tools we use in understanding communication complexity. We will study the Fourier analytic properties of *symmetric* functions with applications in 2 party randomized communication complexity.

Finally the third part of our thesis is about the study of *privacy* in communication complexity. In many scenarios, it is a natural goal to keep the inputs of the players “private” against an outsider or other players while minimizing the number of communicated bits. The study of privacy vs communication complexity is well motivated for instance in auction design and our goal will be to study this trade-off with respect to natural definitions of privacy.

We now give more details about our contributions in the three areas we have mentioned above.

1.1 Motivation and Our Contributions

In this section we will give the motivation behind our work and state our main results in an informal way. This should serve as a summary of the thesis as well as an outline of the remaining chapters. There are 3 subsections. The first deals with the NOF model and covers Chapters 3 and 4 (published in [ACFN12]). The second subsection is about the spectral norm of boolean functions and its applications in 2-party communication complexity and other areas. This subsection covers Chapter 5 (published in [AFH12]). Finally, the third subsection is about privacy concerns in 2-party communication complexity and covers Chapter 6 (published in [ACC$^+$12]).

1.1.1 NOF Model

As mentioned earlier, the NOF model has a number of important theoretical applications and therefore it is one of the most interesting and fundamental models of communication complexity. In this thesis we will focus on two of these applications: boolean circuit complexity and Ramsey theory.

Chapter 3: NOF Communication Complexity of Composed Functions

Recall that P denotes the class of problems computable in polynomial time and NP denotes the class of problems whose solutions can be verified in polynomial time. Boolean circuits constitute a central model of computation, in fact it is considered to be *the* model for non-uniform[3] computation. It is well known that problems in P can be computed by polynomial size circuits, that is circuits with polynomially many AND, OR, and NOT gates. Thus showing that a problem in NP requires superpolynomial size circuits would resolve one of Clay Mathematics Institute's millennium prize problems, i.e., the famous $\mathsf{P} \overset{?}{=} \mathsf{NP}$ problem. Working with circuits rather than Turing Machines is more appealing to mathematicians due to the simple and more natural definition of circuits. Although there has been exciting progress on lower bounds for circuits in the 1980s, no real progress has been made since then. The result that MOD_2 requires exponential size circuits of constant depth remains one of the jewels of complexity theory. Another classic result is exponential size lower bounds for constant depth circuits that are also allowed to have MOD_p gates, where p is a prime. Unfortunately, the current state of affairs is quite embarrassing: we cannot rule out that every function in NP is computable by polynomial size depth 3 circuits composed of only MOD_6 gates!

The class ACC^0 represents functions that are computable by polynomial-size, constant-depth circuits with unbounded fan-in AND, OR, NOT and MOD_m gates. Showing NP is not in ACC^0 is one of the frontiers in complexity

[3] Turing Machine is a uniform model of computation since there is one machine that handles inputs of all lengths. On the other hand circuits are non-uniform because there is a different circuit for each input length.

theory. It is well known that a function in ACC^0 has poly-log(n) k-party deterministic communication complexity, where k is poly-log(n) [HG91, BT94]. In fact the protocol is *simultaneous* where all the players, without interacting, speak once to an external referee who then determines the output based only on the messages s/he receives. Proving that a function in NP requires super-polylogarithmic communication in the simultaneous model for polylogarithmic number of players would result in a major breakthrough. Currently no non-trivial lower bound is known for an explicit function for $k = \log n$ and this has proven to be a formidable barrier (third open problem from earlier).

In this thesis we study this question in the context of *composed* functions. For $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^k \to \{0,1\}$, define

$$f \circ g(x_1, \ldots, x_k) = f(\ldots, g(x_{1,i}, x_{2,i}, \ldots, x_{k,i}), \ldots),$$

where $x_{j,i}$ denotes the ith coordinate of the n-bit string x_j. An easy way to visualize this composed structure is as follows. Let X denote the $k \times n$ matrix such that the jth row contains x_j. Then $f \circ g(x_1, \ldots, x_k)$ is computed by applying g to each column of X one by one, and then applying f to the resulting n-bit string.

Most of the well-known and studied functions in communication complexity have the above composed structure (e.g., Inner Product, Disjointness, Equality, Hamming Distance, Greater Than, etc.). In particular, functions of the form $f \circ \textsc{and}$ (see [Raz95, Raz03, Kla07, She07, SZ09b, LS09, CA08, BHN09]) and $f \circ \textsc{xor}$ (see [Raz95, Kla07, SZ09a, MO09]) have been the focal point of attention, with an emphasis on $\textsc{sym} \circ g$ where $\textsc{sym}$ denotes a symmetric function (a symmetric function is a function whose output depends only on the number of input bits set to 1). All of the major problems are wide open in the setting of composed functions and very little is known even when f and g are restricted to be very special kinds of functions.

We consider the class of functions of the form $\textsc{sym} \circ g$, which contains many interesting functions and it is tempting to conjecture that some of these functions are candidates to break the $\log n$ barrier. Since the *majority* function $\textsc{maj}$ is conjectured to be outside of ACC^0 [Smo87], it is of interest to try to determine the communication complexity of $\textsc{maj} \circ g$ for all g. For instance, Babai, Kimmel and Lokam [BKL95] identify $\textsc{maj} \circ \textsc{maj}$ as a candidate function to be hard for more than $\log n$ many players. Later, in a significantly expanded version of [BKL95], Babai et al. [BGKL03] show that $\textsc{maj} \circ \textsc{maj}$ has an efficient simultaneous protocol when $k > 1 + \log n$. Their upper bound in fact applies to $\textsc{sym} \circ g$ where $\textsc{sym}$ is any symmetric function and g is any symmetric "compressible" function, a small subset of all symmetric functions.[4] In the same paper, the authors ask about the communication complexity of $\textsc{maj} \circ g$ for a specific symmetric g called "the quadratic character of the sum of the bits", which they show is not compressible. And of course the more general question is whether there is any g such that $\textsc{maj} \circ g$ is hard when the number of players is poly-log(n). In this thesis, we remove the symmetry and compressibility conditions on g and show that functions of the form $\textsc{sym} \circ g$ are easy in the simultaneous model when $k > 1 + \log n$, for *any* choice of the inside function g.

- (Theorem 3.1.2) Let $f : \{0,1\}^n \to \{0,1\}$ be a symmetric function and let $g : \{0,1\}^k \to \{0,1\}$ be an arbitrary function. Then $f \circ g$ has simultaneous multiparty protocol of cost poly-log(n) when $k > 1 + \log n$.

Studying the communication complexity of composed functions is of course not solely important for circuit complexity applications. By determining the communication complexity of composed functions, we would conquer a significant part of the communication complexity landscape and solve many interesting open problems. As an example, let's focus on the quantum vs classical communication complexity problem (second open problem from earlier).

[4] A random symmetric function is not compressible with high probability.

The question of whether quantum mechanics can be utilized to perform certain tasks faster than a classical machine is important both from a practical and a philosophical standpoint. The best known algorithms for factoring integers runs in exponential time and the security of the widely used RSA public key cryptographic system relies on the assumption that factoring integers cannot be done efficiently. On the other hand, a remarkable discovery by Shor shows that this task is easy with a quantum computer. Is there a classical counterpart to Shor's quantum algorithm? It is one of the central questions in computational complexity theory to determine the relative power of quantum and classical computation and communication complexity provides an important and elegant setting in which to study this question.

By the definition of the models, quantum communication complexity of a function is always smaller than its classical counterparts, but how big can the gap be? It is conjectured that for total functions, the quantum models are polynomially related to their classical counterparts and research has been focused on establishing these conjectures for natural large families of functions.[5] In an important paper [Raz03], Razborov shows that the conjecture is true for functions of the form $\textsc{sym} \circ \textsc{and}$ in the 2 party setting, where $\textsc{sym}$ denotes a symmetric function. Shi and Zhang [SZ09a] verify the conjecture for $\textsc{sym} \circ \textsc{xor}$ in the 2 party setting. The next big targets are $f \circ \textsc{and}$ and $f \circ \textsc{xor}$ for general f, but handling arbitrary f seems quite difficult at the moment. In the 2 player setting, $\textsc{and}$ and $\textsc{xor}$ are really the only interesting inside functions as other functions are either trivial or reduce to the case of $\textsc{and}$ or $\textsc{xor}$.

Observe that the focus so far in the literature has been to fix an inside function g and vary the outside function f. We propose a new dual approach. We study the *multiparty* communication complexity of composed functions by fixing the outside function to some natural function and vary the inside

[5] For partial functions, i.e. promise functions, exponential gaps are known.

function. This dual approach is particularly purposive in the multiparty setting where the choice for inside function increases double exponentially in k; unlike the 2 party setting, we get a variety of interesting functions other than AND and XOR.

The functions we study are of the form $\text{MAJ} \circ g$, $\text{MOD}_m \circ g$ and $\text{NOR} \circ g$, where the latter two are generalizations of the well known and studied *generalized inner product* GIP $= \text{MOD}_2 \circ \text{AND}$ and *disjointness* DISJ $= \text{NOR} \circ \text{AND}$ respectively. Arguably, these are the king and queen of all functions in communication complexity with a plethora of applications (see [AMS99, MNSW98, CS04, NS06, NW93, BBM11] for DISJ and [HG91, Nis93, FKL$^+$01, Gro98, BNS92] for GIP). We are able to obtain dichotomies, with respect to the choice of g, that characterize the communication complexity of $\text{MAJ} \circ g$, $\text{MOD}_m \circ g$ and $\text{NOR} \circ g$ for every g. In doing so, we show that these functions have polynomially related quantum and classical communication complexities.

Let $g : \{0,1\}^k \to \{0,1\}$ be an arbitrary function with $S \overset{\text{def}}{=} g^{-1}(1)$ being its support set. For $i \in \{0,1\}$, let S_i denote the subset of S that consists of inputs whose Hamming weight has parity i. Below, the statements are for k up to $\approx \frac{1}{2}\log n$ many players. Although all our lower bounds apply for the quantum model, we restrict attention to the classical model.

- (Theorem 3.2.2) If m divides $|S_0| - |S_1|$, then $\text{MOD}_m \circ g$ has an efficient classical deterministic protocol. Otherwise, it is hard in the randomized model.

- (Theorem 3.4.2) If $|S| \neq 1$, $\text{NOR} \circ g$ has an efficient classical randomized protocol. Otherwise, it is hard in the randomized model.

- (Theorem 3.3.2) If $|S_0| = |S_1|$, $\text{MAJ} \circ g$ has an efficient classical deterministic protocol. Otherwise, it is hard in the randomized model.

As a corollary to our characterization of the MAJ $\circ\, g$ functions, we answer an open problem posed by Babai et al. [BGKL03]. Recall that the authors identify an explicit function called *quadratic character of the sum of bits* (denoted by QCSB) that is not compressible and therefore their protocol from [BGKL03] does not work for MAJ $\circ$ QCSB. They ask the question of determining the communication complexity of MAJ $\circ$ QCSB. Our result implies:

- (Corollary 3.3.3) If $k > 1 + \log n$ or $k \equiv 1 \mod 4$, MAJ $\circ$ QCSB has an efficient deterministic protocol. Otherwise, it is hard in the randomized model.[6]

Chapter 4: Ramsey Theory Applications

One of the interesting features of the NOF model is its connection to Ramsey theory, and in particular Szemerédi's Theorem. It has been known since the introduction of the model that the deterministic communication complexity of certain functions are exactly characterized by certain well-known and studied Ramsey numbers [CFL83]. This connection has been utilized to give surprising communication complexity upper bounds via known bounds on Ramsey numbers. In this thesis, for the first time, we exploit the other direction: We give non-trivial bounds on Ramsey numbers via our protocol for SYM $\circ\, g$ functions (Theorem 3.1.2). Due to its technical nature, we cut the discussion here short and leave the details to Chapter 4.

[6] Technically speaking, it is hard up to $\approx \frac{1}{2}\log n$ many players when $k \equiv 3 \mod 4$ and the function is only defined for k being an odd prime number.

1.1.2 Spectral Norm of Symmetric Functions

Chapter 5: Spectral Norm and 2-party Communication Complexity

One of the main tools in theoretical computer science is Fourier analysis of boolean functions. This field has grown tremendously over the last couple of decades and has become an integral part of theoretical computer science. Its applications include, but are not limited to, hardness of approximation, circuit complexity, social choice theory, learning theory and communication complexity.

The main idea in Fourier analysis is to write a boolean function $f : \{0,1\}^n \to \{0,1\}$ as a linear combination of parity functions (also called characters) $\chi_S(x) \overset{\text{def}}{=} (-1)^{\sum_{i \in S} x_i}$, where $S \subseteq [n]$. That is, we write

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x),$$

where $\widehat{f}(S) \in \mathbb{R}$ are real coefficients and the χ_S's are orthonormal with respect to a natural inner product. This kind of orthogonal decomposition into simpler functions and the linear algebraic view in general turns out to be quite fruitful. One important example of the use of Fourier analysis in complexity theory can be summarized as follows. There are natural quantities associated with the Fourier expansion of a function such as the degree (largest $|S|$ such that $\widehat{f}(S)$ is non-zero), sparsity (number of non-zero Fourier coefficients), L_p norms ($\|\widehat{f}\|_p^p = \sum_S |\widehat{f}(S)|^p$), etc. One tries to capture efficient computation with one of these quantities. Often, one is not able to do so exactly but can do it approximately. For example, functions computed by constant depth polynomial size circuits (AC^0 functions) are well approximated by functions with low degree. This can be used to obtain a PAC learning algorithm for AC^0 functions under the uniform distribution. This kind of reduction from computational complexity to *mathematical complexity* has proven to be quite elegant and powerful.

Among the L_p norms, the L_1 norm (also known as the spectral norm) has an important role and captures important upper and lower bounds in different contexts. For example, functions with small L_1 norm can be learned efficiently in a natural setting of learning theory, and they can be computed by small size depth 2 threshold circuits. The spectral norm also has interesting connections to communication complexity, in particular to the 2 party communication complexity of functions of the form $F = f \circ \textsc{xor}$. For example it can be shown that the logarithm of the approximate L_1 norm of f lower bounds the randomized communication complexity of F. Another example is an intriguing conjecture of Grolmusz which says that for any F, the randomized communication complexity of F is always upper bounded by the poly-log of the spectral norm of F [Gro97].

In this thesis we characterize the spectral norm of all symmetric functions and more generally, gain better insight into their Fourier spectrum. Recall that a function is symmetric if the output depends only on the number of input bits set to 1. These functions play a central role in complexity theory as they are usually the starting point of investigation and recently there has been some progress towards understanding their Fourier spectrum, e.g. [ST11, OWZ11].

For a symmetric f, let $f(|x|)$ denote $f(x)$. Let r_0 and r_1 be the minimum integers such that $f(i) = f(i+1)$ for all $i \in [r_0, n - r_1]$ or $f(i) \neq f(i+1)$ for all $i \in [r_0, n - r_1]$. Define $r(f) = \max\{r_0, r_1\}$. We show:

- (Theorem 5.1.1) Let $f : \{0,1\}^n \to \{0,1\}$ be a symmetric function and let $r(f)$ be defined as above. Then, $\log \|\widehat{f}\|_1 = \Theta\left(r(f) \log\left(\frac{n}{r(f)}\right)\right)$.

As an application of this we verify Grolmusz's conjecture mentioned earlier in the setting of *symmetric xor functions*, i.e., functions of the form $F = \textsc{sym} \circ \textsc{xor}$.

- (Corollary 5.1.3) Let $f : \{0,1\}^n \to \{0,1\}$ be a symmetric function and let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be defined as $F(x, y) = f(x \oplus y)$. Then the randomized communication complexity of F is upper bounded by $O(\log^2 \|\widehat{F}\|_1)$.

As a second application, we give a characterization of the parity decision tree size of symmetric functions. A parity decision tree computes a boolean function by querying the parities of subsets of the variables. The size of the tree is the number of leaves in the tree.

- (Corollary 5.1.2) Let $f : \{0,1\}^n \to \{0,1\}$ be a symmetric function. Then the parity decision tree size of f is $2^{\Theta(r(f)\log(n/r(f)))}$.

Note that the lower bound also applies in the case of the usual decision tree size (where one is restricted to query only variables). Decision tree size is an important measure in learning theory; algorithms for learning decision trees efficiently are of great interest both for practical and theoretical reasons. One of the most well-known and studied problems is whether small size decision trees are efficiently learnable from uniformly random examples.

1.1.3 Privacy vs Communication Complexity

Chapter 6: Hardness of Private Communication

The study of *privacy* in computer science is of great importance and “privacy-preserving computation” sits at the core of our investigations. Privacy concerns in the context of communication complexity naturally arise and one good example can be found in auction design. Auction theory is commonly concerned with the goal of incenting bidders to bid *truthfully*, thereby enabling the auctioneer to obtain private information s/he needs to compute an optimal outcome. As many auctions are held online, it is also quite important that the bidders do not have to reveal private information that the auctioneer does not need to compute the outcome. In this thesis, we will be mainly interested in the trade-off between communication complexity and privacy preservation.

In 1989, Kushilevitz [Kus89] initiated the study of *information-theoretic* privacy in communication complexity. Informally, a communication protocol for computing a function $F(x_1, x_2, \ldots, x_k)$ is private if each player does not learn any additional information (in an information theoretic sense) beyond what follows from knowing his/her private input, and the function value $F(x_1, x_2, \ldots, x_k)$. A complete characterization of the privately computable functions was given, but unfortunately, early work ruled out private protocols for most interesting functions [Kus89, BS08]. For example, second-price auctions are not possible with more than two participants, and are extremely inefficient even in the setting of two bidders [CK89, BS08].

The unattainability of perfect privacy for many functions motivated the study of *approximate* privacy, first by Klauck [Kla02] and more recently by Feigenbaum, Jaggard and Schapira [FJS10a]. The relaxation from perfect to approximate privacy is appealing because it renders more functions computable privately, and more closely mirrors real-world situations in which *some* privacy loss may be acceptable. On the other hand, it is more subtle to capture the notion of approximate privacy. While most reasonable definitions of perfect privacy turn out to be equivalent, this is not quite the case with approximate privacy. In particular, the measures of Klauck and Feigenbaum et al. are different and each has its own advantage and characteristics. Our work here is primarily motivated by the more recent work of Feigenbaum et al. [FJS10a]. A second motivation is to understand the connections between the two.

Following Feigenbaum et al. let's now give the definitions of privacy that we will study in this thesis. Let $F(x, y)$ be a two-party communication function, and let P be a deterministic communication protocol for F. The most important feature of a protocol of cost c that computes F is that it partitions the matrix of F, M_F, into at most 2^c *monochromatic* submatrices (also known as *rectangles*). Here monochromatic means that each entry of the submatrix is the same. The players communicate exactly the same bits for the inputs corresponding to any one of these induced submatrices. The privacy loss (or privacy approximation ratio, PAR) on the input (x, y) with respect to P is defined to be the number of inputs that evaluate to $F(x, y)$ divided by the size of the protocol-induced rectangle containing (x, y): $\mathrm{PAR}(x, y) = \frac{|F^{-1}(F(x,y))|}{|P(x,y)|}$. The *worst-case* privacy loss of protocol P is $\max_{(x,y)} \mathrm{PAR}(x, y)$, and the *worst-case* privacy loss of the function F is then the minimum privacy loss over all protocols for F. Perfect privacy of a protocol (as defined in 1989) requires that the privacy approximation ratio (PAR) is 1 for all inputs. The *average-case* privacy is measured by taking the expectation of $\mathrm{PAR}(x, y)$ rather than a maximum.

The problem we will be interested in is the 2nd price Vickrey auction, the canonical example of a *truthful* auction (neither player has an incentive to cheat). The corresponding communication function takes $x, y \in [2^n]$ as input and returns (x, B) if $x \leq y$ (indicating that Player B has won and has to pay x), and (y, A) if $y < x$ (indicating that Player A has won and has to pay y).

Feigenbaum et al. [FJS10a] study the Vickrey auction problem and reveal a possible *inherent* trade-off between worst-case privacy and communication complexity: they describe a family of protocols such that the privacy loss approaches 1 (perfect privacy) as the length of the protocol approaches exponential. We show that the upper bounds presented in [FJS10a] are essentially tight and hence prove that there is indeed an inherent trade-off between privacy and communication complexity for the Vickrey auction problem. [FJS10a] provided a lower bound only for the special case of “bisection-type” protocols.

- (Theorem 6.3.1) For all p, $2 \leq p \leq n/4$, any deterministic protocol for the Vickrey auction problem of cost at most $n2^{\frac{n}{4p}-5}$ obtains worst-case privacy loss at least 2^{p-2}.

Our second contribution demonstrates a similar type of trade-off for the case of average-case approximate privacy. We prove an asymptotically tight lower bound on the average-case approximate privacy of the Vickrey auction problem, showing that the upper bounds from [FJS10a] are essentially tight. Again, [FJS10a] provided lower bounds only for the special case of bisection-type protocols.

- (Theorem 6.3.5) For all $r \geq 1$, any deterministic protocol of cost at most r for Vickrey auction problem has average-case privacy loss of at least $\Omega\left(\frac{n}{\log(r/n)}\right)$.

Our lower bounds show that the approximate privacy of any polynomial (super-linear) cost protocol is still as large as $\Omega(n/(\log n))$.

Lastly, we solve an explicitly stated open problem from [FJS10a] and put an exponential lower bound on the average-case privacy loss of the *set intersection* function. The set intersection function takes two subsets of $[n]$ as input and returns the intersection set.

- (Theorem 6.3.12) Any protocol that computes the *set intersection* function has $2^{\Omega(n)}$ average-case privacy loss under the uniform distribution.[7]

[7] Here the average-case PAR is actually "subjective". We discuss the difference between objective and subjective PAR in Chapter 6.

CHAPTER 2

Background

In this chapter, we will provide the background information required for the presentation of our main results - Chapters 3 to 6. This chapter is broken into three main sections. In the first section we give some general definitions and set some notation which will be used throughout this thesis. The second section is devoted to communication complexity. We will give the formal definitions of the models we are interested in as well as general lower bound techniques and examples. The third section introduces the reader to Fourier analysis of boolean functions. This section is required to follow Chapter 5.

We note that most of the definitions and notation are indexed for easy referencing.

2.1 General Definitions

All the logarithms are to the base 2. For $t \in \mathbb{R}$, $\exp(t)$ denotes e^t. If $z \in \mathbb{C}$, $\bar{z}$ denotes its complex conjugate and $\mathrm{Re}(z)$ denotes its real part. We use $\mathfrak{C}$ to denote the complex conjugation operator so that $\mathfrak{C}(z) = \bar{z}$ and for $n \in \mathbb{Z}$, $\mathfrak{C}^n(z) = z$ if n is even, $\mathfrak{C}^n(z) = \bar{z}$ if n is odd. We define $\omega_m = \exp(2\pi i/m)$ to be an m-th root of unity. When m is clear from the context, we drop the subscript from ω_m and just write ω.

We use the notation $[n]$ to denote either $\{1, 2, \ldots, n\}$ or $\{0, 1, 2, \ldots, n\}$ and the choice will either be clear from the context or the distinction will not matter. When x is a bit string, $|x|$ denotes the number of 1's in x, i.e. the Hamming weight of x. If $x \in \{0,1\}^n$, x_i denotes the ith bit of x, with a notable exception. In the multiparty setting of communication complexity, we will deal with several n-bit strings and therefore in this case we will explicitly define x_i to be an n-bit string. Then, $x_{i,j}$ denotes the jth bit of the n-bit string x_i.

Functions

A big part of this thesis is about boolean valued functions $f : \mathcal{S} \to \{0,1\}$, where $\mathcal{S}$ is some set. For convenience, we will often define the range of a boolean function as $\{1, -1\}$ rather than $\{0, 1\}$ with the understanding that -1 corresponds to 1 and 1 corresponds to 0. In other words, $f(x)$ is represented as $(-1)^{f(x)}$. We now define the functions we will study. For $x \in \{0,1\}^n$:

$$\begin{aligned}
\textsc{and}(x) &= -1 \text{ iff } \sum_i x_i = n \\
\textsc{or}(x) &= -1 \text{ iff } \sum_i x_i > 0 \\
\textsc{nor}(x) &= -1 \text{ iff } \sum_i x_i = 0 \\
\textsc{maj}(x) &= -1 \text{ iff } \sum_i x_i \geq n/2 \\
\textsc{thr}_t(x) &= -1 \text{ iff } \sum_i x_i \geq t \\
\textsc{par}(x) &= -1 \text{ iff } \sum_i x_i \equiv 0 \mod 2 \\
\textsc{mod}_m(x) &= -1 \text{ iff } \sum_i x_i \equiv 0 \mod m
\end{aligned}$$

We also use XOR to denote PAR. Sometimes, for a function f, we will use the notation $f^{n'}$ to indicate that f has an n'-bit input rather than n.

All of the functions above are **symmetric**, which means that the output depends only on the Hamming weight of the input, $|x|$. In other words, the output does not change if we permute the input bits. We denote by SYM an arbitrary symmetric function.

Most of the communication complexity functions we deal with have a composed structure. We recall the definition from the Introduction. Let $(x_1, \ldots, x_k) \in (\{0,1\}^n)^k$. For $f : \{0,1\}^n \to \{1,-1\}$ and $g : \{0,1\}^k \to \{0,1\}$, define

$$f \circ g(x_1, \ldots, x_k) = f(\ldots, g(x_{1,i}, x_{2,i}, \ldots, x_{k,i}), \ldots).$$

In this case we call f the outside function and g the inside function. The famous *disjointness* and *generalized-inner-product* functions are defined as

$$\begin{aligned} \mathrm{DISJ}(x_1,\dots,x_k) &= \textsc{nor}\circ\textsc{and}(x_1,\dots,x_k),\\ \mathrm{GIP}(x_1,\dots,x_k) &= \textsc{par}\circ\textsc{and}(x_1,\dots,x_k), \end{aligned}$$

where NOR is the negation of OR. Note that inside functions are automatically assumed to be $\{0,1\}$ valued. When $k = 2$, the *generalized-inner-product* function is called the *inner-product* function and is denoted by IP.

Some communication functions we study are not composed. For N an n-bit integer, define

$$\mathrm{EXACT}_N(x_1,\dots,x_k) = -1 \text{ iff } x_1 + \cdots + x_k = N,$$

where x_i are viewed as n-bit integers. For an Abelian group G, define $\mathrm{EVAL}_G : G^k \to \{1,-1\}$ as $\mathrm{EVAL}_G(x_1,\dots,x_k) = -1$ iff $x_1 + \cdots + x_k = 0$, where the addition denotes the group operation and 0 is the identity element. Observe that

$$\mathrm{EVAL}_{\mathbb{F}_2^n}(x_1,\dots,x_k) = \textsc{nor}\circ\textsc{xor}(x_1,\dots,x_k)$$

Almost all the functions we study are boolean valued, with the notable exceptions of the functions studied in Chapter 6. The main function we study there is the *Vickrey auction* (also known as the 2nd price auction) and it is defined as follows. For a positive integer n, the n-bit Vickrey auction is defined as $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z} \times \{A, B\}$ where $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{1, 2, \dots, 2^n\}$ and

$$F(x,y) = \begin{cases} (x,B) & \text{if } x \le y,\\ (y,A) & \text{if } y < x. \end{cases}$$

The other non-boolean function we study in Chapter 6 is the *set intersection* function. On inputs $x, y \in \{0,1\}^n$, $\mathrm{INTERSEC}(x,y)$ outputs the set $\{i \in [n] \,:\, x_i = y_i = 1\}$.

Probability Notation

Random variables are denoted with boldface letters, not necessarily capital. If $\mathbf{x}$ is a random variable and μ a distribution, $\mathbf{x} \sim \mu$ means that $\mathbf{x}$ is distributed according to μ. The notation $\mathbf{E}\left[\cdot\right]$ and $\mathbf{Pr}\left[\cdot\right]$ is used for expectation and probability respectively. When the random variable(s) and the distribution(s) are clear from the context, the expectations and the probabilities do not have any subscripts, e.g. $\mathbf{E}\left[f(\mathbf{x})\right]$. If the distribution is clear but we would like to explicitly point out the random variables, we put the random variables as subscript, e.g. $\mathbf{E}_{\mathbf{x}}\left[f(\mathbf{x})\right]$. We also sometimes choose to make the distribution explicit in this notation, e.g. $\mathbf{E}_{\mathbf{x}\sim\mu}\left[f(\mathbf{x})\right]$. The uniform distribution is always denoted by U and the underlying set will always be clear from the context.

Information Theory

Let $\mathbf{X}$ be a random variable with range $\mathcal{X}$. The **entropy** of $\mathbf{X}$, denoted by $H(\mathbf{X})$, is defined as follows:

$$H(\mathbf{X}) \quad \overset{\text{def}}{=} \quad -\sum_{x\in\mathcal{X}} \mathbf{Pr}\left[\mathbf{X}=x\right] \log_2 \Big(\mathbf{Pr}\left[\mathbf{X}=x\right] \Big)$$

Let $\mathbf{Y}$ be another random variable with range $\mathcal{Y}$. For any y in the range of $\mathbf{Y}$, $H(\mathbf{X}|\mathbf{Y}=y)$ is defined as the entropy of $\mathbf{X}$ under the conditional distribution, i.e.

$$H(\mathbf{X}|\mathbf{Y}=y) \overset{\text{def}}{=} -\sum_{x\in\mathcal{X}} \mathbf{Pr}\left[\mathbf{X}=x \mid \mathbf{Y}=y\right] \log \Big(\mathbf{Pr}\left[\mathbf{X}=x \mid \mathbf{Y}=y\right] \Big).$$

Extending the above naturally, we define the notion of **conditional entropy** $H(\mathbf{X}|\mathbf{Y})$ as

$$H(\mathbf{X}|\mathbf{Y}) \overset{\text{def}}{=} \sum_{y\in\mathcal{Y}} \mathbf{Pr}\left[\mathbf{Y}=y\right] H(\mathbf{X}|\mathbf{Y}=y).$$

As intuition suggests, conditioning a random variable $\mathbf{X}$ on another random variable $\mathbf{Y}$ cannot increase its uncertainty on the average. Formally,

Fact 2.1.1. *For any two random variables* $\mathbf{X}$ *and* $\mathbf{Y}$*,*

$$H\big(\mathbf{X}|\mathbf{Y}\big) \;\leq\; H\big(\mathbf{X}\big).$$

The **mutual information** between $\mathbf{X}$ and $\mathbf{Y}$, denoted by $I(\mathbf{X} : \mathbf{Y})$, is defined as

$$I(\mathbf{X} : \mathbf{Y}) \stackrel{\text{def}}{=} H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}).$$

It is straightforward to verify that mutual information is a symmetric quantity, i.e. $I(\mathbf{X} : \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}) = I(\mathbf{Y} : \mathbf{X})$. Fact 2.1.1 implies that mutual information between two random variables is always non-negative. Just like entropy, one can define the **conditional mutual information** between random variables. Let $\mathbf{Z}$ be another random variable. Then,

$$I(\mathbf{X} : \mathbf{Y}\,|\,\mathbf{Z}) \stackrel{\text{def}}{=} H(\mathbf{X}\,|\,\mathbf{Z}) \;-\; H(\mathbf{X}\,|\,\mathbf{Y},\mathbf{Z})$$

We will also need the following simple claim:

Claim 2.1.2. *Let* $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$ *be any random variables. Then,*

$$\big|I(\mathbf{X} : \mathbf{Y}|\mathbf{W}) - I(\mathbf{X} : \mathbf{Y}\,|\mathbf{W},\mathbf{Z})\big| \leq H(\mathbf{Z}).$$

Decision Trees

Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a boolean function. Consider the following 2-player game. Alice gets an input $x \in \{0,1\}^n$ which Bob does not see. Bob's goal is to compute $f(x)$ by querying the bits of x of his choosing (a query can depend on the outcome of previous queries). The minimum number of queries required to compute f is called the query complexity or **decision**

tree complexity of f. The reason for this terminology is that the strategy of Bob for computing f can be represented by a binary tree in which each non-leaf node is labelled by a variable x_i and each edge is labelled by 0 or 1. The leaves are also labelled 0 or 1. The decision tree complexity of f is the height of the shallowest tree that computes f correctly. Another important measure of complexity is the size of the tree, which is defined to be the number of leaves in the tree. The minimum size of a decision tree that computes f is called the **decision tree size** of f.

In a well-known generalization of the above model, one allows Bob to query the parity of a subset of the input bits of his choosing. This leads to the notions of **parity decision tree complexity** and **parity decision tree size** of a boolean function f.

2.2 Communication Complexity

2.2.1 2 Player Deterministic Model

The most basic and fundamental model in communication complexity is the 2 player deterministic model (introduced in [Yao79]). The setting is as follows. We have a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ and two players Alice and Bob. Alice gets $x \in \mathcal{X}$ and Bob gets $y \in \mathcal{Y}$. They want to collaboratively compute $F(x, y)$ by communicating with each other. Their communication consists of bits that are being transferred from one player to the other. They carry out this communication according to a **protocol** that they have agreed upon beforehand. More precisely, the protocol tells each player:

1. Whose turn it is to send a bit; the protocol determines this purely based on the communicated bits thus far, and we assume without loss of generality that Alice sends the first bit.

2. What bit to send; the protocol determines this based on the communicated bits thus far as well as the input of the player sending the bit.

The protocol also determines when communication stops and the value of the output based on the whole transcript of the communicated bits (which implies both players know the output at the end). The resource of interest is the number of communicated bits, or in other words, the length of the transcript. The goal is to compute the function with the shortest transcript possible. It is worth explicitly noting that we put no restriction on the computational capacities of Alice and Bob, and the sole interest is in the number of bits needed to communicate in order to compute the function.

Let P denote a protocol that correctly computes a function F. Denote by $\Pi_P(x,y)$ the **transcript** of protocol P for the input (x,y) (i.e. the sequence of communicated bits). The cost of P is

$$\mathrm{cost}(P) \overset{\mathrm{def}}{=} \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} |\Pi_P(x,y)|.$$

The **deterministic communication complexity** of F, denoted $\mathbf{D}(F)$, is the cost of the most efficient protocol that computes F correctly. That is,

$$\mathbf{D}(F) \overset{\mathrm{def}}{=} \min_{\text{protocol } P \text{ that computes } F} \mathrm{cost}(P).$$

Unless explicitly stated otherwise (for example, we will do so in Chapter 6), we deal with the standard setting of $\mathcal{X} = \mathcal{Y} = \{0,1\}^n$ and $\mathcal{Z} = \{1,-1\}$, and we are interested in how fast $\mathbf{D}(F)$ grows as a function of n. Observe that every function can be trivially computed with $n+1$ bits of communication: Alice sends x to Bob, Bob computes $F(x,y)$ and sends the result back to Alice. Hence for any F:

$$0 \leq \mathbf{D}(F) \leq n+1.$$

In view of this, protocols of cost at most poly-log(n) are considered to be efficient and protocols of larger cost are deemed inefficient. As an example of an efficient protocol, suppose we want to determine if the majority of the bits in x *and* y is 1, i.e. is $|x|+|y| \geq n$? This function can be computed using $\lceil \log n \rceil + 1$ bits since Bob can compute the output if Alice sends him $|x|$. A canonical example of a hard function is the *equality* function which evaluates to -1 if and only if $x = y$. Intuitively one expects that for Alice and Bob to be sure that $x = y$, or detect a difference, they would have to compare x_i and y_i for all $i \in [n]$. That is, our intuition tells us that $\mathbf{D}(\text{EQUALITY}) \geq n$. But is this correct, and if it is, how do we formally prove it?

In order to prove lower bounds on communication complexity, we need to have a combinatorial understanding of what protocols do. To this end, we first observe that a protocol can be conveniently described with a binary tree as follows (see Figure 2.1). Each node v of the tree is labelled with the letter A or B (indicating whether the node belongs to Alice or Bob) and a function f_v. This function is of the form $f_v : \mathcal{X} \to \{0,1\}$ if the label is A or it is of the form $f_v : \mathcal{Y} \to \{0,1\}$ if the label is B, and it determines what bit the corresponding player communicates. Let us trace the behaviour of the protocol to understand the meaning of this tree. As always, Alice gets x and Bob gets y. First, without loss of generality, the root r is always labelled A, which means that Alice is the first to communicate a bit. Then the protocol determines what bit Alice will send by evaluating $f_r(x)$, i.e. Alice sends Bob $f_r(x)$. If $f_r(x)$ is 0, we move to the left child of the root and if $f_r(x) = 1$ we move to the right child. Without loss of generality let's assume we are at the right child, which we denote by v. If v is labelled with A, then it is again Alice's turn to speak. If it is labelled B, it is Bob's turn. And as before, the function f_v tells the player what bit to send. In this fashion we make our way down the tree until we reach a leaf node. Leaf nodes are special and they determine the output of the protocol.

Observe that every protocol can be described with such a tree and this tree description is entirely consistent with the description we provided in the

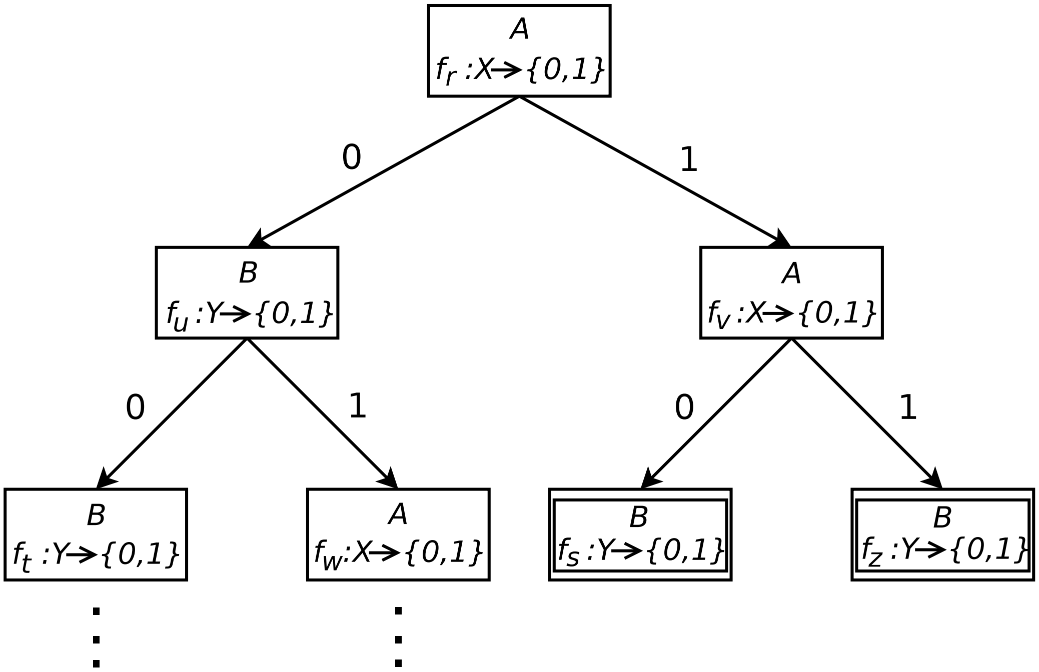

Figure 2.1: A binary tree representing a protocol. Each node is labelled with A or B to indicate whose turn it is to speak. A function associated with a node tells the player what to send. Depending on whether 0 or 1 is sent, we move to the left or the right child of the node. The leaf nodes are indicated with double lines. The functions associated with them determine the output of the protocol.

beginning. In particular, whose turn it is to speak is determined based only on the communicated bits thus far and what a player sends is determined by the communicated bits as well as the input of the player. Obviously the cost of the protocol is the height of the tree.

With this point of view, we will be able to gain a very good understanding of what a protocol does when computing a function F. First we represent F by a $|\mathcal{X}| \times |\mathcal{Y}|$ matrix M_F where the rows are labelled with $x \in \mathcal{X}$, columns are labelled with $y \in \mathcal{Y}$, and $M_F[x,y] = F(x,y)$. A submatrix $\mathcal{S} \times \mathcal{T}$ where $\mathcal{S} \subseteq \mathcal{X}$ and $\mathcal{T} \subseteq \mathcal{Y}$ is called a **rectangle**. The rectangle is said to be **monochromatic** if M_F restricted to $\mathcal{S} \times \mathcal{T}$ has the same value on all of its entries. We will now see that a protocol of cost c that computes F *partitions*[1] M_F into at most 2^c monochromatic rectangles. In fact, this is the most important property of a protocol and all lower bound techniques will be based on this observation.

Proposition 2.2.1. *Let P be a protocol that computes $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ with at most c bits of communication. Then P induces a partition of M_F into at most 2^c monochromatic rectangles.*

To see why this is the case, let's trace once again the behaviour of the protocol down the associated tree. We start at the root which is labelled with A. The root corresponds to the whole matrix $\mathcal{X} \times \mathcal{Y}$. The function f_r is boolean and therefore partitions $\mathcal{X}$ into two sets $\mathcal{X}_0$ and $\mathcal{X}_1$: for all $x \in \mathcal{X}_0$ Alice sends 0 to Bob, and for all $x \in \mathcal{X}_1$ she sends 1. Therefore the left child of r corresponds to the rectangle $\mathcal{X}_0 \times \mathcal{Y}$ and the right child corresponds to $\mathcal{X}_1 \times \mathcal{Y}$. In some sense, if we go to the left child, we eliminate (disregard) the inputs $\mathcal{X}_1 \times \mathcal{Y}$ and our new matrix is $\mathcal{X}_0 \times \mathcal{Y}$ (this is where the input (x,y) lives). If we go to the right child, we eliminate $\mathcal{X}_0 \times \mathcal{Y}$ and our new matrix is

[1] The word *partition* here is important. The rectangles are mutually disjoint and together cover the whole matrix M_F.

$\mathcal{X}_1 \times \mathcal{Y}$. Note that $\mathcal{X}_0 \times \mathcal{Y}$ and $\mathcal{X}_1 \times \mathcal{Y}$ are disjoint. This process inductively continues, so for each node of the tree, there corresponds a rectangle. If a node is the descendent of another, the rectangle of the descendent will be a subset of the other. Otherwise the rectangles are disjoint. Once we reach a monochromatic rectangle, there is no need to partition it further since we can safely declare $F(x, y)$ as the value of this rectangle. Hence each leaf node corresponds to a monochromatic rectangle. Suppose the height of the tree is c, i.e. the protocol has cost c. Then there are at most 2^c leaves. Thus, the protocol partitions M_F into at most 2^c monochromatic rectangles.

It is instructive to see a different proof of the above fact. The following gives an alternative definition of a rectangle.

Proposition 2.2.2. *A set $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ is a rectangle if and only if for all $(x, y), (x', y') \in \mathcal{R}$, we have $(x, y') \in \mathcal{R}$.*

An important observation is that if a protocol produces the same transcript for (x, y) and (x', y'), i.e. $\Pi(x, y) = \Pi(x', y')$, then $\Pi(x, y) = \Pi(x', y') = \Pi(x, y')$. This implies that all the inputs that produce a particular transcript form a rectangle. There are at most 2^c different transcripts and therefore we have at most 2^c monochromatic rectangles that partition M_F.

Proposition 2.2.1 immediately suggests a lower bound strategy: to show a function F has high communication complexity, show that no matter how you partition M_F into monochromatic rectangles, you need many rectangles. Let's denote by $C^D(F)$ the minimum number of rectangles in any monochromatic disjoint cover of M_F. The lower bound strategy can be restated as follows.

Corollary 2.2.3.

$$\mathbf{D}(F) \geq \left\lceil \log C^D(F) \right\rceil .$$

With this tool, it is now easy to show $\mathbf{D}(\text{EQUALITY}) \geq n + 1$. The matrix corresponding to the *equality* function is basically the identity matrix:

the diagonal elements are -1 and the off-diagonal elements are 1. Observe that no monochromatic rectangle can contain more than one -1 since if a rectangle contains the entries (a, a) and (b, b), then it also has to contain (a, b), which corresponds to a 1 entry. This means that we need at least 2^n rectangles to cover the diagonal elements, plus we need at least one rectangle to cover the 1's in the matrix. So in total we need at least $1 + 2^n$ rectangles and hence $\mathbf{D}(\text{EQUALITY}) \geq \lceil \log(1 + 2^n) \rceil = n + 1$.

Although every protocol that computes F induces a partition of M_F into monochromatic rectangles, simple examples show that the converse is not true. So if some monochromatic partitions do not correspond to any protocol, how tight is Corollary 2.2.3? The next theorem states that the gap is not very large.

Theorem 2.2.4.

$$\mathbf{D}(F) \leq O(\log^2 C^D(F)).$$

Let's reiterate that Proposition 2.2.1 and Corollary 2.2.3 are the basis for all lower bound techniques in communication complexity, including the randomized model which we will discuss in the next subsection. In most cases it is not easy to exactly determine $C^D(F)$ so all the various lower bound techniques try to find a suitable lower bound for $C^D(F)$. For instance one might try to upper bound the size of the largest monochromatic rectangle in M_F. If all monochromatic rectangles are small, then we can conclude that we need many rectangles to partition M_F. A more interesting lower bound technique uses the rank of M_F.

Proposition 2.2.5.

$$\mathbf{D}(F) \geq \log \operatorname{rank} M_F.$$

Proof. Suppose a protocol P of cost c computes F and denote by $\mathcal{S}_1 \times \mathcal{T}_1, \ldots, \mathcal{S}_t \times \mathcal{T}_t$ the t monochromatic rectangles that the protocol induces

($t \leq 2^c$). For each of these rectangles $\mathcal{S}_i \times \mathcal{T}_i$, define the $|\mathcal{X}| \times |\mathcal{Y}|$ matrix $M_{\mathcal{S}_i \times \mathcal{T}_i}$ by

$$M_{\mathcal{S}_i \times \mathcal{T}_i}[x, y] = \begin{cases} M_F[x, y] & \text{if } (x, y) \in \mathcal{S}_i \times \mathcal{T}_i \\ 0 & \text{otherwise} \end{cases}$$

These matrices are like the indicator matrices of the rectangles. Obviously we have $M_F = \sum_{i=1}^{t} M_{\mathcal{S}_i \times \mathcal{T}_i}$. By the subadditivity of the rank, we have $\operatorname{rank} M_F \leq \sum_{i=1}^{t} \operatorname{rank} M_{\mathcal{S}_i \times \mathcal{T}_i}$. Since each $M_{\mathcal{S}_i \times \mathcal{T}_i}$ has rank at most 1, we conclude that $\operatorname{rank} M_F$ is at most $t \leq 2^c$, i.e. $c \geq \log \operatorname{rank} M_F$. □

Arguably the most famous open problem in communication complexity is whether the rank lower bound is close to being tight.

Conjecture 2.2.6 (Log Rank Conjecture [LS88])**.** *There is some universal constant k such that*

$$\mathbf{D}(F) \leq O(\log^k \operatorname{rank} M_F).$$

Needless to say there are other lower bound techniques and each has its own advantages depending on the particular function we are dealing with.

One of the well-studied restrictions of the deterministic model is called the **simultaneous** model. Here, the players are not allowed to interact with each other. Upon receiving their inputs, the players send a message to an external referee. The referee, who does not see the players' inputs, determines the output based on these messages. The cost is the number of bits sent to the referee and we denote by $\mathbf{D}^{||}(F)$ the deterministic simultaneous communication complexity of F.

2.2.2 Randomized Model

The previous subsection introduced the most basic communication complexity model. In this subsection we will introduce the randomized model which has a variety of interesting applications.

A natural way to extend the deterministic model to utilize randomness is to allow each player to privately flip coins and make decisions based on the outcomes of those coin flips. Normally, we have to allow some probability of error in computing the function correctly. To make this more concrete, let's say that Alice has access to a random binary string $\mathbf{r}_A$ and Bob has access to a random binary string $\mathbf{r}_B$. Then a randomized protocol computes F with ϵ error if

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \quad \mathbf{Pr}\left[F(x, y) \neq P(x, y)\right] \leq \epsilon,$$

where $P(x, y)$ denotes the output of the protocol and the probability is over the random choices of $\mathbf{r}_A$ and $\mathbf{r}_B$. The cost of a randomized protocol is the maximum number of bits communicated, where the maximum is over all inputs and random strings. It is worth making it clear that the random strings being used by the players do not count towards the cost at all. We denote by $\mathbf{R}^{\epsilon}_{\mathrm{pri}}(F)$ the **randomized communication complexity** of F with ϵ-error, i.e. the cost of the most efficient randomized protocol that computes F with ϵ-error (the subscript 'pri' will be clarified shortly). We are mainly interested in the case where $\epsilon < 1/2$ is some constant. The particular choice of the constant does not matter as it can be shown that it affects the communication complexity by only a constant factor.

Let us revisit the *equality* function and demonstrate the power of randomness. One might be tempted to think that even in the randomized model, the players are bound to check whether $x_i = y_i$ for most of the $i \in [n]$ to convince themselves that the two strings are equal or not (for instance the strings might differ in just one coordinate). On the contrary, by using a clever protocol, the players can compute $\mathrm{EQUALITY}(x, y)$ with high probability using only $O(\log n)$ bits of communication. We describe this protocol now.

To avoid confusion, for this protocol let's denote Alice's input by $a = a_0 a_1 \ldots a_{n-1}$ and Bob's input by $b = b_0 b_1 \ldots b_{n-1}$. The players fix some prime number $p \in [n^2, 2n^2]$. Alice views her input as the polynomial

$$q_A(x) = a_0 + a_1x + a_2x^2 + \cdots a_{n-1}x^{n-1} \mod p$$

over $\mathbb{Z}_p$, and Bob views his input as the polynomial

$$q_B(x) = b_0 + b_1x + b_2x^2 + \cdots b_{n-1}x^{n-1} \mod p.$$

Then Alice chooses uniformly at random an element $\mathbf{z} \in \mathbb{Z}_p$, and sends Bob $\mathbf{z}$ as well as $q_A(\mathbf{z})$. This requires $O(\log n)$ bits of communication. Bob computes $q_B(\mathbf{z})$, compares it to $q_A(\mathbf{z})$, and declares the output to be 1 if they are the same, 0 otherwise. It is easy to see that if $a = b$, then the protocol is always correct. On the other hand, if $a \neq b$, the players make a mistake if $q_A(\mathbf{z}) = q_B(\mathbf{z})$, i.e. $q_A - q_B(\mathbf{z}) = 0$. Note that $q_A - q_B$ is a polynomial of degree at most $n - 1$ and therefore has at most $n - 1$ roots. The players make an error if Alice accidentally picks one of the roots so the probability of error is at most $\frac{n-1}{p} \leq \frac{1}{n}$.

The model we have just introduced is called the “private-coin” model because each player has his/her own private random string. A perhaps less natural but more useful model is the “public-coin” model in which players share a common random string. It is clear that the public-coin model is stronger than the private-coin model and therefore a lower bound in the public-coin model immediately translates into a lower bound in the private-coin model (and we are mainly interested in lower bounds). Furthermore, it is well known that the two models are pretty much equivalent when the error probability is a constant: the communication complexity of a function in the private-coin model is at most $O(\log n)$ more than the communication complexity in the public-coin model [New91]. For these two reasons, and the fact that it is easier to reason about public-coin protocols, our discussion will be about the public-coin model only. Therefore, we drop the subscript ‘pri’ and denote by $\mathbf{R}^{\epsilon}(F)$ the randomized communication complexity of F in the public-coin model.

Going back to the *equality* example, let's show $\mathbf{R}^{\epsilon}(F) = O(1)$ for a constant error probability ϵ. Let $\mathbf{r} \in \{0,1\}^n$ denote the public random string. Alice sends Bob $\langle x, \mathbf{r}\rangle_2 \stackrel{\text{def}}{=} x_1\mathbf{r}_1 + \cdots + x_n\mathbf{r}_n \mod 2$ and Bob compares this value to $\langle y, \mathbf{r}\rangle_2$. If they are the same, he declares the output to be 1, otherwise he outputs 0. If $x = y$ then this protocol never fails. If on the other hand $x \neq y$, then it is easy to see that the inner products will be equal with probability exactly 1/2. So the error probability of the protocol is 1/2. If we repeat this protocol k times with fresh random strings, it is easy to see that the error probability can be reduced to $1/2^k$.

Now that we have seen some interesting upper bounds, let's turn our attention to proving lower bounds. As mentioned earlier, one of the reasons for working with public-coin protocols rather than private-coin protocols is that public-coin protocols are easier to study and understand. A useful way of viewing a public-coin randomized protocol of cost c is as a probability distribution over deterministic protocols, each of cost at most c. Once the random string r is fixed, what the players do is totally deterministic. So the players follow a deterministic protocol P_r that corresponds to the random string r. The success criterion for a randomized protocol is equivalent to saying that for all inputs, at least $1-\epsilon$ fraction of the deterministic protocols should produce the correct answer. Consider a matrix where the rows are labelled with all the possible P_r and the columns are labelled with the inputs (x, y). At the entry corresponding to a particular P_r and (x, y) we put a 1 if $P_r(x, y) = F(x, y)$, and 0 otherwise. The success criterion for the randomized protocol tells us that each column contains at least $1 - \epsilon$ fraction of 1's. So in total, the whole matrix has at least $1-\epsilon$ fraction of 1's. This implies that there must be at least one row that contains at least $1-\epsilon$ fraction of 1's. To sum up, if there is an ϵ-error randomized protocol for F of cost c, then there must be a deterministic protocol P^* of cost at most c such that

$$\mathbf{Pr}\left[F(\mathbf{x}, \mathbf{y}) \neq P^*(\mathbf{x}, \mathbf{y})\right] \leq \epsilon.$$

In fact, it is not difficult to see that the above statement is true for *any* probability distribution over the inputs (x, y). This property of a randomized protocol is the basis for all lower bound techniques because arguing against a deterministic protocol that makes some error is much easier than arguing directly against a randomized protocol. In particular, all the insight we have about deterministic protocols can be put to use in this setting.

Before moving forward, let's make the formal definition of the *distributional* communication complexity model that we have just motivated. Let μ be a distribution over $\mathcal{X} \times \mathcal{Y}$. The ϵ-error **distributional complexity** of F under μ is denoted by $\mathbf{D}_\mu^\epsilon(F)$ and is defined to be the minimum cost of a deterministic protocol P such that

$$\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y}) \neq P(\mathbf{x},\mathbf{y})\right] \leq \epsilon.$$

We have already proved that for any μ, $\mathbf{R}^\epsilon(F) \geq \mathbf{D}_\mu^\epsilon(F)$. It turns out a much stronger relationship holds:

Proposition 2.2.7 ([Yao83])**.**

$$\mathbf{R}^\epsilon(F) = \max_\mu \mathbf{D}_\mu^\epsilon(F).$$

The proof easily follows from von Neumann's Minimax Theorem.

In light of the relationship between randomized communication complexity and distributional communication complexity, we arrive at an obvious lower bound strategy: to prove lower bounds for $\mathbf{R}^\epsilon(F)$, pick your favorite distribution μ and prove a lower bound for $\mathbf{D}_\mu^\epsilon(F)$. As mentioned earlier, this is essentially how all lower bound arguments proceed. Given that a cost c deterministic protocol that computes F partitions M_F into at most 2^c monochromatic rectangles, a protocol that computes F with ϵ fraction of error partitions M_F into at most 2^c "almost" monochromatic rectangles, on average (not all rectangles that the protocol induces must be almost monochromatic but a good fraction must be). To rule out such a possibility with a small c, there are various tactics one can try. Arguably the most famous one is the so called *discrepancy method*. The idea is to show a lower bound for $\mathbf{D}_{\mu}^{\epsilon}(F)$ by showing that every (large enough) rectangle in M_F is balanced in the sense that there are roughly the same fraction of 1's and -1's.

Let's now mathematically formalize the discrepancy method. Let $\mathcal{S} \times \mathcal{T}$ be a rectangle, where $\mathcal{S} \subseteq \mathcal{X}$ and $\mathcal{T} \subseteq \mathcal{Y}$. For a distribution μ over $\mathcal{X} \times \mathcal{Y}$, define the discrepancy of the rectangle $\mathcal{S} \times \mathcal{T}$ with respect to F and μ as the absolute value of the difference between the weight of the 1's and the weight of the -1's in $\mathcal{S} \times \mathcal{T}$, i.e.

$$\begin{aligned}\operatorname{disc}_{\mu}(F, \mathcal{S} \times \mathcal{T}) \overset{\text{def}}{=} \big| &\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y}) = 1 \text{ and } (\mathbf{x},\mathbf{y}) \in \mathcal{S} \times \mathcal{T}\right] \\ &- \mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y}) = -1 \text{ and } (\mathbf{x},\mathbf{y}) \in \mathcal{S} \times \mathcal{T}\right] \big| \\ = &\left| \sum_{(x,y)\in\mathcal{S}\times\mathcal{T}} F(x,y)\mu(x,y) \right|.\end{aligned}$$

The **discrepancy** of F is the maximum discrepancy over all rectangles:

$$\operatorname{disc}_{\mu}(F) \overset{\text{def}}{=} \max_{\mathcal{S}\times\mathcal{T}} \operatorname{disc}_{\mu}(F, \mathcal{S} \times \mathcal{T}).$$

The discrepancy method (see e.g. [CG88]) says that to lower bound $\mathbf{D}_{\mu}^{\epsilon}(F)$, it suffices to upper bound the discrepancy $\operatorname{disc}_{\mu}(F)$.

Proposition 2.2.8 (Discrepancy Method)**.**

$$\mathbf{D}_{\mu}^{\epsilon}(F) \geq \log\left(\frac{1-2\epsilon}{\operatorname{disc}_{\mu}(F)}\right).$$

Proof. Let $\mathbf{D}_{\mu}^{\epsilon}(F) = c$, so there is a deterministic protocol P of cost c that computes F with ϵ error under μ. Let $\mathcal{S}_1 \times \mathcal{T}_1, \ldots, \mathcal{S}_t \times \mathcal{T}_t$, $t \leq 2^c$, be the rectangles that P induces. We denote by $P(\mathcal{S}_i \times \mathcal{T}_i)$ the value the protocol outputs for the inputs $(x, y) \in \mathcal{S}_i \times \mathcal{T}_i$. Then,

$$\begin{aligned}
1 - 2\epsilon &\leq \left|\mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y}) = P(\mathbf{x},\mathbf{y})\right] - \mathbf{Pr}_{(\mathbf{x},\mathbf{y})\sim\mu}\left[F(\mathbf{x},\mathbf{y}) \neq P(\mathbf{x},\mathbf{y})\right]\right| \\
&= \left|\sum_{(x,y)} F(x,y)P(x,y)\mu(x,y)\right| \\
&= \left|\sum_{i=1}^{t} \sum_{(x,y)\in\mathcal{S}_i\times\mathcal{T}_i} F(x,y)P(x,y)\mu(x,y)\right| \\
&= \left|\sum_{i=1}^{t} P(\mathcal{S}_i \times \mathcal{T}_i) \sum_{(x,y)\in\mathcal{S}_i\times\mathcal{T}_i} F(x,y)\mu(x,y)\right| \\
&\leq \sum_{i=1}^{t} |P(\mathcal{S}_i \times \mathcal{T}_i)| \left|\sum_{(x,y)\in\mathcal{S}_i\times\mathcal{T}_i} F(x,y)\mu(x,y)\right| \\
&= \sum_{i=1}^{t} \mathrm{disc}_\mu(F, \mathcal{S}_i \times \mathcal{T}_i) \\
&\leq t \cdot \mathrm{disc}_\mu(F) \leq 2^c \cdot \mathrm{disc}_\mu(F).
\end{aligned}$$

Rearranging, we get $2^c \geq \frac{1-2\epsilon}{\mathrm{disc}_\mu(F)}$. □

Let's see the discrepancy method in action by showing an exponentially small upper bound on the discrepancy of the *inner-product* function IP under the uniform distribution. For a real valued matrix M, let $\|M\|$ denote its spectral norm, i.e. $\|M\| = \max_{u:\|u\|_2=1} \|Mu\|$. It turns out that it is easy to bound the discrepancy of a function under the uniform distribution in terms of the spectral norm of M_F.

Proposition 2.2.9.

$$\mathrm{disc}_U(F) \leq \frac{\|M_F\|}{2^n}.$$

Proof. Let $\mathcal{S} \times \mathcal{T}$ be a rectangle. Denote by $\mathbf{1}_\mathcal{S}$ the indicator vector for $\mathcal{S}$, i.e. the 2^n dimensional vector which has a 1 for positions corresponding to

$\mathcal{S}$ and 0 everywhere else. Similarly for $\mathbf{1}_\mathcal{T}$. By the definition of discrepancy,

$$\mathrm{disc}_U(F, \mathcal{S} \times \mathcal{T}) = \frac{1}{2^{2n}} \left| \sum_{(x,y)\in \mathcal{S}\times\mathcal{T}} F(x,y) \right|.$$

It is not hard to verify that the right hand side is equal to

$$\frac{1}{2^{2n}} |\mathbf{1}_\mathcal{S}^T \cdot M_F \cdot \mathbf{1}_\mathcal{T}| = \frac{1}{2^{2n}} |\langle M_F \mathbf{1}_\mathcal{T}, \mathbf{1}_\mathcal{S} \rangle|.$$

Using the Cauchy-Schwarz inequality, we get $\frac{1}{2^{2n}}|\langle M_F\mathbf{1}_\mathcal{T}, \mathbf{1}_\mathcal{S}\rangle| \leq \frac{1}{2^{2n}} \|M_F \mathbf{1}_\mathcal{T}\| \cdot \|\mathbf{1}_\mathcal{S}\|$. Then by the definition of the spectral norm we conclude:

$$\begin{aligned}\mathrm{disc}_U(F, \mathcal{S} \times \mathcal{T}) &\leq \frac{1}{2^{2n}} \|M_F \mathbf{1}_\mathcal{T}\|_2 \cdot \|\mathbf{1}_\mathcal{S}\|_2 \\ &\leq \frac{1}{2^{2n}} \|M_F\| \|\mathbf{1}_\mathcal{T}\|_2 \|\mathbf{1}_\mathcal{S}\|_2 \\ &\leq \frac{1}{2^{2n}} \|M_F\| \sqrt{|\mathcal{T}|}\sqrt{|\mathcal{S}|} \\ &\leq \frac{\|M_F\|}{2^n}.\end{aligned}$$

□

The spectral norm of M_{IP}, where IP denotes the *inner product* function, is easy to calculate. It is well known that the spectral norm of a matrix M is equal to the largest singular value of M, $\sigma_{\max}(M)$, which in return is equal to the square-root of the largest eigenvalue of $M^T M$. Using the definition of IP, one can easily check that $M_{\mathrm{IP}}^T M_{\mathrm{IP}} = 2^n I$, where I denotes the identity matrix. Therefore for all u, $M_{\mathrm{IP}}^T M_{\mathrm{IP}} u = 2^n u$. This implies $\lambda_{\max}(M_{\mathrm{IP}}^T M_{\mathrm{IP}}) = 2^n$, or in other words, $\|M_{\mathrm{IP}}\| = 2^{n/2}$. Using Proposition 2.2.9, we have $\mathrm{disc}_U(\mathrm{IP}) \leq 1/2^{n/2}$. Plugging this into the Discrepancy Method (Proposition 2.2.8), we conclude

$$\mathbf{R}^\epsilon(\mathrm{IP}) \geq \frac{n}{2} + \log(1 - 2\epsilon).$$

Is the discrepancy method the all powerful method that will give us tight lower bounds for any function? The answer is no and let's explain why. First note that for any function, achieving error probability $1/2$ is trivial since we can just output a random bit. The discrepancy method is a very strong tool in the following sense. If one shows a lower bound of say $\Omega(n)$ on the randomized communication complexity of a function using the discrepancy method, then the lower bound applies to protocols that make $1/2 - 1/\exp(n)$ probability of error, i.e. error exponentially close to $1/2$. For example, in the case of *inner product*, suppose we allow the protocol to make error $\epsilon = 1/2 - 1/2^{\alpha n}$ for some constant $\alpha < 1/2$. Then $\mathbf{R}^{\epsilon}(\mathrm{IP}) \geq n/2 + \log(1 - 2\epsilon) = n/2 - \alpha n = \Omega(n)$. When our primary interest is in constant probability of error, this is an overkill. There are many functions that require $\Omega(n)$ communication complexity when the error probability is a constant but has $O(1)$ communication complexity once we allow the error probability to be $1/2 - 1/\exp(n)$. In particular, it is well known that the discrepancy method cannot yield good lower bounds for any function with small *non-deterministic* communication complexity. A canonical example is the famous *disjointness* function and to handle such functions, one needs to develop more sophisticated tools. On this note, we end our discussion of the 2 party randomized communication complexity model and move on to the non-deterministic model.

2.2.3 Non-Deterministic Model

Non-determinism is a very important notion in computational complexity theory. At a high level, the motivation is to understand whether verifying a given solution to a problem is easier than finding a solution. The answer of course depends on which computational model we are dealing with. In communication complexity, non-determinism can be much more efficient and in this subsection, we will briefly go over non-deterministic communication complexity.

As usual, there are two equivalent ways to view the non-deterministic model. We can view it as a model in which players are allowed to take non-deterministic steps, or we can view it as a proof verification process. We prefer to use the latter version. As before, Alice gets $x \in \mathcal{X}$ and Bob gets $y \in \mathcal{Y}$. We also have a third player called God, who sees the input (x, y) and furnishes a proof string z which is then communicated to both Alice and Bob. Upon receiving z, Alice and Bob communicate with each other and agree on an output. If $F(x, y) = -1$, there must be at least one proof string z that leads Alice and Bob to output -1. On the other hand, if $F(x, y) = 1$, no matter what proof string Alice and Bob receive, they should output 1. We include in the cost the length of z. The **non-deterministic communication complexity** of F, denoted by $\mathbf{N}^{-1}(F)$, is the cost of the most efficient non-deterministic protocol that computes F as described above. The **co-non-deterministic communication complexity** of F is denoted by $\mathbf{N}^{1}(F)$ and is defined to be equal to $\mathbf{N}^{-1}(-F)$, the non-deterministic complexity of the negation of F.[2]

Recall the definition of the *disjointness* function. It is straightforward to see that $\mathbf{N}^{1}(\text{DISJ}) \leq O(\log n)$. God provides an index $i \in [n]$ and Alice and Bob exchange x_i and y_i with each other in order to check if $x_i = y_i = 1$. If x and y are not disjoint, then there is an index i such that $x_i = y_i = 1$. If not, for no index we will have $x_i = y_i = 1$. A similar protocol also shows that $\mathbf{N}^{1}(\text{EQUALITY}) \leq O(\log n)$. On the other hand, intuitively it seems unlikely that $\mathbf{N}^{-1}(\text{EQUALITY})$ is small; how can God furnish a short proof that two strings are equal?

In Subsection 2.2.1, we defined $C^{D}(F)$ as the minimum number of disjoint monochromatic rectangles needed to partition M_F. Define $C^{z}(F)$ as the minimum number of possibly intersecting monochromatic rectangles needed

[2] Note that in the literature, $\mathbf{N}^{-1}(F)$ is almost always denoted by $\mathbf{N}^{1}(F)$ and $\mathbf{N}^{1}(F)$ is denoted by $\mathbf{N}^{0}(F)$. This is due to the range of the function F, which is often $\{0, 1\}$ as opposed to $\{1, -1\}$ as in here.

to cover the z-entries of M_F. This quantity accurately characterizes the non-deterministic communication complexity of F.

Proposition 2.2.10.

$$\log C^z(F) \leq \mathbf{N}^z(F) \leq 2 + \log C^z(F).$$

We skip the proof of this proposition but remark that it is quite straightforward and uses the fact that once the proof string is fixed, Alice and Bob follow a deterministic protocol.

Needless to say, Proposition 2.2.10 is the backbone of all lower bound techniques for the non-deterministic model. Going back to the *equality* example, we see that a monochromatic rectangle can cover at most one -1 entry and therefore we need 2^n rectangles to cover all the -1 entries.

At the end of the previous subsection (Subsection 2.2.2), we mentioned that the discrepancy method fails to give good lower bounds on the randomized communication complexity of functions that have low non-deterministic communication complexity. Let us now make this formal.

Proposition 2.2.11 (see e.g. [Cha08] Lemma 6.17)**.** *Let F be such that* $\min\{\mathbf{N}^1(F), \mathbf{N}^{-1}(F)\} = c$*. Then, under any distribution μ over the inputs,*

$$\operatorname{disc}_\mu(F) \geq \Omega(1/2^c).$$

2.2.4 Multiparty Number on the Forehead Model

There are various ways one can extend the two player model to more players. Given $F : \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_k \to \mathcal{Z}$, the most natural generalization would be to distribute the input $(x_1, x_2, \ldots, x_k)$ so that Player i gets x_i. This is called the "number in the hand" multiparty model; it is an interesting model with nice applications. In this thesis however, we are interested in the so called "number on the forehead" multiparty model in which Player i sees all

x_j with $j \neq i$. We visualize this scenario as x_i being written on the forehead of Player i. Once the input is distributed, the players once again follow a protocol in order to compute $F(x_1, \ldots, x_k)$. The description of a protocol is equivalent to the 2 player model and when a player communicates a bit, all the other players get to see it.

We can generalize the *equality* example seen in the 2 party setting to an arbitrary number of players in the obvious way: let $\mathrm{EQ}_k(x_1, x_2, \ldots, x_k) = 1$ if and only if $x_1 = \cdots = x_k$. When $k = 2$, we saw that the deterministic communication complexity of *equality* is $n + 1$. On the other hand, when $k > 2$, it is easy to see that the communication complexity drops down to just 2 bits. Player 1 checks if $x_2 = x_3 = \cdots = x_k$ and Player 2 checks if $x_1 = x_3 = x_4 = \cdots = x_k$. If both equalities are confirmed, all the strings are equal, otherwise they are not. This example demonstrates the power of the multiparty number on the forehead model. The overlap of information among the players can be exploited to give efficient protocols.

We denote by $\mathbf{D}_k(F)$, $\mathbf{D}_k^{||}(F)$, $\mathbf{R}_k^{\epsilon}(F)$, $\mathbf{D}_{k,\mu}^{\epsilon}(F)$, and $\mathbf{N}_k^{-1}(F)$ the k-party deterministic, deterministic simultaneous, randomized, distributional and non-deterministic communication complexity of F respectively. In the 2 player setting, the single most important property of a protocol was the fact that it induced rectangles. For the k party model with $k \geq 3$, the appropriate generalization of the notion of a rectangle is called a cylinder intersection. A **cylinder** $\mathcal{C}_i$ in the ith direction is a subset of the input space $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ such that membership in $\mathcal{C}_i$ does not depend on the ith coordinate, i.e. if $(x_1, \ldots, x_i, \ldots, x_k) \in \mathcal{C}_i$ then $(x_1, \ldots, x_i', \ldots, x_k) \in \mathcal{C}_i$ for all $x_i' \in X_i$ (see Figure 2.2). A **cylinder intersection** $\mathcal{C}$ is just an intersection of k cylinders, one in each direction, i.e. $\mathcal{C} = \cap_{i=1}^{k} \mathcal{C}_i$ where $\mathcal{C}_i$ is a cylinder in the ith direction. It is important to take a moment and observe that when $k = 2$, this definition corresponds to the notion of a rectangle (see Figure 2.3).

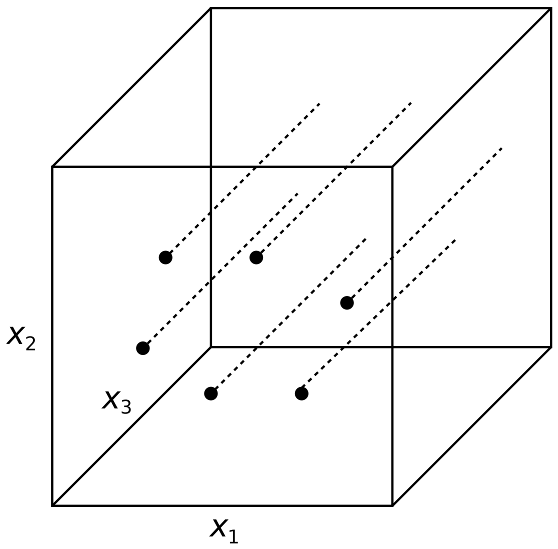

Figure 2.2: A cylinder in the 3rd direction. The bold dots represent a subset of $\mathcal{X}_1 \times \mathcal{X}_2$, which then completely determines the corresponding cylinder.

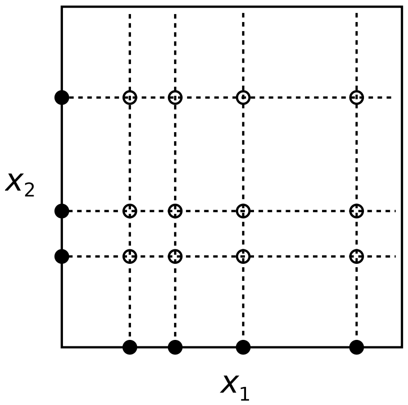

Figure 2.3: A rectangle as the intersection of two cylinders.

In Proposition 2.2.2, we gave an alternative definition of a rectangle. The same characterization holds also for cylinder intersections. A set of k points

$$(x_1', x_2, \dots, x_k), (x_1, x_2', \dots, x_k), \dots, (x_1, x_2, \dots, x_k')$$

in $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$ is called a **star** if $x_i' \neq x_i$ for all $i \in [k]$. The point $(x_1, x_2, \dots, x_k)$ is called the center of the star.

Proposition 2.2.12. *A set $\mathcal{C} \subseteq \mathcal{X}_1 \times \cdots \mathcal{X}_k$ is a cylinder intersection if and only if for every star in $\mathcal{C}$, its center is also contained in $\mathcal{C}$.*

Now it is easy to see that a multiparty protocol induces a partition of M_F into monochromatic cylinder intersections. Here M_F denotes the k-dimensional matrix (often called a tensor) such that $M_F[x_1, \dots, x_k] = F(x_1, \dots, x_k)$ for all $(x_1, \dots, x_k) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$.

Proposition 2.2.13. *Let P be a deterministic protocol that computes $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Z}$ with at most c bits of communication. Then P induces a partition of M_F into at most 2^c monochromatic cylinder intersections.*

Proof. As in the 2 player case, it is easy to see that if the protocol produces the same transcript for all the elements of a star, then the protocol must produce the same transcript for the center of the star as well. Therefore, the set of all points corresponding to a particular transcript forms a cylinder intersection. There are at most 2^c possible transcripts and the statement follows. □

The definition of discrepancy naturally generalizes to cylinder intersections. We will now make the formal definition with respect to complex valued functions $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathbb{C}$ since the definition does not call for a restriction on F to be boolean. Furthermore, in Section 3.2 we will be interested in the discrepancy of complex valued functions.

Given a cylinder intersection $\mathcal{C} = \cap_{i=1}^{k} \mathcal{C}_i$, let ϕ_i denote the characteristic function of $\mathcal{C}_i$, i.e. $\phi_i(x_1, \ldots, x_k) = 1$ if $(x_1, \ldots, x_k) \in \mathcal{C}_i$ and $\phi_i(x_1, \ldots, x_k) = 0$ otherwise. Then $\phi \overset{\text{def}}{=} \prod_{i=1}^{k} \phi_i$ is the characteristic function of $\mathcal{C}$. For a distribution μ over $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$, and a cylinder intersection $\mathcal{C}$, the discrepancy of F with respect to μ and $\mathcal{C}$ is

$$\begin{aligned} \mathrm{disc}_\mu(F, \mathcal{C}) &\overset{\text{def}}{=} \left| \sum_{(x_1, \ldots, x_k) \in \mathcal{C}} F(x_1, \ldots, x_k) \mu(x_1, \ldots, x_k) \right| \\ &= \left| \mathbf{E}_{(\mathbf{x}_1, \ldots, \mathbf{x}_k) \sim \mu} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \phi(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right] \right|. \end{aligned} \tag{2.1}$$

The **discrepancy** of F is the maximum discrepancy over all cylinder intersections:

$$\mathrm{disc}_\mu(F) \overset{\text{def}}{=} \max_{\mathcal{C}} \mathrm{disc}_\mu(F, \mathcal{C}).$$

The discrepancy method generalizes to the multiparty setting with the same proof [BNS92].

Proposition 2.2.14 (Discrepancy Method). *Let $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \{1, -1\}$, and μ a distribution over $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$. Then,*

$$\mathbf{D}^{\epsilon}_{k,\mu}(F) \geq \log \left(\frac{1 - 2\epsilon}{\mathrm{disc}_\mu(F)} \right).$$

In the two party setting, we saw how to upper bound the discrepancy of F using the spectral norm of the matrix M_F. This linear algebraic technique does not work in the multiparty setting because M_F is no longer a matrix and a cylinder intersection is a relatively complicated combinatorial object. There is still however a famous trick one can resort to in order to deal with cylinder intersections: repeatedly apply Cauchy-Schwarz inequality to get rid of the cylinder intersection.

Lemma 2.2.15 ([CT93, Raz00]). *Let $F : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathbb{C}$ and let μ_i be a distribution over $\mathcal{X}_i$. Define the distribution μ as the product of the μ_i, that* *is* $\mu(x_1, \ldots, x_k) = \mu_1(x_1) \cdots \mu_k(x_k)$. *Then,*

$$(\mathrm{disc}_\mu(F))^{2^k} \leq \mathbf{E}_{\substack{\mathbf{x}_1^0, \ldots, \mathbf{x}_k^0 \\ \mathbf{x}_1^1, \ldots, \mathbf{x}_k^1}} \left[\prod_{u \in \{0,1\}^k} \mathfrak{C}^{u_1 + \cdots + u_k}(F(\mathbf{x}_1^{u_1}, \ldots, \mathbf{x}_k^{u_k})) \right], \tag{2.2}$$

where in the expectation, $(\mathbf{x}_i^0, \mathbf{x}_i^1)$ *are distributed according to the product distribution* $\mu_i \times \mu_i$.

Proof. We prove the lemma by induction on k and in order to reduce clutter, we will prove it for real valued functions as opposed to complex valued functions. The proofs are identical. Our induction hypothesis is that the lemma is true for every function with $k-1$ players. Let $\mathcal{C} = \cap_{1 \leq i \leq k} \mathcal{C}_i$ be an arbitrary cylinder intersection with the characteristic function $\phi(x_1, \ldots, x_k) = \phi_1(x_1, \ldots, x_k) \cdots \phi_k(x_1, \ldots, x_k)$. Recall that ϕ_i does not depend on x_i. Then, writing the discrepancy as in (2.1), we have

$$\begin{aligned} \mathrm{disc}_\mu(F, \mathcal{C}) &= \left| \mathbf{E}\left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \prod_{i=1}^{k} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right] \right| \\ &\leq \mathbf{E}_{\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}} \left[\left| \phi_k(\mathbf{x}_1, \ldots, \mathbf{x}_k) \mathbf{E}_{\mathbf{x}_k} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \prod_{i=1}^{k-1} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right] \right| \right]. \end{aligned}$$

Squaring both sides and using the consequence $\mathbf{E}[\mathbf{Z}]^2 \leq \mathbf{E}[\mathbf{Z}^2]$ of Cauchy-Schwarz inequality, we obtain

$$\begin{aligned} &\mathrm{disc}_\mu(F, \mathcal{C})^2 \\ &\leq \mathbf{E}_{\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}} \left[\phi_k(\mathbf{x}_1, \ldots, \mathbf{x}_k)^2 \mathbf{E}_{\mathbf{x}_k} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \prod_{i=1}^{k-1} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right]^2 \right] \\ &= \mathbf{E}_{\mathbf{x}_1, \ldots, \mathbf{x}_{k-1}} \left[\mathbf{E}_{\mathbf{x}_k} \left[F(\mathbf{x}_1, \ldots, \mathbf{x}_k) \prod_{i=1}^{k-1} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_k) \right]^2 \right]. \end{aligned} \tag{2.3}$$

If we let

$$F^{x_k^0, x_k^1}(x_1, \ldots, x_{k-1}) \stackrel{\text{def}}{=} F(x_1, \ldots, x_{k-1}, x_k^0) F(x_1, \ldots, x_{k-1}, x_k^1),$$

and also let

$$\phi_i^{x_k^0,x_k^1}(x_1,\ldots,x_{k-1}) \overset{\text{def}}{=} \phi_i(x_1,\ldots,x_{k-1},x_k^0)\phi_i(x_1,\ldots,x_{k-1},x_k^1)$$

for each $i \in \{1,\ldots,k-1\}$, then we can rewrite (2.3) as

$$\begin{aligned}
&\operatorname{disc}_\mu(F,\mathcal{C})^2 \\
&\le \mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_{k-1}}\left[\mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[F^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\ldots,\mathbf{x}_{k-1})\prod_{i=1}^{k-1}\phi_i^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\ldots,\mathbf{x}_{k-1})\right]\right] \\
&\le \mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[\left|\mathbf{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_{k-1}}\left[F^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\ldots,\mathbf{x}_{k-1})\prod_{i=1}^{k-1}\phi_i^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\ldots,\mathbf{x}_{k-1})\right]\right|\right] \\
&= \mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[\operatorname{disc}_{\mu'}(F^{\mathbf{x}_k^0,\mathbf{x}_k^1},\mathcal{C}')\right]. \qquad (2.4)
\end{aligned}$$

Above, μ' is the product of μ_1 up to μ_{k-1} and $\mathcal{C}'$ is the cylinder intersection defined by $\prod_{i=1}^{k-1}\phi_i^{\mathbf{x}_k^0,\mathbf{x}_k^1}(\mathbf{x}_1,\ldots,\mathbf{x}_{k-1})$. Raising both sides of equation (2.4) to the power of 2^{k-1}, we get

$$\operatorname{disc}_\mu(F,\mathcal{C})^{2^k} \le \mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[\operatorname{disc}_{\mu'}(F^{\mathbf{x}_k^0,\mathbf{x}_k^1},C')\right]^{2^{k-1}}.$$

A repeated application of the Cauchy-Schwarz inequality implies $\mathbf{E}\left[\mathbf{Z}\right]^{2^{k-1}} \le \mathbf{E}\left[\mathbf{Z}^{2^{k-1}}\right]$. Hence,

$$\operatorname{disc}_\mu(F,\mathcal{C})^{2^k} \le \mathbf{E}_{\mathbf{x}_k^0,\mathbf{x}_k^1}\left[\operatorname{disc}_{\mu'}(F^{\mathbf{x}_k^0,\mathbf{x}_k^1},\mathcal{C}')^{2^{k-1}}\right].$$

Now applying the induction hypothesis to $\operatorname{disc}_{\mu'}(F^{x_k^0,x_k^1},\mathcal{C}')^{2^{k-1}}$, we get the desired result. □

The RHS of Inequality 2.2 is important and deserves a name. Let μ be a product distribution over $\mathcal{X}_1\times\cdots\times\mathcal{X}_k$, i.e. $\mu(x_1,\ldots,x_k)=\mu_1(x_1)\cdots\mu_k(x_k)$, where μ_i is a distribution over $\mathcal{X}_i$. We define the **cube measure** of a complex valued function $F:\mathcal{X}_1\times\cdots\times\mathcal{X}_k\to\mathbb{C}$ under μ as

$$\mathcal{E}_\mu(F) = \mathbf{E}_{\substack{\mathbf{x}_1^0,\ldots,\mathbf{x}_k^0\\ \mathbf{x}_1^1,\ldots,\mathbf{x}_k^1}}\left[\prod_{u\in\{0,1\}^k}\mathfrak{C}^{u_1+\cdots+u_k}(F(\mathbf{x}_1^{u_1},\ldots,\mathbf{x}_k^{u_k}))\right].$$

The cube measure is always a non-negative real number. In fact, the quantity $(\mathcal{E}_U(F))^{1/2^k}$, where U is the uniform distribution, is known as the *hypergraph uniformity norm* and is a measure of "quasirandomness" of F. When $F = f \circ \textsc{xor}$, the hypergraph uniformity norm of F corresponds to Gowers uniformity norm of f over $\mathbb{F}_2^n$ (see e.g. [Gow10, Section 2.4] and references therein). Lemma 2.2.15 can now be restated as

$$\operatorname{disc}_\mu(F) \le (\mathcal{E}_\mu(F))^{1/2^k}.$$

Let us see the above inequality in action and show an exponentially small upper bound on the *generalized-inner-product* function GIP.

Theorem 2.2.16.

$$\operatorname{disc}_U(\mathrm{GIP}) \le \exp\left(-\frac{n}{4^k}\right).$$

Proof. Using Lemma 2.2.15, our task is to upper bound the cube measure $\mathcal{E}_U(\mathrm{GIP})$. Since we can decompose GIP as $\textsc{parity} \circ \textsc{and}$, and $\textsc{parity}$ is just multiplication over ±1 valued variables, we have

$$\begin{aligned}\mathcal{E}_U(\mathrm{GIP}) &= \mathbf{E}\left[\prod_{u\in\{0,1\}^k} \mathrm{GIP}(\mathbf{x}_1^{u_1}, \ldots, \mathbf{x}_k^{u_k})\right] \\ &= \mathbf{E}\left[\prod_{u\in\{0,1\}^k} \prod_{i=1}^{n} (-1)^{\textsc{and}(\mathbf{x}_{1,i}^{u_1}, \ldots, \mathbf{x}_{k,i}^{u_k})}\right].\end{aligned}$$

Using independence, we can move the inside product outside to obtain

$$\begin{aligned}\mathcal{E}_U(\mathrm{GIP}) &= \prod_{i=1}^{n} \mathbf{E}\left[\prod_{u\in\{0,1\}^k} (-1)^{\textsc{and}(\mathbf{x}_{1,i}^{u_1}, \ldots, \mathbf{x}_{k,i}^{u_k})}\right] \\ &= (\mathcal{E}_U(\textsc{and}))^n.\end{aligned}$$

Thus, all we need to do is bound the cube measure of the $\textsc{and}$ function on k variables. It is not difficult to see that if for all $j \in \{1, \ldots, k\}$, $\mathbf{x}_{j,i}^0 \ne \mathbf{x}_{j,i}^1$, then the expectation is -1. This happens with probability $1/2^k$. On the other hand, if there is some j such that $\mathbf{x}^0_{j,i} = \mathbf{x}^1_{j,i}$, the product evaluates to 1. Therefore,

$$\mathcal{E}_U(\text{AND}) = \left(1 - \frac{1}{2^k}\right) - \frac{1}{2^k} = 1 - \frac{1}{2^{k-1}}.$$

So $\mathcal{E}_U(\text{GIP}) = (1 - 1/2^{k-1})^n \leq \exp(-n/2^{k-1})$, and the result follows from Lemma 2.2.15. □

Corollary 2.2.17.

$$\mathbf{R}^{\epsilon}_k(\text{GIP}) \geq \frac{n}{4^k} + \log(1 - 2\epsilon).$$

Note that the above lower bound collapses once k reaches $\log n$. This is an unavoidable consequence of Lemma 2.2.15 where we used the Cauchy-Schwarz inequality repeatedly in order to get rid of the cylinder intersection. As all lower bounds in the NOF model use this trick, they all suffer the exponential loss in the number of players. As mentioned in the introduction, proving lower bounds in the NOF model for $\log n$ players is an outstanding open problem.

2.2.5 Communication Complexity Classes

In computational complexity theory we try to classify problems in terms of the resources required to compute their solution. An important part of this classification requires well defined complexity classes, like P, NP, and BPP, which correspond to problems with efficient deterministic, non-deterministic and randomized solutions respectively. In communication complexity, we can define ([BFS86]) analogous complexity classes once we agree on the meaning of "efficient". Conventionally, protocols of cost at most poly-log(n) are considered to be efficient. This naturally leads to the following communication complexity classes corresponding to the different communication complexity models:

Complexity class	P_k^{cc}	NP_k^{cc}	$co\mathsf{NP}_k^{cc}$	BPP_k^{cc}
Complexity measure	$\mathbf{D}_k$	$\mathbf{N}_k^{-1}$	$\mathbf{N}_k^{1}$	$\mathbf{R}_k$

Unlike the Turing Machine world, we have a reasonably good understanding of the relationships between the communication complexity classes since we can actually prove strong lower bounds. For instance, the two player *non-equality* function is in BPP_2^{cc} and NP_2^{cc} but not in P_2^{cc}. Therefore we know that $\mathsf{P}_2^{cc} \neq \mathsf{NP}_2^{cc}$ and $\mathsf{P}^{cc} \neq \mathsf{BPP}_2^{cc}$. We also know that $\mathsf{NP}_2^{cc} \neq \mathsf{BPP}_2^{cc}$ via the *disjointness* function.

2.2.6 Information Complexity

The techniques we have seen so far are some of the highlights of the first generation methods in communication complexity. In recent years, a new method based on information theory, introduced in the seminal paper [CWYS01], has flourished and contributed significantly to the advancement of the field. We will now very briefly touch upon this second generation technique. Our discussion will be limited to the 2 party model since these techniques currently do not extend to the multiparty NOF model.

In a nutshell, information theory methods in communication complexity try to measure how much information Alice and Bob reveal about their inputs to a third party or each other when they follow a communication protocol. There are several ways to measure this quantity but we will for now refer to it informally as *information complexity.* This information is measured in bits and therefore it serves as a lower bound on the communication complexity of a function: if a protocol has cost c, it cannot reveal more than c bits of information. One can then obtain lower bounds on communication complexity by lower bounding the information complexity of a function. This approach puts powerful and intuitive tools from information theory at our disposal.

Let μ be a distribution over the input space $\mathcal{X} \times \mathcal{Y}$, and let P be a protocol that computes a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Recall that $\Pi_P(x, y)$ denotes the transcript that the protocol produces when the input is (x, y). The **external information cost** of a protocol with respect to μ is defined as

$$\mathrm{IC}_\mu^{\mathrm{ext}}(P) \stackrel{\mathrm{def}}{=} I(\mathbf{x}, \mathbf{y} : \Pi_P(\mathbf{x}, \mathbf{y})),$$

where $(\mathbf{x}, \mathbf{y})$ has distribution μ. This intuitively measures how much information a third party learns about Alice's and Bob's inputs by looking at the transcript of the protocol. Another useful measure is the **internal information cost**, which is defined to be

$$\mathrm{IC}_\mu^{\mathrm{int}}(P) \stackrel{\mathrm{def}}{=} I(\mathbf{y} : \Pi_P(\mathbf{x}, \mathbf{y})|\mathbf{x}) + I(\mathbf{x} : \Pi_P(\mathbf{x}, \mathbf{y})|\mathbf{y}).$$

This measures how much information Alice learns about Bob's input plus how much information Bob learns about Alice's input.

Let us restrict our discussion to external information cost. The ϵ-error **information complexity** of a function F with respect to a distribution μ is denoted by $\mathrm{IC}_{\mu,\epsilon}(F)$ and is defined to be the minimum $\mathrm{IC}_\mu^{\mathrm{ext}}(P)$ among all randomized protocols P that compute F with ϵ error. It is straightforward to see that for any distribution μ, $\mathbf{R}^\epsilon(F) \geq \mathrm{IC}_{\mu,\epsilon}(F)$.

To illustrate how this can be used to prove communication complexity lower bounds, let's give a very high level and vague sketch of the lower bound for *disjointness*. As we have seen before, *disjointness* has the composed structure $\mathrm{DISJ} = \textsc{or} \circ \textsc{and}$. Intuitively one expects that any protocol that solves *disjointness* with good accuracy must implicitly solve each of the n instances of the $\textsc{and}$ function. Suppose ν is a distribution over the inputs of a two bit $\textsc{and}$ function and define μ to be the n-fold product of ν, i.e. $\mu = \nu^n$. Then one can hope to show $\mathrm{IC}_{\mu,\epsilon}(\mathrm{DISJ}) = n \cdot \mathrm{IC}_{\nu,\epsilon}(\textsc{and})$. Unfortunately this may not be true in general, for example when ν is not a product distribution over $\{0, 1\} \times \{0, 1\}$. And in the case of *disjointness*, it is essential that ν

is not a product distribution. To get around this problem, one defines an appropriate random variable so that conditioned on it, the "direct sum" property that we hoped for holds. This then reduces our task of showing an $\Omega(1)$ lower bound on the information complexity of the AND function on 2 bits. With some effort, this can be proved by elementary means. The details can be found in [BYJKS04].

2.3 Fourier Analysis of Boolean Functions

The study of boolean functions $f : \{0,1\}^n \to \{0,1\}$ is central to complexity theory and combinatorics as objects of interest in these areas can often be represented as boolean functions. Fourier analysis of boolean functions provides some of the strongest tools in this study with applications to graph theory, circuit complexity, communication complexity, hardness of approximation, machine learning, etc. We will rely on these tools in Chapter 5.

As before, for convenience, we will view the range as $\{1,-1\}$ rather than $\{0,1\}$. The main idea behind Fourier analysis of boolean functions is very simple. We are interested in studying the set of boolean functions $\mathcal{B} = \{f : \{0,1\}^n \to \{1,-1\}\}$. This set by itself does not have much structure and therefore is not easy to reason about. On the other hand, vector spaces have a lot of structure and we understand them very well. Therefore a natural thing to do is to view $\mathcal{B}$ as residing in a vector space, and the natural candidate is the vector space of real valued functions $\mathcal{V} = \{\phi : \{0,1\}^n \to \mathbb{R}\}$. This is a 2^n-dimensional vector space over the reals. Furthermore, we can turn $\mathcal{V}$ into an inner product space by defining an appropriate inner product: for $\phi, \psi \in \mathcal{V}$, define

$$\langle \phi, \psi \rangle \stackrel{\text{def}}{=} \mathbf{E}\left[\phi(\mathbf{x})\psi(\mathbf{x})\right],$$

where the expectation is with respect to the uniform distribution over $\{0,1\}^n$.

Thus we can equivalently write

$$\langle \phi, \psi \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \phi(x)\psi(x).$$

This absolute value of the inner product is often called the *correlation* between ϕ and ψ because when ϕ and ψ are boolean functions, the inner product really measures how well ϕ and ψ are correlated. For boolean functions f and g, we define the **correlation** as

$$\mathrm{Cor}(f, g) \overset{\text{def}}{=} \left| \mathbf{Pr}\left[f(\mathbf{x}) = g(\mathbf{x})\right] - \mathbf{Pr}\left[f(\mathbf{x}) \neq g(\mathbf{x})\right] \right|.$$

Observe that this quantity is always between 0 and 1. It is 1 when $f(x) = g(x)$ for all x or $f(x) = -g(x)$ for all x. It is 0 when $f(x)$ and $g(x)$ agree on exactly half the points x (i.e. knowing $f(x)$ for a random x tells us nothing about $g(x)$). Since f and g are ±1-valued functions, a moment's observation shows that the correlation can be alternatively written as

$$\mathrm{Cor}(f, g) = \left|\mathbf{E}\left[f(\mathbf{x})g(\mathbf{x})\right]\right| = |\langle f, g \rangle|.$$

More generally, for a probability distribution μ over $\{0, 1\}^n$, we define the correlation of f and g under μ as

$$\mathrm{Cor}_\mu(f, g) \overset{\text{def}}{=} \left|\mathbf{E}_{\mathbf{x}\sim\mu}\left[f(\mathbf{x})g(\mathbf{x})\right]\right| = \left| \sum_x f(x)g(x)\mu(x) \right|.$$

Now that we have an inner product, we can hope to define a *useful* orthonormal basis. The Fourier basis consists of the following functions. For each $S \subseteq [n]$, define $\chi_S : \{0, 1\}^n \to \{1, -1\}$ as

$$\chi_S(x) = (-1)^{\sum_{i \in S} x_i}.$$

In other words, the value of $\chi_S(x)$ is the parity of the variables in S, where -1 means the parity is odd and 1 means the parity is even. These functions are often called *characters* and in our setting we have 2^n of them. It is straightforward to verify that

$$\langle \chi_S, \chi_T \rangle = \begin{cases} 0 & \text{if } S \neq T, \\ 1 & \text{if } S = T. \end{cases}$$

So we conclude that the set of characters form on orthonormal basis for the vector space $\mathcal{V}$. This means that we can write every $\phi \in \mathcal{V}$ as a linear combination of the characters:

$$\phi(x) = \sum_{S \subseteq [n]} \widehat{\phi}(S) \chi_S(x).$$

Here, $\widehat{\phi}(S) \in \mathbb{R}$ denotes the coefficient corresponding to χ_S, and these coefficients are called the **Fourier coefficients**. This way of expanding ϕ as a linear combination of the characters is called the **Fourier expansion** of ϕ. Since the characters form on orthonormal basis, it follows that $\widehat{\phi}(S) = \langle \phi, \chi_S \rangle$. We will call the set of Fourier coefficients of ϕ the **Fourier spectrum** of ϕ.

Remark. It is worth noting that Fourier analysis can be applied more generally in the setting of $\mathcal{V} = \{\phi : G \to \mathbb{C}\}$, where G denotes an Abelian group (we view $\{0,1\}^n$ as $\mathbb{F}_2^n$ so we are in the special case of $G = \mathbb{F}_2^n$). A character $\chi : G \to \mathbb{C}$ is any function that satisfies $\chi(gh) = \chi(g)\chi(h)$ for all $g, h \in G$ (when $G = \mathbb{F}_2^n$, the parity functions χ_S are the only functions with this property). The set of all characters form an orthonormal basis for $\mathcal{V}$ with respect to the inner product $\langle \phi, \psi \rangle = \mathbf{E}\left[\phi(\mathbf{x})\overline{\psi(\mathbf{x})}\right]$. Therefore all functions in $\mathcal{V}$ can be written as a linear combination of the characters.

The elements of $\mathcal{V}$ are often referred to as *polynomials*. The reason for this is as follows. If we view the domain of ϕ as $\{1, -1\}^n$ rather than $\{0,1\}^n$, then observe that the characters take the form

$$\chi_S(x) = \prod_{i \in S} x_i.$$

That is, each character is a *multilinear*[3] monomial and the Fourier expansion of ϕ is simply a multilinear polynomial representation of the function. There is no real difference between the two representations and we will stick with the domain $\{0,1\}^n$.

Since $\mathcal{B} \subset \mathcal{V}$, every boolean function $f : \{0,1\}^n \to \{1,-1\}$ also has a Fourier expansion:

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x).$$

In essence, Fourier analysis of boolean functions is the study of boolean functions by looking at the information of how well the function correlates with different parity functions. This point of view turns out to be quite fruitful and we will see several applications of it in this thesis.

Let us now dive a little deeper and try to explore interesting features of Fourier analysis. We start with the most fundamental and essential fact, often called Parseval's Identity, which forms the bridge between the usual representation of a function in terms of the values $\{\phi(x) \mid x \in \{0,1\}^n\}$ and the Fourier representation in terms of the Fourier coefficients $\{\widehat{\phi}(S) \mid S \subseteq [n]\}$. It states that the inner product we defined for $\mathcal{V}$ (i.e. the expected value of the product of the functions) is the usual dot product of the Fourier coefficients.

Fact 2.3.1 (Parseval's Identity)**.** *For $\phi, \psi \in \mathcal{V}$,*

$$\langle \phi, \psi \rangle = \sum_{S \subseteq [n]} \widehat{\phi}(S)\widehat{\psi}(S).$$

Remark. Sometimes the above fact is called Plancherel's Theorem and the special case of $\phi = \psi$ is called Parseval's Identity. For convenience we call the general case Parseval's Identity.

[3] Multilinear means that each variable has exponent 0 or 1.

Proof. The proof simply substitutes the Fourier expansion of ϕ and ψ in the definition of the inner product and then uses the orthonormality of the characters:

$$\begin{aligned}\langle \phi, \psi \rangle = \mathbf{E}\left[\phi(\mathbf{x})\psi(\mathbf{x})\right] &= \mathbf{E}\left[\sum_{S\subseteq[n]} \widehat{\phi}(S)\chi_S(\mathbf{x}) \sum_{T\subseteq[n]} \widehat{\psi}(T)\chi_T(\mathbf{x})\right] \\ &= \sum_{S,T} \widehat{\phi}(S)\widehat{\psi}(T)\mathbf{E}\left[\chi_S(\mathbf{x})\chi_T(\mathbf{x})\right] = \sum_{S} \widehat{\phi}(S)\widehat{\psi}(S).\end{aligned}$$

□

A basic corollary of this fact is that for boolean functions, $\sum_S \widehat{f}(S)^2 = 1$. This is easy to see by substituting $\phi = \psi = f$ in Parseval's Identity. This allows us to view the squares of the Fourier coefficients of a boolean function as a probability distribution over the sets $S \subseteq [n]$. We will call this the **Fourier distribution**. In many different settings, how close this distribution is to the uniform distribution determines how *complex* a function is. There are of course various ways to measure how close a distribution is to the uniform distribution and which one to use depends on the context and application.

The previous paragraph in fact outlines a general theme about how Fourier analysis is used in computational complexity theory. In many different settings, the *hardness* of a function exposes itself in the function's Fourier expansion. In other words, different analytic measures associated with the Fourier coefficients of f can be good approximations to how *complex* the function is e.g. in communication complexity, circuit complexity, learning theory etc. Let us now define some of these useful measures.

The **degree** of a function ϕ is the degree of the multilinear polynomial representation of ϕ. In other words, degree of ϕ, denoted $\deg(\phi)$, is defined as $\max\{|S| : \widehat{\phi}(S) \neq 0\}$. The **monomial complexity** of ϕ is the number of monomials in its polynomial representation, i.e. $|\{S \mid \widehat{\phi}(S) \neq 0\}|$. We denote it by mon($\phi$).

The usual L_p norms are defined as:

$$\|\phi\|_p = \mathbf{E}\left[|\phi(\mathbf{x})|^p\right]^{1/p}.$$

With respect to the Fourier coefficients, we define

$$\|\widehat{\phi}\|_p = \left(\sum_S |\widehat{\phi}(S)|^p\right)^{1/p}.$$

Recall that Parseval's Identity implies $\|\phi\|_2 = \|\widehat{\phi}\|_2$ and for boolean functions this quantity is 1. We characterize this situation by saying that the total L_2 mass of a boolean function is 1. Other interesting L_p norms are the Fourier L_1 norm and the Fourier L_∞ norm. For a boolean function we have

$$1 \leq \|\widehat{f}\|_1 \leq 2^{n/2}.$$

The lower bound follows from the fact that $\|\widehat{f}\|_1 \geq \|\widehat{f}\|_2$ and the upper bound follows from the Cauchy-Schwarz inequality and $\|\widehat{f}\|_2 = 1$. Also we have

$$\frac{1}{2^{n/2}} \leq \|\widehat{f}\|_\infty \leq 1.$$

The lower bound follows from the fact that $\|\widehat{f}\|_2^2 \leq 2^n \max_S \widehat{f}(S)^2$ and the upper bound follows from $\|\widehat{f}\|_\infty \leq \|\widehat{f}\|_2$. The Fourier L_1 and L_∞ norms are measures of how close the Fourier distribution is to the uniform distribution. In fact the Fourier L_1 norm corresponds to the Rényi entropy of order 1/2 of the Fourier distribution:[4]

$$H_{1/2}[\widehat{\phi}^2] = 2\log\left(\sum_S |\widehat{\phi}(S)|\right) = 2\log\|\widehat{\phi}\|_1.$$

The Fourier L_∞ norm corresponds to the min-entropy:

$$H_\infty[\widehat{\phi}^2] = -\log\|\widehat{\phi}\|_\infty^2.$$

[4] For $\alpha > 0$ and $\alpha \neq 1$, the Rényi entropy of order α is defined as $H_\alpha(\mathbf{X}) = \frac{1}{1-\alpha}\log\left(\sum_{x\in\mathcal{X}} \mathbf{Pr}\left[\mathbf{X} = x\right]^\alpha\right)$.

For boolean functions, at the one extreme we have the constant function $f \equiv 1$, with $\|\widehat{f}\|_1 = \|\widehat{f}\|_\infty = 1$. On the other extreme, the *inner-product* function satisfies $|\widehat{f}(S)| = 1/2^{n/2}$ for all S, i.e., its Fourier distribution is uniform. Therefore $\|\widehat{f}\|_\infty = 1/2^{n/2}$ and $\|\widehat{f}\|_1 = 2^{n/2}$.

Let us move on to other useful measures that can serve as the complexity of a boolean function. We say that $\phi \in \mathcal{V}$ **sign represents** a boolean function f if $\phi(x)f(x) > 0$ for all x, in other words, $f(x) = \text{sign}(\phi(x))$ for all x. The **sign degree** of f, denoted $\deg_\pm(f)$, is the minimum degree of a function ϕ that sign represents f. Similarly, the **sign monomial complexity** of f, denoted $\text{mon}_\pm(f)$, is the minimum monomial complexity of a function ϕ that sign represents f. As an example, first consider the *majority* function. Observe that $\textsc{maj}(x) = \text{sign}((-1)^{x_1} + \cdots + (-1)^{x_n} - 0.5)$ and so $\deg_\pm(\textsc{maj}) = 1$ and $\text{mon}_\pm(\textsc{maj}) \leq n+1$. On the other hand, it is quite straightforward to show that the *parity* function satisfies $\deg_\pm(\textsc{par}) = n$. Let ϕ sign represent $\textsc{par}$. Then $\langle \phi, \textsc{par} \rangle > 0$ by definition of sign representation. Because $\textsc{par} = \chi_{[n]}$ and the characters are orthogonal, we have $\langle \phi, \textsc{par} \rangle = 0$ for any ϕ with $\deg(\phi) \leq n-1$. Therefore the function ϕ that sign represents $\textsc{par}$ must have degree n. It is also not too difficult to show that $\text{mon}_\pm(\textsc{ip}) \geq 2^{n/2}$. In fact a classic result of Bruck [Bru90] shows that $\text{mon}_\pm(f) \geq \|\widehat{f}\|_\infty^{-1}$.

A function ϕ **ϵ-approximates** f if for all x, $|\phi(x) - f(x)| \leq \epsilon$. In other words, ϕ approximates f within ϵ in the infinity norm: $\|f - \phi\|_\infty \leq \epsilon$. We can define **$\epsilon$-approximate degree** (**ϵ-approximate monomial complexity**, **ϵ-approximate p-norm**) as the minimum degree (monomial complexity, p-norm) of a function that ϵ-approximates f. We denote these quantities by $\deg_\epsilon(f), \text{mon}_\epsilon(f)$ and $\|\widehat{f}\|_{p,\epsilon}$. We think of ϵ as a fixed constant in the range $[0,1]$ such as $1/3$. A classic result of Paturi [Pat92], that has found many applications in theoretical computer science, characterizes the approximate degree of all symmetric functions. Let $t_0(f) \in [\lfloor n/2 \rfloor]$ and $t_1 \in [\lceil n/2 \rceil]$ be the minimum integers such that $f(i) = f(i+1)$ for all $i \in [t_0(f), n - t_1(f)]$.

Theorem 2.3.2 ([Pat92]). *For $f : \{0,1\}^n \to \{1,-1\}$ a symmetric function,*

$$\deg_{1/3}(f) = \Theta\left(\sqrt{n(t_0(f) + t_1(f))}\right).$$

All these measures $\deg(f)$, $\mathrm{mon}(f)$, $\|\widehat{f}\|_p$, $\deg_\pm(f)$, $\mathrm{mon}_\pm(f)$, $\deg_\epsilon(f)$, $\mathrm{mon}_\epsilon(f)$, $\|\widehat{f}\|_{p,\epsilon}$ can serve as a reasonable measure of complexity of f depending on the particular context. Note that by definition we have

$$\deg_\pm(f) \le \deg_\epsilon(f) \le \deg(f),$$

$$\mathrm{mon}_\pm(f) \le \mathrm{mon}_\epsilon(f) \le \mathrm{mon}(f),$$

and

$$\|\widehat{f}\|_{p,\epsilon} \le \|\widehat{f}\|_p.$$

2.3.1 Noise Stability

In this subsection we will introduce a very important concept in Fourier analysis of boolean functions: the noise operator and noise stability. In many different situations, the noise operator serves as the crucial connection between the combinatorial properties of a boolean function and its Fourier properties.

We begin by defining the noise operator. For $x \in \{0,1\}^n$ and $\rho \in [0,1]$, we say that $\mathbf{y}$ is a ρ-noisy copy of x, denoted $\mathbf{y} \sim_\rho x$, if $\mathbf{y}$ is such that for each $i \in [n]$ independently, we have:

$$\mathbf{y}_i = \begin{cases} x_i & \text{with probability } \rho, \\ 0 & \text{with probability } \frac{1-\rho}{2}, \\ 1 & \text{with probability } \frac{1-\rho}{2}. \end{cases}$$

We also write $\mathbf{y}_i \sim_\rho x_i$ when the coordinates have the above relation. We write $\mathbf{y} \sim_\rho \mathbf{x}$ when $\mathbf{x}$ is uniformly distributed over $\{0,1\}^n$ and $\mathbf{y}$ is then chosen to be a ρ-noisy copy of $\mathbf{x}$. Note that we have symmetry: $\mathbf{y}$ has uniform distribution over $\{0,1\}^n$ and $\mathbf{x} \sim_\rho \mathbf{y}$.

For $\phi : \{0,1\}^n \to \mathbb{R}$, we define the **noise operator** T_ρ to be such that

$$T_\rho \phi(x) = \mathbf{E}_{\mathbf{y} \sim_\rho x} \left[\phi(\mathbf{y}) \right].$$

It is easy to check that T_ρ is linear in the sense that

$$T_\rho(\phi + c\psi) = T_\rho \phi + c T_\rho \psi.$$

Now let us see how T_ρ affects a function's Fourier expansion.

Proposition 2.3.3. *Let* $\phi : \{0,1\}^n \to \mathbb{R}$*. Then,*

$$T_\rho \phi = \sum_{S \subseteq [n]} \rho^{|S|} \widehat{\phi}(S) \chi_S.$$

Proof. Since T_ρ is a linear operator, it suffices to show that $T_\rho \chi_S = \rho^{|S|} \chi_S$, which is quite straightforward:

$$\begin{aligned} T_\rho \chi_S(x) &= \mathbf{E}_{\mathbf{y} \sim_\rho x} \left[\chi_S(\mathbf{y}) \right] = \mathbf{E}_{\mathbf{y} \sim_\rho x} \left[\prod_{i \in S} (-1)^{\mathbf{y}_i} \right] = \prod_{i \in S} \mathbf{E}_{\mathbf{y}_i \sim_\rho x_i} \left[(-1)^{\mathbf{y}_i} \right] \\ &= \prod_{i \in S} \rho (-1)^{x_i} = \rho^{|S|} \chi_S. \end{aligned}$$

□

This proposition shows that the noise operator dampens the high degree Fourier coefficients and the dampening increases exponentially with the degree.

With regards to boolean functions, our main interest will be in how sensitive a function is when noise is applied to its input. To measure this, we look at the correlation of the function with its noisy version. More formally, define the **noise stability** of a function $\phi : \{0,1\}^n \to \mathbb{R}$ to be

$$\mathrm{Stab}_\rho(\phi) \overset{\text{def}}{=} \mathbf{E}_{\mathbf{y} \sim_\rho \mathbf{x}} \left[\phi(\mathbf{x}) \phi(\mathbf{y}) \right].$$

For boolean functions f, this corresponds to

$$\mathbf{Pr}\left[f(\mathbf{x}) = f(\mathbf{y})\right] - \mathbf{Pr}\left[f(\mathbf{x}) \neq f(\mathbf{y})\right].$$

By the definition of the noise operator, we can equivalently write

$$\mathrm{Stab}_\rho(\phi) = \mathbf{E}_{\mathbf{x}\sim U}\left[\phi(\mathbf{x})T_\rho\phi(\mathbf{x})\right] = \langle \phi, T_\rho\phi \rangle.$$

Using Parseval's identity and Proposition 2.3.3, we see that the noise stability of a function has a clean Fourier formula:

$$\mathrm{Stab}_\rho(\phi) = \langle \phi, T_\rho\phi \rangle = \sum_{S\subseteq[n]} \rho^{|S|}\widehat{\phi}(S)^2.$$

This in particular shows that the noise stability of a function is always non-negative, which is not immediately obvious from the original definition of noise stability. Also observe that the above formula implies that noise stable functions must have significant Fourier weight on the low degree coefficients. This intuitively makes sense too since high degree characters are very noise sensitive.

CHAPTER 3

NOF Communication Complexity of Composed Functions

Two of the most well-known and studied functions in the standard two party as well as the multiparty models are the *generalized inner product* function GIP and the *disjointness* function DISJ. The GIP function is a hard function (or conjectured to be hard) in almost every model of communication complexity. As such, strong lower bounds can be proven for many different kinds of boolean circuits using GIP [HG91, Nis93, FKL^{+}01, Gro98]. It is also used in obtaining decision tree lower bounds [Nis93], in the construction of pseudorandom generators, time/space trade-offs for Turing Machines and branching program lower bounds [BNS92].

The DISJ function, unlike GIP, is easy in the non-deterministic model. Proving lower bounds for DISJ in the randomized model even for 2 players was a major challenge. A strong lower bound for 3 players has been proven only very recently [LS09, CA08]. Both the 2 player and multiplayer lower bounds on DISJ lead to the development of interesting techniques and a deeper understanding of communication complexity in general. Apart from this, the interest in studying DISJ also stems from the fact that it is very suitable for reductions: communication complexity lower bounds for DISJ (and slight variations) have been successfully used to give lower bounds in the context of data streaming [AMS99], proof complexity [BPS07], data structures [MNSW98], game theory [CS04, NS06], boolean circuits [NW93], and property testing [BBM11].

The functions GIP and DISJ have the following composed structure. Let $f : \{0,1\}^n \to \{-1,1\}$ and $g : \{0,1\}^k \to \{0,1\}$ be two functions. Define $f \circ g(x_1, \ldots, x_k) = f(\ldots, g(x_{1,i}, x_{2,i}, \ldots, x_{k,i}), \ldots)$, where $x_{j,i}$ denotes the ith coordinate of the n-bit string x_j. In this notation, GIP $= \textsc{mod}_2 \circ \textsc{and}$ and DISJ $= \textsc{nor} \circ \textsc{and}$, where $\textsc{nor}$ is the negation of $\textsc{or}$. Many other important and well-studied functions in communication complexity are also composed. In both the two party and the multiparty models, functions of the form $f \circ \textsc{and}$ have been studied extensively [Raz95, Raz03, Kla07, She07, SZ09b, LS09, CA08, BHN09], with an emphasis on $\textsc{sym} \circ \textsc{and}$, where $\textsc{sym}$ represents a symmetric function. For instance, in the important paper [Raz03], Razborov shows that the 2 party quantum and classical communication complexities of $\textsc{sym} \circ \textsc{and}$ are polynomially related. Functions of the form $f \circ \textsc{xor}$ have also received a lot of attention in the 2 player setting [Raz95, Kla07, SZ09a, MO09], especially the *Hamming distance* problem $\textsc{thr}_t \circ \textsc{xor}$ where $\textsc{thr}_t$ is a threshold function.[1] Notably, Shi and Zhang [SZ09a] obtain 2 party classical and quantum equivalence of functions of the form $\textsc{sym} \circ \textsc{xor}$.

In this chapter, we study the multiparty communication complexity of composed functions with two goals in mind. The first goal is to better understand the power of $\log n$ and more players. The second and more general

[1] Even though $\textsc{xor}$ and $\textsc{mod}_2$ both represent the *parity* function, we use the notation $\textsc{xor}$ for an inside function g and $\textsc{mod}_2$ for an outside function f.

goal is to understand which combinations of the "inside" function g and the "outside" function f lead to hard communication problems and which combinations lead to easy communication problems. The focus of previous research has been on proving lower bounds for composed functions by selecting a "hard" outside function and a convenient inside function (see e.g., [She07, SZ09b, LS09, CA08, BHN09, LZ10]). Our approach is to study composed functions without putting any restriction on g and obtain characterizations for the communication complexity of composed functions with respect to the choice of g. This *dual* approach is particularly interesting in the multiparty setting where the choice for g increases double exponentially in k.

First we study functions of the form $\textsc{sym} \circ g$ in the setting of $\log n$ and more players. As discussed in Chapter 1, a natural question is whether any of these functions can break the $\log n$ barrier. In particular, functions of the form $\textsc{maj} \circ g$ for some specific g have been considered as possible candidates (see e.g., [BKL95, BGKL03]). In Section 3.1 we show that for *any* g, $\textsc{sym} \circ g$ has an efficient simultaneous protocol for more than $\log n$ many players.

Second, we study functions of the form $\textsc{maj}\circ g$, $\textsc{mod}_m \circ g$ and $\textsc{nor}\circ g$ in the setting of less than $\log n$ many players. The latter two are generalizations of GIP and DISJ respectively. We characterize the communication complexity of these functions with respect to the choice of g. This in particular allows us to show that such functions have polynomially related quantum and classical communication complexities. These results are presented in Sections 3.2, 3.3 and 3.4. It is worth mentioning that our upper bounds for $\textsc{maj} \circ g$ and $\textsc{mod}_m \circ g$ have a natural Fourier analytic reinterpretation. Some readers may find this point of view more natural and intuitive and so we present this in Section 3.5.

Although all our lower bounds apply in the quantum model, we will state them using classical communication complexity notation.

3.1 SYM $\circ$ g

In this section we present a deterministic protocol for SYM $\circ\, g$ where SYM denotes an arbitrary symmetric function and g is an arbitrary boolean function. The protocol becomes simultaneous and efficient when $k > 1 + \log n$.

A multiparty non-simultaneous protocol for such a function, GIP $=$ MOD$_2$ $\circ$ AND, was first discovered by Grolmusz [Gro94]. This protocol is non-trivial for all k but only efficient when k reaches $\log n$ (the complexity is similar to Theorem 3.1.2 part (a)). It is not difficult to see that the protocol also works for SYM $\circ$ AND. Later Pudlák [Pud06] gave a non-simultaneous protocol for SYM $\circ$ AND, which can be considered as a very elegant reinterpretation of Grolmusz's protocol (Pudlák's protocol is described in detail in [Cha08]). Babai et al. [BGKL03], using a new idea, obtained a simultaneous protocol for SYM $\circ\, g$ where g is a symmetric and compressible function, when $k > 1 + \log n$ (see [BGKL03, Section 6] for the definition of a compressible function). Although the class of symmetric compressible functions contains natural functions like THR$_t$ and MOD$_m$, this class is only a small portion of all symmetric functions as a random symmetric function is not compressible with high probability. Babai et al. [BGKL03] in fact identify the *quadratic character of the sum of bits* function as a symmetric inside function g for which their method fails.

We improve upon the result of [BGKL03] in two ways. First, we remove the symmetry and compressibility conditions on g and allow inside function(s) to be selected arbitrarily, and second, we provide a non-trivial protocol even when $k \leq 1 + \log n$. This rules out any function of the form SYM $\circ\, g$ as a candidate to break the $\log n$ barrier; this in particular rules out functions of the form MAJ $\circ$ g, which have been studied before as possible candidates.

We obtain our protocols in the non-simultaneous model by extending the ideas of Grolmusz and Pudlák. We combine this with a beautiful lemma of Babai et al. [BGKL03, Lemma 6.10] in order to make our protocols simultaneous. Before we state this lemma and our result, let us first set some notation.

We view an input $(x_1, x_2, \ldots, x_k) \in (\{0,1\}^n)^k$ to the players as a $k \times n$ matrix X where the ith row of X is x_i. We generalize the definition of a composed function in the following way. Define $f \circ \vec{g}$, where $f : \{0,1\}^n \to \{\pm 1\}$, $\vec{g} = (g_1, \ldots, g_n)$ with $g_i : \{0,1\}^k \to \{0,1\}$, by $f \circ \vec{g}(x_1, \ldots, x_k) = f(\ldots, g_i(x_{1,i}, x_{2,i}, \ldots, x_{k,i}), \ldots)$. That is, we apply g_i to the ith column of X, and then apply f to the resulting n-bit string to obtain the output. When all the g_i are the same function g, we recover $f \circ g$. Let $\mathcal{H}_k$ denote the k dimensional hypercube where the vertex set is $\{0,1\}^k$ and there is an edge between two vertices iff their Hamming distance is 1. Given an input in the $k \times n$ dimensional matrix form X, we associate each column of X with the corresponding vertex of $\mathcal{H}_k$.

Lemma 3.1.1 ([BGKL03]). *Suppose $k > 1 + \log n$ and let X be a $k \times n$ boolean matrix given as an input for a k party communication problem. Let n_i be the number of columns of X with Hamming weight i. Then there is a simultaneous deterministic protocol in which each player sends at most $O(k \log n)$ bits to a referee, who then can compute n_i for all $i \in \{0, \ldots, n\}$.*

We note that in the following theorem, it will be clear from the proof that allowing different inner functions for different columns is important even to handle functions $f \circ g$ when the number of players $k \gg \log n$.

Theorem 3.1.2. *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a symmetric function, $g : \{0,1\}^n \to \{0,1\}$ an arbitrary function, and $\vec{g} = (g_1, \ldots, g_n)$ a vector of n functions where $g_i : \{0,1\}^k \to \{0,1\}$ are arbitrary functions. Then,*

(a) $\mathbf{D}_k(f \circ \vec{g}) \leq O(n/2^k \cdot \log n + k \log n)$,

(b) *for* $k > 1 + \log n$*:* $\mathbf{D}_k^{||}(f \circ g) \leq O(\log^3 n)$*,*

(c) *for* $k > 1 + 2\log n$*:* $\mathbf{D}_k^{||}(f \circ \vec{g}) \leq O(\log^3 n)$*.*

Proof. We first prove part **(a)**. Fix an input for $f \circ \vec{g}$ given in $k \times n$ matrix form X. The protocol proceeds in two steps. In the first step, the players determine the column positions of some $u \in \mathcal{H}_k$. Later, they use this to compute the output of $f \circ \vec{g}$.

We now describe the first step. Let $X^{\geq 3}$ denote the $(k-2) \times n$ dimensional submatrix of X where the first two rows are deleted. Since $X^{\geq 3}$ has n columns and there are 2^{k-2} possible strings of length $k-2$, the string $s \in \{0,1\}^{k-2}$ that appears the least number of times as a column of $X^{\geq 3}$ appears at most $n/2^{k-2}$ times. Without any communication, Player 1 and Player 2 can agree on this string (breaking ties in say lexicographical order). Player 2, using at most $n/2^{k-2}$ bits of communication, can send Player 1 the bits on Player 1's forehead corresponding to the positions that string s appears. With this information, Player 1 knows the positions of four vertices $00s$, $01s$, $10s$ and $11s$. Now Player 1 can announce one of these vertices (call it u) and the column indices corresponding to u. The total cost is at most $O((n/2^k)\log n)$.

We proceed to step 2. Observe that the columns corresponding to u are taken care of, that is, we already know the value $g_j(u)$ where j is a column index corresponding to u. Let $S_j = g_j^{-1}(1)$. For $v \in \{0,1\}^k$, let $\mathbf{1}_j(v) = 1$ if v is in column j, and $\mathbf{1}_j(v) = 0$ otherwise. To compute the output of $f \circ \vec{g}$, it suffices to compute

$$\sum_j \sum_{v \in S_j} \mathbf{1}_j(v), \tag{3.1}$$

where the outer sum is over all column indices that do not correspond to u. Consider a shortest path from v to u in $\mathcal{H}_k$: $v = w_1, w_2, \ldots, w_t = u$. Observe that since $\mathbf{1}_j(u) = 0$,

$$\mathbf{1}_j(v) = \sum_{i=1}^{t-1} (-1)^{i+1}(\mathbf{1}_j(w_i) + \mathbf{1}_j(w_{i+1})). \tag{3.2}$$

Each term $(\mathbf{1}_j(w_i) + \mathbf{1}_j(w_{i+1}))$ is known by some player because w_i and w_{i+1} differ only in one coordinate. To compute (3.1), each player announces her part of the sum. Since $\sum_j \sum_{v \in S_j} \mathbf{1}_j(v) \leq n$, it suffices for players to send their part of the sum modulo $n+1$. Therefore this step of the protocol has cost at most $k \cdot \lceil \log(n+1) \rceil$. This completes the proof of part **(a)**. Note that the second step of the protocol is simultaneous while the first step is not. When k is sufficiently large, we bypass the first step using Lemma 3.1.1.

We now prove part **(c)**. Let $\ell = 2+2\log n$. For bit strings u and v, let $u \cdot v$ be the concatenation of u and v. Only the first ℓ players will speak. For each column j, the rows $\ell+1$ to k naturally induce a function $g'_j : \{0,1\}^\ell \to \{0,1\}$; $g'_j(u) = g_j(u \cdot v)$ where $v \in \{0,1\}^{k-\ell}$ appears in column j from row $\ell+1$ to k. Thus our task reduces to finding a protocol for $f \circ \vec{g'}$ with ℓ players. From now on we drop the superscript in g'_j and denote it by g_j.

As before we are interested in computing

$$\sum_{j=1}^{n} \sum_{v \in S_j} \mathbf{1}_j(v). \tag{3.3}$$

Let $\vec{0}$ be the all 0 vertex. Let $v \in S_j$ and let $v = w_1, \ldots, w_t = \vec{0}$ be a shortest path between v and $\vec{0}$. Then we have

$$\mathbf{1}_j(v) = \sum_{i=1}^{t-1} (-1)^{i+1} (\mathbf{1}_j(w_i) + \mathbf{1}_j(w_{i+1})) + (-1)^{|v|} \mathbf{1}_j(\vec{0}). \tag{3.4}$$

Substitute (3.4) into (3.3). Since the quantity in (3.3) is at most n, we can do arithmetic modulo $n+1$. As before, each term $(\mathbf{1}_j(w_i) + \mathbf{1}_j(w_{i+1}))$ in the sum is known to a player so the part of the sum involving these terms can be computed by the players using at most $\ell \cdot \lceil \log(n+1) \rceil$ bits. For each $j \in \{1, \ldots, n\}$, we group the terms involving $\mathbf{1}_j(\vec{0})$ when substituting (3.4) into (3.3) (excluding the $\mathbf{1}_j(\vec{0})$ appearing as $\mathbf{1}_j(w_t)$ in the sum in (3.4)) and let c_j be the coefficient of $\mathbf{1}_j(\vec{0})$ modulo $n+1$. We also need to compute $\sum_j c_j \mathbf{1}_j(\vec{0})$, which can be done as follows. From the original $\ell \times n$ input matrix X, we create a new matrix X' by duplicating the jth column c_j many times. Note that X' has at most n^2 columns so we can apply Lemma 3.1.1 on X' to compute the number of all 0 columns in X', which is exactly what we want. This step has cost $O(\log^3 n)$. So putting things together, we can compute (3.3) with at most $O(\log^3 n)$ bits of communication. The whole protocol is easily seen to be simultaneous. This completes the proof of part **(c)**.

We conclude with the proof of part **(b)**. The strategy is exactly the same as above. We need to calculate $\sum_j c_j \mathbf{1}_j(\vec{0})$. Since all the g_j are the same, $c_j = c$ for all j for some c. So we want to compute $c \sum_j \mathbf{1}_j(\vec{0})$, which is precisely cn_0 (n_0 is defined in the statement of Lemma 3.1.1). We can compute n_0 using Lemma 3.1.1 when $k > 1 + \log n$. So putting things together, we can compute (3.3) using at most $O(k^2 \log n)$ bits of communication. Given part **(c)**, we are done. □

Remark. For functions of the form $\textsc{sym} \circ g$, we can make a small improvement to part **(a)** and show $\mathbf{D}_k(\textsc{sym} \circ g) \leq O(n/2^{k-2} + (k+1) \log n)$ as follows. In light of the proof of part **(b)** above, in step 1 of the protocol, all Player 1 needs to communicate is a vertex u and the number of occurrences of u. The column indices corresponding to u are not needed. Thus the cost of step 1 is at most $n/2^{k-2} + k + \lceil \log(n/2^{k-2}) \rceil = n/2^{k-2} + \lceil \log n \rceil + 2$. Combined with step 2, the total cost is at most $n/2^{k-2} + (k+1) \cdot \lceil \log(n+1) \rceil + 2$. Furthermore, we can also improve part **(c)** when we allow ourselves to be non-simultaneous and show $\mathbf{D}_k(\textsc{sym} \circ \vec{g}) \leq O(\log^2 n)$. To see this, set $\ell = \lceil \log(n+1) \rceil$ in the proof of part **(c)**. Observe that there is a vertex $u \in \{0,1\}^\ell$ that does not appear as a column in the first ℓ rows of the input matrix. Player k announces this vertex using ℓ bits. We replace $\vec{0}$ with u in the proof and note that $\mathbf{1}_j(u) = 0$ for all j. Therefore the desired output can be computed using $\ell + \ell \cdot \lceil \log(n+1) \rceil = \lceil \log(n+1) \rceil^2 + \lceil \log(n+1) \rceil$. These slightly improved upper bounds will be used in Section 4.1.1.

In what follows, we study the communication complexities of $\textsc{mod}_m \circ g$, $\textsc{maj} \circ g$ and $\textsc{nor} \circ g$, for any boolean function g. All these functions are of the form $\textsc{sym} \circ g$ and so for $k > 1 + \log n$, the $O(\log^3 n)$ simultaneous communication complexity upper bound just presented applies to these functions. We note that we will not mention this $O(\log^3 n)$ upper bound explicitly and consider ourselves in the setting of $k \leq 1 + \log n$.

3.2 $\mathrm{MOD}_m \circ g$

Remark. All the lower bounds that will be presented in this and subsequent sections apply in the quantum models. We however stick to the classical communication complexity notation as in Chapter 2.

In this section, we determine the k-party communication complexity of $\textsc{mod}_m \circ g$, for every function g. Babai, Nisan and Szegedy [BNS92] show a lower bound of $\Omega(n/4^k)$ for the k-party randomized communication complexity of generalized inner product $\mathrm{GIP} = \textsc{mod}_2 \circ \textsc{and}$. Their proof is later refined by [CT93, Raz00], where the technique of upper bounding the discrepancy via the cube measure (Lemma 2.2.15) is introduced. Grolmusz [Gro95] extends the analysis of [BNS92] to get an $\Omega(n/4^k)$ lower bound for $\textsc{mod}_m \circ \textsc{and}$, for constant m. Viola and Wigderson [VW08] obtain the same result by extending the analysis of [CT93, Raz00].

We show that in general, the communication complexity of $\textsc{mod}_m \circ g$ is determined by the quantity $\big||S_0| - |S_1|\big|$, where S_i is the subset of the support of g that consists of all inputs whose Hamming weight has parity i. (For the case where $g = \textsc{and}$, considered in the mentioned papers, the support of g is $(1, 1, \ldots, 1)$, so $\big||S_0| - |S_1|\big| = 1$.) We prove a dichotomy for the communication complexity of $\textsc{mod}_m \circ g$. When m divides $|S_0| - |S_1|$, we exhibit an efficient protocol by using ideas from the protocol for $\textsc{sym} \circ g$ presented in the previous section. On the other hand, when m does not divide $|S_0|-|S_1|$, we show an $\Omega(n/m^2 4^k)$ lower bound (ignoring some additive logarithmic factors). The case of m not dividing $|S_0|-|S_1|$ is analysed in two parts. When m and $|S_0|-|S_1|$ are coprime, we use the Discrepancy Method (Proposition 2.2.14) in conjunction with a careful analysis of the cube measure (the expectation in Lemma 2.2.15) to obtain the desired lower bound. We prove that there is also a strong lower bound for randomized protocols in the remaining case (where m and $|S_0|-|S_1|$ are not coprime but m does not divide $|S_0|-|S_1|$) by giving a reduction to the previous case. This reduction also uses ideas from our protocol for $\textsc{sym} \circ g$.

In the analysis of discrepancy, we will make use of the characterization of the $\textsc{mod}_m$ function in terms of exponential sums. Fix $2 \le m \in \mathbb{N}$ and $0 \le a, b \le m-1$. Let $\omega = e^{2\pi i/m}$ be an m-th root of unity. For $(y_1, y_2, \ldots, y_n) \in \{0,1\}^n$ the function $\textsc{exp}_m^{a,b}(y_1, y_2, \ldots, y_n)$ is defined to be

$$\textsc{exp}_m^{a,b}(y_1, y_2, \ldots, y_n) = \omega^{a((\sum_{j=1}^n y_j)-b)}.$$

It is straightforward to check that for any b,

$$\frac{1}{m}\sum_{a=0}^{m-1} \textsc{exp}_m^{a,b}(y_1, y_2, \ldots, y_n) \in \{0,1\}$$

and

$$\frac{1}{m}\sum_{a=0}^{m-1} \textsc{exp}_m^{a,0}(y_1, y_2, \ldots, y_n) = 1 \text{ if and only if } \textsc{mod}_m(y_1, y_2, \ldots, y_n) = -1. \tag{3.5}$$

Before presenting the main result of this section, we first state a fact which we need for our upper bound (when m divides $|S_0|-|S_1|$) and our reduction (when m and $|S_0|-|S_1|$ are not coprime but m does not divide $|S_0|-|S_1|$). This fact essentially follows from the argument presented in the proof of Theorem 3.1.2. Recall that $\mathcal{H}_k$ denotes the k-dimensional hypercube. For each vertex v of the hypercube, we define n_v to be the number of occurrences of v as a column of X.

Fact 3.2.1. *Let $S_0 = \{u_1, \ldots, u_r\}$ and $S_1 = \{v_1, \ldots, v_r\}$ be two subsets of the vertices of $\mathcal{H}_k$ such that for each i, the distance between u_i and v_i is odd. The sum $\sum_{i=1}^{r} n_{u_i} + \sum_{i=1}^{r} n_{v_i} \mod m$ can be computed by the players in the simultaneous model using at most $k \cdot \lceil \log m \rceil$ bits. Similarly, if for each i, the distance between u_i and v_i is even, $\sum_{i=1}^{r} n_{u_i} - \sum_{i=1}^{r} n_{v_i} \mod m$ can be computed in the simultaneous model using at most $k \cdot \lceil \log m \rceil$ bits.*

Proof. Note that we are interested in computing $\sum_{i=1}^{r}(n_{u_i} + n_{v_i}) \mod m$. Each term $(n_{u_i} + n_{v_i})$ can be written as a telescoping sum as in (3.2). Each term in the telescoping sum is known by a player. Since we can do arithmetic modulo m, the desired value can be computed with each player sending their part of the sum modulo m. So the total cost is $k \cdot \lceil \log m \rceil$. The second part holds similarly. □

Theorem 3.2.2. *Let $m \geq 2$ be an integer, $g : \{0,1\}^k \to \{0,1\}$ be a boolean function and $S = \{y \in \{0,1\}^k : g(y) = 1\}$ be its support. Define $S_0 = \{y \in S : y \text{ has even weight}\}$ and $S_1 = \{y \in S : y \text{ has odd weight}\}$. Then the function $\text{MOD}_m \circ g$ satisfies the following:*

(a) *If m divides $|S_0| - |S_1|$, then $\mathbf{D}_k^{||}(\text{MOD}_m \circ g) \leq k \lceil \log m \rceil$.*

(b) *Otherwise, $\mathbf{R}_k^{\epsilon}(\text{MOD}_m \circ g) \geq \frac{5n}{m^2 4^k} + \log(1 - 2\epsilon) - (k+1)\lceil \log m \rceil - 1$.*

Proof. **Part (a):** Suppose that m divides $|S_0| - |S_1|$; we will give an efficient protocol for $\text{MOD}_m \circ g$. Assume without loss of generality that $|S_0| \geq |S_1|$. We choose (arbitrarily) a subset $S_0' \subseteq S_0$ of size $|S_1|$. As the distance between an element of S_0' and an element of S_1 is odd, we can compute $\sum_{v \in S_0'} n_v + \sum_{v \in S_1} n_v \mod m$ using Fact 3.2.1. For the remaining elements in $S_0 - S_0'$, we simply pair them with $\vec{0}$. Therefore, using Fact 3.2.1 once again, we can compute $(|S_0| - |S_1|)n_{\vec{0}} + \sum_{v \in S_0 - S_0'} n_v \equiv \sum_{v \in S_0 - S_0'} n_v \mod m$. Thus, we have computed $\sum_{v \in S_0 \cup S_1} n_v \mod m$, from which the output of $\text{MOD}_m \circ g$ is determined. Observe that the sums $\sum_{v \in S_0'} n_v + \sum_{v \in S_1} n_v \mod m$

and $\sum_{v \in S_0 - S_0'} n_v \mod m$ need not be computed separately and that we can compute $\sum_{v \in S_0 \cup S_1} n_v \mod m$ in one shot using $k \lceil \log m \rceil$ bits of communication. (See Section 3.5 for a reinterpretation of this protocol in terms of computing polynomials.)

Part (b), Case 1: We consider two cases, depending on whether m and $|S_0| - |S_1|$ are coprime or not. The first case is when m and $|S_0| - |S_1|$ are coprime.

For $(y_1, y_2, \cdots, y_n) \in \{0,1\}^n$, define $f_m(y_1, \ldots, y_n) = \sum_j y_j \mod m$. Also for $b \in \{0, 1, \ldots, m-1\}$, let $f_m^b(y_1, \ldots, y_n) = 1$ if $\sum_j y_j \equiv b \mod m$, and 0 otherwise. Note that f_m^b are 0/1 valued functions rather than ± 1 valued like $\textsc{mod}_m$. We define $F_m = f_m \circ g$ and $F_m^b = f_m^b \circ g$.

The strategy is as follows. Assume g is not constant. First note that by an elementary argument, one can show that the fraction of points x with $F_m(x) = b$ is roughly (with an exponentially small error) $1/m$ for all $b \in \{0, 1, \ldots, m-1\}$. It is possible to show that the same holds within any cylinder intersection that is not very small by analysing the cube measure of the functions $\textsc{exp}_m^{a,b} \circ g$ with respect to the uniform distribution. This step uses the assumption that $|S_0| - |S_1|$ and m are coprime. It follows that in any sufficiently large cylinder intersection, the number of points x with $F_m^0(x) = 1$ is roughly the same as the number of points x with $F_m^1(x) = 1$. Define the distribution μ that puts equal weight to all x with $F_m^0(x) = 1$ and $F_m^1(x) = 1$. All other points get 0 weight. The discrepancy $\text{disc}_\mu(\textsc{mod}_m \circ g)$ can now be easily upper bounded and this yields the desired lower bound via the Discrepancy Method (Proposition 2.2.14).

We now flesh out the above strategy. Let $\mathcal{C}$ be a cylinder intersection and U denote the uniform distribution over $(\{0,1\}^n)^k$. We will denote by ϕ the characteristic function of $\mathcal{C}$. For any b, we have

$$\begin{aligned}\mathbf{E}\left[F_m^b(\mathbf{x})\phi(\mathbf{x})\right] &= \mathbf{E}\left[\frac{1}{m}\sum_{a=0}^{m-1}\mathrm{EXP}_m^{a,b}\circ g(\mathbf{x})\phi(\mathbf{x})\right]\\ &= \frac{1}{m}\sum_{a=0}^{m-1}\mathbf{E}\left[\mathrm{EXP}_m^{a,b}\circ g(\mathbf{x})\phi(\mathbf{x})\right],\end{aligned}$$

where all the expectations are with respect to the uniform distribution. The term corresponding to $a = 0$ contributes $\frac{1}{m}\frac{|\mathcal{C}|}{2^{nk}}$ to the sum, and thus we can write

$$\frac{1}{m}\frac{|\mathcal{C}|}{2^{nk}} - error \leq \mathbf{E}\left[F_m^b(\mathbf{x})\phi(\mathbf{x})\right] \leq \frac{1}{m}\frac{|\mathcal{C}|}{2^{nk}} + error, \tag{3.6}$$

where $error = \frac{1}{m}\sum_{a=1}^{m-1}\left|\mathbf{E}\left[\mathrm{EXP}_m^{a,b}\circ g(\mathbf{x})\phi(\mathbf{x})\right]\right|$. Note that the terms of this sum are exactly $\mathrm{disc}_U(\mathrm{EXP}_m^{a,b}\circ g, \mathcal{C})$, which can be upper bounded using the cube measure (Lemma 2.2.15). The following lemma gives an upper bound on the cube measure of $\mathrm{EXP}_m^{a,b}\circ g$.

Lemma 3.2.3. *Assume m and $|S_0| - |S_1|$ are coprime. Then for any $a \in \{1, 2, \ldots, m-1\}$ and $b \in \{0, 1, \ldots, m-1\}$,*

$$\mathcal{E}_U(\mathrm{EXP}_m^{a,b}\circ g) \leq \frac{1}{e^{8n/(m^2 2^k)}}.$$

We defer the proof of this lemma to the end of the section to not break the flow. We can now upper bound the error:

$$error < \frac{1}{e^{8n/(m^2 4^k)}}.$$

From this, it easily follows that the number of points with $F_m^0(x) = 1$ is very close to the number of points with $F_m^1(x) = 1$, with exponentially small error:

$$\begin{aligned}\left|\mathbf{E}\left[(F_m^0(\mathbf{x}) - F_m^1(\mathbf{x}))\phi(\mathbf{x})\right]\right| &= \left|\mathbf{E}\left[F_m^0(\mathbf{x})\phi(\mathbf{x})\right] - \mathbf{E}\left[F_m^1(\mathbf{x})\phi(\mathbf{x})\right]\right|\\ &\leq 2\cdot error.\end{aligned}$$

Recall the definition of μ, and let $\alpha > 0$ be the non-zero weight that μ assigns to a point in the support of F_m^0 and F_m^1. Then,

$$\begin{aligned}
\mathrm{disc}_\mu(\textsc{MOD}_m \circ g, C) &= \left|\sum_x \textsc{MOD}_m \circ g(x)\phi(x)\mu(x)\right| \\
&= \alpha \cdot \left|\sum_{\substack{x: \\ F_m^0(x)=1 \text{ or } F_m^1(x)=1}} \textsc{MOD}_m \circ g(x)\phi(x)\right| \\
&= \alpha \cdot \left|\sum_x (F_m^0(x) - F_m^1(x))\phi(x)\right| \\
&= \alpha \cdot 2^{nk} \cdot \left|\mathbf{E}\left[(F_m^0(\mathbf{x}) - F_m^1(\mathbf{x}))\phi(\mathbf{x})\right]\right| \\
&\le \alpha \cdot 2^{nk} \cdot 2 \cdot \mathit{error}.
\end{aligned}$$

We can get a bound on $\alpha \cdot 2^{nk}$ as follows. Note that the whole input space $(\{0,1\}^n)^k$ is a cylinder intersection and so we can use (3.6) to obtain $\mathbf{E}\left[F_m^0(\mathbf{x})\right] = 1/2^{nk} \cdot |\mathrm{support}(F_m^0)| = 1/m \pm \mathit{error}$. Similarly we have $1/2^{nk} \cdot |\mathrm{support}(F_m^1)| = 1/m \pm \mathit{error}$. Since $\alpha \cdot 2^{nk} = 2^{nk}/(|\mathrm{support}(F_m^0)| + |\mathrm{support}(F_m^1)|)$, we get

$$\alpha \cdot 2^{nk} \le \frac{1}{2/m - 2 \cdot \mathit{error}}.$$

Putting things together we get

$$\frac{1}{\mathrm{disc}_\mu(\textsc{MOD}_m \circ g, \mathcal{C})} \ge \frac{1/m - \mathit{error}}{\mathit{error}} \ge \frac{e^{8n/(m^2 4^k)}}{m} - 1 \ge \frac{2^{11n/(m^2 4^k)}}{m} - 1.$$

Finally, we can apply the discrepancy method to conclude

$$\mathbf{R}_k^\epsilon(\textsc{MOD}_m \circ g) \ge \frac{11n}{m^2 4^k} + \log(1 - 2\epsilon) - \log m - 1. \tag{3.7}$$

Part (b), Case 2: We now consider the case where m does not divide $|S_0| - |S_1|$, but $\gcd(m, |S_0| - |S_1|) > 1$. The lower bound here is obtained via a reduction to the previous case. We assume for the remainder of the proof that $|S_0| - |S_1| > 0$. The case $|S_0| - |S_1| < 0$ can be handled in the same way. Let $1 < d = \gcd(m, |S_0| - |S_1|)$, and let $m = dq$ and $|S_0| - |S_1| = dr$, where q and r are coprime integers. Because m does not divide $|S_0| - |S_1|$, $q \geq 2$. Our strategy is to use a protocol for $\text{MOD}_m \circ g$ in order to construct a protocol for $\text{MOD}_q \circ g'$ for some function g' for which we can apply the lower bound on the randomized communication complexity given in (3.7).

We start by partitioning the set S_0 into sets $S_0', T_1, \ldots, T_d$ with $|S_0'| = |S_1|$ and $|T_1| = \cdots = |T_d| = r$. Let g' be the function whose support is T_1. Note that the support of g' has size r and consists only of inputs of even Hamming weight. So we can apply the lower bound (3.7) to $\text{MOD}_q \circ g'$ since q and r are coprime.

Using a protocol for $\text{MOD}_m \circ g$ (with inputs from $\{0,1\}^{k \times n}$), we will construct a protocol for $\text{MOD}_q \circ g'$ as follows. Let the input for $\text{MOD}_q \circ g'$ be $X \in \{0,1\}^{k \times n'}$ (we'll make the relation between n and n' explicit shortly). Recall that for each $v \in \{0,1\}^k$, n_v denotes the number of occurrences of v as a column of X. First, using Fact 3.2.1 we can compute $\sum_{v \in S_0' \cup S_1} n_v \mod m$ using $k \lceil \log m \rceil$ bits of communication. Again using Fact 3.2.1, for any $\ell \in \{2, \ldots, d\}$, the difference $\sum_{v \in T_\ell} n_v - \sum_{v \in T_1} n_v \mod m$ can also be computed at a cost of $k \lceil \log m \rceil$ bits. As a result, we can compute

$$\sum_{v \in S_0' \cup S_1} n_v + \sum_{\ell=2}^{d} \Big(\sum_{v \in T_\ell} n_v - \sum_{v \in T_1} n_v \Big) \equiv \sum_{v \in S} n_v - d \sum_{v \in T_1} n_v \mod m.$$

Let $s = s(X)$ denote this number. Observe that $\sum_{v \in T_1} n_v \equiv 0 \mod q$ if and only if $d \sum_{v \in T_1} n_v \equiv 0 \mod m$. So $\sum_{v \in T_1} n_v \equiv 0 \mod q$ if and only if $\sum_{v \in S} n_v \equiv s \mod m$. The latter can be determined by running the protocol for $\text{MOD}_m \circ g$ on the input which is obtained from X (viewed as an $k \times n'$ array) by appending $m - s$ columns all of which belong to S.

In short, the protocol for $\text{MOD}_q \circ g'$ on inputs from $(\{0,1\}^{n'})^k$ consists of two steps: First, the players compute s. Then they simulate the protocol for $\text{MOD}_m \circ g$ on the input of size $(\{0,1\}^n)^k$ specified above, where $n = n' + (m - s)$.

Suppose that we can use c bits to compute $\textsc{mod}_m \circ g(X)$ when X is of size $k \times n$. Then the cost of the above protocol is $c + k\lceil \log m \rceil$. Using the fact that $n' = n - (m - s) > n/2$, and (3.7), we conclude

$$c + k\lceil \log m \rceil \geq \frac{5n}{m^2 4^k} + \log(1 - 2\epsilon) - \log m - 1.$$

That is,

$$\mathbf{R}_k^{\epsilon}(\textsc{mod}_m \circ g) \geq \frac{5n}{m^2 4^k} + \log(1 - 2\epsilon) - (k+1)\lceil \log m \rceil - 1.$$

□

Corollary 3.2.4. *If $g : \{0,1\}^k \to \{0,1\}$ has even support size, then $\mathbf{D}_k^{||}(\textsc{mod}_2 \circ g) \leq k$. Otherwise, $\mathbf{R}_k^{\epsilon}(\textsc{mod}_2 \circ g) \geq \frac{n}{4^k} + \log(1 - 2\epsilon) - k - 2$.*

Proof of Lemma 3.2.3

By definition of the cube measure, we have

$$\begin{aligned}
\mathcal{E}_U(\textsc{exp}_m^{a,b} \circ g) &= \mathbf{E}_{\substack{\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_k^0 \\ \mathbf{x}_1^1, \mathbf{x}_2^1, \dots, \mathbf{x}_k^1}} \left[\prod_{u \in \{0,1\}^k} \mathfrak{C}^{u_1 + \cdots + u_k} (\textsc{exp}_m^{a,b} \circ g(\mathbf{x}_1^{u_1}, \dots, \mathbf{x}_k^{u_k})) \right] \\
&= \mathbf{E}_{\substack{\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_k^0 \\ \mathbf{x}_1^1, \mathbf{x}_2^1, \dots, \mathbf{x}_k^1}} \left[\prod_{u \in \{0,1\}^k} \mathfrak{C}^{u_1 + \cdots + u_k} \left(\omega^{a \sum_{j=1}^n g(\mathbf{x}_{1,j}^{u_1}, x_{2,j}^{u_2}, \dots, \mathbf{x}_{k,j}^{u_k}) - ab} \right) \right].
\end{aligned}$$

In the exponent of ω, we can safely ignore ab since exactly half of the terms in the product are conjugated. So without loss of generality we assume $b = 0$.

The first standard step is to write the exponential sum in the cube measure as an n-fold product, interchange the two products, and then finally interchange the n-fold product and the expectation using the independence of the coordinates. This way, the cube measure $\mathcal{E}_U(\textsc{exp}_m^{a,0} \circ g)$ can be written as the n-fold product of the cube measure of $\omega^{a \cdot g(y_1, \dots, y_k)}$. That is,

$$\mathcal{E}_U(\textsc{exp}_m^{a,0} \circ g) = \left(\mathbf{E}_{\substack{\mathbf{y}_1^0, \dots \mathbf{y}_k^0 \\ \mathbf{y}_1^1, \dots, \mathbf{y}_k^1}} \left[\omega^{a \sum_{(u_1, \dots, u_k) \in \{0,1\}^k} (-1)^{u_1 + \cdots + u_k} \cdot g(\mathbf{y}_1^{u_1}, \dots, \mathbf{y}_k^{u_k})} \right] \right)^n,$$

where in the expectation, $\mathbf{y}_j^0$ and $\mathbf{y}_j^1$ are independent and uniformly distributed over $\{0,1\}$.

Observe that for every setting of $\mathbf{y}_j^0$, $\mathbf{y}_j^1$ (for $1 \le j \le k$) where $\mathbf{y}_j^0 = \mathbf{y}_j^1$ for some j, the sum in the exponent is 0, and thus the expression inside the expectation evaluates to 1. This happens with probability $(1 - \frac{1}{2^k})$. Now consider a setting of $\mathbf{y}_j^0$, $\mathbf{y}_j^1$ (for $1 \le j \le k$) where $y_j^0 \ne y_j^1$ for all $1 \le j \le k$. Simply write y_j for y_j^0. Then we can nicely write y_j^u as $y_j \oplus u$, for $u \in \{0,1\}$. Consequently,

$$\begin{aligned}&\sum_{(u_1,\dots,u_k)\in\{0,1\}^k} (-1)^{u_1+\cdots+u_k} g(y_1^{u_1},\dots,y_k^{u_k})\\ &\qquad = \sum_{(u_1,\dots,u_k)\in\{0,1\}^k} (-1)^{u_1+\cdots+u_k} g(y_1 \oplus u_1,\dots,y_k \oplus u_k).\end{aligned}$$

By letting $v_i = y_i \oplus u_i$, the last sum becomes

$$\begin{aligned}&(-1)^{y_1+\cdots+y_k} \sum_{(v_1,\dots,v_k)\in\{0,1\}^k} (-1)^{v_1+\cdots+v_k} g(v_1,\dots,v_k)\\ &\qquad = (-1)^{y_1+\cdots+y_k} \sum_{(v_1,\dots,v_k)\in S} (-1)^{v_1+\cdots+v_k} = (-1)^{y_1+\cdots+y_k}(|S_0| - |S_1|).\end{aligned}$$

This is either $|S_0| - |S_1|$ or $|S_1| - |S_0|$, depending on the parity of $y_1 + y_2 + \cdots + y_k$. Among all tuples $(y_1, y_2, \dots, y_k)$, exactly half of them have even parity. As a result,

$$\begin{aligned}&\mathbf{E}_{\substack{\mathbf{y}_1^0,\dots\mathbf{y}_k^0\\ \mathbf{y}_1^1,\dots,\mathbf{y}_k^1}} \left[\omega^{a\sum_{u\in\{0,1\}^k}(-1)^{u_1+\cdots+u_k}\cdot g(\mathbf{y}_1^{u_1},\dots,\mathbf{y}_k^{u_k})}\right]\\ &= (1 - \frac{1}{2^k}) + \frac{\omega^{a(|S_0|-|S_1|)} + \omega^{a(|S_1|-|S_0|)}}{2^{k+1}}\\ &= (1 - \frac{1}{2^k}) + \frac{\mathrm{Re}(\omega^{a(|S_0|-|S_1|)})}{2^k}\\ &= 1 - \frac{1 - \cos\left(\frac{2\pi}{m}\cdot a(|S_0| - |S_1|)\right)}{2^k}\\ &= 1 - \frac{2\sin^2(a(|S_0| - |S_1|)\pi/m)}{2^k}.\end{aligned}$$

Because m and $|S_0|-|S_1|$ are coprime, $a(|S_0|-|S_1|)\pi/m$ is not a multiple of π, for $1 \leq a \leq m-1$. So $\sin^2(a(|S_0|-|S_1|)\pi/m) \geq \sin^2(\pi/m) \geq 4/m^2$. (Here we use the fact that $\sin(x) \geq 2x/\pi$ for $0 \leq x \leq \pi/2$.) Thus,

$$\mathcal{E}_U(\textsc{exp}_m^{a,0} \circ g) \leq \big(1 - \frac{8}{m^2 2^k}\big)^n \leq \frac{1}{e^{8n/(m^2 2^k)}}.$$

3.3 MAJ $\circ\, g$

It is not difficult to show that the MAJ $\circ\, g$ functions are the hardest among the functions of the form SYM $\circ\, g$.[2] Recall that for a function f, the notation $f^{n'}$ means that f's input is an n'-bit string.

Proposition 3.3.1. *Let $g : \{0,1\}^k \to \{0,1\}$ be a boolean function and $f : \{0,1\}^n \to \{-1,1\}$ be a symmetric function on n variables. For any $\epsilon \geq 0$,*

$$\mathbf{R}_k^{\epsilon'}(f \circ g) \leq \mathbf{R}_k^{\epsilon}(\textsc{maj}^{2n} \circ g) \cdot \lceil \log(n+1) \rceil ,$$

where $\epsilon' = \epsilon \lceil \log(n+1) \rceil$.

Proof. If g is constant, the statement clearly holds. We assume g is not constant in the following. By a binary search strategy we will show how to use a communication protocol for MAJ$^{2n} \circ g$ to compute a function $f \circ g$. Consider a randomized protocol with cost c computing MAJ$^{2n} \circ g$ with error ϵ. We are going to use this protocol to build a protocol that determines the number, w, of ones in $\{g(x_{1,1}, \ldots, x_{k,1}), \ldots, g(x_{1,n}, \ldots, x_{k,n})\}$. Since f is symmetric, $f \circ g(x_1, \ldots, x_k)$ can then be computed from w without communication.

The binary search algorithm for computing w proceeds in stages. During the search we maintain the condition that $w \in [\ell, u]$ for some interval $[\ell, u]$ whose length is halved after each stage. Initially, $\ell = 0$ and $u = n$. Suppose

[2] The fact that MAJ is "hardest" among all symmetric functions is not unique to communication complexity.

that at some stage we have $\ell \leq w \leq u$. In order to determine the values ℓ', u' for the next stage, we will determine whether $w \leq \left\lfloor \frac{\ell+u}{2} \right\rfloor$ or not. Then, if $w \leq \left\lfloor \frac{\ell+u}{2} \right\rfloor$, we set $\ell' = \ell$ and $u' = \left\lfloor \frac{\ell+u}{2} \right\rfloor$, otherwise we set $\ell' = \left\lfloor \frac{\ell+u}{2} \right\rfloor + 1$ and $u' = u$. Clearly, it takes at most $\lceil \log(n+1) \rceil$ stages to arrive at the exact value of w.

Players use a protocol for $\textsc{maj}^{2n} \circ g$ to compare w and $\left\lfloor \frac{\ell+u}{2} \right\rfloor$ as follows. As g is not constant, we can define auxiliary input variables $x'_1, \ldots, x'_k$, all of which are bit strings of length n, such that the number of ones in $g(x'_{1,1}, \ldots, x'_{k,1}), \ldots, g(x'_{1,n}, \ldots, x'_{k,n})$ is exactly $n - \left\lfloor \frac{\ell+u}{2} \right\rfloor$. Now run the protocol for $\textsc{maj}^{2n} \circ g$ on the input $x_1 x'_1, \ldots, x_k x'_k$, where each $x_i x'_i$ is a $2n$-bit string obtained by concatenating x_i and x'_i. Clearly the output of this protocol tells us whether $w \leq \left\lfloor \frac{\ell+u}{2} \right\rfloor$ or not.

We now analyse the error and communication cost of this protocol. Since there are $\lceil \log(n+1) \rceil$ stages, the total cost is at most $\lceil \log(n+1) \rceil$ times the cost for the majority protocol. Also, by a union bound, the protocol makes an error with probability at most $\lceil \log(n+1) \rceil \cdot \epsilon$. □

We can combine Proposition 3.3.1 with our lower bounds for $\textsc{mod}_m \circ g$ functions (Theorem 3.2.2) to obtain a dichotomy for the communication complexity of $\textsc{maj} \circ g$ for every g.

Theorem 3.3.2. *Let $g : \{0,1\}^k \to \{0,1\}$ be a boolean function and $S = \{y \in \{0,1\}^k : g(y) = 1\}$ be its support. Define $S_0 = \{y \in S : y \text{ has even weight}\}$ and $S_1 = \{y \in S : y \text{ has odd weight}\}$. Then the function $\textsc{maj} \circ g$ satisfies the following:*

- *If $|S_0| = |S_1|$, then $\mathbf{D}^{||}_k(\textsc{maj}^n \circ g) \leq k \cdot \lceil \log(n+1) \rceil$.*
- *Otherwise, $\mathbf{R}^{1/3}_k(\textsc{maj}^n \circ g) \geq \Omega\left(\frac{n}{(k \log k)^2 \cdot 4^k \log n \log\log n}\right)$.*

Proof. The case where $|S_0| = |S_1|$ follows from Fact 3.2.1 by setting $m = n+1$. (See Section 3.5 for a reinterpretation of this protocol in terms of computing polynomials.)

Now consider the case where $|S_0| \neq |S_1|$. We use Proposition 3.3.1 to prove a lower bound on the randomized communication complexity of $\textsc{maj}^{2n} \circ g$. Observe that for some large enough constant c, $\prod_{p \leq ck \log k : p \text{ prime}} p > 2^k \geq ||S_0| - |S_1||$ because there are at least k primes in the set $\{2, 3, \ldots, ck \log k\}$. Thus, there exists a prime $m \leq ck \log k$ that does not divide $|S_0| - |S_1|$. Now applying Proposition 3.3.1 with $\epsilon = \frac{1}{3\lceil \log(n+1) \rceil}$, together with Theorem 3.2.2, and also using $m \leq k \log k$ and $k \leq \log n$, we get

$$\begin{aligned}\mathbf{R}_k^{\epsilon}(\textsc{maj}^{2n} \circ g) &\geq \mathbf{R}_k^{1/3}(\textsc{mod}_m \circ g) / \lceil \log(n+1) \rceil \\ &\geq \Omega\left(\frac{n}{(k \log k)^2 \cdot 4^k \log n} - \log \log n\right).\end{aligned}$$

By a standard boosting argument (i.e., repeating a protocol with constant error probability t times and taking the majority vote to reduce the error probability to exponentially small in t) we have

$$\mathbf{R}_k^{1/3}(\textsc{maj}^{2n} \circ g) \geq \Omega\left(\frac{n}{(k \log k)^2 \cdot 4^k \log n \ \log \log n}\right).$$

Finally, since $\textsc{maj}^{2n+1} \circ g$ is at least as hard as $\textsc{maj}^{2n} \circ g$, we obtain the desired result. □

To illustrate the above theorem, we apply it to some natural choices of inner functions g (omitting the details). Recall $\textsc{thr}_t(y_1, \ldots, y_k) = 1$ if $\sum y_i \geq t$ and $\textsc{thr}_t(y_1, \ldots, y_k) = 0$ otherwise. If g is the threshold function $\textsc{thr}_t$ for some $0 < t < n$, then it is simple to show that $\textsc{maj} \circ \textsc{thr}_t$ is always a hard function as long as the number of players is at most $\approx \frac{1}{2} \log n$. The functions $\textsc{maj} \circ \textsc{mod}_m$ exhibit an interesting behaviour: For even m, the function $\textsc{maj} \circ \textsc{mod}_m$ is always hard as long as the number of players is at most $\approx \frac{1}{2} \log n$. By contrast, for odd m, it has an efficient protocol for some values of k, namely when k is an odd multiple of m.

Theorem 3.3.2 can also be used to determine the communication complexity of a class of functions considered by Babai et al. [BGKL03]. For an odd prime k, define the function $\textsc{qcsb}_k : \{0,1\}^k \to \{0,1\}$ by $\textsc{qcsb}_k(y_1,\dots,y_k) = 1$ if and only if $y_1 + \cdots + y_k$ is a quadratic residue modulo k. Recall that $z \in \mathbb{F}_k$ is a quadratic residue if there exists $a \in \mathbb{F}_k$ such that $z = a^2$. The authors of [BGKL03] prove that $\textsc{qcsb}_k$ is not 'compressible', so their protocol does not apply for $\textsc{sym} \circ \textsc{qcsb}_k$. They leave as an open question the problem of finding good upper or lower bounds for the communication complexity of the function $\textsc{maj} \circ \textsc{qcsb}_k$. The following corollary completely determines the hardness of this function for any number of players k, except in the range between $\approx \frac{1}{2}\log n$ and $\log n$.

Corollary 3.3.3. *Let k be an odd prime.*

- *If $k \equiv 1 \mod 4$, then $\mathbf{D}_k^{||}(\textsc{maj} \circ \textsc{qcsb}_k) \leq O(k \log n)$.*
- *If $k \equiv 3 \mod 4$, then $\mathbf{R}_k^{1/3}(\textsc{maj} \circ \textsc{qcsb}_k) \geq \Omega\left(\frac{n}{(k\log k)^2 4^k \log n \log\log n}\right)$.*
- *If $k > 1 + \log n$, then $\mathbf{D}_k^{||}(\textsc{maj} \circ \textsc{qcsb}_k) \leq O(\log^3 n)$.*

Proof. Let S be the support of $\textsc{qcsb}_k$ and define S_0 and S_1 as in Theorem 3.3.2. It is known that when $k \equiv 1 \mod 4$, $z \in \{0,\dots,k-1\}$ is a quadratic residue modulo k if and only if $-z \equiv k - z \mod k$ is a quadratic residue modulo k; see e.g., [Sho09, Theorem 2.21]. As k is odd, z is even if and only if $k-z$ is odd. In other words, the function $(y_1,\dots,y_k) \mapsto (1-y_1,\dots,1-y_k)$ defines a bijection between S_0 and S_1. Thus, $|S_0| = |S_1|$ whenever $k \equiv 1 \mod 4$. Otherwise, if $k \equiv 3 \mod 4$, then the number $|S|$ of quadratic residues modulo k is odd; see e.g., [Sho09, Theorem 2.20]. This implies that $|S_0| \neq |S_1|$. For $k > 1 + \log n$, we can use Theorem 3.1.2. □

3.4 NOR $\circ$ g

In this section, we obtain a simple and perhaps surprising characterization for the k-player randomized communication complexity of $\textsc{nor} \circ g$. We use the following lower bound of Sherstov [She13] for the randomized communication of the disjointness function DISJ $= \textsc{nor} \circ \textsc{and}$:

Theorem 3.4.1 ([She13])**.**

$$\mathbf{R}_k^{1/3}(\mathrm{DISJ}) \geq \Omega\left(\frac{\sqrt{n}}{2^k k}\right).$$

First we observe that the above lower bound for disjointness applies - via a simple reduction - to $\textsc{nor} \circ g$ where g's support size is 1. We complement this with an efficient randomized protocol for $\textsc{nor} \circ g$ when g's support size is more than one.

Theorem 3.4.2. *Let $g : \{0,1\}^k \to \{0,1\}$ be a boolean function and $S = \{y \in \{0,1\}^k \ : \ g(y) = 1\}$ be its support. For some constant $\epsilon < 1/2$,*

- *If $|S| = 1$, $\mathbf{R}_k^{1/3}(\textsc{nor} \circ g) \geq \Omega\left(\frac{\sqrt{n}}{2^k k}\right)$,*
- *Otherwise, $\mathbf{R}_k^{\epsilon}(\textsc{nor} \circ g) \leq O(k)$.*

Proof. For the first part, let $S = \{v\}$ with $v \in \{0,1\}^k$. Then, we can solve $\textsc{nor} \circ \textsc{and}$ on input X by first flipping all the input bits of the rows i for which $v_i = 0$ and then run a protocol for $\textsc{nor} \circ g$. The lower bound then follows from Theorem 3.4.1.

For the upper bound, first assume that $|S|$ is even. In this case, by Corollary 3.2.4, we have a deterministic protocol P for $\textsc{mod}_2 \circ g$ of cost k. We will use this protocol P as a subroutine to compute $\textsc{nor} \circ g$. As before, denote by X the $k \times n$ dimensional matrix representing the input. Denote by X_r a random matrix obtained from X by deleting every column independently with probability $1/2$. The players can agree on X_r without any communication using their public random bits. We output -1 if $P(X_r) = -1$ and output 1 otherwise.

Observe that if $\textsc{nor} \circ g(X) = -1$, then $\textsc{nor} \circ g(X_r) = -1$, and so $\textsc{mod}_2 \circ g(X_r) = -1$. In this case our protocol does not make an error. If

$\textsc{nor} \circ g(X) = 1$, then the bit string $g(X)$ is not the all-zero string and thus the parity of a random subset is uniformly distributed on $\{0, 1\}$, i.e., $\textsc{mod}_2 \circ g(X_r) = 1$ with probability $1/2$. So in this case, the error probability is $1/2$. Repeating this protocol t times would reduce the error probability to $1/2^t$.

Now assume $|S|$ is odd and greater than 1. Divide S into two non-disjoint parts S_1 and S_2 of even size each. Let g_1 be the boolean function with support S_1 and g_2 be the boolean function with support S_2. Observe that $\textsc{nor} \circ g(X) = -1$ if and only if both $\textsc{nor} \circ g_1(X) = -1$ and $\textsc{nor} \circ g_2(X) = -1$. Since we covered the case of even support size, we are done. □

3.5 The Polynomial View of Protocols

Before moving on to the concluding remarks of this chapter, we feel it is worth pointing out that some of the upper bounds we presented have a very natural reinterpretation in terms of protocols computing polynomials. We now explain this using the Fourier analysis notation we set in Chapter 2.

Let's recall the very high level strategy of our protocol for $\textsc{sym} \circ g$ functions. As before, we view the input to the players as a $k \times n$ matrix X and we are interested in the number of columns of X that fire a 1 when g is applied. Let N be this number. Then we write N as a sum (of integer values) such that each term in the sum is known by a player. Furthermore, we use the fact that $N \leq n$ in order to do computation modulo $n + 1$. This way, when the players announce their aggregate parts, each player communicates at most $O(\log n)$ bits.

Going back to our characterization for $\textsc{maj} \circ g$ functions (Theorem 3.3.2), we have an efficient protocol provided that $|S_0| = |S_1|$, where S_i denotes the k-bit strings in the support of g whose Hamming weight has parity i. In fact, in this case the efficient protocol applies to all functions of the form $\textsc{sym} \circ g$.

Let's now reprove this fact. First, consider the Fourier expansion of g:

$$g(x) = \sum_{S \subseteq [k]} \widehat{g}(S) \chi_S(x).$$

Then we can write N as

$$N = \sum_{i=1}^{n} \sum_{S \subseteq [k]} \widehat{g}(S) \chi_S(x^i),$$

where x^i denotes the input variables corresponding to column i of X. The important observation now is that the condition $|S_0| = |S_1|$ implies that $\widehat{g}(S) = 0$ for $S = [k]$ (this is straightforward to check using the definition of $\widehat{g}(S)$). In other words, we have written N as a polynomial of degree at most $k-1$. This in return implies that the value of every monomial of this polynomial is known by some player. Thus, we have written N as a sum, where each term in the sum is known by a player. We still have to be careful, however, since the Fourier coefficients are not integers. We have to make sure that when a player sends his/her aggregate part, the number of bits is not too many. Each Fourier coefficient can be viewed as a fraction with denominator 2^k, which is less than n. Therefore we can multiply the above equality by 2^k and turn each term into an integer. With this, the sum is at most $2^k n \leq n^2$ and we can do computation modulo $2^k n + 1$. So each player again only needs to send $O(\log n)$ bits when announcing their part of the sum.

In the case of $\text{MOD}_m \circ g$ functions, we can exploit the fact that all we need to compute is $N \mod m$, i.e., we can do computation modulo m. Recall that in our characterization for these functions (Theorem 3.2.2), we had an efficient protocol when m divides $|S_0| - |S_1|$. This condition implies that $\widehat{g}([n]) = 0$ when working modulo m, and therefore we can use the same analysis as above.

3.6 Conclusion and Open Problems

The most well-studied communication problems like *generalized inner product* and *disjointness* have a composed structure with an outer function f and an inner function g. Recently, this structure has been exploited by several authors to prove hardness in the NOF model. A natural question that arises is what combination of f and g results in hardness. Almost all previous work focused on fixing the inner function g with a convenient property that allows one to prove hardness for a range of outer functions f. In this work, we address the dual and natural problem of studying families of functions that arise from varying the inner function g. We obtain complete characterizations of hard and easy functions in three of these families: $\textsc{maj} \circ g$, $\textsc{mod}_m \circ g$ and $\textsc{nor} \circ g$. Our characterizations show that hard functions in each of these families, somewhat unexpectedly, exhibit simple and elegant structure. Furthermore, as a corollary, we show that these functions have polynomially related quantum and classical communication complexities.

A key component of our characterization is a new simultaneous protocol for $\textsc{sym} \circ g$ that is efficient for every g, when the number of players is more than $\log n$. This rules out the possibility of composing a symmetric function with any inner function to take us past the $\log n$ barrier for proving strong lower bounds. In particular, Babai et. al., ten years ago, posed an open problem of determining the communication complexity of the function $\textsc{maj} \circ \textsc{qcsb}$, where $\textsc{qcsb}$ is the quadratic residuosity function. Combining our protocol for $\textsc{sym} \circ g$ with our characterization of $\textsc{maj} \circ g$, we are able to completely answer this question. While this may sound as a setback to the hope of going past the $\log n$ barrier, it highlights the importance of considering *block composition* where the inner function acts on a block of columns rather than one column as presented in this chapter. This leads to a natural open question: Is there an inner function g that acts on two columns such that $\textsc{maj} \circ g$ is hard for more than $\log n$ players?

Another natural and important avenue is to extend our dichotomy results to other outside functions. A good starting point is to consider $\textsc{thr}_t \circ g$ and obtain a dichotomy for any t. Our results for $\textsc{nor} \circ g$ and $\textsc{maj} \circ g$ imply characterizations for $\textsc{thr}_1 \circ g$ and $\textsc{thr}_{n/2} \circ g$. These characterizations are quite different. As t goes from 1 to $n/2$, when and how does the characterization change? We note that we ask this question in the randomized communication complexity setting as a characterization of $\textsc{thr}_1 \circ g$ in the deterministic setting is likely to be very challenging due to its connections with very challenging open problems in Ramsey theory. In the next chapter we explore these connections.

CHAPTER 4

Ramsey Theory Applications

This chapter is devoted to the interesting connection of the NOF model with Ramsey theory. In particular, our protocol for functions of the form $\textsc{sym} \circ g$ implies bounds on important Ramsey numbers. Before presenting our results, we first give the history and background information regarding these Ramsey numbers.

The famous Van der Waerden's Theorem, which can be considered as the seed for modern Ramsey theory, states that for all c and k, there exists a large enough $N(c, k)$ such that no matter how you color the integers $[N]$ with c colors, there is always a monochromatic length k arithmetic progression. Erdős and Turán [ET36] conjectured more generally that every large enough subset of $[N]$ must contain a k-term arithmetic progression. More precisely, they conjectured that for all δ and k, there is a large enough $N(\delta, k)$ such that every subset A of $[N]$ of density δ (i.e., of size δN) contains an arithmetic progression of length k. Roth [Rot53] proved the Erdős-Turán conjecture for the special case of $k = 3$ in 1953. It was not until 1975 that a proof for the general case was discovered by Szemerédi [Sze75]. This theorem is considered to be one of the jewels of mathematics.

It is of course a natural question to ask how fast N grows in Van der Waerden's and Szemerédi's Theorems. We will be interested in the following two Ramsey numbers. For a finite Abelian group G, define $c_k(G)$ to be the minimum number of colors we can use to color G so that no k-term arithmetic progression is monochromatic. Also let $r_k(G)$ be the cardinality of the largest subset of G that contains no length k arithmetic progression. Let $N = |G|$. We use the notation $c_k(N)$ and $r_k(N)$ when working over $[N]$ rather than an Abelian group G. Van der Waerden's Theorem and Szemerédi's Theorem are equivalent to showing $c_k(N) = \omega(1)$ and $N/r_k(N) = \omega(1)$ respectively. Obviously we have $N/r_k(N) \leq c_k(N)$. Obtaining good quantitative bounds on $c_k(N)$ and $r_k(N)$ is one of the major challenges in combinatorics.

The best known bounds for $r_k(N)$ are as follows (we write the bounds in terms of $N/r_k(N)$ as the interest is in this fraction). Sanders [San11] recently showed that

$$\frac{N}{r_3(N)} \geq \Omega\left(\frac{\log N}{(\log\log N)^5}\right),$$

and the best upper bound comes from Behrend's construction of a set without a 3-term progression [Beh46] (in [Elk11], Elkin obtains a minor improvement):

$$\frac{N}{r_3(N)} \leq O\left(2^{\sqrt{8\log N}}(\log N)^{1/4}\right).$$

For general k, the best bounds are

$$\frac{N}{r_k(N)} \geq \Omega\left((\log\log N)^{t_k}\right),$$

(t_k is a positive constant that depends only on k) due to Gowers [Gow01], and

$$\frac{N}{r_k(N)} \leq C \cdot 2^{O((\log N)^{1/\log k}+\log\log N)},$$

for a constant C, due to O'Bryant [O'B11].

It has been observed several times that the above *lower bound* results are in fact easier and cleaner to handle when working over $\mathbb{F}_p^n$ as one can exploit linear algebraic tools. As Green notes [Gre05], another motivation to work in the finite field setting is inspired by Bourgain's work [Bou99], which can be interpreted to show how to convert results obtained in the finite fields setting to arbitrary groups.

Very recently Bateman and Katz [BK12], in a breakthrough work, show that

$$\frac{N}{r_3(\mathbb{F}_3^n)} \geq \Omega\left((\log N)^{1+\epsilon}\right).$$

Non-trivial upper bounds are harder to come by in the finite field setting. Behrend's construction does not work over $\mathbb{F}_p^n$. The best upper bound we have for $N/r_3(\mathbb{F}_3^n)$ is much weaker and is about $N^{0.28}$, which comes from design theory; see e.g., [Gre05, Section 4]. It is reasonable to expect, both in the setting of $[N]$ and $\mathbb{F}_p^n$, that the lower bounds are far from being tight. For instance, Green [Gre05] conjectures that

$$\frac{N}{r_3(\mathbb{F}_3^n)} \geq N^{\delta},$$

for an absolute constant δ.

A well known generalization of Van der Waerden's Theorem and Szemerédi's Theorem is called the *multidimensional version* or the *corners* problem. In the k dimensional setting, our space is G^k rather than G, and the structure we are looking for is a *corner* rather than an arithmetic progression. A k dimensional *corner* is a set of $k+1$ points in G^k of the form

$$(x_1, x_2, \ldots, x_k), (x_1+\lambda, x_2, \ldots, x_k), (x_1, x_2+\lambda, \ldots, x_k), \ldots, (x_1, x_2, \ldots, x_k+\lambda),$$

for some non-zero $\lambda \in G$.

Let $c_k^{\angle}(G)$ be the minimum number of colors we can use to color G^k so that no k-dimensional corner is monochromatic. Also let $r_k^{\angle}(G)$ be the cardinality of the largest subset of G^k that contains no k-dimensional corner. As before, we use the notation $c_k^{\angle}(N)$ and $r_k^{\angle}(N)$ when working over $[N]^k$. To eliminate any confusion, we note explicitly that a k dimensional corner consists of $k+1$ points and therefore there is a correspondence between a k dimensional corner and a $(k+1)$-term arithmetic progression. In particular, via a standard reduction, we have $r_{k+1}(G) \leq r_k^{\angle}(G)/N^{k-1}$. For example, for $k = 2$, let A be a subset of G that does not contain a 3-term arithmetic progression. Then let $A' \subset G^2$ be the set of pairs (x, y) such that $x + 2y$ is in A. It is easy to see that if A' contains a 2 dimensional corner, then A contains a 3-term arithmetic progression.

In a far reaching extension of Szemerédi's Theorem, Gowers [Gow07] obtains an explicit lower bound on $N^k/r_k^{\angle}(N)$, but the bound is of Ackerman type and we do not state it here.[1] This bound remains best known for arbitrary fixed k. In the two dimensional case (which can be thought of as the generalization of Roth's Theorem [Rot53], i.e., Szemerédi's Theorem for $k = 3$.), Shkredov [Shk06b, Shk06a] obtains the bound

$$\frac{N^2}{r_2^{\angle}(N)} \geq (\log\log N)^{\epsilon}.$$

The best upper bound comes from Behrend's construction via a reduction. In the finite field setting, a better lower bound is obtained by Lacey and McClain [LM07]:

$$\frac{N^2}{r_2^{\angle}(\mathbb{F}_p^n)} \geq \frac{\log\log N}{\log\log\log N}.$$

To the best of our knowledge, no non-trivial upper bound on $N^k/r_k^{\angle}(\mathbb{F}_p^n)$ is mentioned in the literature.

There is an interesting connection between the coloring number for corners and multiparty communication complexity. Define the EXACT_N func-

[1] The bound Gowers obtains is similar to what Szemerédi obtains in the setting of progressions. This is because Gowers generalizes Szemerédi's Regularity Lemma to hypergraphs and this step is responsible for the horrendous bound.

tion to be equal to -1 if and only if $x_1 + \cdots + x_k = N$, where x_i are the inputs, each an n-bit integer in $[N]$. Chandra Furst and Lipton [CFL83] show that the $k+1$ party deterministic communication complexity of EXACT_N is essentially equal to $\log c_k^{\angle}(N)$. The known lower bounds on $N^k/r_k^{\angle}(N)$ [FK78, Gow07] imply superconstant lower bounds on $c_k^{\angle}(N)$ and using this, they conclude that the deterministic k-party communication complexity of EXACT_N is superconstant for all constant k. Furthermore, they convert the known upper bound on $N/r_3(N)$ due to Behrend into an upper bound on $c_2^{\angle}(N)$ and obtain a surprising *non-explicit* protocol of cost $O(\sqrt{n})$ for the EXACT_N function for 3 players. Although this and other kinds of communication complexity bounds have been proven using Ramsey theory (e.g., [CFL83, Pud03, Tes03, CKK$^+$07, BGG06]), no bounds on Ramsey numbers have been proven via communication complexity bounds before.

For an Abelian group G, define $\mathrm{EVAL}_G : G^k \to \{\pm 1\}$ to be equal to -1 if and only if $x_1 + \cdots + x_k = 0$, where the $x_i \in G$ are the inputs, and 0 denotes the identity element of G. As observed in [BGG06], the proof of [CFL83] also shows that the $k+1$ party communication complexity of EVAL_G is essentially equal to $\log c_k^{\angle}(G)$. Here, we are interested in upper bounds on $c_k^{\angle}(N)$ and $c_k^{\angle}(\mathbb{F}_2^n)$, which in return give upper bounds for $N^k/r_k^{\angle}(N)$ and $N^k/r_k^{\angle}(\mathbb{F}_2^n)$.

In this chapter we will show the following.

- Section 4.1.1: We observe that $\mathrm{EVAL}_{\mathbb{F}_2^n}$ is the same function as $\textsc{nor} \circ \textsc{xor}$. Using our deterministic protocol for functions of the form $\textsc{sym} \circ g$ from the previous chapter, we get the upper bound $N^k/r_k^{\angle}(\mathbb{F}_2^n) \leq c_k^{\angle}(\mathbb{F}_2^n) \leq O(N^{1/2^{k-1}} \log^{k+2} N)$ (Corollary 4.1.3). As far as we are aware, this result gives the first non-trivial upper bound and we suspect that it is essentially tight. For $k \geq \log\log N$, our bounds imply the following strong bounds: $N^k/r_k^{\angle}(\mathbb{F}_2^n) \leq c_k^{\angle}(\mathbb{F}_2^n) \leq O((\log N)^{3+\log\log N})$. The coloring induced by the protocol does not give an explicit large set without a corner. Inspired by our protocol, we provide such an explicit set with a simple description (Theorem 4.1.4). Our results can be considered as the first application of communication complexity to Ramsey theory.

- Section 4.1.2: Recall that Behrend [Beh46] obtained the upper bound $N/r_3(N) \leq O(2^{\sqrt{8\log N}}(\log N)^{1/4})$. This result does not imply any bounds for $c_3(N)$. We observe that Behrend's idea can be used to give an explicit coloring of $[N]$ and obtain the upper bound $c_3(N) \leq 2^{\sqrt{8\log N}}(2\log N)^{1/2}$. This upper bound, via a standard reduction, also gives an upper bound for $c_2^{\angle}(N)$. Using this, we present an *explicit* protocol of cost $O(\sqrt{n})$ for the EXACT_N function for 3 players. As mentioned before, [CFL83] gets the same upper bound with a non-explicit protocol using a probabilistic argument.

In the following section, we will first state formally the relationship between the communication complexities of EXACT_N and EVAL_G with $c_k^{\angle}(N)$ and $c_k^{\angle}(G)$. Afterwards, we will present our results as outlined above.

4.1 Upper bounds on coloring numbers for corners

First, we state a result by Chandra, Furst and Lipton that connects multiparty communication complexity with coloring numbers for corners:

Lemma 4.1.1 ([CFL83]).

$$\log\left(c_k^{\angle}\left(\left\lceil\frac{N-1}{k}\right\rceil\right)\right) \leq \mathbf{D}_{k+1}(\mathrm{EXACT}_N) \leq k + \log(c_k^{\angle}(N)).$$

As observed in [BGG06], such a connection, with essentially the same proof, also holds for the EVAL_G function. We provide a proof for completeness.

Lemma 4.1.2.

$$\log(c_k^{\angle}(G)) \leq \mathbf{D}_{k+1}(\mathrm{EVAL}_G) \leq k + \log(c_k^{\angle}(G)).$$

Proof. Upper bound: Fix a coloring of G^k with $c_k^{\angle}(G)$ colors so that there is no monochromatic corner. Denote the players' input by $x_1, \ldots, x_{k+1}$. For $i = 1, \ldots, k$, define $x_i' = -\sum_{j \neq i} x_j$, where the addition represents the operation of the group. Observe that $\mathrm{EVAL}_G(x_1, \ldots, x_{k+1}) = 1$ if and only if $x_i = x_i'$ for all $i = 1, \ldots, k$. Now, for $i = 1, \ldots, k$, Player i computes the color of $(x_1, \ldots, x_i', \ldots, x_k)$. Player $k+1$ computes the color of $(x_1, \ldots, x_k)$. One player announces her color and the rest compare it with their color. If the colors are the same, they accept. Otherwise they reject. If $\mathrm{EVAL}_G(x_1, \ldots, x_{k+1}) = 1$ then obviously all the colors are the same. If $\mathrm{EVAL}_G(x_1, \ldots, x_{k+1}) = 0$, then setting $z = -\sum_{j=1}^{n} x_i$, we have $x_i' = x_i + z$ for all $i \in \{1, \ldots, n\}$. Thus the $k+1$ points that the players compute the colors for form a corner. By assumption, this corner is not monochromatic and the correctness of the protocols follows. The number of bits communicated is clearly as advertised.

Lower bound: Let c be the cost of an optimal $(k+1)$-party protocol for EVAL_G. We will color G^k with 2^c colors so that no corner is monochromatic. The coloring is as follows. We know the protocol partitions the input space G^{k+1} into at most 2^c cylinder intersections, each of which has the same value with respect to EVAL_G's output. We color a point $(x_1, \ldots, x_k)$ in G^k with the label of the cylinder intersection that contains $(x_1, \ldots, x_k, -(x_1 + \cdots + x_k))$. To show that this is indeed a legal coloring, suppose there is a corner which is monochromatic:

$$(x_1, x_2, \ldots, x_k),$$

$$(x_1 + \lambda, x_2, \ldots, x_k),$$

$$(x_1, x_2 + \lambda, \ldots, x_k),$$

$$\vdots$$

$$(x_1, x_2, \ldots, x_k + \lambda).$$

These are colored respectively with the colors of

$$(x_1, x_2, \ldots, x_k, -(x_1 + \cdots + x_k) - \lambda + \lambda),$$

$$(x_1 + \lambda, x_2, \ldots, x_k, -(x_1 + \cdots + x_k) - \lambda),$$

$$(x_1, x_2 + \lambda, \ldots, x_k, -(x_1 + \cdots + x_k) - \lambda),$$

$$\vdots$$

$$(x_1, x_2, \ldots, x_k + \lambda, -(x_1 + \cdots + x_k) - \lambda).$$

This is a star, contained in a cylinder intersection with value 1, and its center is $(x_1, x_2, \ldots, x_k, -(x_1 + \cdots + x_k) - \lambda)$. Hence the center must also be in the same cylinder intersection and must have the value 1. But this is not true since the sum of the coordinates is λ which is non-zero by definition. □

4.1.1 Finite field setting

In Section 3.1, we presented a protocol for functions of the form SYM $\circ\, g$. Observe that $\text{EVAL}_{\mathbb{F}_2^n}$ can be written as NOR$\circ$XOR and therefore the protocol described in Theorem 3.1.2 works for $\text{EVAL}_{\mathbb{F}_2^n}$. Using Lemma 4.1.2, we get an upper bound on $c_k^{\angle}(\mathbb{F}_2^n)$, and this in return gives a lower bound on $r_k^{\angle}(\mathbb{F}_2^n)$. The bounds below are obtained using the remark made right after the proof of Theorem 3.1.2.

Corollary 4.1.3. *Let $N = 2^n$. For any k,*

$$c_k^{\angle}(\mathbb{F}_2^n) \leq 16N^{1/2^{k-1}} \log^{k+2} N.$$

In particular, when $k > \log n$,

$$c_k^{\angle}(\mathbb{F}_2^n) \leq 32(\log N)^{3+\log\log N}.$$

The coloring above does not give an explicit large set that does not contain a corner. A color class defined by a communication protocol as described in the proof of Lemma 4.1.2 corresponds to a set of inputs for NOR $\circ$ XOR that evaluate to -1 for which the communication transcript is the same. We now consider the protocol of Theorem 3.1.2 and try to build such a large set of inputs. To ensure that the different inputs lead to the same communication in the first step of the protocol, we fix the number of times the all-zero column occurs to 0. As the second step of the protocol only depends on the number of times each column appears, fixing these numbers defines a set without corner. This set does not however have the desired size. Below, we show that it is sufficient to fix the number of columns with Hamming weight i for all $i \in \{0, 1 \ldots, k\}$ and this yields a much larger corner-free set.

For $X = (x_1, \ldots, x_k) \in (\mathbb{F}_2^n)^k$, we denote by $n_i(X)$ the number of columns of X with Hamming weight i, i.e., $n_i(X) = |\{j \in \{1, \ldots, n\} : \sum_{\ell=1}^k X_{\ell,j} = i\}|$, where the sum $\sum_{\ell=1}^k X_{\ell,j}$ should be understood as an operation over the integers. Let $N_i = \left\lfloor \frac{\binom{k}{i}}{2^k-1} n \right\rfloor$ for $i \in \{1, \ldots, k-1\}$ and $N_k = n - \sum_{i=1}^{k-1} N_i$ and

$$S^k = \left\{X \in (\mathbb{F}_2^n)^k : \forall i \in \{1, \ldots, k\}, n_i(X) = N_i\right\}. \tag{4.1}$$

Observe that this implies that for all $X \in S^k$, $n_0(X) = 0$ and $n_i(X) \geq 1$ for $i \in \{1, \ldots, k\}$.

Theorem 4.1.4. *Let $n \geq 2$ and $2 \leq k \leq \lceil \log n \rceil$, and let $N = 2^n$. The set S^k defined above does not contain a corner, and*

$$|S^k| \geq C_k \frac{N^k}{N^{-\log(1-2^{-k})} \log^{k/2} N}$$

where C_k only depends on k. For $k = \lceil \log n \rceil$,

$$|S^k| \geq \frac{N^k}{(\log N)^{C \log \log N}}.$$

for some constant $C > 0$.

Proof. We first prove that S^k does not contain a corner. This part of the proof does not make use of the particular values chosen for N_i, in fact we prove that S^k as defined in (4.1) does not contain a corner for any choice of $N_1, \ldots, N_k$ satisfying $\sum_i N_i = n$. Recall that $n_0(X) = 0$ for all $X \in S^k$, and this is crucial for the argument. Assume that there exists $X \in S^k$ and non-zero $\lambda \in \mathbb{F}_2^n$ such that for all $\ell \in \{1, \ldots, k\}$, $X + \lambda^\ell \in S^k$ where $\lambda^\ell \in (\mathbb{F}_2^n)^k$ is zero except for the ℓ-th row where it is equal to λ. Consider the columns of X corresponding to indices j such that $\lambda_j = 1$. Let t denote the minimum Hamming weight among these columns. Note that $t > 0$. By the minimality of t, the columns of X with Hamming weight $t-1$ remain intact in $X + \lambda^\ell$ for all $\ell \in \{1, \ldots, k\}$. So $n_{t-1}(X + \lambda^\ell) \geq n_{t-1}(X) = N_{t-1}$ for every ℓ. On the other hand, observe that by the choice of t, there is some ℓ' such that $n_{t-1}(X + \lambda^{\ell'}) > n_{t-1}(X)$. This is a contradiction.

We now move on to estimate the size of S^k. The values of N_i were picked so that S^k is as large as possible while keeping the size estimation simple. We will prove generally that for any $k \geq 2$,

$$|S^k| \geq (2^k - 1)^n \cdot \frac{1}{n^{k/2}} \cdot \frac{1}{2} \frac{e^{-2k-2k^2}}{\sqrt{2\pi}^{k-1}(1+k)}.$$

Then, to obtain the advertised bound, we write

$$(2^k - 1)^n \geq (2^n)^k (1 - 2^{-k})^n = \frac{N^k}{N^{-\log(1-2^{-k})}},$$

and we define $C_k = \frac{1}{2} \frac{e^{-2k-2k^2}}{\sqrt{2\pi}^{k-1}(1+k)}$. To obtain the bound for $k = \lceil \log n \rceil$, we observe that $N^{-\log(1-2^{-k})} \leq (1 - 1/n)^{-n} \leq 4$.

We use Stirling's approximation: for all $n \geq 1$

$$\left(\frac{n}{e}\right)^n \sqrt{2\pi n} \leq n! \leq e \cdot \left(\frac{n}{e}\right)^n \sqrt{2\pi n}.$$

We define the reals α_i such that $N_i = \frac{\alpha_i}{2^k - 1} n$. Note that $\alpha_i \leq \binom{k}{i}$ for all

$i \in \{1, \dots, k-1\}$ and $N_k \leq \frac{1}{2^k-1}n + k$ so that $\alpha_k \leq 1 + \frac{2^k-1}{n}k$.

$$\begin{aligned}
|S^k| &= \binom{n}{N_1 \quad N_2 \quad \cdots \quad N_k} \cdot \binom{k}{1}^{N_1} \cdots \binom{k}{k}^{N_k} \\
&\geq \frac{\left(\frac{n}{e}\right)^n \sqrt{2\pi n}}{e^k \left(N_1 e^{-1}\right)^{N_1} \cdots \left(N_k e^{-1}\right)^{N_k} \sqrt{(2\pi)^k N_1 \cdots N_k}} \\
&\cdot \binom{k}{1}^{N_1} \cdots \binom{k}{k}^{N_k} \\
&\geq \frac{\left(\frac{n}{e}\right)^n \sqrt{2\pi n}}{e^k \left(\frac{\alpha_1}{2^k-1} n e^{-1}\right)^{N_1} \cdots \left(\frac{\alpha_k}{2^k-1} n e^{-1}\right)^{N_k} \sqrt{(2\pi)^k N_1 \dots N_k}} \\
&\cdot \binom{k}{1}^{N_1} \cdots \binom{k}{k}^{N_k} \\
&\geq \frac{(2^k-1)^n \sqrt{2\pi n}}{e^k \alpha_k^{N_k} \sqrt{(2\pi)^k N_1 \dots N_k}}.
\end{aligned}$$

Observe that

$$\alpha_k^{N_k} \leq \left(1 + \frac{(2^k-1)k}{n}\right)^{\frac{n}{2^k-1}+k} \leq e^{k+2k^2},$$

where we used the fact that $(2^k-1)/n \leq 2$ as $k \leq \lceil \log n \rceil$. Moreover,

$$N_1 \cdots N_k \leq \frac{\binom{k}{1} \cdots \binom{k-1}{k}(1+k)}{(2^k-1)^k} n^k \leq \frac{2^{k^2} \cdot (1+k) n^k}{(2^k-1)^k},$$

which gives the desired bound. □

4.1.2 An explicit $O(\sqrt{n})$-protocol for EXACT_N for 3 players

Using an elegant argument, [Beh46] shows that for any N sufficiently large, there is a subset of $[N] = \{1, 2, \dots, N\}$ of size at least

$$\Omega\left(\frac{N}{2^{\sqrt{8\log N}}(\log N)^{1/4}}\right) \tag{4.2}$$

that does not contain any (nontrivial) 3-term arithmetic progressions. We observe that Behrend's argument can actually be made to give an *explicit* coloring of $[N]$ using at most

$$2^{\sqrt{8\log N}}(2\log N)^{1/2} \tag{4.3}$$

colors such that there is no monochromatic 3-term arithmetic progression. Furthermore, Behrend's argument also shows that in our coloring there is a color class of the size stated in (4.2). This coloring will be used to obtain an explicit protocol for EXACT_N for three players.

Note that Behrend's result has been used in [CFL83] to show the existence of a $O(\sqrt{n})$-cost protocol for the EXACT_N function for three players. The high-level outline is as follows. By mapping $(x, y) \in [N]\times[N]$ to $x+2y \in [N]$, the large set which exists by Behrend's argument can be used to obtain a large subset of $[N] \times [N]$ that does not contain a corner. Then a probabilistic argument shows that with high probability, a sufficiently large number of translations of this subset will cover the whole space $[N] \times [N]$. Each of these translations is colored by a distinct color, and this shows the existence of a protocol of cost $O(\sqrt{n})$.

Our observation shows that we can bypass the probabilistic step above. Moreover, the explicit protocol we obtain might give insight into how three players can cooperatively offer a much more efficient protocol than two players.

For completeness, we start by sketching Behrend's construction. Consider a subset T of points $\mathbf{x} = (x_0, \ldots, x_{d-1}) \in \mathbb{Z}^d$ that lie on a sphere $x_0^2 + x_1^2 + \cdots + x_{d-1}^2 = t$. Observe that T cannot contain a 3-term arithmetic progression, i.e., we cannot have $\mathbf{x} + \mathbf{y} = 2\mathbf{z}$ for distinct $\mathbf{x}, \mathbf{y}, \mathbf{z} \in T$. By imposing a constraint $x_i < m$ for all $i \in \{0, \ldots, d-1\}$, one can then choose t appropriately so as to obtain a large set $T \subseteq \{0, 1, \ldots, m-1\}^d$ without 3-term arithmetic progressions. The final step is to map this set T to positive integers. This is done by seeing the vector $\mathbf{x}$ as the digits of an integer written base $2m$. More precisely, $\mathbf{x}$ is associated to the positive integer $x = 1 + x_0 + x_1(2m) + \cdots + x_{d-1}(2m)^{d-1}$. It is then simple to prove that a 3-term arithmetic progression for the integers x, y, z directly corresponds to a 3-term arithmetic progression for the vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$. One can then obtain the lower bound in (4.2) on the size of the set by choosing m and d appropriately. Now we show how to use these ideas to obtain a *coloring* of $[N]$ with no monochromatic arithmetic progression of length 3.

As described above, m and d are some parameters to be determined later. For each $x \in [N]$ we write $x - 1$ in base $(2m)$ as

$$x - 1 = x_0 + x_1(2m) + x_2(2m)^2 + \ldots + x_{d-1}(2m)^{d-1}, \tag{4.4}$$

where $0 \leq x_i < 2m$ for $0 \leq i < d$. Then, our coloring for $[N]$ is as follows. Define $S(x)$ to be the subset of indices i such that $x_i < m$, and define

$$t(x) = \Big(\sum_{j \in S(x)} x_j^2 \Big) + \Big(\sum_{j \notin S(x)} (x_j - m)^2 \Big).$$

Now we color x with the pair $(S(x), t(x))$.

Lemma 4.1.5. *In the above coloring of $[N]$ there is no monochromatic 3-term arithmetic progression. Moreover, for $d = \sqrt{2 \log N}$ and $m = 2^{\sqrt{\frac{1}{2}\log N} - 1}$, the total number of colors needed is at most $2^{\sqrt{8 \log N}}(2 \log N)^{1/2}$.*

Proof. The fact that there is no monochromatic 3-term arithmetic progression can be seen as follows. Suppose that x, y, z have the same color, and that $x + y = 2z$. First, since x, y, z have the same color, we have

$$S(x) = S(y) = S(z).$$

From this and the hypothesis that $x + y = 2z$, we can prove by induction on i that $x_i + y_i = 2z_i$, for all $0 \leq i < d$. From this it follows that $x_i^2 + y_i^2 \geq 2z_i^2$

and $(x_i - m)^2 + (y_i - m)^2 \geq 2(z_i - m)^2$, and equality holds if and only if $x_i = y_i = z_i$. As a result, $t(x) + t(y) \geq 2t(z)$, and equality holds if and only if $x = y = z$. Now because x, y, z have the same color,

$$t(x) = t(y) = t(z),$$

so equality does indeed hold. It follows that $x = y = z$. This shows that there is no monochromatic (nontrivial) 3-term arithmetic progression.

Now the total number of colors is at most $2^d(d(m-1)^2 + 1)$, because there are 2^d possible sets $S(x)$, and $t(x) \leq d(m-1)^2$. So for the values of d and m given in the lemma, it is easy to verify that the total number of colors needed is as stated in (4.3).

□

Note that the above setting of the parameters is optimal for the total number of colors needed in our coloring. This is because we need $(2m)^d \geq N$ in order to write $N - 1$ as in (4.4).

By a standard argument, i.e., mapping each $(x, y) \in [N]^2$ to $x + 2y$, we can exhibit an explicit coloring of $[N]^2$ without a monochromatic corner. Here we will use this to describe an explicit $O(\sqrt{n})$-protocol for the 3 player communication problem EXACT_{2^n}. Recall that in this problem there are three players: Alice, Bob, and Charlie, with inputs x, y, z $(0 \leq x, y, z \leq 2^n)$ respectively on their foreheads. The players want to determine whether $x + y + z = 2^n$.

Our protocol is obtained by combining the above explicit coloring and the argument from [CFL83] (that shows how to obtain a protocol from a coloring, as in the proof of Lemma 4.1.2). It is based on the following observation. Let

$$x' = 2^n + y - z, \quad y' = 2^{n+1} - x - 2z, \quad z' = x + 2y.$$

Then $y' + z' = 2x'$, i.e., y', x', z' form a 3-term arithmetic progression. Moreover, $x + y + z = 2^n$ if and only if $x' = y' = z'$. In addition, x', y', and

z' can be computed by Alice, Bob, and Charlie, respectively, without any communication.

The Protocol:

- Alice sends the color of x';
- Bob and Charlie send one bit each indicating whether y' and z' have the same color as x';
- The players conclude that $x + y + z = 2^n$ if and only if x', y', z' have the same color.

Here the colors of x', y', z' are determined as above, but note that $x', y', z' \in [2N]$. So we set $d = \sqrt{2(n+1)}$ and $m = 2^{\sqrt{(n+1)/2}}$. The cost of the protocol is at most

$$2 + \lceil \log(2^d dm^2) \rceil \leq 2\sqrt{2(n+1)} + \frac{1}{2}\log(n+1) + 4.$$

4.2 Open Problems

There are several interesting open problems related to the topics studied in this chapter. We state a few of them here.

Getting good bounds on $c_k^{\angle}(G)$ and $r_k^{\angle}(G)$ is a major challenge. Can one make progress on this using the connection with communication complexity? Observe that EVAL_G has $O(1)$ complexity in the randomized model as it reduces to the 2 player EQUALITY function, which is the canonical example of a function with a very efficient randomized protocol. Hence, to show a good lower bound on $\mathbf{D}_k(\mathrm{EVAL}_G)$, one needs to use a lower bound technique that does not apply to randomized protocols. So far, the only strong lower bound technique we have in the NOF model is the discrepancy method (and its extension called the generalized discrepancy method) which proves lower bounds for the randomized model. Can we develop new lower bound techniques that work only for the deterministic model? It is a major open problem in the NOF model to exhibit an explicit function which is easy in the randomized model but hard in the deterministic model, even for 3 players.[2] The EVAL_G function is of course a natural candidate.

Chandra, Furst and Lipton [CFL83] showed that the EXACT_N function for 3 players has an $O(\sqrt{n})$-cost protocol. Our protocol for $\mathrm{EVAL}_{\mathbb{F}_2^n}$ has cost $\Theta(n)$ when k is a constant, but it has cost $O(\log^2 n)$ when $k \geq \log n$. Does EXACT_N have an efficient protocol for $\log n$ many players? Is it possible to get a $o(n)$ cost protocol for $\mathrm{EVAL}_{\mathbb{F}_2^n}$ for 3 players? We suspect that the answer to the latter question is no.

Our protocol for $\mathrm{EVAL}_{\mathbb{F}_2^n}$ does not work for $\mathrm{EVAL}_{\mathbb{F}_3^n}$. Can one get a similar bound for $\mathrm{EVAL}_{\mathbb{F}_3^n}$, and for $\mathrm{EVAL}_{\mathbb{F}_p^n}$ in general? The complexity of EVAL_G for other G is also interesting to study.

[2] It is proved in [BDPW10], by a clever counting argument, that such functions exist.

CHAPTER 5

Spectral Norm and 2-party Communication Complexity

5.1 Introduction

This chapter marks the end of our discussion of the NOF model and we move on to the second part of our thesis which is about Fourier analysis of symmetric boolean functions. Here we will give a combinatorial characterization for the spectral norm of all symmetric functions and discuss its applications to decision tree complexity and communication complexity of these functions. We will conclude with an intriguing open question (an extension of our main result) and mention some of its interesting consequences.

It is important to review the basics of Fourier analysis of boolean functions before moving on (Section 2.3).

5.1.1 Spectral Norm of Boolean Functions

As $\sum_S \widehat{f}(S)^2 = 1$ for a boolean function f, it is often useful to view the squares of the Fourier coefficients as a probability distribution over the subsets $S \subseteq [n]$. The spectral norm corresponds to the Rényi entropy of order $1/2$ of the squares of the Fourier coefficients, $H_{1/2}[\widehat{f}^2] = 2\log\left(\sum_S |\widehat{f}(S)|\right) = 2\log\|\widehat{f}\|_1$. It provides useful upper and lower bounds on the *complexity* of a function in settings such as learning theory, circuit complexity, and communication complexity. It is particularly useful in settings where PARITY is considered a function of low complexity. We list some of the applications below.

In the setting of learning theory, the spectral norm is used in conjunction with the *Kushilevitz-Mansour Algorithm* [KM91] (see [KV94] for an introduction to computational learning theory). This algorithm, using membership queries, learns efficiently a concept class $\mathcal{F}$ where the Fourier spectrum of every function in $\mathcal{F}$ is concentrated on a small set of characters (This set can be different for different functions.). Kushilevitz and Mansour observe that an upper bound on the spectral norm implies such a concentration, and obtain:

> If $\mathcal{F} = \{f : \{0,1\}^n \to \{-1,1\} \mid \|\widehat{f}\|_1 \leq s\}$, then $\mathcal{F}$ is learnable with membership queries in time $\mathrm{poly}(n, s, 1/\epsilon)$.

Using the above result, they show that functions computable by small size parity decision trees (see the end of Section 2.1 for the definition of parity decision tree size) are efficiently learnable with membership queries. This is done by observing that a function computable by a size s parity decision tree satisfies $\|\widehat{f}\|_1 \leq s$. This inequality is also interesting since it provides a lower bound in terms of the spectral norm on the size of any parity decision tree computing f.

Threshold circuits (i.e., circuits composed of threshold gates) constitute an important model of computation (in part due to their resemblance to neural networks), and they have been studied extensively. A classical result of Bruck and Smolensky [BS92] states that a function with small spectral norm can be represented as the sign of a polynomial with few monomials. This in turn implies that functions with small spectral norm can be computed by depth 2 threshold circuits of small size. The result of Bruck and Smolensky has found other interesting applications (see for example [SB91, GHR92, Gro99, OS08]).

We now turn our attention to communication complexity. One of the most famous conjectures in communication complexity is the Log Rank Conjecture which states that the deterministic communication complexity of a function $F : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ is upper bounded by $\log^c \operatorname{rank} M_F$ where the matrix M_F is defined as $M_F[x,y] = F(x,y)$. Grolmusz [Gro97] makes a similar intriguing conjecture for the randomized communication complexity:

> There is a constant c such that the randomized communication complexity of $F : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ is upper bounded by $\log^c \|\widehat{F}\|_1$.

In the same paper, Grolmusz is able to prove a much weaker upper bound of $O(\|\widehat{F}\|_1^2 \delta(n))$ with $\exp(-c\delta(n))$ probability of error. Even this weaker result has interesting applications in circuit complexity and decision tree complexity (see [Gro97] for more details).

Another major open problem in communication complexity is whether the classical and quantum communication complexity of total boolean functions $f : \mathcal{X} \times \mathcal{Y} \to \{-1,1\}$ (i.e., functions defined on all of $\mathcal{X} \times \mathcal{Y}$) are polynomially related. It is conjectured that this is so and research has been focused on establishing it for natural large families of functions. In an important paper [Raz03] Razborov showed that the conjecture is true in the 2 party setting for functions of the form $\textsc{sym} \circ \textsc{and}$ where $\textsc{sym}$ denotes a symmetric function.

Shi and Zhang [SZ09a] verified the conjecture for SYM $\circ$ XOR, again in the 2 party setting. The next big targets are $f \circ$ AND and $f \circ$ XOR for general f, but handling arbitrary f seems quite difficult.

A variant of the spectral norm, *the approximate spectral norm*, is intimately related to the communication complexity of "xor functions". The ϵ-approximate spectral norm of f, denoted $\|\widehat{f}\|_{1,\epsilon}$, is the smallest spectral norm of a function $\phi : \{0,1\}^n \to \mathbb{R}$ such that $\|f - \phi\|_\infty \leq \epsilon$. It is known (see for example [LS09]) that $\log \|\widehat{f}\|_{1,\epsilon}$ lower bounds the quantum bounded error communication complexity of $f \circ$ XOR. We expect that the lower bound $\log \|\widehat{f}\|_{1,\epsilon}$ is tight, and that this quantity characterizes the communication complexity of xor functions. More discussion on the communication complexity of xor functions, and how it relates to this work is given in Section 5.5.

This ends our discussion of the use of the spectral norm in learning theory, circuit complexity and communication complexity. We conclude this subsection by mentioning a relatively recent result that studies the spectral norm of boolean functions. Green and Sanders [GS08] show that every boolean function whose spectral norm is bounded by a constant can be written as a sum of constantly many ± indicators of cosets. This gives an interesting characterization of boolean functions with small spectral norm.

5.1.2 Fourier Spectrum of Symmetric Functions

As argued in previous chapters, symmetric functions are at the heart of complexity theory as natural functions like AND, OR, MAJORITY, and MOD$_m$ are all symmetric. They are often the starting point of investigation because the symmetry of the function can be exploited. On the other hand, they can also have surprising power. In several settings, functions such as PARITY and MAJORITY represent "hard" functions. Given their central role, it is of interest to gain insight into the Fourier spectrum of symmetric functions.

There are various nice results related to the Fourier spectrum of symmetric functions. We cite a few of them here. A beautiful result of Paturi [Pat92] tightly characterizes the approximate degree of every symmetric function, and this has found many applications in theoretical computer science [Raz03, BBC$^+$01, She09, dW08, She11]. Kolountzakis *et al.* [KLM$^+$09] studied the so called *minimal degree* of symmetric functions and applied their result in learning theory. Shpilka and Tal [ST11] later simplified and improved the work of Kolountzakis *et al.* Recently, O'Donnell, Wright and Zhou [OWZ11] verified an important conjecture in the analysis of boolean functions, the Fourier Entropy/Influence Conjecture, in the setting of symmetric functions. In fact we make use of their key lemma in this chapter.

5.1.3 Our Results and Proof Overview

We give a combinatorial characterization of the spectral norm of symmetric functions. For $x \in \{0,1\}^n$, recall $|x| = \sum x_i$. For a function $f : \{0,1\}^n \to \{-1,1\}$, let r_0 and r_1 be the minimum integers less than $n/2$ such that $f(x)$ or $f(x) \cdot \text{PARITY}(x)$ is constant for x with $|x| \in [r_0, n - r_1]$. Define $r(f) \overset{\text{def}}{=} \max\{r_0, r_1\}$. We show that $\log \|\widehat{f}\|_1$ is of the same order of magnitude as $r(f)\log(n/r(f))$:

Theorem 5.1.1 (Main Theorem)**.** *For any symmetric function $f : \{0,1\}^n \to \{-1,1\}$, we have*

$$\log \|\widehat{f}\|_1 = \Theta\left(r(f)\log\left(\frac{n}{r(f)}\right)\right)$$

whenever $r(f) > 1$. If $r(f) \leq 1$, then $\|\widehat{f}\|_1 = \Theta(1)$.

As an application, we give a characterization of the parity decision tree size of symmetric functions. As described in Section 2.1, a parity decision tree computes a boolean function by querying the parities of subsets of the variables. The size of the tree is simply the number of leaves in the tree.

Corollary 5.1.2. *Let $f : \{0,1\}^n \to \{-1,1\}$ be a symmetric function. Then the parity decision tree size of f is $2^{\Theta(r(f)\log(n/r(f)))}$.*

The proof of this corollary is presented in Section 5.4. Note that the lower bound also applies in the case of the usual decision tree size (where one is restricted to query only variables). Decision tree size is an important measure in learning theory; algorithms for learning decision trees efficiently is of great interest both for practical and theoretical reasons. One of the most well-known and studied problems is whether small size decision trees are efficiently learnable from uniformly random examples.

As a second application, using the protocol of Shi and Zhang [SZ09a, Proposition 3.4], and the observation that $\|\widehat{F}\|_1 = \|\widehat{f}\|_1$ when $F = f \circ \text{XOR}$, we verify Grolmusz's conjecture mentioned earlier in Section 5.1.1 in the setting of *symmetric xor functions.*

Corollary 5.1.3. *Let $F : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ be a function of the form* SYM ∘ XOR*, where* SYM *denotes an arbitrary symmetric function. Then for constant ϵ,*

$$\mathbf{R}^{\epsilon}(F) \leq O(\log^2 \|\widehat{F}\|_1).$$

Proof Overview

As the proof of our main result is technical, it is instructive to present an overview of the proof before diving into the details. To this end, we now give an outline for the proof of Theorem 5.1.1.

The upper bound is quite straightforward and is given in Lemma 5.3.1. The lower bound is handled in two different cases: when $r(f)$ is bounded away from $n/2$ (Lemma 5.3.3) and when $r(f)$ is close to $n/2$ (Lemma 5.3.7).

We refer to the Fourier spectrum of f restricted to the sets $S \subseteq [n]$ of size k as the *k-th level* of the Fourier spectrum. Note that for a symmetric f, we have $\widehat{f}(S) = \widehat{f}(T)$ whenever $|S| = |T|$. Therefore the Fourier spectrum is maximally spread out in each level. The overall strategy for the lower bound is to show an appropriate lower bound on the L_2 mass of the Fourier spectrum on a middle level. Middle levels have many Fourier coefficients, and therefore contribute significantly to the spectral norm provided there is enough L_2 mass on them. An important tool in our analysis is the use of certain discrete derivatives of f. Identify $\{0,1\}^n$ with $\mathbb{F}_2^n$ and let $e_1, \ldots, e_n$ denote the standard vectors in $\mathbb{F}_2^n$. For $i \neq j$, define $f_{ij}(x) \stackrel{\text{def}}{=} f(x + e_i + e_j) - f(x)$. We observe that

$$\sum_{i \neq j} \mathbf{E}\left[f_{ij}^2(\mathbf{x})\right] = 8 \sum_S |S|(n - |S|)\widehat{f}(S)^2.$$

The quantity on the LHS, and therefore the RHS, can be lower bounded using $r(f)$ (Lemma 5.3.2). As the coefficient $|S|(n - |S|)$ increases as $|S|$ approaches $n/2$, we are able to give a lower bound on the L_2 mass of the Fourier spectrum on the middle levels. This approach gives tight bounds for $r(f)$ bounded away from $n/2$, but not for a function such as MAJORITY.

To handle functions f with $r(f)$ close to $n/2$, we use ideas from [OWZ11]. The main lemma of [OWZ11] states that the first derivatives of a symmetric function are noise sensitive (see Section 2.3.1 for the definition of noise stability). We observe that this is also true for the derivatives f_{ij}. This allows us to derive the inequality

$$\sum_S |S|(n - |S|)\widehat{f}(S)^2(\rho^{|S|} + \rho^{n-|S|}) \leq \frac{8}{\sqrt{\pi c}} \cdot \sum_S |S|(n - |S|)\widehat{f}(S)^2,$$

where $\rho = (1 - c/n)$. The quantity $\rho^{|S|} + \rho^{n-|S|}$ is decreasing in $|S|$ for $|S| \leq n/2$. Thinking of c as a large constant, we see that the dampening of the middle levels with $\rho^{|S|} + \rho^{n-|S|}$ decreases the value of the sum significantly.

From this, we can lower bound the L_2 mass of the middle levels. Note that if $\sum_S |S|(n - |S|)\widehat{f}(S)^2$ is small to begin with ($r(f)$ is small), the above inequality is not useful. On the other hand if $r(f)$ is large, $\sum_S |S|(n - |S|)\widehat{f}(S)^2$ is large, and the strategy just described gives good bounds.

5.2 Preliminaries

For a boolean function f, we define $W_k[f] = \sum_{|S|=k} |\widehat{f}(S)|^2$. We simply use W_k when f is clear from the context. For a symmetric function, we often write $f(k)$ for $f(x)$ with $\sum_i x_i = k$ and $k \in [n]$. We use h to denote the binary entropy function $h(\alpha) = -\alpha \log(\alpha) - (1 - \alpha)\log(1 - \alpha)$. We will use the following simple estimates for binomial coefficients (See [MU05, Lemma 9.2]): Let $\alpha \in [0, 1]$ such that αn is an integer. Then

$$\sum_{k=0}^{\alpha n} \binom{n}{k} \leq 2^{nh(\alpha)}, \tag{5.1}$$

and

$$\frac{2^{nh(\alpha)}}{n+1} \leq \binom{n}{\alpha n}. \tag{5.2}$$

If $\alpha \in [0, 1/2]$ is arbitrary, then

$$\frac{2^{nh(\alpha)}}{n(n+1)} \leq \binom{n}{\lfloor \alpha n \rfloor} \leq 2^{nh(\alpha)}. \tag{5.3}$$

The following fact is also easy and classical. For every constant $c > 0$, there exists a constant $C > 0$ such that for any $n \geq 1$,

$$\binom{n}{\lfloor n/2 + c\sqrt{n} \rfloor} \geq C\frac{2^n}{\sqrt{n}}. \tag{5.4}$$

Definition 5.2.1. For any $f : \{0,1\}^n \to \mathbb{R}$, we define

$$R(f) \overset{\text{def}}{=} \sum_{S \subseteq [n]} |S|(n - |S|)\widehat{f}(S)^2.$$

For $a \in \mathbb{F}_2^n$, we define the derivative of $f : \mathbb{F}_2^n \to \mathbb{R}$ in the direction a as

$$\Delta_a f : x \mapsto f(x+a) - f(x).$$

Let $e_1, \ldots, e_n$ denote the standard vectors in $\mathbb{F}_2^n$, and let $f : \{0,1\}^n \to \mathbb{R}$. For all $i \neq j$, define

$$f_{ij} \stackrel{\text{def}}{=} \Delta_{e_i+e_j} f. \tag{5.5}$$

Lemma 5.2.2. *For every $f : \{0,1\}^n \to \mathbb{R}$, we have*

$$\sum_{i \neq j} \mathbf{E}\left[f_{ij}^2(\mathbf{x})\right] = 8R(f).$$

Proof. We have

$$f_{ij}(x) = \sum_S \widehat{f}(S)\chi_S(x)(\chi_S(e_i+e_j)-1) = \sum_{S:|S\cap\{i,j\}|=1} -2\widehat{f}(S)\chi_S(x),$$

which by Parseval's identity implies

$$\mathbf{E}\left[f_{ij}^2(\mathbf{x})\right] = \sum_{S:|S\cap\{i,j\}|=1} 4\widehat{f}(S)^2.$$

Summing over all pairs $i \neq j$, we obtain

$$\sum_{i \neq j} \mathbf{E}\left[f_{ij}^2(\mathbf{x})\right] = 8 \sum_{S \subseteq [n]} |S|(n-|S|)\widehat{f}(S)^2.$$

□

5.3 Proof of Theorem 5.1.1

As mentioned earlier the upper bound is proved in Lemma 5.3.1. The proof of the lower bound is divided into two parts: Lemma 5.3.3 handles the case where r is bounded away from $n/2$ and Lemma 5.3.7 the case when r is close to $n/2$.

5.3.1 Upper Bound

Lemma 5.3.1. *For all $n \geq 1$ and every symmetric function $f : \{0,1\}^n \to \{-1,1\}$,*

$$\log \|\widehat{f}\|_1 \leq 2 \cdot r(f) \log(n/r(f)) + 3.$$

Proof. By definition of r_0 and r_1, there exists a function

$$g \in \{-1, 1, -\textsc{parity}, +\textsc{parity}\}$$

such that $f(k) = g(k)$ for all $k \in [r_0, r_1]$. By linearity of the Fourier transform, we have for any $S \subseteq [n]$,

$$\begin{aligned}
\widehat{f}(S) &= \widehat{g}(S) + \widehat{f-g}(S) \\
&= \widehat{g}(S) + \frac{1}{2^n} \sum_{k=0}^{n} (f(k) - g(k)) \sum_{|x|=k} \chi_S(x) \\
&= \widehat{g}(S) + \frac{1}{2^n} \sum_{k=0}^{r_0-1} (f(k) - g(k)) \sum_{|x|=k} \chi_S(x) \\
&\quad + \frac{1}{2^n} \sum_{k=n-r_1+1}^{n} (f(k) - g(k)) \sum_{|x|=k} \chi_S(x)
\end{aligned}$$

Thus,

$$\begin{aligned}
|\widehat{f}(S)| &\leq |\widehat{g}(S)| + \frac{1}{2^n} \sum_{k=0}^{r_0-1} 2\binom{n}{k} + \frac{1}{2^n} \sum_{k=n-r_1+1}^{n} 2\binom{n}{k} \\
&\leq |\widehat{g}(S)| + 2 \cdot \frac{2^{h(r_0/n)n} + 2^{h(r_1/n)n}}{2^n},
\end{aligned}$$

where for the last inequality, we used (5.1). Summing over all subsets $S \subseteq [n]$, we get

$$\|\widehat{f}\|_1 \leq 1 + 2(2^{h(r_0/n)n} + 2^{h(r_1/n)n}) \leq 1 + 4 \cdot 2^{h(r/n)n}.$$

As $h(t) \leq -2t \log t$ when $t \leq 1/2$, we obtain $\log \|\widehat{f}\|_1 \leq 3 + 2r \log(n/r)$. □

5.3.2 Lower Bound

We start by making some simple observations.

Lemma 5.3.2. *Let $f : \{0,1\}^n \to \{-1,1\}$ be a symmetric function, and define $r_0 = r_0(f)$ and $r_1 = r_1(f)$. Then*

$$R(f) \geq \left((n-r_0+1)(n-r_0)\binom{n}{r_0-1} + (n-r_1+1)(n-r_1)\binom{n}{r_1-1} \right) 2^{-n}. \tag{5.6}$$

Moreover, assuming that $f(s) = 1$ for all $s \in \{r_0, \ldots, n-r_1\}$, we have

$$\sum_{S \neq \emptyset} \widehat{f}(S)^2 \leq 4 \left(\sum_{s<r_0} \binom{n}{s} + \sum_{s<r_1} \binom{n}{s} \right) 2^{-n}. \tag{5.7}$$

Proof. Define f_{ij} as in (5.5). As f is symmetric, we only need to consider f_{12}.

$$\begin{aligned}
\mathbf{E}\left[f_{12}^2(\mathbf{x})\right] &= \mathbf{E}_{\mathbf{x}_3\ldots\mathbf{x}_n}\Bigg[\frac{1}{4} \cdot \big(f_{12}^2(00\mathbf{x}_3\ldots\mathbf{x}_n) + f_{12}^2(01\mathbf{x}_3\ldots\mathbf{x}_n) \\
&\qquad\qquad + f_{12}^2(10\mathbf{x}_3\ldots\mathbf{x}_n) + f_{12}^2(11\mathbf{x}_3\ldots\mathbf{x}_n)\big)\Bigg] \\
&= \frac{1}{4}\mathbf{E}_{\mathbf{x}_3\ldots\mathbf{x}_n}\big[\,(f(00\mathbf{x}_3\ldots\mathbf{x}_n) - f(11\mathbf{x}_3\ldots\mathbf{x}_n))^2 \\
&\qquad\qquad + (f(11\mathbf{x}_3\ldots\mathbf{x}_n) - f(00\mathbf{x}_3\ldots\mathbf{x}_n))^2\,\big] \\
&\geq \frac{1}{2}\left(\binom{n-2}{r_0-1} \cdot 2^{-(n-2)} \cdot 4 + \binom{n-2}{n-r_1-1} \cdot 2^{-(n-2)} \cdot 4\right) \\
&= 8 \cdot \left(\frac{(n-r_0+1)(n-r_0)}{n(n-1)} \cdot \binom{n}{r_0-1}\right. \\
&\qquad\qquad \left. + \frac{(n-r_1+1)(n-r_1)}{n(n-1)} \cdot \binom{n}{r_1-1}\right)2^{-n}.
\end{aligned}$$

To obtain the second equality, observe that

$$f_{12}^2(01x_3\ldots x_n) = f_{12}^2(10x_3\ldots x_n) = 0$$

because f is symmetric. Inequality (5.6) follows by applying Lemma 5.2.2.

In order to establish inequality (5.7), we show a lower bound on the principal Fourier coefficient of f:

$$\widehat{f}(\emptyset) \geq 1 - 2\left(\sum_{s<r_0}\binom{n}{s} + \sum_{s>n-r_1}\binom{n}{s}\right)2^{-n},$$

which implies that

$$\widehat{f}(\emptyset)^2 \geq 1 - 4\cdot\left(\sum_{s<r_0}\binom{n}{s} + \sum_{s<r_1}\binom{n}{s}\right)2^{-n}.$$

□

Lower Bound: $r \ll n/2$

Lemma 5.3.3. *For every symmetric function* $f : \{0,1\}^n \to \{-1,1\}$ *with* $r = r(f)$,

$$\log\|\widehat{f}\|_1 \geq \Omega\left(\left(1 - \frac{2r-2}{n}\right)\cdot r\log(n/r)\right).$$

Proof. Observe that we can assume without loss of generality that $f(s) = 1$ for all $s \in \{r_0, \ldots, n - r_1\}$. In fact, to handle the case $f = -1$ or $f = \pm$PARITY in $[r_0, n - r_1]$, it suffices to multiply the function by -1 or by $\pm$PARITY, respectively. This does not affect the spectral norm of the function.

We prove the statement by showing that a significant portion of the L_2 mass of $\widehat{f}$ sits in the middle levels from m to $n - m$ for a well-chosen m depending on $r(f)$.

Define $\alpha_0 = \frac{r_0-1}{n} < 1/2$ and $\alpha_1 = \frac{r_1-1}{n}$. We also let

$$m_0 = \left\lfloor n/2\cdot(1 - \sqrt{4\alpha_0 - 6\alpha_0^2 + 4\alpha_0^3})\right\rfloor$$

and

$$m_1 = \left\lfloor n/2\cdot(1 - \sqrt{4\alpha_1 - 6\alpha_1^2 + 4\alpha_1^3})\right\rfloor.$$

By Lemma 5.3.2, we have $\sum_{k>0} W_k \leq 4 \cdot \left(\sum_{s<r_0} \binom{n}{s} + \sum_{s<r_1} \binom{n}{s}\right) 2^{-n}$. Let U_k and V_k be so that $W_k = U_k + V_k$ and $\sum_{k>0} U_k \leq 4 \cdot 2^{-n} \sum_{s<r_0} \binom{n}{s}$ and $\sum_{k>0} V_k \leq 4 \cdot 2^{-n} \sum_{s<r_1} \binom{n}{s}$. Our objective is now to obtain a lower bound on $\sum_{k=m_0}^{n-m_0} k(n-k)U_k + \sum_{k=m_1}^{n-m_1} k(n-k)V_k$ using Lemma 5.3.2

$$\begin{aligned}
&\sum_{k=m_0}^{n-m_0} k(n-k)U_k + \sum_{k=m_1}^{n-m_1} k(n-k)V_k \\
&\quad = R(f) - \sum_{k \notin [m_0, n-m_0]} k(n-k)U_k - \sum_{k \notin [m_1, n-m_1]} k(n-k)V_k \\
&\quad \geq (n-r_0)(n-r_0+1)\binom{n}{r_0-1} 2^{-n} - (m_0-1)(n-m_0+1)4 \cdot 2^{-n} \sum_{s<r_0} \binom{n}{s} \\
&\quad + (n-r_1)(n-r_1+1)\binom{n}{r_1-1} 2^{-n} - (m_1-1)(n-m_1+1)4 \cdot 2^{-n} \sum_{s<r_1} \binom{n}{s}.
\end{aligned} \tag{5.8}$$

Define $A_0 \overset{\text{def}}{=} (n-r_0)(n-r_0+1)\binom{n}{r_0-1}2^{-n} - (m_0-1)(n-m_0+1)4 \cdot 2^{-n} \sum_{s<r_0} \binom{n}{s}$, and let A_1 be its analogue for r_1 so that the right hand side of (5.8) equals $A_0 + A_1$.

Observe that $\binom{n}{s} = \frac{s+1}{n-s}\binom{n}{s+1}$, and $\frac{s+1}{n-s} \leq \frac{r_0-1}{n-(r_0-1)} = \frac{\alpha_0}{1-\alpha_0}$ for $s < r_0 - 1$.

Thus

$$\begin{aligned}
A_0 &\geq \binom{n}{r_0-1}2^{-n}\cdot \\
&\left((n-\alpha_0 n-1)(n-\alpha_0 n)-4(m_0-1)(n-m_0+1)\frac{1}{1-\alpha_0/(1-\alpha_0)}\right) \\
&= \binom{n}{r_0-1}2^{-n}\cdot \\
&\left(n^2(1-\alpha_0)^2-(1-\alpha_0)n-4(m_0-1)(n-m_0+1)\frac{1-\alpha_0}{1-2\alpha_0}\right) \\
&\geq \binom{n}{r_0-1}2^{-n}\cdot \\
&\left(n^2\left((1-\alpha_0)^2-(1-(4\alpha_0-6\alpha_0^2+4\alpha_0^3))\frac{1-\alpha_0}{1-2\alpha_0}\right)-(1-\alpha_0)n\right) \\
&= \binom{n}{r_0-1}2^{-n}(1-\alpha_0)\left(n^2\left((1-\alpha_0)-(1-2\alpha_0+2\alpha_0^2)\right)-n\right) \\
&= \binom{n}{r_0-1}2^{-n}(1-\alpha_0)\left(\alpha_0(1-2\alpha_0)n^2-n\right). \qquad (5.9)
\end{aligned}$$

Analogously, we have

$$A_1 \geq \binom{n}{r_1-1}2^{-n}(1-\alpha_1)\left(\alpha_1(1-2\alpha_1)n^2-n\right). \qquad (5.10)$$

We now assume that $r_0 \geq r_1$. Observe that we then have $m_0 \leq m_1$. Combining (5.8) and (5.9), we get

$$n^2\sum_{k=m_0}^{n-m_0} W_k \geq \sum_{k=m_0}^{n-m_0} k(n-k)W_k \geq \binom{n}{r_0-1}2^{-n}(1-\alpha_0)\left(\alpha_0(1-2\alpha_0)n^2-n\right).$$

Note that for symmetric functions $\|\widehat{f}\|_1 = \sum_{k=0}^{n} \sqrt{\binom{n}{k} W_k}$, and thus

$$\begin{aligned}
\|\widehat{f}\|_1 &\geq \sum_{k=m_0}^{n-m_0} \sqrt{\binom{n}{k} W_k} \geq \sqrt{\binom{n}{m_0} \sum_{k=m_0}^{n-m_0} W_k} \\
&\geq \sqrt{\binom{n}{m_0}\binom{n}{r_0-1} 2^{-n} \frac{(1-\alpha_0)\left(\alpha_0(1-2\alpha_0)n^2-n\right)}{n^2}} \\
&\geq \sqrt{\binom{n}{\left\lfloor n/2(1-\sqrt{4\alpha_0-6\alpha_0^2+4\alpha_0^3})\right\rfloor}\binom{n}{\alpha_0 n}} \\
&\cdot \sqrt{2^{-n}\frac{(1-\alpha_0)\left(\alpha_0(1-2\alpha_0)n^2-n\right)}{n^2}}. \qquad (5.11)
\end{aligned}$$

Using (5.2) and (5.3), we obtain

$$\|\widehat{f}\|_1^2 \geq \frac{2^{n\left(h\left(\frac{1}{2}-\frac{1}{2}\sqrt{4\alpha_0-6\alpha_0^2+4\alpha_0^3}\right)+h(\alpha_0)-1\right)}}{n(n+1)^2} \cdot \frac{(1-\alpha_0)\left(\alpha_0(1-2\alpha_0)n^2-n\right)}{n^2}.$$

As a result

$$\begin{aligned}
\log\|\widehat{f}\|_1 \geq \frac{n}{2}&\left(h\left(\frac{1}{2}-\frac{1}{2}\sqrt{4\alpha_0-6\alpha_0^2+4\alpha_0^3}\right)+h(\alpha_0)-1\right) \\
&+\frac{1}{2}\log\frac{(1-\alpha_0)\left(\alpha_0(1-2\alpha_0)n^2-n\right)}{n^3(n+1)^2}.
\end{aligned}$$

Claim 5.3.4. *There exists a constant $c > 0$ such that for every $\alpha_0 \in (0, 1/2)$,*

$$h\left(\frac{1}{2}-\frac{1}{2}\sqrt{4\alpha_0-6\alpha_0^2+4\alpha_0^3}\right)+h(\alpha_0)-1 \geq c(1-2\alpha_0)\cdot\alpha_0\cdot\log(1/\alpha_0). \qquad (5.12)$$

Proof. Using the inequality $|h(x_2)-h(x_1)| \leq h(x_2-x_1)$ which holds for every $0 < x_1 < x_2 < 1$, we have

$$h\left(\frac{1}{2}-\frac{1}{2}\sqrt{4\alpha_0-6\alpha_0^2+4\alpha_0^3}\right)+h(\alpha_0)-1 \geq h(\alpha_0)-h\left(\frac{1}{2}\sqrt{4\alpha_0-6\alpha_0^2+4\alpha_0^3}\right).$$

By looking at the Taylor expansion, it is easy to see that there exists an $\epsilon > 0$, such that for every $\alpha_0 \in [0, \epsilon] \cup \left[\frac{1}{2}-\epsilon, \frac{1}{2}\right]$ we have

$$h(\alpha_0)-h\left(\frac{1}{2}\sqrt{4\alpha_0-6\alpha_0^2+4\alpha_0^3}\right) \geq \frac{1}{2}(1-2\alpha_0)\cdot\alpha_0\cdot\log(1/\alpha_0).$$

On the other hand, there exists a constant $c_\epsilon > 0$ such that when $\alpha_0 \in (\epsilon, 1/2-\epsilon)$, both $h(\alpha_0) - h\left(\frac{1}{2}\sqrt{4\alpha_0 - 6\alpha_0^2 + 4\alpha_0^3}\right)$ and the right-hand side of (5.12) belong to $[c_\epsilon, 1]$. Taking $c \overset{\text{def}}{=} 1/c_\epsilon$ finishes the proof. □

Using this claim, we obtain

$$\log \|\widehat{f}\|_1 \geq c(1-2\alpha_0) \cdot \alpha_0 \log(1/\alpha_0) \cdot \frac{n}{2} + \frac{1}{2} \log \frac{(1-\alpha_0)\,(\alpha_0(1-2\alpha_0)n^2 - n)}{n^3(n+1)^2}.$$

This proves the desired result provided $r(f)$ is larger than some constant. Next we handle small (constant) values of $r(f)$. We start with the case $r(f) = 1$. In this case, it is easy to see that $\|\widehat{f}\|_1 = O(1)$. Next, we consider $r(f) = 2$. Let $g_k(x) = -1$ iff $|x| = \sum_i x_i = k$. For the function g_1, we have for $S \neq \emptyset$,

$$\begin{aligned}
\widehat{g_1}(S) &= \frac{1}{2^n} \cdot -2 \sum_{|x|=1} \chi_S(x) \\
&= \frac{-2}{2^n} \cdot \sum_{i=1}^{n} (-1)^{\mathbf{1}_{i \in S}} \\
&= \frac{-2}{2^n}(n - |S| - |S|) = \frac{-2(n-2|S|)}{2^n}.
\end{aligned}$$

Hence,

$$\begin{aligned}
\|\widehat{g_1}\|_1 &= 1 - 2\frac{n}{2^n} + \frac{2}{2^n} \sum_{k=1}^{n} \binom{n}{k} |n-2k| \\
&= \Theta(\sqrt{n}),
\end{aligned}$$

by observing that a constant fraction of the probability mass of the binomial distribution lies in the interval $[n/2 - 2\sqrt{n}, n/2 - \sqrt{n}]$. Similarly, one can show that $\|\widehat{g_1} + \widehat{g_{n-1}}\|_1 = \Theta(\sqrt{n})$. All other functions with $r(f) = 2$ are obtained from these two functions by adding functions g_0 or g_n and by multiplying by a constant or the parity function.

We now consider the case $r(f) \geq 3$, but constant. We perform an analysis similar to the proof of Lemma 5.3.3. We can assume that $r_0 \geq r_1$. We take $m_0 = \left\lfloor n/2(1 - \sqrt{5\alpha_0 - 6\alpha_0^2})\right\rfloor$. As in (5.9), we obtain the bound

$$A_0 \geq \binom{n}{r_0 - 1} 2^{-n}(1 - \alpha_0)(2\alpha_0 n^2 - n).$$

Hence, the analogue of inequality (5.11) becomes

$$\begin{aligned}\|\widehat{f}\|_1 &\geq \sqrt{\binom{n}{m_0}\binom{n}{r_0 - 1} 2^{-n}\frac{(1 - \alpha_0)(2\alpha_0 n^2 - n)}{n^2}} \\ &\geq \sqrt{\binom{n}{\left\lfloor n/2(1 - \sqrt{5\alpha_0 - 6\alpha_0^2})\right\rfloor}\binom{n}{\alpha_0 n} 2^{-n}\frac{(1 - \alpha_0)(2\alpha_0 n^2 - n)}{n^2}}.\end{aligned}$$

But $\left\lfloor n/2(1 - \sqrt{5\alpha_0 - 6\alpha_0^2})\right\rfloor = n/2 - \Theta(\sqrt{n})$ and thus $\binom{n}{\left\lfloor n/2(1 - \sqrt{5\alpha_0 - 6\alpha_0^2})\right\rfloor} = \Omega(2^n/\sqrt{n})$ (see inequality (5.4)). As a result,

$$\begin{aligned}\|\widehat{f}\|_1 &\geq \Omega\left(\sqrt{\frac{1}{\sqrt{n}}\binom{n}{\alpha_0 n}\frac{1}{n}}\right) \\ &\geq \Omega\left(\sqrt{\binom{n}{r_0 - 1} n^{-3/2}}\right),\end{aligned}$$

which proves the lemma. □

Lower Bound: $r \approx n/2$

For the case $r \approx n/2$, we use a result of [OWZ11] that states that the derivative of a symmetric boolean function is noise sensitive. Here, we use the noise sensitivity of the derivative f_{ij}. The following lemma is an analogue of [OWZ11, Theorem 6].

Lemma 5.3.5. *Let f be a symmetric boolean function and f_{ij} be defined as in* (5.5)*. Then for $\rho = 1 - c/n$, we have*

$$\sum_S \widehat{f_{ij}}(S)^2 \rho^{|S|} \leq \frac{4}{\sqrt{\pi c}} \cdot \sum_S \widehat{f_{ij}}(S)^2, \tag{5.13}$$

for any $c \in [1, n]$. Summing over all i, j with $i \neq j$, we get

$$8 \sum_S |S|(n - |S|)\widehat{f}(S)^2 \rho^{|S|} \leq \frac{4}{\sqrt{\pi c}} \cdot 8R(f). \tag{5.14}$$

Proof. The proof is the same as the proof of [OWZ11, Theorem 6] except that we use f_{ij} instead of the derivative. We have

$$\sum_S \widehat{f_{ij}}(S)^2 \rho^{|S|} = \mathbf{E}_{\mathbf{x}}\left[f_{ij}(\mathbf{x}) \mathbf{E}_{\mathbf{y}}\left[f_{ij}(\mathbf{y})\right]\right]$$

where in the expectations, $\mathbf{x}$ is uniform and $\mathbf{y} \sim_\rho \mathbf{x}$. Note that we can write for any x

$$\begin{aligned} \left|\mathbf{E}_{\mathbf{y}}\left[f_{12}(\mathbf{y})|x\right]\right| = \Bigg|\mathbf{E}_{\mathbf{y}_3 \ldots \mathbf{y}_n}\Bigg[&(\mathbf{Pr}\left[\mathbf{y}_1\mathbf{y}_2 = 00|x\right] - \mathbf{Pr}\left[\mathbf{y}_1\mathbf{y}_2 = 11|x\right]) \\ &\cdot\ (f(11\mathbf{y}_3 \ldots \mathbf{y}_n) - f(00\mathbf{y}_3 \ldots \mathbf{y}_n)) \ \Bigg|\ x\Bigg]\Bigg| \\ \leq \left|\mathbf{E}_{\mathbf{y}_3 \ldots \mathbf{y}_n}\left[f(11\mathbf{y}_3 \ldots \mathbf{y}_n) - f(00\mathbf{y}_3 \ldots \mathbf{y}_n)|x\right]\right|. \end{aligned}$$

To find an upper bound for this expression, it suffices to replace the use of [OWZ11, Lemma 1] by the following claim.

Claim 5.3.6. *Let $E = \{i \in [m] : i \equiv 0 \mod 2\}$ and $O = \{i \in [m] : i \equiv 1 \mod 2\}$. Let $p_1, \ldots, p_m$ be a non-negative unimodal sequence and $g : [m] \to \{-1, 0, 1\}$ with the property that the sets $g^{-1}(1) \cap E$ and $g^{-1}(-1) \cap E$ are interleaving, and the sets $g^{-1}(1) \cap O$ and $g^{-1}(-1) \cap O$ are interleaving. Then $|\sum_{i=1}^m p_i g(i)| \leq 2\max\{p_i\}$.*

To prove the claim, we simply write $|\sum_{i=1}^m p_i g(i)| \leq |\sum_{i \in O} p_i g(i)| + |\sum_{i \in E} p_i g(i)|$. Now [OWZ11, Lemma 1] implies that each term is upper-bounded by $\max\{p_i\}$. □

We are now ready to prove the following result.

Lemma 5.3.7. *There exists a constant $\gamma < 1/2$ such that for any symmetric boolean function f with $r(f) \geq \gamma n$, we have* $\log \|\widehat{f}\|_1 = \Omega(n)$.

Proof. Let $\rho = 1 - c/n$ where c is a constant chosen later, and let n be large enough so that $\rho \geq 1/2$. We apply (5.14) to $g \overset{\text{def}}{=} f \cdot \textsc{parity}$:

$$\sum_S |S|(n - |S|)\widehat{g}(S)^2 \rho^{|S|} \leq \frac{4}{\sqrt{\pi c}} \cdot R(g).$$

Note that $\textsc{parity} = \chi_{[n]}$ which shows $\widehat{f}([n] \setminus S) = \widehat{g}(S)$ for all S, and in particular $R(g) = R(f)$. So we can rewrite the above inequality as

$$\sum_S |S|(n - |S|)\widehat{f}(S)^2 \rho^{n-|S|} \leq \frac{4}{\sqrt{\pi c}} \cdot R(f). \tag{5.15}$$

Summing (5.14) and (5.15), we get

$$\sum_S |S|(n - |S|)\widehat{f}(S)^2 (1 - \rho^{|S|} - \rho^{n-|S|}) \geq \left(1 - \frac{8}{\sqrt{\pi c}}\right) R(f). \tag{5.16}$$

Let $\beta < 1/2$ be a positive constant to be chosen later. We have

$$\begin{aligned}\sum_{|S| \leq \beta n} |S|(n - |S|)\widehat{f}(S)^2 (\rho^{|S|} + \rho^{n-|S|}) &\geq \sum_{|S| \leq \beta n} |S|(n - |S|)\widehat{f}(S)^2 (\rho^{\beta n} + \rho^{(1-\beta)n}) \\ &\geq \sum_{|S| \leq \beta n} |S|(n - |S|)\widehat{f}(S)^2 (1/2 \cdot e^{-c\beta} + 1/2 \cdot e^{-c(1-\beta)}).\end{aligned}$$

For the first equality, we used the fact that $\rho^{|S|} + \rho^{n-|S|}$ is decreasing in $|S|$ for $|S| \leq n/2$. For the second inequality, we used the inequality $(1 - c/n)^{\beta n} \geq e^{-c\beta}/2$ when $1 - c/n \geq 1/2$. Similarly, we have

$$\begin{aligned}&\sum_{|S| \geq (1-\beta)n} |S|(n - |S|)\widehat{f}(S)^2 (\rho^{|S|} + \rho^{n-|S|}) \\ &\qquad \geq \sum_{|S| \geq (1-\beta)n} |S|(n - |S|)\widehat{f}(S)^2 (e^{-c\beta}/2 + e^{-c(1-\beta)}/2).\end{aligned}$$

Summing the two inequalities, we obtain

$$\begin{aligned}&\sum_{|S| \notin (\beta n, (1-\beta)n)} |S|(n - |S|)\widehat{f}(S)^2 (\rho^{|S|} + \rho^{n-|S|}) \\ &\qquad \geq \frac{e^{-c\beta} + e^{-c(1-\beta)}}{2} \sum_{|S| \notin (\beta n, (1-\beta)n)} |S|(n - |S|)\widehat{f}(S)^2.\end{aligned}$$

Combining this with (5.16), we obtain

$$\begin{aligned}
&\sum_{\beta n \le |S| \le (1-\beta)n} |S|(n-|S|)\widehat{f}(S)^2(1-\rho^{|S|}-\rho^{n-|S|}) \\
&\quad = \sum_{S} |S|(n-|S|)\widehat{f}(S)^2(1-\rho^{|S|}-\rho^{n-|S|}) \\
&\qquad - \sum_{|S| \notin (\beta n,(1-\beta)n)} |S|(n-|S|)\widehat{f}(S)^2(1-\rho^{|S|}-\rho^{n-|S|}) \\
&\quad \ge (1-\frac{8}{\sqrt{\pi c}})R(f) - (1-e^{-c\beta}/2-e^{-c(1-\beta)}/2) \\
&\qquad \cdot \sum_{|S| \notin (\beta n,(1-\beta)n)} |S|(n-|S|)\widehat{f}(S)^2.
\end{aligned}$$

As $e^{-c\beta}/2+e^{-c(1-\beta)}/2<1$, this leads to

$$\begin{aligned}
&\sum_{\beta n \le |S| \le (1-\beta)n} |S|(n-|S|)\widehat{f}(S)^2(1-\rho^{|S|}-\rho^{n-|S|}) \\
&\qquad \ge \left(e^{-c\beta}/2+e^{-c(1-\beta)}/2-\frac{8}{\sqrt{\pi c}}\right)R(f).
\end{aligned}$$

Consequently,

$$\frac{n^2}{4}\sum_{\beta n \le |S| \le (1-\beta)n} \widehat{f}(S)^2 \ge R(f)(e^{-c\beta}/2+e^{-c(1-\beta)}/2-8/\sqrt{\pi c}).$$

By picking $c=10^4$ and $\beta=10^{-4}\ln 2$, we have $\frac{e^{-c\beta}+e^{-c(1-\beta)}}{2}-\frac{8}{\sqrt{\pi c}} \ge \frac{1}{10}$. We conclude that $\sum_{\beta n \le k \le (1-\beta)n} W_k \ge \frac{4R(f)}{10n^2}$, and thus

$$\|\widehat{f}\|_1 = \sum_{k=0}^{n}\sqrt{\binom{n}{k}W_k} \ge \sqrt{\binom{n}{\beta n}R(f)\frac{4}{10n^2}}.$$

Using (5.6), it follows that

$$\|\widehat{f}\|_1 = \Omega\left(\sqrt{\binom{n}{\beta n}\binom{n}{r-1}2^{-n}}\right) = \Omega\left(2^{(h(\beta)+h(\alpha)-1)\frac{n}{2}}(n+1)^{-1}\right),$$

where $\alpha=(r-1)/n$. If α is such that $h(\alpha) \ge 1-h(\beta)/2$, we obtain the desired bound $\log\|\widehat{f}\|_1=\Omega(n)$. □

5.4 Proof of Corollary 5.1.2

We start by observing that we can assume that $f(x)$ is constant whenever $|x| \in [r_0, n - r_1]$. In fact, if this is not the case, then $f \cdot \textsc{parity}(x)$ will be constant when $|x| \in [r_0, n-r_1]$. But $f(x)$ can be computed from $f \cdot \textsc{parity}(x)$ using only one query to $\textsc{parity}(x)$, which multiplies the size of the tree by at most 2. In the remainder of the proof, we assume $f(x)$ is constant for $|x| \in [r_0, n - r_1]$.

We start by proving the lower bound. It is simple to prove that $\|\widehat{f}\|_1$ is a lower bound on the parity decision tree size of f [KM91, Lemma 5.1]. For completeness, we provide a sketch of a proof. As all the possible inputs that lead to some leaf L have the same value for f, we can write f as a sum over all leaves of the tree $f(x) = \sum_L f(L)\mathbf{1}_L(x)$, where the function $\mathbf{1}_L$ takes value 1 if the input belongs to the leaf L and is 0 otherwise. By linearity of the Fourier transform and the triangle inequality, we have $\|\widehat{f}\|_1 \leq \sum_L |f(L)| \|\widehat{\mathbf{1}_L}\|_1$. Now observe that the inputs corresponding to L (that we also call L) are inputs that satisfy some parity conditions on subsets belonging to some subspace $\mathcal{S}$. Then, we have $\widehat{\mathbf{1}_L}(S) = \pm\frac{|L|}{2^n}$ for any $S \in \mathcal{S}$. Note that the number of such subsets is $2^n/|L|$. But if $S \notin \mathcal{S}$, then $\sum_{x \in L} \chi_S(x) = 0$. It follows that $\|\widehat{\mathbf{1}_L}\|_1 = 1$ and that $\|\widehat{f}\|_1$ is a lower bound on the size of the tree.

Using Theorem 5.1.1, this proves the lower bound stated in Corollary 5.1.2, except in the case where $r(f) = 1$. For this case, observe that we can assume that a leaf at depth d corresponds to 2^{n-d} possible inputs; see e.g., [KM91, Lemma 5.1]. So we have at most two input bit strings that have a value for f that is different from the value f takes when $x \in [r_0(f), n-r_1(f)]$. This proves that the depth of the tree is at least $n - 1$ and completes the proof of the lower bound.

For the upper bound, we give a decision tree of size at most $4\binom{n}{r_0(f)} + 4\binom{n}{r_1(f)}$ for computing f. We start by considering a complete binary tree of depth n. Level i of the tree corresponds to querying the i-th input bit x_i. The number of leaves of the tree is 2^n. Clearly, one can compute any function using such a tree. We are going to use the values $r_0(f)$ and $r_1(f)$ to remove unnecessary nodes from the tree. Note that each node at level i can be labelled by a bit string of length i. We remove all the nodes that have r_0 ones and at least r_1 zeros, and the nodes that have r_1 zeros and at least r_0 ones, together with all their children. All of these nodes correspond to inputs x for which $|x| \in [r_0, n - r_1]$, so the value of f is a constant that only depends on f.

It now remains to compute the number of leaves of the constructed decision tree. The number of leaves at a level $i < n$ is 0 if $i < r_0 + r_1$ and $\binom{i-1}{r_0-1} + \binom{i-1}{r_1-1}$ if $i \geq r_0 + r_1$. At level n, we have all the remaining nodes that can have at most r_0 ones or at most r_1 zeros, thus at most $\binom{n}{r_0} + \binom{n}{r_1}$ leaves. Thus, the total number of leaves is at most

$$\begin{aligned}
\sum_{i=r_0+r_1}^{n-1} & \binom{i-1}{r_0-1} + \binom{i-1}{r_1-1} + \binom{n}{r_0} + \binom{n}{r_1} \\
& \leq \sum_{i=r_0-1}^{n-1} \binom{i}{r_0-1} + \sum_{i=r_1-1}^{n-1} \binom{i}{r_1-1} + \binom{n}{r_0} + \binom{n}{r_1} \\
& = 2 \cdot \left(\binom{n}{r_0} + \binom{n}{r_1} \right).
\end{aligned}$$

We can then obtain the stated result by (5.3) and the fact that $h(x) \leq -2x \log x$ for $x \in (0, 1/2]$.

5.5 Future Work

A natural next step is to extend Theorem 5.1.1 to *approximate* spectral norm. Indeed this would have interesting implications. Recall that the ϵ-approximate spectral norm of a boolean function f is the smallest spectral norm of a function ϕ with $\|f - \phi\|_\infty \leq \epsilon$, i.e., for all x, $|f(x) - \phi(x)| \leq \epsilon$. Trivially $\|\widehat{f}\|_{1,\epsilon}$ is smaller than $\|\widehat{f}\|_1$. We conjecture that it cannot be much smaller.

Conjecture 5.5.1. *For all symmetric functions $f : \{0,1\}^n \to \{\pm 1\}$,*

$$\log \|\widehat{f}\|_1 = \Theta^*(\log \|\widehat{f}\|_{1,1/3})$$

where Θ^* suppresses factors between 1 and $\log n$.

We now discuss some of the applications of the above conjecture in conjunction with Theorem 5.1.1.

Analog of Paturi's Result for Monomial Complexity

A famous result of Paturi [Pat92] characterizes the approximate degree of all symmetric functions. Recall that the degree of a function f is the largest $|S|$ such that $\widehat{f}(S)$ is non-zero. The ϵ-approximate degree is then the smallest degree of a function ϕ with $\|f - \phi\|_\infty \leq \epsilon$. Let t_0 and t_1 be the minimum integers such that $f(i) = f(i+1)$ for all $i \in [t_0, n - t_1]$.

Theorem 5.5.2 ([Pat92])**.** *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a symmetric function and let t_0 and t_1 be defined as above. Then, $\deg_{1/3}(f) = \Theta(\sqrt{n(t_0 + t_1)})$.*

Paturi's result has found numerous applications in theoretical computer science [Raz03, BBC$^+$01, She09, dW08, She11].

The monomial complexity of a boolean function f, denoted $\mathrm{mon}(f)$, is the number of non-zero Fourier coefficients of f. The approximate monomial complexity is then also defined as the smallest monomial complexity of a function that approximates f in the ℓ_∞ norm. The monomial complexity appears naturally in various areas of complexity theory, and it is desirable to obtain simple characterizations for natural classes of functions. An argument similar to the one in [BS92] shows that $\mathrm{mon}_\epsilon(f) \leq \frac{2n}{\epsilon^2}\|\widehat{f}\|_1^2$ for every $\epsilon > 0$.

Combining this with Conjecture 5.5.1 and Theorem 5.1.1 would show that $r(f)$ characterizes the approximate monomial complexity of f:

Conjecture 5.5.3 (Consequence of Conjecture 5.5.1)**.** *For a symmetric function $f : \{0,1\}^n \to \{\pm 1\}$,*

$$\log \mathrm{mon}_{1/3}(f) = \Theta^*(r(f)).$$

Communication Complexity of Xor Functions

Recall the Log Rank Conjecture mentioned in the introduction. This conjecture has an analogous version for the randomized communication complexity model: "Log Approximation Rank Conjecture". The ϵ-approximate rank of a matrix M is denoted by $\mathrm{rank}_\epsilon(M)$, and is the minimum rank of a matrix that ϵ approximates M. It is known that $\mathbf{R}^\epsilon(F) \geq \log \mathrm{rank}_{\epsilon'}(M_F)$, where ϵ' is a constant that depends on ϵ and M_F is the matrix representation of F. Log Approximation Rank Conjecture states that this lower bound is tight:

Conjecture 5.5.4 (Log Approximation Rank Conjecture)**.** *There is a universal constant c such that for any 2 party communication problem F,*

$$\log \mathrm{rank}_{\epsilon'}(M_F) \leq \mathbf{R}^\epsilon(F) \leq \log^c \mathrm{rank}_{\epsilon'}(M_F).$$

The important paper of Razborov [Raz03] established this conjecture for the functions SYM $\circ$ AND. In fact, Razborov showed that the quantum and classical randomized communication complexities of such functions are polynomially related. Later, Shi and Zhang [SZ09a], via a reduction to the case SYM$\circ$AND, showed the quantum/classical equivalence for symmetric xor functions SYM$\circ$XOR. They show that the randomized and quantum bounded error communication complexities of F are both $\Theta(r(f))$, up to polylog factors. However, their result does not verify the Log Approximation Rank Conjecture for symmetric xor functions.

Conjecture 5.5.1 along with Theorem 5.1.1 would verify the Log Approximation Rank Conjecture for symmetric xor functions (This follows from the protocol of Shi and Zhang [SZ09a, Proposition 3.4] for symmetric xor functions, and the facts $\|M_F\|_{tr,\epsilon} = 2^n\|\widehat{f}\|_{1,\epsilon}$ and $\text{rank}_\epsilon(M_F)^{1/2} \geq \|M_F\|_{tr,\epsilon}/(1+\epsilon)2^n$.). Furthermore, we would obtain a direct proof of the result of Shi and Zhang. This is very desirable since a major open problem is to understand the communication complexity of $f \circ \textsc{xor}$ for general f (with no symmetry condition on f). There is a sentiment that this should be easier to tackle than $f \circ \textsc{and}$ as xor functions seem more amenable to Fourier analytic techniques. A direct proof of the result of Shi and Zhang gives more insight into the communication complexity of xor functions.

Agnostically Learning Symmetric Functions

Let $\mathcal{C}$ be a concept class and $\phi_i : \{-1,1\}^n \to \mathbb{R}$ be functions for $1 \leq i \leq s$ such that every $f : \{-1,1\}^n \to \{-1,1\}$ in $\mathcal{C}$ satisfies $\|f - \sum_{i=1}^s c_i\phi_i\|_\infty \leq \epsilon$, for some reals c_i. The smallest s for which such ϕ_i's exist corresponds to the ϵ-approximate rank of $\mathcal{C}$. If each $\phi_i(x)$ is computable in polynomial time, then $\mathcal{C}$ can be agnostically learned under any distribution in time $\text{poly}(n,s)$ and with accuracy ϵ [KKMS08] (see the paper for the definition of agnostic learning).

Klivans and Sherstov [KS10] proved strong lower bounds on the approximate rank of the concept class of disjunctions $\{\bigvee_{i\in S} x_i : S \subseteq [n]\}$ and majority functions $\{\textsc{maj}(\pm x_1, \pm x_2, \ldots, \pm x_n)\}$ thereby ruling out the possibility of applying the algorithm of [KKMS08] to agnostically learning these concept classes.

Theorem 5.1.1 together with Conjecture 5.5.1 provides additional negative results and gives strong lower bounds on the approximate rank of the concept class consisting of symmetric functions f with large $r(f)$.

CHAPTER 6

Hardness of Private Communication

As significant parts of our lives migrate to the internet, the study of *privacy* becomes an important area of research in many disciplines. There is a huge body of work studying different aspects of privacy in different contexts. In this chapter, we will be concerned with privacy in the context of communication complexity. A prime example of interest comes from auctions, in which there are several participants who wish to obtain some item(s). The participants place bids and the auctioneer then decides on the winner and the price the winner has to pay based on the rules of the auction. For example, in the Vickrey auction (also known as the 2nd price auction), the highest bidder wins and pays the price of the second highest bid. Vickrey auction is quite famous in the game theory literature since it is the canonical example of a *truthful* mechanism, i.e., the best strategy for each participant, regardless of what other players choose to do, is to bid their true valuation for the item. In auctions like these, the auctioneer is concerned with obtaining information from the participants that allows her to compute the outcome of the auction. On the other hand, the participants do not want to give away any information about their inputs (valuations) that is not necessary to compute the outcome.

Let's look into the above example a little deeper. Suppose we have two players and a single item. The players have their private valuations (a positive integer) for this item. We'll view their valuation as their input. In the most basic protocol, the players would communicate their inputs to the auctioneer and the auctioneer would compare the two values and decide on the winner as well as the price the winner has to pay. Although this protocol is communication-wise quite efficient[1], both players' inputs are completely revealed to the auctioneer. In contrast, consider the following protocol which proceeds in rounds. In round i, the players send a bit each indicating whether their input is greater than i. When we reach a round where one of the players indicates their input is not greater than i, the protocol ends. The auctioneer can then decide on the winner and the price the winner has to pay (say we break ties in a predetermined way). Notice that in this protocol, the auctioneer learns what she needs to learn in order to decide the outcome, but she does not learn any additional information. In particular, she does not learn the input of the winner, but only the input of the loser. And the input of the loser is required information to determine how much the winner has to pay. Even though this protocol is "perfectly private", it suffers in the cost of communication since the protocol can have cost exponential in the length of the players' inputs.

To make the above discussion a bit more formal, let $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = [2^n]$

[1] In this setting, this is considered to be an efficient protocol because the output of the protocol is essentially the same length as the inputs to the players.

and define $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z} \times \{A, B\}$ as

$$F(x, y) = \begin{cases} (x, B) & \text{if } x \leq y, \\ (y, A) & \text{if } y < x. \end{cases}$$

This function corresponds to the Vickrey auction. The matrix representation of F is shown in Figure 6.1. Informally, we say that F has a perfectly private communication protocol if the communication transcript does not reveal any information about the players' inputs other than what is revealed by the function's output. Formally, this corresponds to saying that the monochromatic rectangles induced by the protocol exactly coincides with the rectangles shown in the figure. If, for example, the protocol induces two rectangles for a particular output, say $(1, B)$, knowing that we are in one rectangle and not the other gives us information about Player B's input that the function output itself does not give. It is easy to check that the communication-wise inefficient protocol that we described above induces exactly the rectangles

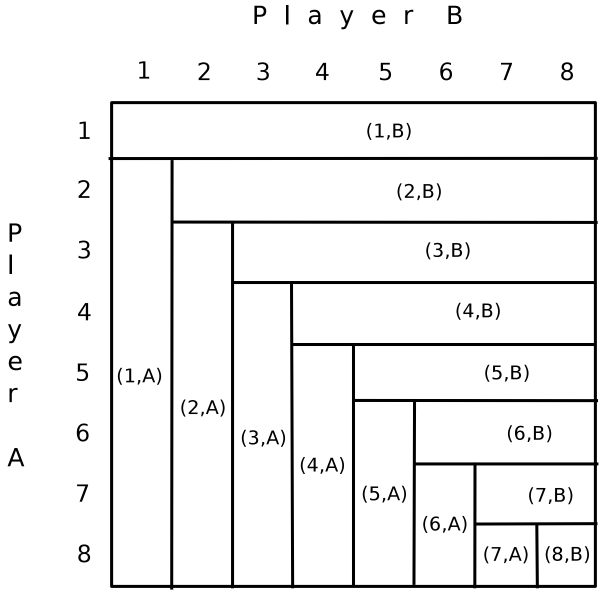

Figure 6.1: The matrix for Vickrey auction when $n = 3$.

shown in the figure and is indeed a perfectly private protocol. On the other hand, the efficient protocol in which both players reveal their inputs induces a separate rectangle for each entry of the matrix. As such, it is the worst possible protocol in terms of privacy.

Studying only perfect privacy has several shortcomings. First, it is easy to see that many functions do not have perfectly private protocols and therefore requiring perfect privacy is not reasonable. Second, even if a function has a perfectly private protocol, it may require infeasible amount of communication. Third, although perfect privacy is nice to pursue in an ideal world, in the real world, some loss of privacy can be quite acceptable. All these reasons motivate the study of *approximate* privacy and trade-offs between privacy and communication complexity.

Feigenbaum et al. [FJS10a] define two notions of approximate privacy. The *worst-case privacy approximation ratio* is defined to be the worst ratio between the size of a region corresponding to a particular output and the size of a protocol induced rectangle within that region. In the *average-privacy approximation ratio*, rather than taking the worst ratio, we look at the average ratio (formal definitions of these notions will be made in the next section). The authors study the privacy approximation ratios of several natural auctions and functions. In the case of Vickrey auction, they are able to analyse some specific protocols and obtain upper and lower bounds on their approximate privacy. However, they do not obtain a general result that applies to *any* protocol that computes the Vickrey auction.

In this chapter, we show that any protocol computing the Vickrey auction has an inherent trade-off between privacy and communication complexity by obtaining asymptotically tight lower bounds (in terms of the communication cost) on worst-case and average-case privacy approximation ratios of these protocols. Furthermore, we obtain an exponential lower bound on the average-case privacy approximation ratio of any protocol computing the *set* *intersection* function, independent of the cost of the protocol. This solves an explicitly stated open problem from [FJS10a].

The rest of this chapter is organized as follows. In Section 6.1 we formally define the privacy measures that we are interested in. First we define the worst-case privacy approximation ratio, then we define the average-case privacy approximation ratio, and in the last part, we discuss the connection of average-case privacy approximation ratio with other information theoretic notions of privacy. In Section 6.2, we review our initial discussion on the Vickrey auction and state some preliminary results. Section 6.3 is devoted to the presentation of our main results. First we prove worst-case privacy approximation ratio lower bound for the Vickrey auction. Next we prove average-case privacy approximation ratio lower bound for the Vickrey auction. And lastly, we prove average-case privacy approximation ratio lower bound for the Intersection function by using the connection between average-case privacy approximation ratio and information theoretic notions of privacy. We conclude in Section 6.4.

6.1 Privacy Measures

6.1.1 Worst-Case Privacy Approximation Ratio

We first set some notation. Given $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, each input (x, y) is associated with the *region* $R_{x,y}$ of all inputs in the preimage of $F(x, y)$, i.e.,

$$R_{x,y} = \{(x', y') \in \mathcal{X} \times \mathcal{Y} \mid F(x, y) = F(x', y')\}.$$

For any value $z \in \mathcal{Z}$, we let $R_z = F^{-1}(z)$ be the preimage of z. The set of all regions of a function F is $\mathcal{R}(F) = \{R_{x,y} \in \mathcal{X} \times \mathcal{Y}\}$. Let P be a communication protocol that computes the function F. Recall that $\Pi_P(x, y)$ denotes the transcript that the protocol produces on input (x, y). We let $P_{x,y}$

denote the protocol-induced rectangle that contains (x, y), or in other words,

$$P_{x,y} = \{(x', y') \in \mathcal{X} \times \mathcal{Y} \mid \Pi_P(x, y) = \Pi_P(x', y')\}.$$

Note that $P_{x,y} \subseteq R_{x,y}$ since P correctly computes F.

A perfectly private communication protocol for F will reveal only the output of F and no additional information. Every two inputs (x, y) and (x', y') such that $F(x, y) = F(x', y')$ should be indistinguishable from each other [Kus89, CGGK94]. Approximate privacy provides a measure of how much indistinguishability has been lost. The following definition captures the privacy loss of a communication protocol with respect to a third party observer (eavesdropper) who overhears the messages sent between the players. This measure is referred to as **objective**.

Definition 6.1.1. [FJS10a] A protocol P for a function F on $\mathcal{X} \times \mathcal{Y}$ has **worst-case objective privacy approximation ratio** (PAR) defined by

$$\mathrm{PAR}(P) = \max_{(x,y)} \frac{|R_{x,y}|}{|P_{x,y}|} = \max_{(x,y)} \mathrm{PAR}(P, x, y),$$

where for each input (x, y), $\mathrm{PAR}(P, x, y) = \frac{|R_{x,y}|}{|P_{x,y}|}$ denotes that input's own privacy approximation ratio. Often we do not specify the protocol P when it is clear from context.

The PAR measure of privacy can be extended to **subjective PAR**, which measures the privacy that the players lose *to each other.*

Definition 6.1.2. [FJS10a] A protocol P for a function F on $\mathcal{X} \times \mathcal{Y}$ has **worst-case subjective privacy approximation ratio** ($\mathrm{PAR}^{\mathrm{sub}}$) defined by:

$$\mathrm{PAR}^{\mathrm{sub}}(P) = \max\left\{ \max_{(x,y)} \frac{|R_{x,y} \cap \mathcal{X} \times \{y\}|}{|P_{x,y} \cap \mathcal{X} \times \{y\}|}, \max_{(x,y)} \frac{|R_{x,y} \cap \{x\} \times \mathcal{Y}|}{|P_{x,y} \cap \{x\} \times \mathcal{Y}|} \right\}.$$

6.1.2 Average-Case Privacy Approximation Ratio

For a probability distribution D on $\mathcal{X} \times \mathcal{Y}$ and a protocol P for a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, Feigenbaum et al. [FJS10a] define the average-case PAR as the quantity

$$\mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}\left[\frac{|R_{\mathbf{x},\mathbf{y}}|}{|P_{\mathbf{x},\mathbf{y}}|}\right].$$

Here we consider the following alternative definition.

Definition 6.1.3. For a probability distribution D on $\mathcal{X} \times \mathcal{Y}$ and a protocol P for a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, let the **average-case objective privacy approximation ratio** of protocol P for function F be:

$$\operatorname{avg}_D \mathrm{PAR}(P) = \mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}\left[\frac{|R_{\mathbf{x},\mathbf{y}}|_D}{|P_{\mathbf{x},\mathbf{y}}|_D}\right],$$

where for $S \subseteq \mathcal{X} \times \mathcal{Y}$, $|S|_D = \sum_{(x,y)\in S} D(x,y)$. Furthermore, we let the **average-case subjective privacy approximation ratio** of protocol P for function F be:

$$\operatorname{avg}_D \mathrm{PAR}^{\mathrm{sub}}(P) =$$
$$\max\left\{\mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}\left[\frac{|R_{\mathbf{x},\mathbf{y}} \cap \mathcal{X} \times \{\mathbf{y}\}|_D}{|P_{\mathbf{x},\mathbf{y}} \cap \mathcal{X} \times \{\mathbf{y}\}|_D}\right], \mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}\left[\frac{|R_{\mathbf{x},\mathbf{y}} \cap \{\mathbf{x}\} \times \mathcal{Y}|_D}{|P_{\mathbf{x},\mathbf{y}} \cap \{\mathbf{x}\} \times \mathcal{Y}|_D}\right]\right\}.$$

As opposed to Feigenbaum et al. we measure the size of subsets of $\mathcal{X} \times \mathcal{Y}$ relative to the measure D. This definition coincides with the definition of Feigenbaum et al. for the uniform distribution. Their paper does not give any results for distributions other than uniform, so our definition is consistent with their results. Similarly, our main results are also for the uniform distribution. We believe that our modified measure has several advantages. It allows natural alternative characterizations, and (as we will see in the next subsection) it is related to other known measures of privacy.

One benefit of Definition 6.1.3 is that it allows us to write average-case PAR as a sum that is convenient to work with. Consider a protocol P for a function F. For a region $R \in \mathcal{R}(F)$ let $\mathrm{cut}_P(R) = |\{P_{x,y} \,|\, (x,y) \in R\}|$ be the number of protocol-induced rectangles contained within R. The following statement is implicit in Feigenbaum et al. [FJS10a] for the case of uniform distribution and objective PAR.

Proposition 6.1.4. *For any function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, protocol P for F and any probability distribution D on $\mathcal{X} \times \mathcal{Y}$,*

$$\mathrm{avg}_D\,\mathrm{PAR}(P) = \sum_{R \in \mathcal{R}(f)} |R|_D \cdot \mathrm{cut}_P(R),$$

and

$$\begin{aligned} \mathrm{avg}_D\,\mathrm{PAR}^{\mathrm{sub}}(P) \;\; = \;\; \max\Bigg\{ & \sum_{y \in \mathcal{Y}, R \in \mathcal{R}(F)} |R \cap \mathcal{X} \times \{y\}|_D \cdot \mathrm{cut}_P(R \cap \mathcal{X} \times \{y\}), \\ & \sum_{x \in \mathcal{X}, R \in \mathcal{R}(F)} |R \cap \{x\} \times \mathcal{Y}|_D \cdot \mathrm{cut}_P(R \cap \{x\} \times \mathcal{Y}) \Bigg\}. \end{aligned}$$

Proof. For any protocol-induced rectangle A, $\sum_{(x,y)\in A} D(x,y) \cdot \frac{1}{|A|_D} = 1$. Hence,

$$\begin{aligned} \mathrm{avg}_D\,\mathrm{PAR}(P) &= \mathbf{E}_{(\mathbf{x},\mathbf{y})\sim D}\left[\frac{|R_{\mathbf{x},\mathbf{y}}|_D}{|P_{\mathbf{x},\mathbf{y}}|_D}\right] \\ &= \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} D(x,y) \cdot \frac{|R_{x,y}|_D}{|P_{x,y}|_D} \\ &= \sum_{R\in\mathcal{R}(F)} \sum_{(x,y)\in R} \frac{D(x,y)\cdot |R|_D}{|P_{x,y}|_D} \\ &= \sum_{R\in\mathcal{R}(F)} |R|_D \Bigg(\sum_{(x,y)\in R} \frac{D(x,y)\cdot 1}{|P_{x,y}|_D} \Bigg) \\ &= \sum_{R\in\mathcal{R}(F)} |R|_D \cdot \mathrm{cut}_P(R), \end{aligned}$$

where for the last equality, divide the R in the inner sum into the protocol induced rectangles inside R.

The case of subjective PAR is analogous. □

6.1.3 Average-Case PAR and Mutual Information

The definition of average-case PAR is closely related to previously studied concepts in communication complexity such as information cost [BBCR10] (as discussed in Section 2.2.6) and information-theoretic privacy introduced by Klauck [Kla02]. The main distinction is that these concepts measure in terms of bits, and PAR does not. Next we recapitulate some of these measures and show their relationship to average-case PAR.

Among these notions, Klauck's privacy measure [Kla02] is most closely related to average-case PAR. Let D be a probability distribution on $\mathcal{X} \times \mathcal{Y}$ and let $(\mathbf{x}, \mathbf{y}) \sim D$. Klauck [Kla02] gives the following definition of privacy of a protocol.

$$\mathrm{PRIV}_D(P) = \max\{I(\mathbf{x} : \Pi_P(\mathbf{x}, \mathbf{y}) | \mathbf{y}, F(\mathbf{x}, \mathbf{y})), I(\mathbf{y} : \Pi_P(\mathbf{x}, \mathbf{y}) | \mathbf{x}, F(\mathbf{x}, \mathbf{y}))\}.$$

The relationship between this measure and our average-case PAR is given by the following theorem.

Theorem 6.1.5. *For a probability distribution D on $\mathcal{X} \times \mathcal{Y}$ and a protocol P for a function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, the following holds:*

$$\mathrm{PRIV}_D(P) \leq \log(\mathrm{avg}_D \, \mathrm{PAR}^{\mathrm{sub}}(P)).$$

Proof. By symmetry, it suffices to show that $I(\mathbf{x} : \Pi_P(\mathbf{x}, \mathbf{y}) | \mathbf{y}, F(\mathbf{x}, \mathbf{y})) \leq \log(\mathrm{avg}_D \, \mathrm{PAR}^{\mathrm{sub}}(P))$.

$$\begin{aligned}
& I(\mathbf{x} : \Pi_P(\mathbf{x}, \mathbf{y}) | \mathbf{y}, F(\mathbf{x}, \mathbf{y})) \\
& \leq H(\Pi_P(\mathbf{x}, \mathbf{y}) | \mathbf{y}, F(\mathbf{x}, \mathbf{y})) \\
& = \sum_{y \in \mathcal{Y}, z \in \mathcal{Z}} \Pr[\mathbf{y} = y, F(\mathbf{x}, \mathbf{y}) = z] \cdot H(\Pi_P(\mathbf{x}, \mathbf{y}) | \mathbf{y} = y, F(\mathbf{x}, \mathbf{y}) = z) \\
& \leq \sum_{y \in \mathcal{Y}, z \in \mathcal{Z}} |R_z \cap \mathcal{X} \times \{y\}|_D \cdot \log(\mathrm{cut}_P(R_z \cap \mathcal{X} \times \{y\})) \\
& \leq \log(\mathrm{avg}_D \, \mathrm{PAR}^{\mathrm{sub}}(P)),
\end{aligned}$$

The first inequality holds by simple algebra. The second inequality holds because, for any $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$, $\Pr[\mathbf{y} = y, F(\mathbf{x}, \mathbf{y}) = z] = |R_z \cap \mathcal{X} \times \{y\}|_D$ and $H(\Pi_P(\mathbf{x}, \mathbf{y})|\mathbf{y} = y, F(\mathbf{x}, \mathbf{y}) = z) \leq \log(\mathrm{cut}_P(R_z \cap \mathcal{X} \times \{y\}))$. The final inequality follows from concavity of logarithm and Proposition 6.1.4. □

Hence, one can use lower bounds on PRIV to derive lower bounds for average-case PAR. For example, recall the *disjointness* function DISJ : $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, which on inputs $x, y \in \{0,1\}^n$ is defined to be one if $\{i \in [n] \ : \ x_i = y_i = 1\}$ is empty and zero otherwise. Klauck [Kla02] shows that for any protocol P for the disjointness problem, $\mathrm{PRIV}_D(P) \geq \Omega(\sqrt{n}/\log n)$, where D is uniform on strings of hamming weight $\sqrt{n}$. Using the above lower bound, we immediately obtain $\mathrm{avg}_D\, \mathrm{PAR}^{\mathrm{sub}}(P) \geq 2^{\Omega(\sqrt{n}/\log n)}$ for any protocol P for DISJ.

There are two other well studied measures that are closely related to our average-case PAR: the *external* and *internal information cost* ($\mathrm{IC}^{\mathrm{ext}}$ and $\mathrm{IC}^{\mathrm{int}}$, resp.) that we briefly discussed in Section 2.2.6. The external information cost was defined in [CWYS01] where the internal cost was also used implicitly. Later, using this measure, Bar-Yossef et al. [BYJKS04] obtained $\Omega(n)$ lower bounds on the randomized communication complexity of DISJ_n. The internal information cost was formalized in [BBCR10]. For a protocol P for function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ and a distribution D on $\mathcal{X} \times \mathcal{Y}$, they are defined respectively as follows:

$$\mathrm{IC}^{\mathrm{ext}}_D(P) = I(\mathbf{x}, \mathbf{y} : \Pi_P(\mathbf{x}, \mathbf{y}))$$

$$\mathrm{IC}^{\mathrm{int}}_D(P) = I(\mathbf{x} : \Pi_P(\mathbf{x}, \mathbf{y})|\mathbf{y}) + I(\mathbf{y} : \Pi_P(\mathbf{x}, \mathbf{y})|\mathbf{x}).$$

As one can see the internal information cost is closely related to the privacy measure PRIV of Klauck. The only substantial difference is that PRIV is conditioned on the value of the function whereas $\mathrm{IC}^{\mathrm{int}}$ is not. When f is a boolean function, they are asymptotically identical.

Proposition 6.1.6. *For any probability distribution D on $\mathcal{X} \times \mathcal{Y}$ and any protocol P for a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$:*

$$\mathrm{PRIV}_D(P) - \log|\mathcal{Z}| \leq \mathrm{IC}_D^{\mathrm{int}}(P) \leq 2 \cdot (\mathrm{PRIV}_D(P) + \log|\mathcal{Z}|).$$

The proposition follows from Claim 2.1.2.

6.2 Vickrey Auction

Vickrey auction (also known known as 2nd price auction) arises in mechanism design, and is a canonical example of a *truthful* mechanism: neither player has incentive to cheat, as long as the auction is computed correctly. Let's recall the definition from the introduction of this chapter. For a positive integer n, the n-bit Vickrey auction is defined as $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z} \times \{A, B\}$ where $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{1, 2, \ldots, 2^n\}$ and

$$F(x, y) = \begin{cases} (x, B) & \text{if } x \leq y, \\ (y, A) & \text{if } y < x. \end{cases}$$

Two players, Alice and Bob, have private values x and y, respectively. These private values indicate the amount of money that the item is worth to each of them. If $x \leq y$, then Bob wins, and the price that he pays is x. (Thus, $F(x, y) = (x, B)$ means that Bob wins and pays x for the item.) Similarly, if $x > y$, then Alice wins, and the price that she pays is y. Vickrey auction remains truthful for more than two players, but is not computable with perfect privacy (PAR $= 1$) for more than two players [BS08].

Perfect privacy for two-player Vickrey auction is achieved by the successive English bidding protocol, in which bids start at 1 and increase by 1 in each round, and the first player to drop out of bidding reveals his entire private value. Note that this incurs no loss of privacy, since that value is part of the function output. This protocol has cost 2^{n+1} for the n-bit Vickrey auction, and is known to be the only protocol which obtains perfect privacy PAR $= 1$ for Vickrey auction.

Theorem 6.2.1. *[Kus89] Perfect privacy for two-player n-bit Vickrey auction is only achievable by the 2^{n+1}-cost English bidding protocol.*

Notice that the range of F is of size 2^{n+1} and that F is surjective, so that there must be at least 2^{n+1} distinct leaves in any protocol tree for F. Thus any protocol for F requires cost at least $n + 1$. An example of a protocol achieving cost $n+1$ is the bisection protocol that proceeds by binary search on an interval containing the smaller input [FJS10a]. However, the bisection protocol obtains PAR $= 2^n$, the worst possible loss of privacy for this function.

Although the bisection protocol loses a lot of privacy in the worst-case, it achieves exponentially better privacy in the average-case.

Proposition 6.2.2. *Let P denote the bisection protocol for the n-bit Vickrey auction. For any probability distribution D on $[2^n] \times [2^n]$, we have*

$$\operatorname{avg}_D \mathrm{PAR}(P) \leq n + 1.$$

Proof. Each region R of the n-bit Vickrey auction is covered by at most $n+1$ rectangles induced by the bisection protocol, i.e., $\mathrm{cut}_P(R) \leq n + 1$. The claim follows by Proposition 6.1.4. □

Let's now make a couple of observations regarding the subjective vs objective PARs of the Vickrey auction. Objective and subjective worst-case PAR coincide because all regions are rectangles with width or height one (we omit the trivial proof):

Lemma 6.2.3. *Let P be any protocol for the Vickrey auction. Then $\mathrm{PAR}(P) = \mathrm{PAR}^{\mathrm{sub}}(P)$.*

A similar equivalence holds for average-case PAR under the uniform distribution:

Lemma 6.2.4. *Let P be any protocol for the n-bit Vickrey auction and U be the uniform probability distribution on $[2^n] \times [2^n]$. Then*

$$\mathrm{avg}_U \, \mathrm{PAR}^{\mathrm{sub}}(P) \leq \mathrm{avg}_U \, \mathrm{PAR}(P) \leq 2 \, \mathrm{avg}_U \, \mathrm{PAR}^{\mathrm{sub}}(P).$$

Proof. To prove the relationship for the average-case PAR, consider an input $(x, y) \in [2^n] \times [2^n]$. If $x \leq y$ then $R_{x,y} \cap \{x\} \times \mathcal{Y} = R_{x,y}$ and $R_{x,y} \cap \mathcal{X} \times \{y\} = \{(x, y)\}$. If $x > y$ then $R_{x,y} \cap \{x\} \times \mathcal{Y} = \{(x, y)\}$ and $R_{x,y} \cap \mathcal{X} \times \{y\} = R_{x,y}$. Identically for $P_{x,y}$ instead of $R_{x,y}$. Hence if $x > y$,

$$\frac{|R_{x,y} \cap \mathcal{X} \times \{y\}|}{|P_{x,y} \cap \mathcal{X} \times \{y\}|} = \frac{|R_{x,y}|}{|P_{x,y}|}$$

and

$$\frac{|R_{x,y} \cap \{x\} \times \mathcal{Y}|}{|P_{x,y} \cap \{x\} \times \mathcal{Y}|} = 1 \leq \frac{|R_{x,y}|}{|P_{x,y}|}.$$

On the other hand, if $x \leq y$, then

$$\frac{|R_{x,y} \cap \mathcal{X} \times \{y\}|}{|P_{x,y} \cap \mathcal{X} \times \{y\}|} = 1 \leq \frac{|R_{x,y}|}{|P_{x,y}|}$$

and

$$\frac{|R_{x,y} \cap \{x\} \times \mathcal{Y}|}{|P_{x,y} \cap \{x\} \times \mathcal{Y}|} = \frac{|R_{x,y}|}{|P_{x,y}|}.$$

Thus, $\mathrm{avg}_U \, \mathrm{PAR}^{\mathrm{sub}}(P) \leq \mathrm{avg}_U \, \mathrm{PAR}(P)$. For the upper bound

$$\sum_{x,y} \frac{1}{4^n} \cdot \frac{|R_{x,y}|}{|P_{x,y}|} = \sum_{x>y} \frac{1}{4^n} \cdot \frac{|R_{x,y} \cap \mathcal{X} \times \{y\}|}{|P_{x,y} \cap \mathcal{X} \times \{y\}|} + \sum_{x \leq y} \frac{1}{4^n} \cdot \frac{|R_{x,y} \cap \{x\} \times \mathcal{Y}|}{|P_{x,y} \cap \{x\} \times \mathcal{Y}|}.$$

Hence, $\mathrm{avg}_U \, \mathrm{PAR}(P) \leq 2 \, \mathrm{avg}_U \, \mathrm{PAR}^{\mathrm{sub}}(P)$. □

6.3 Main Results: PAR Lower Bounds

6.3.1 Worst-Case PAR Lower Bound for Vickrey Auction

The two protocols (English bidding protocol and bisection protocol) discussed in the previous section suggest that any protocol computing Vickrey auction should have a trade-off between cost and privacy. Protocol steps which resemble those of the English bidding protocol partition the inputs in an unbalanced way, so that most inputs follow one branch of the protocol tree, and few inputs follow the other branch. Such steps preserve privacy but do not make much progress (in an imbalanced partition, on the larger side the protocol still has a lot of work to do in order to compute the function). On the other hand, protocol steps that resemble those of the bisection protocol (binary search), partition the inputs in a nearly balanced way. Such steps make good progress, but are bad for privacy (dividing the remaining inputs in half increases the PAR by a factor of 2). This is the high-level intuition behind the proof of the worst-case PAR lower bound for Vickrey auction.

Theorem 6.3.1. *For every n and p, $2 \leq p \leq n/4$, any deterministic protocol for two-player Vickrey auction with communication cost (length) less than $n2^{\frac{n}{4p}-5}$ obtains worst-case PAR at least 2^{p-2}.*

Here the variable p serves as a parameter, explicitly linking the protocol length to the achievable PAR. For instance, if we put $p = \sqrt{n}$, then we conclude by Theorem 6.3.1 that either the protocol communicates $2^{\Omega(\sqrt{n})}$ bits in the worst case, or the worst-case privacy loss is $2^{\Omega(\sqrt{n})}$. This theorem shows that for Vickrey auction, there is an inherent trade-off between communication complexity and privacy. Note that the trade-off holds for both the objective and subjective worst-case PAR due to Lemma 6.2.3.

In the rest of this section, we prove the theorem.

We will assume without loss of generality that in the protocol, the players take turns and send one bit per message. Any protocol can be put into this form by at most doubling the length of the protocol. Moreover, note that our protocol is assumed to be deterministic and to have zero error.

Let M denote the communication matrix for Vickrey auction (Figure 6.1). Recall from Chapter 2 that every communication protocol can be visualized as a binary tree. Each node v of the tree is associated with a rectangle (submatrix) $T(v) = T_A(v) \times T_B(v) \subseteq \mathcal{X} \times \mathcal{Y}$. The root node r is associated with the entire matrix $T_A(r) \times T_B(r) = \mathcal{X} \times \mathcal{Y} = M$. Each leaf node l is associated with a monochromatic submatrix $T_A(l) \times T_B(l)$. Each internal node v has two children, v_0 and v_1. If the protocol calls for Alice to speak at node v, then the bit sent by Alice at v induces a partition of $T_A(v)$ into two pieces, $T_A(v_0)$ and $T_A(v_1)$. The submatrix associated with v_0 is $T_A(v_0) \times T_B(v)$, and the submatrix associated with v_1 is $T_A(v_1) \times T_B(v)$. Similarly if Bob speaks at node v, then the submatrix associated with v_0 is $T_A(v) \times T_B(v_0)$ and the submatrix associated with v_1 is $T_A(v) \times T_B(v_1)$.

We can define a PAR value for an input (x, y) with respect to a node v of the tree: for any $(x, y) \in T(v)$,

$$\mathrm{PAR}_v(x, y) = \frac{|R_{x,y}|}{|R_{x,y} \cap T(v)|}.$$

If v is a leaf, then for any $(x, y) \in T(v)$, $\mathrm{PAR}_v(x, y) = \mathrm{PAR}(x, y)$. The following simple claim will be useful:

Claim 6.3.2. *$\forall (x, y) \in T(v)$, $\mathrm{PAR}(x, y) \;\geq\; \mathrm{PAR}_v(x, y)$.*

We now describe the strategy of the proof. We are given a protocol for the Vickrey auction and its associated protocol tree. Starting at the root, we will follow a path down the protocol tree. Our decision to go to the left or right child of a node will depend on the behaviour of the protocol at that node. If we simplify the argument a bit, the choice is essentially made as follows. If the protocol, at a particular node, makes an unbalanced partition (e.g., the perfectly private English bidding protocol), then we will choose the child that corresponds to the bigger part. We call this step "useless" since the protocol still has work to do in order to correctly compute this big part. On the other hand, if the protocol makes a balanced partition (e.g., the bisection protocol), we will choose the child that incurs the most privacy loss. By making the appropriate choices to go left or right, we will end up at a node (i.e. a rectangle) such that the inputs in the rectangle will either have large privacy loss $\mathrm{PAR}(x, y) \geq 2^{p-2}$, or will require at least $n2^{\frac{n}{4p}-5}$ bits of communication. The theorem then follows from this.

Let's now get into the details of this strategy. For each node v, we will maintain three sets $S(v), L_A(v), L_B(v) \subseteq [2^n]$, where S stands for "small" and L stands for "large". We define

$$L_A(v) = T_A(v) \cap [2^n] \backslash [2^{n-p}],$$

$$L_B(v) = T_B(v) \cap [2^n] \backslash [2^{n-p}],$$

and

$$S(v) = T_A(v) \cap T_B(v) \cap [2^{n-p}].$$

So $L_A(v)$ and $L_B(v)$ track the large inputs in v, and $S(v)$ tracks the small inputs (except we don't distinguish between players A and B). At the root r, $S(r) = [2^{n-p}]$ and $L_A(r) = L_B(r) = [2^n] \backslash [2^{n-p}]$.

The submatrix $S(v) \times S(v)$ plays an important role in our strategy and it has two properties to keep in mind. First, observe that every entry is contained in a large region (it sits at the upper-left corner of M) and therefore these entries have the potential to incur a big (exponential) privacy loss. Second, the submatrix $S(v) \times S(v)$ has exactly the same structure/shape as the original Vickrey auction matrix M, albeit smaller (but still large enough). Because of these two properties of $S(v) \times S(v)$, we will actually keep track of these inputs and we will base our decision to choose the left or right child of a node v by looking at whether the protocol partitions $S(v)$ in a balanced way.

The purpose of the sets $L_A(v)$ and $L_B(v)$ is to help us keep track of the privacy loss of inputs in $S(v) \times S(v)$. In particular, for any (x, y) in $S(v) \times S(v)$ where Player A wins (i.e. $x > y$), privacy loss for (x, y) increases as the size of $L_A(v)$ decreases. Similarly for those inputs where Player B wins, privacy loss increases as the size of $L_B(v)$ decreases. More precisely, for any (x, y) in $S(v) \times S(v)$ with $x > y$,

$$\mathrm{PAR}(x, y) = \frac{|R_{x,y}|}{|P_{x,y}|} \geq \mathrm{PAR}_v(x, y) = \frac{|R_{x,y}|}{|R_{x,y} \cap T(v)|} \geq \frac{2^n - 2^{n-p}}{|L_A(v)| + 2^{n-p}}, \quad (6.1)$$

where the last inequality holds because $|R_{x,y}| \geq 2^n - 2^{n-p}$ and $|R_{x,y} \cap T(v)| \leq |L_A(v)| + 2^{n-p}$. Similarly, for any (x, y) in $S(v) \times S(v)$ with $x \leq y$,

$$\mathrm{PAR}(x, y) \geq \mathrm{PAR}_v(x, y) \geq \frac{2^n - 2^{n-p}}{|L_B(v)| + 2^{n-p}}.$$

Let's now make precise how we traverse down the protocol tree. There are two cases, depending on whether it is Player A's or Player B's turn to send a message. We will first describe the case where at node v, it is Player A's turn to speak. Player A sends Player B some bit b which partitions her inputs $T_A(v)$ into two pieces. Since $S(v)$ and $L_A(v)$ are always subsets of $T_A(v)$, this induces a partition of $S(v)$ into $S^0(v)$ and $S^1(v)$ and $L_A(v)$ into $L_A^0(v)$ and $L_A^1(v)$. Let $\alpha = 2^{\frac{-n}{4p}}$. We determine if a step made *progress* or was *useless* in the following way:

- If $\alpha|S(v)| \leq |S^0(v)| \leq (1-\alpha)|S(v)|$ (hence $\alpha|S(v)| \leq |S^1(v)| \leq (1-\alpha)|S(v)|$), then we say this step made **progress** on $S(v)$. In this case, the set $S(v)$ is partitioned into roughly balanced pieces. Select i such that $|L_A^i(v)| \leq \frac{1}{2}|L_A(v)|$.
- Otherwise, pick i such that $|S^i(v)| \geq (1-\alpha)|S(v)|$. In this case, we call it a **useless** step.

The second case is when it is Player B's turn to speak and it is completely analogous to the previous case. Now $T_B(v)$ is partitioned into two pieces, inducing a partition of $S(v)$ into $S^0(v)$ and $S^1(v)$, and a partition of $L_B(v)$ into $L_B^0(v)$ and $L_B^1(v)$. We pick i as above, but with L_A^i replaced with L_B^i.

We keep traversing the protocol tree as described above until one of the two events happens for the first time:

- Player A (or Player B) has made p progress steps, so $L_A(v)$ (or $L_B(v)$) has been halved at least p times.
- The strategy reaches a leaf node, and can go no further.

The theorem will now follow from the two lemmas below.

Lemma 6.3.3. *Let our strategy reach node v and find Player A (or Player B) took p progress steps on the way. Then, for each $(x, y) \in S(v) \times S(v)$ such that $x > y$ (or $x \le y$) $\mathrm{PAR}(x, y) \ge 2^{p-2}$.*

Lemma 6.3.4. *If our strategy reaches a leaf node v without Player A or Player B taking p progress steps, then for every $(x, y) \in T(v)$, the protocol communicates at least $\frac{\ln 2}{4}(n - 2p)2^{n/4p}$ bits.*

Proof of Lemma 6.3.3. Let r be the root node of our protocol tree. Note that $|L_A(r)| = 2^n - 2^{n-p}$. Let φ be the path in the protocol tree from r to v that our strategy chooses such that player A takes p progress steps along φ. Consider any pair of adjacent nodes u, w in path φ such that Player A makes progress in going from u to w. Then, we know $|L_A(w)| \le \frac{1}{2}|L_A(u)|$. Hence,

$$|L_A(v)| \le \frac{1}{2^p}|L_A(r)| = \frac{2^n - 2^{n-p}}{2^p}.$$

For inputs $(x, y) \in S(v) \times S(v)$ on which Alice wins, Inequality (6.1) implies

$$\mathrm{PAR}(x, y) \ge \frac{2^n - 2^{n-p}}{\frac{2^n - 2^{n-p}}{2^p} + 2^{n-p}} \ge 2^{p-2}$$

The analysis when Player B makes p progress steps proceeds very similarly. $\square$

Proof of Lemma 6.3.4. The strategy reaches a leaf node v traversing a path φ, and $|S(v)| = 1$. (If $|S(v)| > 1$, then there is more than one possible answer, and so the computation is not yet finished.) In this case, Player A and Player B each took fewer than p progress steps. Let q be the total number of useless steps followed to get to v. (The protocol is at most $2p + q$ long.) On each progress step (u, w) in path φ, by definition, $|S(w)| \geq \alpha|S(u)|$. On each useless step (u, w), $|S(w)| \geq (1 - \alpha)|S(u)|$. This gives a lower bound on the size of set $S(v)$. Hence $|S(v)| \geq 2^{n-p}\alpha^{2p}(1 - \alpha)^q$.

Assume that $q < \frac{\ln 2}{4}(n - 2p)2^{\frac{n}{4p}}$. The calculation below shows that $|S(v)| > 1$, thus deriving a contradiction to the fact that v is a leaf node where the protocol ends.

$$\begin{aligned}
|S(v)| &\geq 2^{n-p}\alpha^{2p}(1-\alpha)^q \\
&> 2^{n-p}(2^{-\frac{n}{4p}})^{2p}(1 - 2^{-\frac{n}{4p}})^{\frac{\ln 2}{4}(n-2p)2^{\frac{n}{4p}}} \\
&= 2^{\frac{n}{2}-p}(1 - 2^{\frac{-n}{4p}})^{\frac{\ln 2}{4}(n-2p)2^{\frac{n}{4p}}} \\
&> 2^{\frac{n}{2}-p}e^{-2^{-\frac{n}{4p}}\frac{\ln 2}{2}(n-2p)2^{\frac{n}{4p}}} \\
&= 2^{\frac{n}{2}-p}e^{-(\ln 2)(\frac{n}{2}-p)} \\
&= 1,
\end{aligned}$$

where for the last inequality, we used the fact that $(1 - x) > e^{-2x}$ for $x \in (0, 1/2]$. $\square$

Thus we know that either there is an input with privacy loss at least 2^{p-2} or an input with communication at least $\frac{\ln 2}{4}(n - 2p)2^{n/4p} \geq \frac{n}{12}2^{n/4p}$ bits. Since we might have doubled the communication cost by assuming that the players take turns to send a bit, we obtain the resulting lower bound.

6.3.2 Average-Case PAR Lower Bound for Vickrey Auction

In this section we prove a lower bound on the average-case PAR of Vickrey auction under the uniform distribution. The restriction to uniform distribution is not surprising since if the distribution is concentrated say on a single input, one should not expect large loss of privacy. Also note that by Lemma 6.2.4 our bound applies to both objective and subjective average-case PAR.

Theorem 6.3.5. *For all $n, r \geq 1$, any deterministic protocol of length at most r for the two-player n-bit Vickrey auction problem has average-case PAR at least $\Omega(\frac{n}{\log(r/n)})$ (over the uniform distribution of inputs).*

This bound is asymptotically tight since the $\frac{n}{r}$-bisection protocol achieves asymptotically the same upper-bound (see [FJS10a]).

The rest of this section is devoted to the proof of Theorem 6.3.5.

Proposition 6.1.4 characterizes the average-case PAR as the weighted sum of $\text{cut}_P(R)$ over all regions R of the function (recall that $\text{cut}_P(R)$ denotes the number of rectangles within R that the protocol has induced). We will lower bound average-case PAR by summing only over the "large" regions. In particular, we will sum only over regions $R_{x,y}$ with $x, y \leq 2^{n-1}$. These regions together cover 3/4 the area of $\mathcal{X} \times \mathcal{Y}$, and each has size between 2^{n-1} and 2^n (i.e., they all have the same weight up to a factor of 2). Let's call this collection of regions $\mathcal{L}$. Then,

$$\text{avg}_U \, \text{PAR}(P) \geq \frac{2^{n-1}}{4^n} \sum_{R \in \mathcal{L}} \text{cut}_P(R). \tag{6.2}$$

The sum above suggests that counting the number of induced rectangles of a protocol that computes the Vickrey auction is the key to lower bounding the average-case PAR. To count the number of induced rectangles, we will abstract it away into the *Ball Partition Problem.* As we will see shortly, a lower bound on the Ball Partition Problem will then yield a lower bound on the average-case PAR of Vickrey auction.

Definition 6.3.6 (Ball Partition Problem)**.** For integers $N, r \geq 1$, there are N balls and r rounds. All of the balls begin in one big set. In each round, the balls in each current set are partitioned into (at most) two new sets. The cost of partitioning the balls in any set S into sets S_1 and S_2 is $\min(|S_1|, |S_2|)$. After r rounds, each of the N balls shall be in a singleton set. The total cost of the game is the sum of the cost, over all r rounds, of every partition made during each round. We denote the minimal possible cost by $B(N, r)$.

The interesting values of r lie in a particular range. For $r < \log_2 N$, the game cannot be finished at any cost. For $r > N$, the game can easily be finished with minimal cost $B(N, r) = N - 1$: cut away 1 ball from the largest set at every round. However, for intermediate values $\log N \leq r \leq N$, one might ask: what is the smallest possible cost c achievable in r rounds?

Theorem 6.3.7. *For the Ball Partition Problem,* $B(N, r) \geq \frac{N \log N}{4 \log(\frac{4r}{\log N})}$.

The above lower bound is asymptotically optimal. We will prove this at the end of this section (Proposition 6.3.10) to not break the flow.

The following lemma states that the number of rectangles induced by a cost r protocol that computes the $(\log N)$-bit Vickrey auction is lower bounded by $B(N, r)$.

Lemma 6.3.8. *Let $N, r \geq 1$ be integers and let $B(N, r)$ be the minimal cost of the Ball Partition Problem on N balls in r rounds. Then for any deterministic r-bit protocol P for $(\log N)$-bit Vickrey auction,*

$$\sum_{R \in \mathcal{R}} \mathrm{cut}_P(R) \geq B(N, r).$$

From Theorem 6.3.7 and Lemma 6.3.8 it is easy to derive the average-case PAR lower bound for Vickrey auction (Theorem 6.3.5). Let P be a cost r

protocol that solves the n-bit Vickrey auction and let $N = 2^n$. Consider the upper left quarter of the Vickrey auction matrix which corresponds to the $(n-1)$-bit Vickrey auction. The protocol P solves the $(n-1)$-bit Vickrey auction as a subproblem and therefore using Lemma 6.3.8 we have

$$\sum_{R \in \mathcal{L}} \mathrm{cut}_P(R) \geq B(N/2, r).$$

Plugging this into Inequality (6.2), we obtain

$$\mathrm{avg}_U \, \mathrm{PAR}(P) \geq \frac{1}{2N} B(N/2, r).$$

Now applying the lower bound from Theorem 6.3.7 for $B(N/2, r)$ gives us the desired lower bound on the average-case PAR.

To complete the proof of Theorem 6.3.5, we now present the proofs of Lemma 6.3.8 and Theorem 6.3.7.

Proof of Lemma 6.3.8. A cost r protocol for the $(\log N)$-bit Vickrey auction gives us a solution to the Ball Partition Problem with N balls and r rounds. Let's first see why this is true, and then argue why the cost of the solution is at most the number of protocol-induced rectangles. The statement of the lemma then follows.

We will consider the diagonal elements of the Vickrey auction matrix as "balls" and view the protocol as partitioning these diagonal elements. Note that at the end of the protocol, each diagonal element must be in a separate protocol-induced rectangle as they each belong to a different region. Recall the following notation used in the proof of Theorem 6.3.1. A protocol is associated with a protocol tree where each node v corresponds to a combinatorial rectangle $T(v) = T_A(v) \times T_B(v) \subseteq \mathcal{X} \times \mathcal{Y}$. Define $D_v = T_A(v) \cap T_B(v)$ to be the diagonal elements contained in the rectangle corresponding to v. At the root, D_v is equal to $[N]$. If the root has children v_1 and v_2, then D_v gets partitioned into D_{v_1} and D_{v_2}. Continuing in this fashion, the diagonal elements travel down the protocol tree. Whenever there is a node v with children v_1 and v_2 such that $D_{v_1} \neq \emptyset \neq D_{v_2}$, the diagonal elements (balls) get partitioned further. Since the tree has height (cost) r and at the end each diagonal element ends up in a different leaf node, the protocol provides a solution to the Ball Partition Problem for N balls and r rounds.

Now let's argue why the cost of this solution is at most the number of induced rectangles. We will start by analysing the first step of the protocol and see how many rectangles it creates. This will be sufficient to derive a lower bound on the number of induced rectangles in terms of $B(N, r)$. Initially we have $2N$ rectangles, one for each region. Let's assume Player A first sends a bit (i.e., horizontally cuts the matrix). How many new rectangles does this create? As mentioned above, at the root, D_v is equal to $[N]$ and its two children v_1 and v_2 is such that $D_v = D_{v_1} \dot{\bigcup} D_{v_2}$. Let $x_1 = \max(D_{v_1})$ and $x_2 = \max(D_{v_2})$. Assume that $x_1 < x_2$. For every $y \in D_{v_1}, y \neq x_1$, $(x_1, y) \in R_{y+1,y} \cap T(v_1)$ and also $(x_2, y) \in R_{y+1,y} \cap T(v_2)$. So for each $y \in D_{v_1}$, $R_{y+1,y}$ is "cut" into two and there are $|D_v| - 1$ such y's. If $x_2 < x_1$, we would switch the roles of v_1 and v_2. Hence we conclude that this step of the protocol creates at least $\min(|D_{v_1}|, |D_{v_2}|) - 1$ *new* rectangles on top of the $2N$ we started with.

If at the first step, Player B speaks, the argument is similar. Let $y_1 = \max(D_{v_1})$ and $y_2 = \max(D_{v_2})$, and assume $y_1 < y_2$. Then for every $x \in D_{v_1}$, $(x, y_1) \in R_{x,x} \cap T(v_1)$ and also $(x, y_2) \in R_{x,x} \cap T(v_2)$. Thus in this case one does not even lose the -1 additive term.

Each node of the protocol tree that splits into two potentially creates new induced rectangles. Suppose that we are at a node v (not necessarily the root) that splits into v_1 and v_2 so that $D_{v_1} \neq \emptyset \neq D_{v_2}$. Note that $D_v \times D_v$ is a smaller version of the Vickrey auction that the protocol still needs to solve. And via the same argument that we just presented, we know that the split of v into v_1 and v_2 creates at least $\min(|D_{v_1}|, |D_{v_2}|) - 1$ new

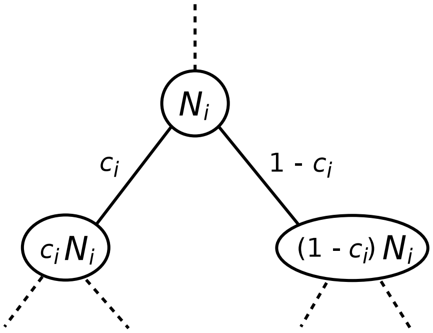

Figure 6.2: An arbitrary node in the ball-partitioning tree.

rectangles. There are exactly $N-1$ nodes v that split into v_1 and v_2 with $D_{v_1} \neq \emptyset \neq D_{v_2}$. Thus overall, the number of induced rectangles is at least $2N + B(N,r) - (N-1) = B(N,r) + N + 1$. We ignore the additive $(N+1)$ term and obtain the desired lower bound on the number of induced rectangles.

□

Proof of Theorem 6.3.7. We will examine the entropy of the partitions at each round. This permits an abstraction away from a particular ball-partition instance, in order to obtain general properties. This will lead to a lower bound on the objective function $B(N,r)$, the cost of the Ball Partition Problem.

It will be useful to associate with the Ball Partition Problem in r rounds a full binary tree of depth r where each set obtained at round t is associated to a distinct node at level t, and the remaining nodes are associated with the empty set. The association should be so that a node associated with a set S has its children associated with sets S_1 and S_2 obtained from S during the partitioning. We label each node i, by the size of the associated set, N_i, and we label edges by the fraction of balls that travel "over" that edge from the parent to the child node. (See Figure 6.2: a node labelled N_i with children labelled $c_i N_i$ and $(1-c_i)N_i$ will have edges to those children labelled c_i and $1-c_i$, respectively.)

The tree's root node is labelled N; each leaf is labelled 1 or 0. (The 0 leaves are a result of assuming the binary tree is full; if some ball is partitioned into a singleton set in round $i < r$, then in each subsequent round it is "partitioned" into two sets: the singleton set and the empty set.)

Remark. At each level of the tree, the sum of the node labels equals N. Thus the sum of labels of all the non-leaf nodes in the tree is rN.

Consider the path followed by any ball b from the root to a leaf. It traverses edges labelled d_1^b, d_2^b, $\ldots$, d_r^b, where $\prod_{i=1}^r d_i^b = \frac{1}{N}$.

Multiplying this number for all balls gives a nice symmetrization which is true for all trees representing solutions to the Ball Partition Problem.

$$\left(\frac{1}{N}\right)^N = \prod_{b \text{ a ball}} \prod_{i=1}^r d_i^b \tag{6.3}$$

Consider some non-leaf node i of the tree, with edges to its children labelled c_i and $1-c_i$ (Figure 6.2). Together, these edges contribute $(c_i)^{c_i N_i}(1-c_i)^{(1-c_i)N_i}$ to the right-hand side of equation (6.3). (If $c_i = 0$ this term equals 1 by definition.) Without loss of generality, assume each $c_i \leq 1/2$. Equation (6.3) can be rewritten as:

$$\begin{aligned} \left(\frac{1}{N}\right)^N &= \prod_{\text{non-leaf node } i} (c_i)^{c_i N_i}(1-c_i)^{(1-c_i)N_i} \\ -N\log N &= \sum_i N_i(-H(c_i)) \end{aligned} \tag{6.4}$$

Here $H(x) = x \log \frac{1}{x} + (1-x)\log\frac{1}{1-x}$ is the binary entropy of x.

Since the leaf nodes are not included in the sum, $\sum_{\text{non-leaf node } i} N_i = rN$ (by the **Remark** above). Let $c = \sum_i \frac{c_i N_i}{rN}$ be the average cost of a cut in the Ball Partition Problem. Then the cost of the entire tree is $B(N,r) = crN$. Since H is concave, $\sum_i \frac{N_i}{rN} H(c_i) \leq H(\sum_i \frac{c_i N_i}{rN}) = H(c)$.

$$N \log N = rN \sum_i \frac{N_i}{rN} H(c_i) \leq rNH(c) \tag{6.5}$$

For the sake of contradiction, suppose that the cost of the tree $B(N, r) = crN < \frac{N \log N}{4 \log(\frac{4r}{\log N})}$. Then the average cost of a cut is $c < \frac{\log N}{4r \log(\frac{4r}{\log N})}$. This c can be rewritten as $c = \frac{x}{-\log x}$ for $x = \frac{\log N}{4r}$. Combining equation (6.5) and Lemma 6.3.9 (below),

$$\frac{\log N}{r} \leq H(c) = H\Big(\frac{x}{-\log x}\Big) < 4x = 4\frac{\log N}{4r} = \frac{\log N}{r}$$

The inequality makes this a contradiction. Therefore every tree of depth $\leq r$ must incur cost $\geq \frac{N \log N}{4 \log(\frac{4r}{\log N})}$. □

Lemma 6.3.9. *For* $0 < x \leq \frac{1}{2}$*, the binary entropy* $H\big(\frac{x}{-\log x}\big) < 4x$.

Proof. For $0 < x \leq \frac{1}{2}$, $\log \frac{1}{x} \geq 1$ so clearly $0 < \big(\frac{x}{-\log x}\big) \leq \frac{1}{2}$. Let $y = \frac{x}{-\log x}$.

Expanding,

$$H(y) = y \log \frac{1}{y} + (1-y) \log \frac{1}{1-y}$$

For $0 < y \leq \frac{1}{2}$, it is not difficult to see that $-\log(1-y) \leq 2y$ and $1 - y < 1$.

$$H(y) \leq y \log \frac{1}{y} + (1-y)2y < y \log \frac{1}{y} + 2y$$

Substituting for y and expanding,

$$H\left(\frac{x}{\log \frac{1}{x}}\right) \;<\; x\left(\frac{\log \log \frac{1}{x}}{\log \frac{1}{x}}\right) + x\left(\frac{\log \frac{1}{x}}{\log \frac{1}{x}}\right) + 2x\left(\frac{1}{\log \frac{1}{x}}\right)$$

Examination reveals that for $0 < x \leq \frac{1}{2}$, the parenthesized coefficients are each ≤ 1. Hence $H(\frac{x}{\log \frac{1}{x}}) < 4x$. □

This completes the proof of the main theorem of this section. Now, as promised, we show that the lower bound for $B(N, r)$ is asymptotically tight.

Proposition 6.3.10. *Let* N *and* r *be integers such that* $2 \log N \leq r$*. For the Ball Partition Problem,* $B(N, r) \leq O\left(\frac{N \log N}{\log(\frac{r}{\log N})}\right)$.

Proof. Ignoring the rounding issues, at each round we can split each non-singleton set S into two sets of sizes $\alpha|S|$ and $(1-\alpha)|S|$, for $\alpha = (\log N)/r \leq 1/2$. It follows that within r rounds, each set contains at most one element as $(1-\alpha)^r N \leq Ne^{-\alpha r} < 1$. The total cost of the ball partitioning is the sum of sizes of all the smaller sets obtained in each partition. This corresponds to the number of elements in these sets (counting multiplicity). Each element can appear in at most $\log_{1/\alpha} N = (\log N)/\log(r/\log N)$ of the smaller sets as the size of the set containing the element shrinks by factor of α on each such occasion. Hence, the total cost is at most $N \cdot (\log N)/\log(r/\log N)$. Always rounding the size of the smaller set up will introduce a constant factor in the final bound. □

6.3.3 Average-Case PAR Lower Bound for Set Intersection

In this section we will show an average-case PAR lower bound for the *set-intersection* function using the relationship described in Proposition 6.1.6 together with the known lower bounds on internal information cost of DISJ. This proves a conjecture from [FJS10a], which says that the average-case subjective PAR for the set intersection function under the uniform distribution is exponential in n. Function INTERSEC : $\{0,1\}^n \times \{0,1\}^n \to \mathcal{P}([n])$ on inputs $x, y \in \{0,1\}^n$ gives the set $\{i \in [n] \ : \ x_i = y_i = 1\}$.

We make use of the following lower bound on the internal information cost of DISJ. Let ν be the uniform distribution supported on $\{(0,1),(1,0),(0,0)\}$. Let τ be the distribution generated by taking the n-fold product of ν. In other words, τ is the uniform distribution supported on pairs of strings that are disjoint.

Theorem 6.3.11. *[Bra11] Let P be any randomized protocol that computes disjointness* DISJ *with error probability $< 1/3$. Then,* $\mathrm{IC}^{\mathrm{int}}_{\tau}\big(P\big) = \Omega(n)$.

Using the above theorem, we show the following bound for set-intersection.

Theorem 6.3.12. *Let P be any deterministic protocol that computes set intersection* INTERSEC*. Then, for U the uniform distribution,* $\mathrm{PRIV}_U\big(P\big) = \Omega(n)$*.*

Proof. We prove this by contradiction. Assume that we have a protocol P to solve $\mathrm{INTERSEC}_m$ on m-bit inputs with little privacy loss under the uniform distribution. The main idea of the argument is to come up with an appropriate reduction from set disjointness DISJ_n on n bits to set intersection $\mathrm{INTERSEC}_m$. This reduction will need to satisfy the following features: solving intersection on the reduced instance should solve set-disjointness on the original input instance. The reduced instance should not blow up too much in size, i.e., $m = \Theta(n)$. Finally, and most importantly, distribution τ on input instances to set-disjointness should generate (by our reduction) the uniform distribution on set-intersection. This last step seems difficult to do via a deterministic reduction. So we aim to get a workaround as follows.

Let $\mathbf{\Pi}$ be the random variable denoting the transcript generated by P. Then, our assumption on P gives the following for some constant β which we fix at the end:

$$I_U\big(\mathbf{x} : \mathbf{\Pi} \,|\, \mathbf{y}, \mathrm{INTERSEC}(\mathbf{x},\mathbf{y})\big) + I_U\big(\mathbf{y} : \mathbf{\Pi} \,|\, \mathbf{x}, \mathrm{INTERSEC}(\mathbf{x},\mathbf{y})\big) < \beta m.$$

The uniformly distributed pairs of m-bit random strings $(\mathbf{x},\mathbf{y})$ can be alternatively generated by first selecting a random subset $\mathbf{A}$ of $[m]$ where each element is in the set independently with probability 1/4. For each $i \in \mathbf{A}$, we set $(\mathbf{x}_i, \mathbf{y}_i) = (1,1)$. Then, for each coordinate $i \in \mathbf{A}^c = [m] - \mathbf{A}$, $(\mathbf{x}_i, \mathbf{y}_i)$ is picked independently according to ν. Let τ denote the joint distribution $(\mathbf{x},\mathbf{y},\mathbf{A})$ sampled as described. Let $(\mathbf{x},\mathbf{y}|\mathbf{A})$ denote pair of random variables that are distributed according to $\mathbf{x}$,$\mathbf{y}$ conditioned on $\mathbf{A}$ as above, and let the underlying distribution on this pair be denoted by $\tau_{\mathbf{A}}$. Thus, our assumption becomes equivalently:

$$\mathbf{E}_{\mu_A}\left[I_{\tau_{\mathbf{A}}}\big(\mathbf{x} : \mathbf{\Pi} \,|\, \mathbf{y}, \mathbf{A}\big) + I_{\tau_{\mathbf{A}}}\big(\mathbf{y} : \mathbf{\Pi} \,|\, \mathbf{x}, \mathbf{A}\big)\right] < \beta m,$$

where μ_A is the distribution on $\mathbf{A}$. Applying the Chernoff bound on the deviation of $|\mathbf{A}|$ from its expectation, one concludes:

$$\mathbf{E}_{\mu_A}\left[I_{\tau_{\mathbf{A}}}\big(\mathbf{x} : \mathbf{\Pi} \,|\, \mathbf{y}, \mathbf{A}\big) + I_{\tau_{\mathbf{A}}}\big(\mathbf{y} : \mathbf{\Pi} \,|\, \mathbf{x}, \mathbf{A}\big) \,\middle|\, |\mathbf{A}| \leq m/2\right] < \frac{\beta m}{1 - \exp(-\Omega(m))}$$

Thus, there exists some fixed set A of size at most $m/2$ such that

$$I_{\tau_A}\big(\mathbf{x} : \mathbf{\Pi} \,|\, \mathbf{y}, \mathbf{A} = A\big) + I_{\tau^a}\big(\mathbf{y} : \mathbf{\Pi} \,|\, \mathbf{x}, \mathbf{A} = A\big) \; < \; \beta' m. \tag{6.6}$$

This set A is going to provide us with the workaround needed for the deterministic reduction. We define our reduction now with respect to A. Set $n = m - |A| \geq m/2$. Let P' be a protocol that solves set-disjointness as follows: Given two n-bit strings (u, v), protocol P' first embeds u and v naturally into $A^c = [m] - A$. Let the embedded strings be called $x(u)$ and $y(v)$ which each player can generate privately on its own. Then, the players run the protocol P on $\big(x(u), y(v)\big)$. Let J be the intersection set that P returns. Clearly, $\mathrm{DISJ}_n(u, v) = 1$ iff $|J| = |A|$. Finally, note if $(\mathbf{u}, \mathbf{v})$ are generated according to τ, then the mapped strings $\big(\mathbf{x}(\mathbf{u}), \mathbf{y}(\mathbf{v})\big) \sim (\mathbf{x}, \mathbf{y} | \mathbf{A} = A)$. Hence, (6.6) implies that $\mathrm{IC}_\tau(P) \leq \beta' m \leq 2\beta' n$. By setting β' to be a small enough constant, we derive a contradiction to Theorem 6.3.11. This completes the argument. □

By using Theorem 6.1.5, this immediately yields the following theorem, conjectured by Feigenbaum et al. [FJS10b].

Theorem 6.3.13. *For all $n \geq 1$, and any protocol P computing the set-intersection* INTERSEC *on n bits, the average-case subjective PAR is exponential in n under the uniform distribution:* $\mathrm{avg}_U\, \mathrm{PAR}^{\mathrm{sub}}(P) = 2^{\Omega(n)}$.

6.4 Conclusion

The study of privacy is an important area of practical and theoretical computer science. From a theoretical perspective, the first step is to find the right formulation for it. Usually there is not *one* right definition because the right definition depends on the particular aspects of privacy that one is interested in. For example, different privacy models stem from the question: privacy from whom? Additional differences can arise from different considerations of what knowledge should be kept private (e.g., does the output of the function contribute to privacy loss?).

In this chapter we have discussed the following privacy measures:

$$\mathrm{PAR}, \mathrm{PAR}^{\mathrm{sub}}, \operatorname{avg}\mathrm{PAR}, \operatorname{avg}\mathrm{PAR}^{\mathrm{sub}}, \mathrm{PRIV}, \mathrm{IC}^{\mathrm{int}}, \mathrm{IC}^{\mathrm{ext}}\,.$$

Notions like PAR, avg PAR and $\mathrm{IC}^{\mathrm{ext}}$ measure privacy from an eavesdropper whereas $\mathrm{PAR}^{\mathrm{sub}}$, $\operatorname{avg}\mathrm{PAR}^{\mathrm{sub}}$, $\mathrm{IC}^{\mathrm{int}}$ and PRIV measure privacy from other participants. Furthermore, recall that the main difference between $\mathrm{IC}^{\mathrm{int}}$ and PRIV is that PRIV does not consider the knowledge of the function value as a privacy loss.

There are aspects of privacy that the above measures do not capture. For example, every bit of the input may not be equally private (the most significant bit of a salary is much more privacy revealing than the least significant bit). Which bits or parts of the input are private can depend on the context of the problem as well as other participants.

Regardless of these difficulties in capturing the “right” definition of privacy, we believe that worst-case and average-case privacy approximation ratios studied in this chapter are intuitive, useful and in many situations the right measures of privacy in the setting of communication complexity. Therefore understanding these measures in more depth (e.g., obtaining privacy vs communication cost trade-offs for other functions, as well as better understanding the relationship of these measures with other privacy measures) is an important challenge. There are interesting directions that one can pursue along these lines:

- This chapter was only concerned with deterministic protocols that compute the function correctly on all inputs. What about protocols that make a certain fraction of error? What about randomized protocols?
- Can we develop general techniques for obtaining privacy vs communication cost trade-offs?
- What other notions of approximate privacy are natural and useful to study? How do these new notions compare to the other ones?

CHAPTER 7

Conclusion

Computation is a fundamental aspect of our universe. In the last 70 years we finally took the steps to formalize the notion of computation and rigorously understand it. This quest has played a crucial role in the invention of first computers which have evolved into indispensable devices that we now use everyday. Furthermore, not only has it added a new perspective to old philosophical questions, it has also created new ones just as deep and exciting. Computational complexity theory plays one of the starring roles in all this development with its connections to the nature of mathematical knowledge, artificial intelligence, foundations of quantum mechanics, closed timelike curves, etc... (see the survey by Aaronson titled "Why Philosophers Should Care About Computational Complexity" [Aar11]).

Computational complexity theory has matured significantly over the last several decades even though we still seem as far away from answering questions like $\mathsf{P} \overset{?}{=} \mathsf{NP}$ as we were back then. As discussed in the Introduction, one of the highlights of this young research area has been communication complexity, which has grown into a picklock for the field. With applications in circuit complexity, proof complexity, machine learning, game theory, data structures, pseudorandom generators and other areas, communication complexity is a true hero of computational complexity theory and theoretical computer science in general. In this thesis we presented our contributions to communication complexity in three main parts in which we studied three different aspects of communication complexity.

In the first part, we studied the so called 'number on the forehead' (NOF) model of multiparty communication complexity. This is one of the most important models with applications to circuit complexity, Ramsey theory, pseudorandom generators and branching programs. Without a doubt, the holy grail in this area is breaking the $\log n$ barrier, i.e., exhibiting an explicit function that is hard in the NOF model when the number of players is $\log n$. In Chapter 3 we showed that the NOF model with $\log n$ many players is more powerful than previously thought. We ruled out some candidates previously considered to break the $\log n$ barrier by presenting an efficient protocol for the set of composed functions $\textsc{sym} \circ g$, where $\textsc{sym}$ is an arbitrary symmetric function and g is any function. Furthermore, we presented an interesting application of this result to Ramsey theory in Chapter 4.

Composed functions have a special role in communication complexity since most of the functions that drew significant interest have this structure. In particular, the king and queen of communication complexity *generalized-inner-product* GIP $= \textsc{mod}_2 \circ \textsc{and}$ and *disjointness* DISJ $= \textsc{nor} \circ \textsc{and}$ are composed. Given a composed function $f \circ g$, the most basic and central question to study is what combinations of f and g lead to hard or easy communication functions. The focus in the literature so far has been to fix g to be a natural function like $\textsc{and}$ and then figure out what kinds of f make $f \circ g$ hard. In this thesis we proposed a new dual approach. We fixed f to be a natural function like MOD_2, NOR, MAJ, and obtained strong upper and lower bounds on the communication complexity of $f \circ g$ for every possible g. As there are many choices for g in the NOF model, this approach gives us new insights about the communication complexity of composed functions.

In the second part of the thesis, we studied the Fourier analytic properties of symmetric functions (Chapter 5). Fourier analysis provides us a different and very useful way of looking at boolean functions. It is by now one of the main tools used in computational complexity theory and the situation is no different in communication complexity. In particular the communication complexity of $f \circ \text{XOR}$ is intimately connected to the Fourier analytic properties of f. In this thesis we gave a combinatorial characterization for the Fourier L_1-norm of all symmetric functions and discussed its applications to communication complexity and decision tree complexity. We hope that our result will be extended to *approximate* L_1-norm. As discussed in the conclusion of Chapter 5, this would have several important applications in computational complexity theory.

In the third part of the thesis (Chapter 6), we studied a different aspect of communication complexity: the trade-off between communication cost and privacy of the players. We used the notions of *worst-case approximate privacy* and *average-case approximate privacy* introduced by Feigenbaum et al. [FJS10a]. These serve as very reasonable formalizations of privacy as argued in [FJS10a] and this thesis. One area where players wish to keep their inputs private while minimizing the communication cost is combinatorial auctions. In this setting we studied the Vickrey auction, which is the canonical example of a *truthful* mechanism. We showed essentially tight trade-offs between the cost of any protocol computing the Vickrey auction and the worst-case approximate privacy as well as the average-case approximate privacy of the players. We also showed an exponential lower bound on the average-case approximate privacy for the *set-intersection* function, regardless of the cost of the protocol; this solved an explicitly stated open problem from [FJS10a].

At the end of each chapter from 3 to 6, we stated some open problems that we personally thought were both interesting and within reach. In the final pages of this thesis, let's step back and recall some of the big questions that are at the forefront of communication complexity.

Quantum vs classical communication complexity for total boolean functions is one of the most intriguing questions in the field. It is conjectured that quantum and classical communication complexities are polynomially related but progress on it has been limited. The conjecture is open even in the 2 player setting and it would be a breakthrough to establish it for functions of the form $f \circ \textsc{and}$ or $f \circ \textsc{xor}$ for all f. We believe that functions of the form $f \circ \textsc{xor}$ should be easier to tackle because of the intimate relationship between the communication complexity of $f \circ \textsc{xor}$ and the Fourier analytic properties of f. Fourier analysis of boolean functions have matured tremendously over the last couple of decades and it probably holds the keys that would unlock some of the mysteries in communication complexity.

In our biased view, the most important problem in communication complexity is breaking the $\log n$ barrier. We can prove strong lower bounds for several functions for up to $\approx \frac{1}{2} \log n$ players and interestingly, all these lower bounds apply in the randomized model since they are based on the discrepancy method. In fact, we don't know of an explicit function that is hard in the deterministic model but easy in the randomized model for even 3 players. The EVAL_G function has been conjectured to have this property, for any G: we know EVAL_G has an efficient randomized protocol for any G, and it is conjectured to be hard in the deterministic model. The nice thing about this conjecture is that we are free to choose our favourite G. And the smart choice for G would be a "quasirandom"[1] G (see [Gow08]) as one can hope to

[1] A group is called *quasirandom* if it is far away from being an Abelian group in a certain sense. Informally these groups have good mixing properties. More formally, these

exploit this property to prove deterministic communication complexity lower bounds for EVAL_G. Here, we would like to take the above conjecture a step further and identify EVAL_G, for a quasirandom G, as a candidate to break the $\log n$ barrier. As far as we know, EVAL_G has never been considered as a candidate to break the $\log n$ barrier before.

The idea of looking at quasirandom G is not new and was proposed by Gowers [Gow08] in order to get good bounds on the Ramsey number $r_k(G)$. Tao [Tao12] recently accomplished this for $k = 3$. Can Tao's result be lifted to give good bounds for $r_2^{\angle}(G)$? This would show that EVAL_G is hard in the deterministic model for 3 players. More ambitiously, can it be lifted to give good bounds for $r_k^{\angle}(G)$, where the dependence on k is good? This would break the $\log n$ barrier.

are groups whose non-trivial irreducible representations have high dimension.

Bibliography

[Aar11] Scott Aaronson. Why philosophers should care about computational complexity. *CoRR*, abs/1108.1791, 2011.

[ACC$^+$12] Anil Ada, Arkadev Chattopadhyay, Stephen A. Cook, Lila Fontes, Michal Koucký, and Toniann Pitassi. The hardness of being private. In *IEEE Conference on Computational Complexity*, pages 192–202, 2012. To appear in ACM Transactions on Computation Theory.

[ACFN12] Anil Ada, Arkadev Chattopadhyay, Omar Fawzi, and Phuong Nguyen. The nof multiparty communication complexity of composed functions. In *International Colloquium on Automata, Languages, and Programming*, pages 13–24, 2012. To appear in Computational Complexity.

[AFH12] Anil Ada, Omar Fawzi, and Hamed Hatami. Spectral norm of symmetric functions. In *APPROX-RANDOM*, pages 338–349, 2012.

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58:137–147, 1999.

[BBC$^+$01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, July 2001.

[BBCR10] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 67–76, 2010.

[BBM11] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. In *In IEEE Conference on Computational Complexity*, pages 210–220, 2011.

[BDPW10] Paul Beame, Matei David, Toniann Pitassi, and Philipp Woelfel. Separating deterministic from randomized multiparty communication complexity. *Theory of Computing*, 6(1):201–225, 2010.

[Beh46] Felix A. Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 32:331–332, 1946.

[BFS86] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *Proceedings of Symposium on Foundations of Computer Science*, pages 337–347, 1986.

[BGG06] Richard Beigel, William Gasarch, and James Glenn. The multiparty communication complexity of exact t: Improved bounds and new problems. In Rastislav Královic and Pawel Urzyczyn, editors, *Mathematical Foundations of Computer Science 2006*, volume 4162 of *Lecture Notes in Computer Science*, pages 146–156. Springer Berlin / Heidelberg, 2006.

[BGKL03] László Babai, Anna Gál, Peter G. Kimmel, and Satyanarayana V. Lokam. Communication complexity of simultaneous messages. *SIAM Journal on Computing*, 33:137–166, 2003.

[BHN09] Paul Beame and Dang-Trinh Huynh-Ngoc. Multiparty communication complexity and threshold circuit size of AC^0. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 53–62, 2009.

[BK12] Michael Bateman and Nets Hawk Katz. New bounds on cap sets. *Journal of the American Mathematical Society*, 25:585–613, 2012.

[BKL95] László Babai, Peter G. Kimmel, and Satyanarayana V. Lokam. Simultaneous messages vs. communication. In *In 12th Annual Symposium on Theoretical Aspects of Computer Science*, pages 361–372. Springer, 1995.

[BNS92] László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992.

[Bou99] Jean Bourgain. On triples in arithmetic progression. *Geometric and Functional Analysis*, pages 968–984, 1999.

[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*, 37:845–869, June 2007.

[Bra11] Mark Braverman. Interactive information complexity. *Electronic Colloquium on Computational Complexity*, (123), 2011.

[Bru90] Jehoshua Bruck. Harmonic analysis of polynomial threshold functions. *SIAM Journal of Discrete Mathematics*, 3:168–177, 1990.

[BS92] Jehoshua Bruck and Roman Smolensky. Polynomial threshold functions, ac0 functions, and spectral norms. *SIAM Journal on Computing*, 21(1):33–42, February 1992.

[BS08] Felix Brandt and Tuomas Sandholm. On the Existence of Unconditionally Privacy-Preserving Auction Protocols. *ACM Transactions on Information and System Security*, 11(2):1–21, May 2008.

[BT94] Richard Beigel and Jun Tarui. On ACC. *Computational Complexity*, 4:350–366, 1994.

[BYJKS04] 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, June 2004.

[CA08] Arkadev Chattopadhyay and Anil Ada. Multiparty communication complexity of disjointness. Technical report, In Electronic Colloquium on Computational Complexity (ECCC) TR08–002, 2008.

[CFL83] Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multi-party protocols. In *Proceedings of ACM Symposium on Theory of Computing*, pages 94–99, 1983.

[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, April 1988.

[CGGK94] Benny Chor, Mihály Geréb-Graus, and Eyal Kushilevitz. On the structure of the privacy hierarchy. *Journal of Cryptology*, 7:53–60, 1994.

[Cha08] Arkadev Chattopadhyay. *Circuits, Communication and Polynomials*. PhD thesis, McGill University, 2008.

[CK89] Benny Chor and Eyal Kushilevitz. A Zero-One Law for Boolean Privacy (extended abstract). In *Proceedings of ACM Symposium on the Theory of Computing*, pages 62–72, 1989.

[CKK$^+$07] Arkadev Chattopadhyay, Andreas Krebs, Michal Koucky, Mario Szegedy, Pascal Tesson, and Denis Thérien. Languages with bounded multiparty communication complexity. In *Proceedings of Conference on Theoretical Aspects of Computer Science*, pages 500–511, 2007.

[CS04] Vincent Conitzer and Tuomas Sandholm. Communication complexity as a lower bound for learning in games. In *International Conference on Machine Learning*, pages 24–, 2004.

[CT93] Fan R.K. Chung and Prasad Tetali. Communication complexity and quasi randomness. *SIAM Journal on Discrete Mathematics*, 6(1):110–123, 1993.

[CWYS01] Amit Chakrabarti, Anthony Wirth, Andrew Yao, and Yaoyun Shi. Informational complexity and the direct sum problem for simultaneous message complexity. *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 270–278, 2001.

[dW08] Ronald de Wolf. A note on quantum algorithms and the minimal degree of ϵ-error polynomials for symmetric functions. *Quantum Information and Computation*, 8(10):943–950, November 2008.

[Elk11] Michael Elkin. An improved construction of progression-free sets. *Israel Journal of Mathematics*, 184:93–128, 2011.

[ET36] Paul Erdős and Paul Turán. On some sequences of integers. *Journal of the London Mathematical Society*, 1(4):261, 1936.

[FJS10a] Joan Feigenbaum, Aaron D Jaggard, and Michael Schapira. Approximate Privacy: Foundations and Quantification. *Proceedings of the 11th Conference on Electronic Commerce*, pages 167–178, 2010.

[FJS10b] Joan Feigenbaum, Aaron D Jaggard, and Michael Schapira. Approximate Privacy: PARs for Set Problems. *DIMACS Technical Report 2010-01*, pages 1–34, 2010.

[FK78] Harry Furstenberg and Yitzhak Katznelson. An ergodic Szemerédi theorem for commuting transformations. *Journal d'Analyse Mathematique*, 34:275–291, 1978.

[FKL$^+$01] Jürgen Forster, Matthias Krause, Satyanarayana V. Lokam, Rustam Mubarakzjanov, Niels Schmitt, and Hans ulrich Simon. Relations between communication complexity, linear arrangements, and computational complexity. In *Foundations of Software Technology and Theoretical Computer Science*, pages 171–182, 2001.

[GHR92] Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, pages 277–300, 1992.

[Gow01] W. Timothy Gowers. A new proof of Szemerédi's theorem. *Geometric and Functional Analysis*, 11:465–588, 2001.

[Gow07] W. Timothy Gowers. Hypergraph regularity and the multidimensional Szemerédi theorem. *Annals of Mathematics*, 166:897–946, 2007.

[Gow08] W. Timothy Gowers. Quasirandom groups. *Combinatorics, Probability and Computing*, 17(3):363–387, May 2008.

[Gow10] W. Timothy Gowers. Decompositions, approximate structure, transference, and the Hahn-Banach theorem. *Bulletin of the London Mathematical Society*, 42:573–606, 2010.

[Gre05] Ben Green. *Surveys in Combinatorics 2005*, chapter Finite field models in additive combinatorics, pages 1–27. London Mathematical Society Lecture Notes 327. Cambridge Univ Press, 2005.

[Gro94] Vince Grolmusz. The BNS lower bound for multi-party protocols is nearly optimal. *Information and Computation*, 112:51–54, 1994.

[Gro95] Vince Grolmusz. Separating the communication complexities of MOD m and MOD p circuits. In *Proceedings of Symposium on Foundations of Computer Science*, pages 278–287, 1995.

[Gro97] Vince Grolmusz. On the power of circuits with gates of low l1 norms. *Theoretical Computer Science*, 188(1-2):117–128, November 1997.

[Gro98] Vince Grolmusz. Circuits and multi-party protocols. *Computational Complexity*, 7:1–18, 1998.

[Gro99] Vince Grolmusz. Harmonic analysis, real approximation, and the communication complexity of boolean functions. *Algorithmica*, 23(4):341–353, 1999.

[GS08] Ben Green and Tom Sanders. Boolean functions with small spectral norm. *Geometric and Functional Analysis*, 18(1):144–162, 2008.

[HG91] Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:610–618, 1991.

[KKMS08] Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[Kla02] Hartmut Klauck. On quantum and approximate privacy. In *Proceedings of Symposium on Theoretical Aspects of Computer Science*, pages 335–346, 2002.

[Kla07] Hartmut Klauck. Lower bounds for quantum communication complexity. *Siam Journal on Computing*, 37:20–46, 2007.

[KLM$^+$09] Mihail N Kolountzakis, Richard J Lipton, Evangelos Markakis, Aranyak Mehta, and Nisheeth K Vishnoi. On the fourier spectrum of symmetric boolean functions. *Combinatorica*, 29(3):363–387, 2009.

[KM91] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. In *Proceedings of ACM Symposium on Theory of Computing*, pages 455–464, 1991.

[KS10] Adam R. Klivans and Alexander A. Sherstov. Lower bounds for agnostic learning via approximate rank. *Computational Complexity*, 19(4):581–604, 2010.

[Kus89] Eyal Kushilevitz. Privacy and communication complexity. *30th Annual Symposium on Foundations of Computer Science*, pages 416–421, 1989.

[KV94] Michael Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, 1994.

[LM07] Michael T. Lacey and William McClain. On an argument of Shkredov on two-dimensional corners. *Online Journal of Analytic Combinatorics*, 2007.

[LS88] László Lovász and Michael Saks. Lattices, Möbius functions and communication complexity. In *29th Annual Symposium on Foundations of Computer Science*, pages 81–90. IEEE Computer Society, 1988.

[LS09] Troy Lee and Adi Shraibman. Disjointness is hard in the multiparty number-on-the-forehead model. *Computational Complexity*, 18:309–336, 2009.

[LZ10] Troy Lee and Shengyu Zhang. Composition theorems in communication complexity. In *Proceedings of International Colloquium Conference on Automata, Languages and Programming*, pages 475–489, 2010.

[MNSW98] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37 – 49, 1998.

[MO09] Ashley Montanaro and Tobias Osborne. On the communication complexity of XOR functions. *Arxiv preprint arXiv:0909.3392*, 2009.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge Univ Press, 2005.

[New91] Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, July 1991.

[Nis93] Noam Nisan. The communication complexity of threshold gates. *Combinatorica*, 1993.

[NS06] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129:192–224, 2006.

[NW93] Noam Nisan and Avi Wigderson. Rounds in communication complexity revisited. *Siam Journal on Computing*, 22:211–219, 1993.

[O'B11] Kevin O'Bryant. Sets of integers that do not contain long arithmetic progressions. *The Electronic Journal of Combinatorics*, 18(1):59, 2011.

[OS08] Ryan O'Donnell and Rocco A. Servedio. Extremal properties of polynomial threshold functions. *Journal of Computer System Sciences*, 74(3):298–312, May 2008.

[OWZ11] Ryan O'Donnell, John Wright, and Yuan Zhou. The fourier entropy-influence conjecture for certain classes of boolean functions. In *Proceedings of International Colloquim on Automata, Languages and Programming*, pages 330–341, 2011.

[Pat92] Ramamohan Paturi. On the degree of polynomials that approximate symmetric Boolean functions (preliminary version). In *Proceedings of ACM Symposium on Theory of Computing*, pages 468–474, 1992.

[Pud03] Pavel Pudlák. An application of hindman's theorem to a problem on communication complexity. *Combinatorics, Probability and Computing*, 12:661–670, November 2003.

[Pud06] Pavel Pudlák, 2006. Personal communication.

[Raz95] Ran Raz. Fourier analysis for probabilistic communication complexity. *Computational Complexity*, 5:205–221, 1995.

[Raz00] Ran Raz. The BNS-Chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.

[Raz03] Alexander Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya: Mathematics*, 67(1):145–159, 2003.

[Rot53] Klaus F. Roth. On certain sets of integers. *Journal of The London Mathematical Society-second Series*, s1-28:104–109, 1953.

[San11] Tom Sanders. On Roths theorem on progressions. *Annals of Mathematics*, 174:619–636, 2011.

[SB91] Kai-Yeung Siu and Jehoshua Bruck. On the power of threshold circuits with small weights. *SIAM Journal on Discrete Mathematics*, 4(3):423–435, 1991.

[She07] Alexander A. Sherstov. The pattern matrix method for lower bounds on quantum communication. In *In Proceedings of Symposium on Theory of Computing*, pages 85–94, 2007.

[She09] Alexander A. Sherstov. Approximate inclusion-exclusion for arbitrary symmetric functions. *Comput. Complex.*, 18(2):219–246, 2009.

[She11] Alexander A. Sherstov. The multiparty communication complexity of set disjointness. Technical report, In Electronic Colloquium on Computational Complexity (ECCC) TR11–145, 2011.

[She13] Alexander A. Sherstov. Communication lower bounds using directional derivatives. In *Proceedings of ACM Symposium on Symposium on Theory of Computing*, pages 921–930, 2013.

[Shk06a] Ilya D. Shkredov. On a generalization of Szemerédi's theorem. *Proceedings of the London Mathematical Society*, 93:723–760, 2006.

[Shk06b] Ilya D. Shkredov. On a problem of Gowers. *Izvestiya: Mathematics*, 70:385, 2006.

[Sho09] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, 2009.

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

[ST11] Amir Shpilka and Avishay Tal. On the minimal fourier degree of symmetric boolean functions. In *Proceedings of IEEE Conference on Computational Complexity*, pages 200–209, 2011.

[SZ09a] Yaoyun Shi and Zhiqiang Zhang. Communication complexities of symmetric XOR functions. *Quantum Information and Computation*, 9:255–263, 2009.

[SZ09b] Yaoyun Shi and Yufan Zhu. Quantum communication complexity of block-composed functions. *Quantum Information and Computation*, 9:444–460, May 2009.

[Sze75] Endre Szemerédi. On sets of integers containing no k elements in arithmetic progression. In *International Congress of Mathematicans*, 1975.

[Tao12] Terence Tao. Mixing for progressions in non-abelian groups. http://arxiv.org/abs/1212.2586, 2012.

[Tes03] Pascal Tesson. *Computational complexity questions related to finite semigroups and monoids.* PhD thesis, McGill University, 2003.

[VW08] Emanuele Viola and Avi Wigderson. Norms, xor lemmas, and lower bounds for polynomials and protocols. *Theory of Computing*, 4(1):137–168, 2008.

[Yao79] Andrew C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of ACM Symposium on Theory of Computing*, pages 209–213, 1979.

[Yao83] Andrew C. Yao. Lower bounds by probabilistic arguments. In *Proceedings of Symposium on Foundations of Computer Science*, pages 420–428, 1983.

Index