Circuits, Communication and Polynomials

Arkadev Chattopadhyay

Doctor of Philosophy

School of Computer Science

McGill University

Montreal,Quebec

2008-08-29

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

ACKNOWLEDGEMENTS

First, I would like to express my deep gratitude to my supervisor Denis Thérien for picking me up as a graduate student, introducing me to theoretical computer science, supporting me financially and infecting me with his love for circuit complexity. I thank him for asking me some of the most beautiful questions on boolean circuits. It is a real pity that I did not solve any of them, despite Denis's constant encouragement, but I earnestly hope that one day I will. I also thank him for inviting me, each year for the last six years, to his fantastic workshop at Barbados.

Second, I would like to thank my co-supervisor Pascal Tesson for sharing his knowledge and enthusiasm for communication complexity. I also thank him for inviting and hosting me at Quebec city. He has given numerous comments and provided several corrections to earlier drafts of this thesis that have greatly improved its readability. I aslo thank him for funding some of my scientific visits.

I gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) during the period 2004-2007.

The work presented in this thesis has benefitted a lot from several people, directly and indirectly. I hope that I may be excused for not remembering every discussion that I have had at a workshop or a conference that triggered a certain chain of thought later. First, I would like to thank my collaborators with whom I directly obtained results presented here: Anil Ada, Navin Goyal, Kristoffer Arnsfelt Hansen, Michal Koucky, Andreas Krebs, Pavel Pudlák, Mario Szegedy, Pascal Tesson and Denis Thérien. It was fun collaborating and I learnt a lot from each of

them. I also thank Mario for inviting me twice to Rutgers University and for several stimulating discussions.

My understanding of the subject has greatly been influenced by several other colleagues: I have had interesting discussions with Eric Allender, David Barrington, Matei David, Ricard Gavaldà, Fred Green, Pierre McKenzie, Toni Pitassi, Mike Saks and Alexander Sherstov. In particular, I will like to thank David, Fred and Ricard for teaching me several things and Toni for inviting me to Toronto. I also had several enlightening discussions with Henri Darmon and Andrew Granville at the expense of boring them with my point of view.

McGill University's computer science department is a wonderful place to do research. In particular, I have been privileged to have had great teachers like Luc Devroye, Prakash Panangaden, Bruce Reed, Jacques Verstraete and Adrian Vetta. Bruce taught me probabilistic combinatorics and I sincerely thank him for inviting me to his workshop in Banff.

The administrative staff of the department are a wonderful bunch. In particular, I thank Diti, Lucy, Lise and Sheryl for all the help that they have provided. I also thank Andrew, Kailesh, Ron and the other systems staff of the department who have always been eager to help.

I would like to thank Denis's other students Anil Ada, Laszlo Egri and Mark Mercer, for helping create a very interesting and fun environment. I also thank Anil for providing valuable comments on an earlier draft of the thesis. He also helped me tremendously with drawing figures in LaTeX.

My parents developed in me a taste for abstract things and encouraged intellectual curiosity. My father inspired an early but deep interest in the natural sciences that has endured. This was further stimulated by the intense polemical environment present at my maternal uncles'.

Finally, I want to sincerely thank my wife for putting up with me during real difficult times. She shouldered, without complaints, the financial burden that I unleashed upon her out of the blue, when I decided to come back to school. She also patiently, single-handedly and tirelessly took care of our daughter, who was born while this thesis was still unborn, letting me focus on the job.

ABSTRACT

In this thesis, we prove unconditional lower bounds on resources needed to compute explicit functions in the following three models of computation: constant-depth boolean circuits, multivariate polynomials over commutative rings and the 'Number on the Forehead' model of multiparty communication. Apart from using tools from diverse areas, we exploit the rich interplay between these models to make progress on questions arising in the study of each of them.

Boolean circuits are natural computing devices and are ubiquitous in the modern electronic age. We study the limitation of this model when the depth of circuits is fixed, independent of the length of the input. The power of such constant-depth circuits using gates computing modular counting functions remains undetermined, despite intensive efforts for nearly twenty years. We make progress on two fronts: let m be a number having r distinct prime factors none of which divides ℓ. We first show that constant depth circuits employing AND/OR/MOD$_m$ gates cannot compute efficiently the MAJORITY and MOD$_\ell$ function on n bits if 'few' MOD$_m$ gates are allowed, i.e. they need size $n^{\Omega(\frac{1}{s}(\log n)^{1/(r-1)})}$ if s MOD$_m$ gates are allowed in the circuit. Second, we analyze circuits that comprise only MOD$_m$ gates. We show that in sub-linear size (and arbitrary depth), they cannot compute AND of n bits. Further, we establish that in that size they can only very poorly approximate MOD$_\ell$.

Our first result on circuits is derived by introducing a novel notion of computation of boolean functions by polynomials. The study of degree as a resource in polynomial representation of boolean functions is of much independent interest. Our notion, called the weak generalized representation, generalizes all previously studied notions of computation by polynomials over finite commutative rings. We prove that over the ring $\mathbb{Z}_m$, polynomials need $\Omega(\log n)^{1/(r-1)}$ degree to represent, in our sense, simple functions like MAJORITY and MOD_ℓ. Using ideas from arguments in communication complexity, we simplify and strengthen the breakthrough work of Bourgain showing that functions computed by $o(\log n)$-degree polynomials over $\mathbb{Z}_m$ do not even correlate well with MOD_ℓ.

Finally, we study the 'Number on the Forehead' model of multiparty communication that was introduced by Chandra, Furst and Lipton [CFL83]. We obtain fresh insight into this model by studying the class CC_k of languages that have constant k-party deterministic communication complexity under every possible partition of input bits among parties. This study is motivated by Szegedy's [Sze93] surprising result that languages in CC_2 can all be extremely efficiently recognized by very shallow boolean circuits. In contrast, we show that even CC_3 contains languages of arbitrarily large circuit complexity. On the other hand, we show that the advantage of multiple players over two players is significantly curtailed for computing two simple classes of languages: languages that have a neutral letter and those that are symmetric.

Extending the recent breakthrough works of Sherstov [She07, She08b] for two-party communication, we prove strong lower bounds on multiparty communication complexity of functions. First, we obtain a bound of $n^{\Omega(1)}$ on the k-party randomized communication complexity of a function that is computable by constant-depth circuits using AND/OR gates, when k is a constant. The bound holds as long as protocols are required to have better than inverse exponential (i.e. $2^{-n^{o(1)}}$) advantage over random guessing. This is strong enough to yield lower bounds on the size of an important class of depth-three circuits: circuits having a MAJORITY gate at its output, a middle layer of gates computing arbitrary symmetric functions and a base layer of arbitrary gates of restricted fan-in.

Second, we obtain $n^{\Omega(1)}$ lower bounds on the k-party randomized (bounded error) communication complexity of the Disjointness function. This resolves a major open question in multiparty communication complexity with applications to proof complexity. Our techniques in obtaining the last two bounds, exploit connections between representation by polynomials over reals of a boolean function and communication complexity of a closely related function.

ABRÉGÉ

Nous cherchons dans cette thèse à établir des bornes inférieures sur la quantité de ressources de calcul nécessaires au calcul de certaines fonctions explicites. Cette étude est centrée sur trois modèles importants: les circuits booléens de profondeur bornée, les polynômes multivariés dans des anneaux commutatifs et le modèle de complexité de communication à plusieurs joueurs appelé “modèle de données sur le front”. Pour avancer sur ces questions, nous utilisons une variété d'outils mathématiques mais exploitons aussi les riches interactions entre l'étude de ces trois modèles.

Les circuits booléens sont des engins de calcul très naturels et sont omniprésents dans l'ère technologique. Nous étudions les limites de tels circuits lorsque leur profondeur est bornée par une constante ne dépendant pas de la longueur des données. Malgré vingt ans de recherche sur le sujet, leur puissance dans ce cas est encore très mal comprise lorsque les portes composant les circuits calculent des sommes modulo un entier. Nous progressons sur deux fronts. Nous considérons d'abord que les circuits de profondeur bornée employant des portes ET/OU/MOD$_m$. Nous montrons qu'ils ne peuvent calculer efficacement les fonctions MAJORITÉ et MOD$_\ell$ (pour ℓ et m co-premiers) lorsque le nombre de portes MOD$_m$ est limité. D'autre part, nous considérons les circuits ne contenant que des portes MOD$_m$ et prouvons qu'un tel circuit ne peut calculer la fonction ET sur n bits lorsque sa taille est $o(n)$ et ce, peut-importe sa profondeur. Nous montrons même que ces circuits ne peuvent calculer que des approximations très pauvres de la fonction MOD$_\ell$.

Notre premier résultat sur les circuits est basé sur une nouvelle notion de calcul

d'une fonction par des polynômes. Dans ce type d'étude, le degré des polynômes est vu comme une ressource de calcul à minimiser. Notre notion de représentation faible généralisée étend toutes les notions précédentes de représentations par des polynômes sur l'anneau commutatif $\mathbb{Z}_m$. Nous montrons que, dans ce nouveau cadre, les fonctions MAJORITÉ et MOD_ℓ ne peuvent être représentées par des polynômes de petit degré. Par ailleurs, nous utilisons des idées venant de la complexité de communication pour simplifier et renforcer les percées de Bourgain qui a montré que les polynômes de $\mathbb{Z}_m$ de degré $o(\log n)$ n'ont qu'une faible corrélation avec la fonction MOD_ℓ.

Finalement, nous étudions le modèle de communication multipartie "données sur le front" proposé par Chandra, Furst et Lipton [CFL83]. Nous tentons de mieux comprendre la nature du modèle en considérant la classe CC_k des langages de complexité bornée dans le modèle déterministe et "pire partition" pour k joueurs. Ces travaux sont motivés par les résultats surprenants de Szegedy [Sze93] qui montrent en particulier que les langages de CC_2 peuvent tous être reconnus efficacement par des circuits booléens de très petite profondeur. Nous montrons qu'à l'opposé, il existe des langages de CC_3 qui ont une complexité de circuit arbitraire. Cependant, nous prouvons aussi que l'avantage des joueurs multiples est grandement limité lorsque le langage à reconnaître est symmétriques ou muni d'une lettre neutre.

En généralisant les résultats récents et novateurs de Shershtov [She07, She08b] sur le modèle à deux joueurs, nous obtenons de fortes bornes inférieures sur la complexité de communication pour k joueurs de fonctions explicites. Pour toute constante k, nous établissons d'abord une borne de $n^{\Omega(1)}$ sur la complexité de protocoles

randomisés pour k joueurs, calculant une fonction calculé par des circuits ET/OU de taille polynomiale et de profondeur constante. Cette borne reste valide pour tout protocole dont l'avantage par rapport à une réponse aléatoire est supérieure à l'inverse d'une fonction exponentielle (i.e. $2^{-n^{o(1)}}$). Le résultat est suffisamment fort pour obtenir des bornes inférieures sur la taille d'une classe importante de circuits, soit ceux formés d'une porte MAJORITÉ en sortie, d'un niveau intermédiaire formé de portes calculant une fonction symmétrique arbitraire et d'un niveau de base où l'entrance des portes utilisées est bornée.

De plus, nous obtenons une borne inférieure de $n^{\Omega(1)}$ sur la complexité à k joueurs des protocoles randomisés (avec erreur bornée) pour la fonction DISJOINTNESS. Cette borne résoud une question très importante qui a des applications nombreuses, entre autre dans le domaine de la complexité des preuves. Nos résulats exploitent les liens entre les représentations de fonctions booléennes par des polynômes réels et la complexité de communication de fonctions qui leur sont intimement liées.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Origins of the Theory of Computation

Every reasonable curriculum in elementary school is replete with tricks to "compute". Starting with skills to perform arithmetic operations like addition, multiplication, division in primary school, through taking square-roots of numbers (up to a required precision) and solving quadratic equations in middle school, kids move on to learn performing much more sophisticated computational tasks like differentiating or integrating whole functions. Indeed, many people like to measure a child's progress in school by testing how quickly he/she can perform such tasks. In light of this, it may seem surprising that it took until the beginning of the last century for someone to ask the right question that made people realize that, something as fundamental as computation had gone unformalized!

In 1900, David Hilbert posed the following problem[1] to the leading figures of the period in mathematics : Is there a finitary procedure to determine if a given multivariate polynomial with integral coeffcients has an integral solution? Hilbert, as the legend goes, was expecting a positive answer. In retrospect now, one may well say that 'fortunately' the answer was 'no'. Had the answer been 'yes' and

[1] It featured as the tenth problem in Hilbert's list of twenty three problems.

had someone discovered such a procedure, arguably that would have delayed the beginning of the inevitable ‘Theory of Computation’.

Hilbert’s question led Turing, more than thirty five years later, to provide a satisfactory model of computation now known as the Turing machine. The Turing machine remains the universal model of computation as we understand today. Everything that can be done by a real computer or any other known devices[2] can be ‘reasonably efficiently’ performed on a Turing machine. A procedure running on a Turing machine is called an *algorithm*. On the other hand, Turing’s work led to such remarkable conclusions as that not every task has an algorithm that halts on all inputs. Using this result about Turing machines, in 1972, Matiassevich resolved Hilbert’s tenth problem in the negative, building upon the earlier breakthrough work of Davis, Putnam and Robinson.

While Turing’s work and Hilbert’s problem were motivated from foundational questions of mathematical logic, the notion of ‘efficient computation’ is easily motivated from more mundane affairs. Many salesmen have wondered how to chalk out an itinerary such that they touch upon every city precisely once and return to their starting point. Modern network designers are routinely confronted with the problem of determining an optimal cost network with a given redundancy. Secretaries have a hard time scheduling a time table meeting everyone’s demands. Indeed, life would have been much more pleasant if several such tasks from different spheres of activity

[2] Strictly speaking, devices whose operations are limited by the classical laws of physics.

had efficient algorithms. Unfortunately, all these tasks seem intractably difficult in the sense that every known algorithm for them runs for very long before they output a solution. In particular, the number of steps that the algorithm executes before giving the correct answer tends to grow exponentially with the size of the input, measured in any reasonable sense.

One of the basic goals of computer science and the guiding theme of Computational Complexity is to classify algorithmic problems into *complexity classes* according to the amount of minimum *resources* needed to solve them in a given computational model. The most powerful model or device that is considered for such task is the Turing machine. The two resources that have classically been looked at, corresponding to the running time and memory requirements respectively of a modern computer, are *time* and *space* measured with respect to the size of the input. The usage of resources is defined by the behavior of the algorithm on the worst-case input (as opposed to let us say its behavior on the average[3] input). The universally accepted mathematical concept of efficient (and feasible) computation is the notion of algorithms running in polynomial time. This gives rise to the widely known class P that contains those *decision* problems that admit polynomial time algorithms. None of the problems mentioned in the last paragraph, when defined formally as decision problems in a reasonable way, *seem* to be in P. However, there

[3] Average-case complexity is an interesting growing sub-field of Computational Complexity, surveyed by Bogdanov and Trevisan [BT06] recently.

are no known arguments that show there does not exist polynomial time algorithms for these problems.

On the other hand, these problems share the property that every guessed solution can be efficiently verified. For instance, given an itinerary a salesman can quite easily verify if it satisfies the need of touching every city precisely once. Computation where guessing is allowed gives rise to the important notion of *non-determinism*. The class of problems whose guessed solution can be verified in polynomial time by a Turing machine is the celebrated class NP. The Holy Grail of computational complexity theory, and an outstanding problem in modern mathematics, is to separate (or collapse) these two classes.

1.2 The Theory of Lower Bounds

Proving impossibility results about computation is a formidable challenge. Much of computer science is filled with various tricks on how to perform certain things rather than to show the impossibility of the existence of tricks to achieve a task. Indeed, powerful algorithms exist drawing upon entirely counter-intuitive ideas from various branches of classical mathematics. The tremendous rate of growth of such tricks (see for example [LU97, AKS04, Rei05, CKSU05, AHT07]) strongly suggests that we have barely scratched the surface of algorithmic techniques. In this light, Turing's theorem about the existence of non-computable tasks does seem quite impressive. It is surprising that his result follows simply by employing the technique invented by Cantor to prove the non-existence of a bijection from the set of reals to the set of natural numbers. This powerful method is called *diagonalization* in logic. Interesting and fundamental separation results like the time and space *hierarchy theorems* have been discovered, also employing the method of diagonalization. These results roughly say that the class of functions computable by a Turing machine strictly grows if either more time or more space is allowed.

Unfortunately, diagonalization has strong limitations. In particular, diagonalization proofs *relativize* i.e. if two complexity classes A and B are separated using diagonalization, then for every language $\mathcal{C}$, A with access to $\mathcal{C}$ for free (denoted by $A^{\mathcal{C}}$) is different from B with similar access to $\mathcal{C}$ (denoted by $B^{\mathcal{C}}$). A very interesting result of [BGS75] establishes that there exists languages $\mathcal{C}, \mathcal{D}$ such that $\mathrm{P}^{\mathcal{C}} = \mathrm{NP}^{\mathcal{C}}$ and $\mathrm{P}^{\mathcal{D}} \neq \mathrm{NP}^{\mathcal{D}}$. This result proves that P cannot be separated from NP using a pure diagonalization argument. This made researchers look for *non-relativizing* techniques.

One way of developing new methods is to consider explicit functions and prove lower bounds against them in other natural (and simpler) models of computation. Interesting natural models bring out new facets of computation. The effort of understanding their limitations often forges links with other disciplines of mathematics. More surprisingly, and perhaps a little discomfortingly, it highlights how little we understand computation when we are unable to determine the complexity of a function in a simple model. Arguably, this goes on to show that although the P vs. NP question defined our field, it is by no means the only question. While proving lower bounds for explicit functions in natural models of computation is of fundamental importance, the theory of lower bounds is just in its infancy.

We contribute to the further development of this theory by exploring three well-known and important models of computation: boolean circuits of constant depth, low degree multivariate, multilinear polynomials over rings and the ‘Number on the Forehead’ (NOF) model of multiparty communication.

An important feature of the Turing machine is its *uniformity*, i.e. for every task, *one* algorithm handles inputs of every possible length. This is an extreme degree of uniformity. One could enforce a milder notion of uniformity by having a family of algorithms, one for every input length and then have a relationship between each such algorithm in the family. Vollmer [Vol99] provides an exposition of this approach to circuit complexity. On the other hand, our approach with all three models is that we consider *non-uniform* versions as opposed to the Turing machine model. In other words, we consider a family of algorithms (i.e a circuit or a protocol or a polynomial as the case may be), one for each input length n and there does not exist any a priori relationship among algorithms in the family. Disregarding uniformity allows one to focus on the combinatorial weakness of a model. We believe such investigations bring out deep combinatorial questions that are interesting in their own right. Such questions then allow fruitful exchange with other areas of mathematics, making available a wider tool-set to make progress.

1.3 Boolean Circuits

Although the Turing machine is the model employed by theoreticians to argue about computation in general, it is fair to say that it is not used in practice as a device. In contrast, circuits indeed are implemented by engineers and are ubiquitous in modern life. The integrated circuit, abbreviated as IC, has revolutionized our electronic age. They are the building blocks of not just modern computers, but every sophisticated device. We describe this natural model of computation more formally below.

A circuit is a directed acyclic graph whose nodes are *gates* and edges are *wires*, where each gate computes a boolean function of the wires feeding into it. In general, circuits have multiple outputs. In this work, we focus on circuits computing a boolean function. Hence, our circuits have a special node with out-degree 1 called the output gate. The value it outputs on a particular input instance is the output of the circuit on that input. As stated before, a circuit operates on inputs of a fixed length n. More precisely, we consider a family of circuits $\{\ldots, C_n, \ldots\}$, one for each input length. Similarly, when we define a boolean function, we do so by defining one for each input length. To keep our notation simple, we do not explicitly mention the input length as in most cases it can be easily understood from the context. For example, we define the *THRESHOLD* function as $\mathrm{THR}_k(x) = 1$ iff $\sum_{i=1}^{n} x_i \geq k$, where k is a positive integer. Here, k need not be fixed. In fact, $\mathrm{THR}_{\lceil n/2 \rceil}$ is called the MAJORITY function. Similarly, $\mathrm{MOD}_q(x) = 0$ iff $\sum_{i=1}^{n} x_i \equiv 0 \pmod{q}$, for any positive integer q. The following figure shows a circuit having only AND and OR gates computing the MOD_2 function (also known as PARITY) for the input length $n = 3$. It works by exhaustively verifying if the input instance corresponds to any one of strings with odd parity.

The *size* of a circuit is the number of non-input gates used. The *depth* of a circuit is the maximum of all input node to output node distances. The *fan-in* of a

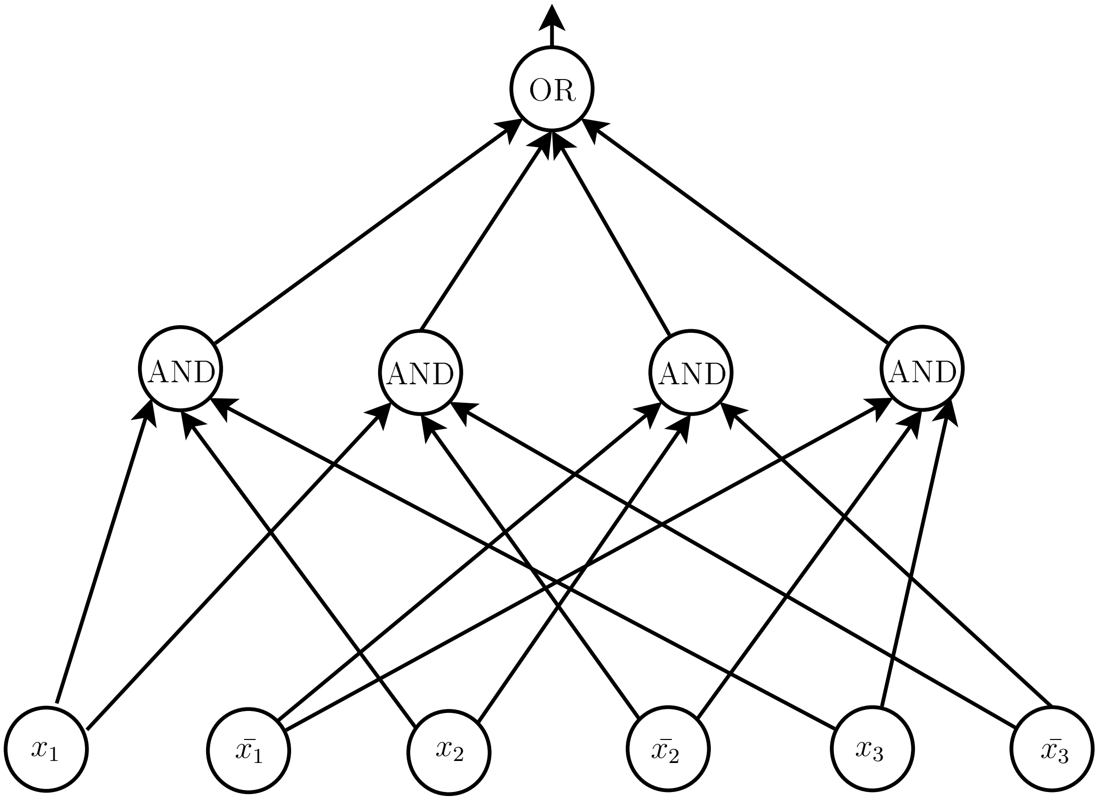

Figure 1–1: A circuit of size 5 and depth 2 computing PARITY of 3 bits

gate is its in-degree. The figure above, depicts a family of circuits whose size grows exponentially in the input length n and whose depth remains a constant.

Size in circuits roughly corresponds to time in Turing machines. Indeed, it is not hard to verify that any problem that can be solved in time $T(n)$ with a Turing machine can be solved by circuits comprising AND/OR gates of size $(T(n))^2$, which follows from the proof of the famous Cook's Theorem. In fact, circuits of just size $O(T(n)\log T(n))$, as shown by [PF77], can simulate an algorithm running on the Turing machine for time $T(n)$. Proving strong lower bounds on the size of circuits thus yields strong lower bounds on the running time of algorithms on a Turing machine. Several researchers in the eighties felt that circuits provide a clean combinatorial handle on computation as one can avoid dealing with messy features of Turing machines like moving heads and changing states. This feeling received a big impetus from the celebrated work of Razborov [Raz86]. Using a beautiful combinatorial argument, he showed that monotone circuits, i.e. circuits having AND/OR gates that do not access negated input variables, cannot compute the CLIQUE function in polynomial size.

The restriction to monotone circuits does not appear serious because the target function is itself monotone, i.e. if we add edges to our graph it does not destroy any clique that was present in the original graph. Intuitively, one expects that monotone functions have near-optimal circuits that are monotone. However, Razborov showed that MATCHING does not have polynomial size monotone circuits. A famous algorithm due to Edmonds shows that MATCHING has a polynomial time algorithm and hence poly-size non-monotone circuits[4] . This destroyed the intuition about computing monotone functions. Indeed, the progress on general circuits has been abysmally low. The best known lower bound on the size of unrestricted circuits computing an explicit function is less than $4.5n$ [LR01].

1.3.1 Circuits of Constant Depth

Depth in circuits corresponds to the notion of 'parallel time'. Indeed, the delay in propagating signal in digital devices is roughly proportional to the depth of the underlying circuit (assuming that all gates involved have same latency). Investigating depth needed for computing functions is thus a natural research direction. This direction was quite fruitfully pursued in the eighties. It has yielded some of the most beautiful results in the theory of lower bounds.

An obvious starting point here is to consider circuits of constant depth. It is not hard to see that if gates of such circuits have bounded fan-in, then the function computed can only depend on a constant number of input variables. On the other hand, allowing AND/OR gates of unbounded fan-in with constant-depth results in non-trivial computational power. Proving lower bounds against such circuits require even more non-trivial insight into computation. A series of dramatic work by Ajtai [Ajt83], Furst, Saxe and Sipser [FSS84], Yao [Yao85] and Håstad [Hås86] established the fundamental result that such circuits in sub-exponential size (i.e. $2^{n^{o(1)}}$ size)

[4] Much more recently, there are indications that MATCHING may be doable efficiently *in parallel* (see [AHT07])

cannot compute the parity of n boolean variables. More generally, the result shows that modular counting using AND/OR gates is inherently difficult.

On the other hand, modular counters are very much part of the basic building blocks in modern digital hardware design. A natural next step is to precisely determine what advantage is gained by allowing modular counting gates into our circuits. For any integer $m \geq 2$, define a MOD_m gate to be a boolean gate that outputs 1 if the number of its input bits that are set to one is not divisible by m. Unfortunately, the powerful techniques introduced in [Ajt83, FSS84, Yao85, Hås86] fail to work well in the presence of MOD_m gates. The best that one could say using this method was worked out in [Hås86] where it was essentially shown that few PARITY gates (fewer than $\Omega((\log n)^{3/2})$) does not help in significantly reducing (below $2^{\Omega(\log n)^{3/2}}$) the size of a constant-depth circuit computing the MAJORITY function.

MAJORITY has two crucial properties. It is a robust function whose value does not get determined by revealing the assignment to any sublinear number of its input bits. This is quite different from the AND and OR functions whose value gets fixed if any of its input bit is fixed to 0 and 1 respectively. The crucial technical ingredient of the works of [Ajt83, FSS84, Yao85, Hås86] showed that this weakness of AND and OR gates are propagated in some sense to the whole circuit if it is of constant depth and is entirely composed of these gates. The second property of the MAJORITY function is that it is severely *aperiodic*. MOD_m gates are of course *periodic* with a small period of m for any constant m. This makes MAJORITY a tempting target on which to prove lower bounds for size of circuits comprising AND/OR/MOD_m gates.

Developing a powerful machinery for approximating boolean functions by multivariate, multilinear polynomials of low degree over finite fields, Razborov [Raz87] proved exponential size lower bounds on the size of circuits having AND/OR/PARITY gates for computing MAJORITY. Building on this breakthrough work, Smolensky [Smo87] generalized the argument by replacing the PARITY with MOD$_{p^k}$ gates, where p is any arbitrary fixed prime and k is a fixed positive integer. A special case of Smolensky's argument yields (with a slight degradation of parameters) a new proof of the earlier exponential lower bounds on the size of constant depth circuits computing PARITY. After more than twenty years of its discovery, the Razborov-Smolensky argument remains a true gem of theoretical computer science.

Yet, the seemingly innocuous extension to composite modular counting has resisted attacks from a long list of several researchers (for example, see [BS95, BS99, BST90, Gre04, Gro94b, Gro98, GT00, KW91, HM04, MPT91, Smo90, ST06]). No non-trivial lower bounds are known for general constant depth circuits that employ MOD$_m$ gates when m has two distinct prime factors. While three generations of algorithm designers have in frustration called NP-complete problems intractable, it remains consistent with our current knowledge that circuits comprising *only* MOD$_6$ gates in depth three and linear size can compute these problems. Separating such depth-three circuits from NP is indeed one of the current frontiers in the theory of lower bounds.

Another direction, also very natural, is to consider constant depth circuits augmented with gates computing MAJORITY. We call them MAJ gates. The influential work of Minsky and Papert [MP88] considered a special case of such circuits called perceptrons. These are boolean gates that generalize a MAJ gate: every input to a perceptron is weighted by some real number and the gate outputs one iff the weighted sum of its inputs is positive and otherwise outputs -1. Artificial neural networks, using the perceptron as a building block, have been widely studied in the Artificial Intelligence and Machine-Learning communities as a reasonable model of neural activity in the human brain. It is known that such constant depth neural networks can be efficiently simulated by circuits comprising ordinary unweighted MAJ gates.

A series of results [ABFR94, BRS91b, Bei94, BS94] in the early nineties established strong lower bounds on constant-depth circuits augmented with *few* MAJ gates. Specifically, these series of results showed that circuits comprising AND,OR and MAJ gates cannot compute[5] in sub-exponential size the MOD_m function as long as the number of MAJ gates is restricted to $n^{o(1)}$. On the other hand, it is known that allowing more MAJ gates increases significantly the computational power of such circuits. In linear size and depth-two, circuits comprising *only* MAJ gates compute the MOD_m function, for every m. More surprisingly, in depth-three and quasi-polynomial size (i.e. $n^{O((\log n)^d)}$ for some constant d), circuits with only MAJ gates compute every function that can be computed by circuits of quasi-polynomial size and constant depth having AND/OR/MOD_m gates [Yao90, BT94].

This brings us to another frontier in the theory of lower bounds. Currently, we cannot prove a superlinear lower bound on the size of depth-three circuits comprising

[5] In fact, Barrington and Straubing [BS94] show that such circuits cannot even approximate well the MOD_m function.

only MAJ gates computing any function in NP. In other words, for every intractable problem, there *may* exist a shallow depth and small size neural network that solves the problem.

1.4 Polynomials over Rings

Multivariate polynomials over rings are classical objects in mathematics that have been studied in a wide variety of contexts since long. More recently, they have aroused major interest in the computing community after a string of impressive results in circuit complexity [Raz87, Smo87, ABFR94], interactive proofs [LFKN92], communication complexity [She07], learning theory [LMN93, Kli02] and quantum computing [AS04, BCW98, Raz03, She08b] have been obtained with polynomials playing a central role.

Many of these results use polynomials as a tool to analyze a given problem. A little differently, the Razborov-Smolensky argument for showing limitations of constant-depth circuits having AND/OR/MOD$_p$ gates, implicitly views polynomials as non-uniform models of computation. The work of Barrington, Beigel and Rudich [BBR94] and Nisan and Szegedy [NS94] initiated a systematic study of the power of polynomials in representing/computing boolean functions.

More precisely, let a polynomial P over $\mathbb{Z}_m$ with n variables $x_1, \ldots, x_n$ represent the boolean function $f : \{0,1\}^n \to \{0,1\}$ if there exists an *accepting set* $A \subseteq \mathbb{Z}_m$ such that $f(x) = 1$ iff $P(x) \in A$, for each $x \in \{0,1\}^n$. It is worth noting that since our interest is on the behavior of P over the boolean hypercube where $x_i^2 = x_i$ for each variable x_i, we conveniently henceforth assume w.l.o.g that P is *multilinear*. The resource that is of interest in this model is the *degree* of P. The basic question of the subject is “How much degree is needed by a polynomial to represent the boolean function f over $\mathbb{Z}_m$?” when m is fixed. This quantity is called the MOD$_m$-degree of f.

The work of Razborov-Smolensky provides answers to such questions, when m is a prime power. For instance, one can show that the OR function has $\Omega(n)$ degree if m is a prime power. But the method fails, as explained in detail by Barrington [Bar92], as soon as m contains two distinct prime factors. Quite surprisingly, the model of polynomials reveals a non-trivial computational advantage of composite numbers over their primal counterparts. Barrington et.al. [BBR94] show that there exists a polynomial of degree $O(n^{1/r})$ over $\mathbb{Z}_m$ computing the OR function when m has r distinct prime factors. Similar advantages to represent the MOD$_q$ function, for some special q that are co-prime with m, have been subsequently discovered by Hansen [Han06b].

Our lack of understanding of the computational power of modular counting is best exemplified in the setting of low degree polynomials. Indeed, it is perplexing that no function $f \in \mathrm{NP}$ is known such that the MOD$_6$-degree of f is super-logarithmic i.e. $\omega(\log n)$. A simple counting argument, on the other hand, reveals that most functions have linear degree.

1.5 ‘Number on the Forehead’ Model of Communication

A beautiful theory of communicating processes has been developed starting with the seminal paper of Yao [Yao79]. In the model proposed by Yao, there are two players, Alice and Bob, who wish to collaboratively compute a boolean function f. The problem is that the set of input bits of the function is partitioned into two sets

X_A and X_B. Alice has only access to the bits of X_A and Bob to those in X_B. They decide, a priori, upon a protocol for communicating with each other with the goal that both of them can determine the value of f on any assignment to its input bits. Further, they want to minimize the amount of bits they need to exchange with each other for achieving this goal. In order to entirely focus on bits communicated as a resource, Alice and Bob are endowed with unlimited computational power in terms of time and space. The simple question that is of intrinsic interest is "How many bits do Alice and Bob need to communicate to compute f with the best protocol?". The amount of communication taking place is measured with respect to the size of the set of input bits assigned to each player. Assuming that each player holds n-bits of information, every function can be computed trivially by communicating $n + 1$ bits.

Exploration around this theme has uncovered a rich underlying structure of the model. A thorough exposition of this theory, now known as Communication Complexity, is given in the excellent book by Kushilevitz and Nisan [KN97]. Surprisingly, an ever expansive set of diverse applications of this theory to other fields in theoretical computer science is being discovered. For instance, a powerful technique to prove lower bounds on the depth of monotone boolean circuits was developed using a variant of this model by Karchmer and Wigderson [KW88] that was further developed in the work of [KRW95, RW92, RM97]. Very interesting trade-off results between the resources of time and space have been derived using communication complexity in the work of [BSSV00, BV02]. Connections with randomness extraction from imperfect random sources was established in the work of Vazirani [Vaz85], Chor and Goldreich [CG85]. Indeed, the list of applications goes on and on and Communication Complexity has been fondly called the 'Swiss-army Knife' of complexity theorists.

The two-party model of Yao extends to the multiparty model in more than one way. The first one is called the 'Number in the Hand' model where the set of input bits is partitioned into k sets $X_1, \ldots, X_k$. Player i gets X_i. In this model, the more players there are, the less information is directly accessible to each player (assuming each player gets access to equal number of bits). This is known to weaken the power of the two-player model, although it has been studied for applications in areas like data-streams [CKS03, CCM08]. Our concern here is with the other extension to multiparty introduced by Chandra, Furst and Lipton [CFL83] called the 'Number on the Forehead' (NOF) model. In this model, input bits of X_i are held on the forehead of Player i. In other words, each player has access to all input bits (written on the foreheads of other players) except those that are held on his own forehead.

There are several features that make the model quite powerful. In particular, there is an *overlap* of information accessible to players which can be used to save communication significantly even with three players. Grolmusz [Gro94a] devised a clever protocol exhibiting the surprising power of $\log n$ players, where n is the number of bits written on the forehead of each player. Other non-obvious k-party protocols have been discovered (see, for example, [Amb96, CFL83]). Proving both lower bounds and upper bounds for this model is very challenging. On the other hand, many rewarding applications of strong lower bounds on the multiparty communication complexity of a function exist. They can be used to prove lower bounds on resources needed in various other important models like branching programs [CFL83], constant-depth circuits [HG91] and proof systems [BPS05]. In fact, many other such applications are known, while proving the lower bounds themselves in the model have evaded efforts [KN97].

One such application is of great interest for the research described in this thesis. Recall that no superlinear lower bounds exist on the size of depth-three circuits comprising only MOD_6 gates. It is however known from the work of Yao [Yao90] and Beigel-Tarui [BT94], that super-polylogarithmic (i.e. $(\log n)^{\omega(1)}$) lower bounds on the k-party communication complexity of a function f for some very restricted protocols is enough to show that constant-depth circuits having $\mathrm{AND}/\mathrm{OR}/\mathrm{MOD}_m$ gates cannot compute f in quasipolynomial size, provided $k = (\log n)^{O(1)}$. The seminal work of Babai, Nisan and Szegedy [BNS92] introduced[6] a powerful method, called the Discrepancy Method, to obtain the first strong lower bounds on the multiparty communication complexity of functions. However, the technique in [BNS92] stopped short of proving non-trivial bounds for $\log n$ players. It is now believed that fundamentally new ideas are needed to sail past the $\log n$ players barrier.

On the other hand, there is evidence that we do not quite understand the model even when fewer players are involved. There are several simple and natural functions whose three-party communication complexity is not known. In fact, until recently,

[6] The Discrepancy Method existed in mathematics before the work of [BNS92]. Here we mean that it was introduced to multiparty communication complexity by [BNS92].

no superlogarithmic (i.e. $\omega(\log n)$) lower bound was known for three players for these functions. A systematic study of the different aspects of this model is compelling in its own right.

1.6 Our Contributions

Constant-depth circuits. In Chapter 3, we make progress towards understanding the computational power of circuits of constant depth comprising AND,OR and MOD_m gates, when m is an arbitrary fixed positive integer. We approach this from two directions. In the first part of the chapter, we probe the limitations of such circuits when the number of MOD_m gates allowed in the circuit is restricted. We show that indeed computing MAJORITY and MOD_ℓ by such circuits requires superpolynomial size when ℓ contains a prime factor that does not divide m. This result is expressed formally in Theorem 3.1. The result first appeared in joint work with Kristoffer Arnsfelt Hansen [CH05] and at the time represented the best known lower bounds on the size of such circuits (with few MOD_m gates) computing MOD_ℓ. It still remains the best known lower bound for computing MAJORITY. The main technical novelty introduced in this part is a connection with a new notion of computation of boolean functions by polynomials that we describe in the next section.

In the second part of Chapter 3, we shed light on the limitations of modular counting by allowing only MOD_m gates in our circuits. We show that (non-constant) functions computed by such circuits of sublinear size (and arbitrary depth) should have a large support set (see Theorem 3.4). Consequently they cannot compute AND in sublinear size, as AND has a support set of size one. Such a result was first proved by Thérien [Thé94], but our bounds are sharper and our techniques are different.

The main technical ingredient used is a result about linear maps that is stated in Theorem 3.19. We further show that such circuits in sublinear size cannot compute MOD_ℓ when m and ℓ are co-prime. This result is a significant improvement over the previous best lower bound of $\log n$ due to Smolensky [Smo90]. Smolensky's result said nothing about the approximability of MOD_ℓ by such circuits. On the other hand, Theorem 3.5 shows that such circuits of sublinear size do not even approximate MOD_ℓ well: a MAJORITY gate needs to seek votes from exponentially many such circuits to correctly compute MOD_ℓ. We derive this result by proving a Uniformity Lemma (see Lemma 3.20) for every system of linear polynomials. Uniformity Lemmas are interesting in their own right and we prove ours using an exponential sum argument. We believe that exponential sums will play a crucial role in developing new techniques for circuit complexity. Results in this part are based on a joint work with Navin Goyal, Pavel Pudlák and Denis Thérien [CGPT06].

In Chapter 7, we prove lower bounds on the size of some depth-three circuits that follow as a consequence of our work on Communication Complexity in Chapter 6. Recall that we do not know if depth-three circuits comprising only MAJ gates can compute every function in NP. On the other hand, Yao [Yao90] has shown that such depth-three circuits in quasipolynomial size can simulate every function computable by constant-depth circuits of quasipolynomial size and comprising AND,OR and MOD_m gates, even when the fan-in of the bottom gates are restricted to polylogarithmic. In contrast, we show that if the bottom fan-in is further restricted to $o(\log\log n)$ then such circuits cannot compute much simpler functions efficiently.

In particular, in quasipolynomial size they cannot compute a function that is computed by a linear size depth-three circuit comprising *only* AND and OR gates (see Theorem 7.1). This result first appeared in [Cha07b].

Polynomials over rings. In Chapter 3, we relax the notion of computation by a polynomial over $\mathbb{Z}_m$ of a boolean function to a weak computation that allows for errors. The polynomial is allowed to give false negative answers but no false positives and it must output a positive answer on at least one input. This model generalizes all models of computation by polynomials over finite rings considered so far in the literature. We prove lower bounds on the degree needed by any polynomial over $\mathbb{Z}_m$ to represent the MAJORITY (Theorem 3.10) and MOD_ℓ (Theorem 3.11, m, ℓ are co-prime) function in this generalized sense. Our argument for establishing Theorem 3.11 makes a novel combination of a combinatorial argument due to Tardos and Barrington [TB98] and a Fourier theoretic argument due to Green [Gre00]. As we show, our bounds are strong enough to yield lower bounds on the size of circuits with MOD_m gates computing the same functions. These bounds are not known to follow directly from either the work of [TB98] or [Gre00].

In Chapter 7, we simplify the breakthrough work of Bourgain [Bou05] that settled a long line of research [CGT96, Gre99, AB01, Gre04] on the *correlation* of low degree polynomials over $\mathbb{Z}_m$ and MOD_ℓ function. In this model, polynomials are allowed to err on both positive (true) and negative (false) inputs of the boolean function that they represent. However, unlike the previous model, we count the number of errors that the polynomial makes. Bourgain's work proves exponentially small upper bounds on the correlation between functions computed by low degree polynomials over $\mathbb{Z}_m$ and MOD_ℓ. We sharpen this result (see Lemma 7.7). Moreover, we show a close correspondence between the proof technique of the seminal result of Babai, Nisan and Szegedy [BNS92] for obtaining upper bounds on discrepancy in the context of communication complexity and our argument to upper bound the correlation of polynomials over $\mathbb{Z}_m$ with MOD_ℓ. In retrospect, the result on correlation should have been obtained much earlier.

Communication complexity. One can naturally define the notion of protocols deterministically, non-deterministically and randomly computing functions. Our work concerns all three models and their relationship to each other.

In Chapter 5, we obtain new insight into the multiparty model by considering the class of functions that can be computed deterministically by k players in constant cost (denoted by CC_k), for some fixed k. A priori, there is no reason to suspect that this class is related in some way to circuit complexity classes. Yet, Szegedy [Sze93] obtained several beautiful algebraic and combinatorial characterizations for the class CC_2. Consequently, he was able to show that every function in CC_2 can be computed by linear size shallow circuits comprising AND/OR/MOD_m gates. In contrast, Corollary 5.6 shows, making use of specially crafted codes, that even three players in constant cost can compute functions with exponentially large circuit complexity, ruling out any simple characterization for CC_k with $k \geq 3$. Our proof of this result exploits the following two features of the model: a) Overlap of information, i.e. every input bit is visible to two other players. b) Each player knows the precise position of every input bit that it sees. While it was already known that removal of the first feature renders the model weaker than the two player case, the significance of the second feature had never been investigated before. We consider two simple classes of functions in which intuitively one expects that the second feature does not afford any advantage. Using Ramsey theoretic arguments we prove the following: a) Every function f having a neutral letter that is in CC_k, for some fixed k, is regular[7] (see Theorem 5.7). b) A symmetric function is in CC_k, for some fixed k, iff it is in CC_2 (Theorem 5.8). These results first appeared in a joint work with Andreas Krebs, Michal Koucky, Mario Szegedy, Pascal Tesson and Denis Thérien [CKK+07].

In the first part of Chapter 6, we prove strong lower bounds for the multiparty communication complexity of some simple functions that had resisted attacks from several researchers in the past. In particular, there was no known function computed efficiently in constant depth by circuits comprising AND/OR gates that required large three party communication. Extending the work of Sherstov [She07], we exhibit such a simple function that requires large communication by even randomized protocols that are required to perform better than random guessing by a very thin margin (see Theorem 6.1). The main technical component of this work, called the Orthogonality-Discrepancy Lemma, is a new relationship between the property of a boolean function being orthogonal to low degree polynomials and the discrepancy of a closely related function (see Lemma 6.8). This allows passage from a well-known

[7] Function f has neutral letter e, if inserting or deleting e at any place in each input word does not change the value of f on the word. Note that a boolean function f induces a language L_f in an obvious way i.e. $x \in L_f$ iff $f(x) = 1$. Function f is called regular precisely if L_f is regular.

algebraic measure of complexity of boolean functions, called *voting degree*, to communication complexity (see Corollary 6.12). Such a passage was first devised, in the context of two-player communication protocols, by Sherstov [She07]. Our result first appeared in [Cha07b].

In the second part of Chapter 6, we exhibit a function whose non-deterministic communication complexity is small ($\log n$) but requires large ($n^{\Omega(1)}$) communication by k-party randomized protocols achieving a bounded advantage over random guessing (see Theorem 6.2). This settles a major open question in multiparty communication complexity (see [BPS, BPSW06, BDPW07, Cha07a, VW07b]). Determining the relative power of determinism, non-determinism and randomization is a central theme of theoretical computer science. The celebrated P vs. NP question is an exploration of this theme in the Turing machine model. Our result answers a question on the same theme in the model of multiparty communication. Further, it proves superpolynomial lower bounds on the length of proofs in an important class of proof systems, called Lovász-Schrijver proofs (see [BPS] for details). Our result appeared as a joint work with Anil Ada [CA08]. A similar result has been independently obtained by Lee and Shraibman [LS08]. Finally, in Section 6.6, we extend the recent work of Shi and Zhu [SZ07] to the multiparty model. It was not known if such an extension existed and was suggested as a direction of investigation in the recent survey by Sherstov [She08a]. We, on the other hand, show that our extension is powerful enough to also yield $n^{\Omega(1)}$ lower bounds on the k-party communication complexity of Disjointness. This provides a second proof of an important result.

CHAPTER 2
Background for Boolean Circuits

In this chapter, we formally define the complexity classes for boolean circuits. We also recall some of the known arguments for proving lower bounds for constant depth circuits that serve as the starting point of our investigation. We further highlight the difficulties that are faced when one tries to employ similar arguments for more powerful circuits. In the process, we develop the Razborov-Smolensky theory of polynomial representation of boolean functions.

We point out that while our review of complexity classes is brief and targeted towards placing our work in the larger context, an interested reader can consult any excellent textbook on Computational Complexity (for example [AB09, Pap94]) to get a more thorough treatment of issues.

2.1 Boolean Circuits

We recall from Chapter 1 that the first key resource of the model of boolean circuits is its size i.e. the total number of gates used in the circuit. Size, quite closely, corresponds to running time in Turing machines. The class of boolean functions that can be computed by boolean circuits of polynomial size is denoted by P/poly. This corresponds to the non-uniform version of the class P defined for the Turing machine. Most proponents of the conjecture P $\neq$ NP, in fact, have the stronger belief that the class NP is not contained in even P/poly. This stronger statement is a more natural target to aim for in the context of boolean circuits.

As we said earlier, Razborov's result [Raz86], showing that monotone circuits of polynomial size cannot decide if an input graph has a clique of prescribed size, is the closest that we have come to proving this conjecture.

Unfortunately, Razborov himself [Raz89] showed that the method of approximations that he employed to obtain his results cannot yield super-linear lower bounds on the size of non-monotone circuits. Subsequently, other obstacles in the form of "natural proofs" [RR97] were identified. Recently, Aaronson and Wigderson [AW08] pointed out an additional barrier called 'algebrization'. The idea of these papers is to show that most known lower bound proofs naturalize [RR97] and algebrize [AW08]. Further, they show that, widely believed cryptographic assumptions get violated if one finds such proofs (that algebrize or naturalize) showing that a function in NP is not contained in P/poly. However, fresh hope emerges from the very recent work of Chow [Cho08] that shows there are no known barriers to obtaining such a results by 'slightly tweaking' natural proofs. In any case, most of the the complexity classes that we study in this work are not known to present any great barrier. Yet, progress on them has been limited.

Our interest is to consider circuits of restricted depth. Besides being a natural restriction, such circuits also intuitively capture the notion of highly parallel computation. For every integer $i \geq 0$, let NC^i denote the class of circuits that have polynomial size, $O(\log n)^i$ depth and use binary AND and OR gates. Define

NC $= \cup_i$NC^i. The following sums up the known relationship (among non-uniform classes)[1] :

$$\mathrm{NC}^0 \subsetneq \mathrm{NC}^1 \subseteq \mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{NC}^2 \subseteq \cdots \subseteq \mathrm{NC} \subseteq \mathrm{P/poly} \tag{2.1}$$

A fairly straightforward counting argument shows that a random function, with probability asymptotically tending to one, needs exponential size circuits to be computed (even when depth is unrestricted). It however is a recurring theme of the subject, that finding an explicit function that cannot be computed using limited resources is very challenging even though one knows that most functions are hard for the model. No explicit function in NP is known to be not in NC^1. In fact, as we shall see below, we cannot prove any such explicit function to be not contained in even some subclasses of NC^1 where circuits are further restricted to have *constant depth.*

A word about our 'abuse' of notation for circuit complexity classes is in order. Assume A is a circuit class. We use A with two different connotations. The first refers to the class of functions that have polynomial size computations over circuits of a certain type over which the complexity class A is defined. In the second use, A means the underlying circuit model (as opposed to a class of functions). This

[1] NC stands for "Nick's class" as coined by Steve Cook to honor Nick Pippenger. Pippenger reciprocated the gesture by coining "Steve's class" (SC). We will not have the occasion to consider the class SC in this work.

is illustrated by the following two simple examples that respectively invoke these connotations: The function MAJORITY is in NC^1. The function PARITY can be computed by linear size NC^1 circuits. The particular sense in which we are referring to a circuit complexity class is clear from the context.

2.1.1 Circuits of Constant Depth

Before we move on, let us fix some more terminology. Conventionally, theoretical computer scientists have visualized the flow of information in a circuit upwards i.e. the input variables are at the bottom and the output gate is at the top[2]. Henceforth, we further assume that our circuits are layered in the following sense: Layer 0 consists of input variables and their negations. Each gate in Layer i receives its inputs only from gates in Layer $i-1$, for $i \geq 1$. Each gate in Layer 1 is called a *bottom gate*. The maximum fan-in of a bottom gate is called the *bottom fan-in* of the circuit. The fan-in of the output gate is called the *top fan-in* of the circuit. Let gates of a circuit of depth k have gates of type G_i at Layer i. We denote such a circuit by $G_k \circ G_{k-1} \circ \cdots \circ G_1$.

Note that NC^0 is the class of functions computable by circuits with constant depth, polynomial size and binary fan-in AND/OR gates. Thus, such functions do not even depend on all of the input bits. Consequently, this class is quite weak[3]:

[2] It seems to us that depicting the flow of information from top to bottom is more reasonable. To save confusion, we however follow convention.

[3] Note that this class is interesting in other contexts. For instance, there is evidence now that many cryptographic primitives can be computed in NC^0 [AIK04].

for instance, they do not contain the simple boolean AND and OR functions. This motivates the introduction of the class AC^0 : functions computable by circuits having unbounded fan-in AND/OR gates and constant depth. It is worthwhile to note that such circuits in depth-2 and exponential size can compute every boolean function. More interestingly, they can add two n-bit integers in depth five and cubic size. In depth two and polynomial size, they can compute THR_k for any constant k by an exhaustive verification. Much more surprisingly, in polynomial size they can compute $\mathrm{THR}_{(\log n)^c}$, for any constant c (see [FKPS85, WWY90, RW91]). There are other quite non-trivial algorithms that can be executed by such circuits. One may well expect[4] that proving lower bounds on resources in a model that allows such subtle computations to take place, will be a challenge!

A natural question to probe, is if the weakness of a bounded depth circuit is closely related to the weakness of its constituent gates. The weakness of an AND (OR) gate is that fixing any one of its input to 0 (1) fixes its output. This gives the hope that if an AC^0 circuit has not too many gates, then it should be possible to fix the output of the circuit by just fixing a few input variables to zero and one. If that were true, such circuits in small size would not compute a 'robust' function like PARITY, which does not get fixed even when all but one variable gets fixed. This intuition first got formalized and verified by the work of Ajtai [Ajt83] followed by

[4] It is worth noting that most of these positive results with AC^0 were obtained after strong lower bounds had been shown.

that of Furst, Saxe and Sipser [FSS84]. Furst, Saxe and Sipser deliver a beautiful probabilistic argument by introducing the powerful notion of a *random restriction.*

We sketch below the essence of the argument in [FSS84]. Let $\rho = \{0, 1, *\}^n$ define a restriction of the input variables, where an assignment of $*$ to a variable signifies that the restriction leaves it free (i.e. does not set it). Define a probability distribution μ on restrictions in the following way: Independently assign each variable to $*$ with probability $1/\sqrt{n}$ and with equal probability, i.e. $(1 - 1/\sqrt{n})/2$, assign it to 1 and to 0. Define a gate *wide* if it has fan-in at least $c \ln n$ and otherwise call it *narrow.* It is not hard to verify then the following key observation:

Observation 2.1 *A restriction ρ chosen randomly according to μ satisfies the following:*

- *A wide AND/OR gate is not forced to 0/1 with probability $o(n^{-c/4})$.*
- *Each narrow gate has more than c inputs assigned to $*$ with probability at most $o(n^{-c/4})$.*

Additionally, we expect a random ρ to leave $\sqrt{n}$ input variables of the circuit assigned to $*$. Thus, with $c = 8k$, a circuit of size n^k when hit by a random restriction, results in a circuit with at least $\sqrt{n}/2$ variables left free and all of whose base gates have fan-in at most c. As a final step, Furst, Saxe and Sipser analyze depth-2 circuits whose base gates have small (constant) fan-in. With a more involved argument, they show the following:

Lemma 2.2 *For fixed integers $c, k > 0$, there exists a constant $b_c = 4k + 2^{4k} b_{c-1}$ satisfying the following: Every depth-2 circuit of size n^k, all of whose base circuits of depth 1 have fan-in at most c, when hit with a restriction chosen randomly according* *to μ computes a function of at most b_c input variables with probability at least $1 - o(n^{-k})$.*

The power of the lemma above becomes evident by applying it repeatedly to obtain a switching effect as following: applying Observation 2.1, we first find a restriction that leaves enough (at least $\sqrt{n}/2$) variables free and decreases the bottom fan-in to a constant c. Applying a second round of random restriction on the erstwhile free variables, Lemma 2.2 ensures that each depth-2 circuit computes a function of a constant (i.e. b_c) number of variables. Every such function can be written as both a AND $\circ$ OR and a OR $\circ$ AND circuit of size at most 2^{b_c} (which is a constant). This allows us to switch from a circuit of type AND $\circ$ OR to a circuit of type OR $\circ$ AND or vice-versa. Once the depth-2 circuits are appropriately switched, the second and third layers can be merged decreasing the depth of our original circuit by one, i.e. we move from depth d to $d-1$. Meanwhile our bottom fan-in has changed from c to b_c (a double exponential blow-up in k). The bootstrapping process is complete and we go on applying successive random restrictions, each of which decreases the depth of our circuit by one, increases its size by a constant factor and increases the bottom fan-in (that still remains a constant). At each step, we also decrease the size of the set of free variables by about a quadratic factor. We do this a constant $(d-2)$ number of times to reach a state where our restricted circuit is computing a function of constant number of variables despite the fact that there are about $\Omega(n^{1/2^{d-1}})$ variables remaining free. The robustness of a function like PARITY finishes the argument by supplying a contradiction. The restricted circuit has to compute either

PARITY or its complement on the remaining variables which contradicts the fact that it is computing a function of merely a constant number of variables.

Tracing the various blow-ups carefully as one applied Lemma 2.2 to successive restrictions, one concludes a lower bound of $n^{\Omega(\log n)}$ on the size of AC^0 circuits computing PARITY. The key ingredient in the argument is the ability to switch from a DNF to a CNF with small blow-up. A lemma like Lemma 2.2 that allows one to do so is called a Switching Lemma in the literature. Switching Lemmas have played a major role in obtaining lower bounds in various other models. In the context of constant-depth circuits, after improvements made by Yao [Yao85], work on them in the mid-eighties culminated in the powerful work of Håstad [Hås86]. Håstad's Switching Lemma also yields optimal (exponential) lower bounds on the size of AC^0 circuits computing PARITY.

We state a version of the Switching Lemma that is due to Beame [Bea94] and is convenient to use in our work. In order to do so, let us recall the well-known notion of a *decision tree*. A decision tree is a rooted binary tree, each of whose internal nodes are labeled by one of the n input variables. For every node, one of its outgoing arc is labeled 0 and the other 1. The leaves of a decision tree are labeled 0 and 1 and along each path from the root to a leaf no label on a node is repeated. Given an assignment of input variables, computation by a tree proceeds along a path in the following way: starting from the root, each node queries the variable used to label it and then follows the arc labeled with the answer to reach the next node. The process is repeated with the next node until we hit a leaf at which point the tree outputs the label of the leaf. It is easy to verify that the set of inputs that correspond to a computation along a given path in the tree is disjoint from the set of inputs that correspond to computation along a different path. This simple feature of a decision tree makes it very handy for our applications.

As with any other tree, the height of a decision tree is the length of the longest path and it size is the number of internal nodes. It is straightforward to verify that every boolean function has a decision tree of at most linear height and exponential size. The resources in this model are the height and size of the tree. They are, of course, not unrelated as for instance a tree of logarithmic depth can have at most polynomial size.

Remark 2.3 *A boolean function computed by a decision tree of height h has a DNF (and a CNF) formula with each term of size at most h.*

Armed with these notions, we are ready to express the powerful effect of random restrictions on constant depth circuits. Let R_n^ℓ be the set of all restrictions that leave precisely ℓ of n variables free.

Lemma 2.4 (Beame's Switching Lemma) *Let f be a DNF (or CNF) formula in n variables with terms of length at most r. Let $\ell = pn$ and pick ρ uniformly at random from R_n^ℓ. Then the probability that f_ρ cannot be computed by a decision tree of height at most d is less than $(7pr)^d$.*

Beame's version of the Switching Lemma readily yields an exponential lower bound on size of constant-depth circuits computing PARITY.

Corollary 2.5 *A circuit of depth d, using unbounded fan-in AND/OR gates cannot compute PARITY if it has size less than $2^{\Omega(n^{1/d-1})}$.*

Proof:[adapted from Beame [Bea94]] Let the size of the circuit computing PARITY be S. We successively apply random restrictions, one for each layer of the circuit. Let $p_1 = 1/14$ and $p_i = 1/(14 \log S)$ for $i = 2, \ldots, d-1$. W.l.o.g. assume that the base layer is of OR gates. Each OR gate can be thought of as a DNF with term size 1. We apply the Switching Lemma with $p = p_1 = 1/14$ and $d = \log S$ and $r = 1$ to each OR gate in the first layer. Under a random restriction from $R_n^{n/14}$, each restricted gate fails to be computed by a decision tree of height at most $\log S$ with probability less than $2^{-\log S} = S$. Since there are at most S gates in the first layer, there exists a restriction $\rho_1 \in R_n^{n/14}$ that succeeds in restricting the height of decision tree to $\log S$ for each OR gate at the base layer.

We show by induction of depth that there exists $(d-1)$ successive restrictions $\rho_1, \rho_2, \ldots, \rho_{d-1}$ with $\rho_i \in R_{n_{i-1}}^{n_i}$ where $n_i = p_i n_{i-1}$ and $n_0 = n$, such that after applying ρ_i the output of each gate at the ith layer is computed by a decision tree of height at most $\log S$. The base case of this induction has been established above for the base layer, i.e. $i = 1$. If the $i+1$th layer is that of AND (OR) gates, we compute the corresponding CNF (DNF) formula for each restricted gate in the ith layer from its decision tree as per Remark 2.3. Thus, the output of each gate in the $i+1$th layer of the restricted circuit can be again expressed as a CNF (DNF) formula. Then apply the Switching Lemma to each such formula by choosing a random restriction from $R_{n_i}^{n_{i+1}}$. Again, each formula fails to be restricted to a decision tree of height $\log S$ with probability less than $(1/2p_{i+1} \log S)^{\log S} = S$. So, there exists a restriction that does not fail for any formula. This completes the induction.

Thus, after applying $d-1$ restrictions, we have n_{d-1} variables free and a depth-2 circuit with bottom fan-in $\log S$ either computing PARITY or its complement on these free variables. Hence, $\log S \geq n_{d-1} = n/(14(14 \log S)^{d-2})$, yielding the required bound on S. ∎

The above bound is not only an exponential lower bound for constant d but is quite close to being optimal as the following fact shows:

Fact 2.6 *Let $d > 0$ be an even number. There is an AC^0 circuit of size $2^{O(n^{2/d})}$ and depth d that computes PARITY.*

Proof: The circuit is built using a simple divide and conquer strategy. The circuit has $d/2$ sections and the outputs of Section i are fed into the inputs of Section $i+1$. Each section has depth 2 and the total number of inputs in Section i is $n_i = n/(n^{2(i-1)/d})$. Further, n_i is split into equal blocks of size $n^{2/d}$. In Section i, we compute in parallel the parity of each block. This is accomplished by using the obvious depth-2 exponential size circuit for each block. Thus, the total number of gates in a section is $2^{n^{2/d}} \times (n_i/n^{2/d}) < n2^{n^{2/d}}$. As there are $d/2$ sections, we get a total depth of d and total size less than $nd2^{n^{2/d}}$. ∎

2.1.2 Modular and Threshold Counting gates

The previous circuit for PARITY can be easily modified to show that in $\log n$ depth, one can compute PARITY in linear size using *binary fan-in* AND/OR gates[5] .

[5] In fact, using the same divide and conquer strategy, every regular language can be computed in linear size and logarithmic depth using bounded fan-in AND/OR gates. The non-boolean letters of the alphabet may be encoded as boolean strings in any reasonable way.

Thus, Parity witnesses a clean separation of AC^0 from NC^1. This is one of the rare unconditional explicit separations of complexity classes. Several other natural functions are also not in AC^0 (as first observed in [FSS84]) because PARITY *reduces* to them.

The notion of a reduction is a very standard one in complexity theory to express the relative hardness of two problems. This is the notion that gives rise to the idea of completeness of a problem in a complexity class (for instance NP-completeness). In the context of circuits, we say a boolean function f AC^0 reduces to function g, denoted by $f \leq^{\mathrm{AC}^0} g$, if one can compute f in constant depth and polynomial size using AND/OR gates and gates computing the function g.

Observation 2.7 *$Thr_t \leq^{AC^0}$ MAJORITY.*

Proof: If $t \leq n/2$ ($t > n/2$), then by feeding $(\frac{n}{2} - 1)$ constant ones (zeroes) to a MAJ gate, we make it compute Thr_t. ∎

Observation 2.8 (Furst, Saxe and Sipser [FSS84]) *PARITY $\leq^{AC^0}$ MAJORITY.*

Proof: The basic intuition is that MAJORITY allows you to count precisely the number of ones occurring in a boolean string. This is because the number of ones in a n-bit string x is t iff $\mathrm{Thr}_t(x) = 1$ and $\mathrm{Thr}_{t+1}(x) = 0$. Thus,

$$\mathrm{PARITY}(x) = \vee_{0 \leq 2i \leq n}\big(\mathrm{Thr}_{2i}(x) \wedge \neg\mathrm{Thr}_{2i+1}(x)\big)$$

Observing that $\neg\mathrm{Thr}_k(x_1, \ldots, x_n) = \mathrm{Thr}_{n-k+1}(\neg x_1, \neg x_2, \ldots, \neg x_n)$ and using Observation 2.7, we are done. ∎

The argument above shows something slightly stronger. A boolean function is called a *symmetric function* if its value depends just on the number of input bits set to 1. PARITY, MAJORITY, Thr_t are all symmetric functions. The argument above shows the following:

Fact 2.9 *Let SYMM be an arbitrary symmetric function. Then, SYMM $\leq^{AC^0}$ MAJORITY.*

In the light of these observations, a series of natural questions emerge from the separation of AC^0 from NC^1: How does the computational power of the model get changed, if we allow PARITY or other modular counting gates in addition to AND/OR gates in our circuit? How is it altered, if we allow gates computing MAJORITY (denoted by MAJ) in our circuits? Define $\mathrm{ACC}^0[m]$ to be the class of functions computed by constant depth polynomial size circuits consisting of unbounded fan-in AND, OR and MOD_m gates. Barrington [Bar86] defined ACC^0 as $\cup_{m\geq 2}\mathrm{ACC}^0[m]$. Define TC^0 to be the class of functions that can be computed by circuits using only MAJ gates in constant depth and polynomial size. Note that by our previous observations, augmenting TC^0 circuits with AND/OR/SYMM gates does not give us additional power, where a SYMM gate computes an arbitrary symmetric function. In fact, Hajnal et.al. [HMP+93] observe that slightly modifying the proof of Observation 2.8 shows that every symmetric function can be computed by TC^0 circuits in depth-2 and linear size. Thus, the class of functions computable by constant-depth circuits of polynomial size using gates computing arbitrary symmetric functions is precisely TC^0. A non-trivial fact is that MAJORITY of n bits can be computed by a circuit of polynomial size and $O(\log n)$ depth that has only binary fan-in AND/OR gates. To sum up, we have the following refined view:

$$\mathrm{NC}^0 \subsetneq \mathrm{AC}^0 \subsetneq \mathrm{ACC}^0 \subseteq \mathrm{TC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{L/poly} \subseteq \mathrm{NL/poly} \subseteq \mathrm{P/poly}$$

ACC^0 is the smallest naturally arising complexity class which cannot be separated from NP. Yet, no function in ACC^0 is known whose computation makes 'clever' use of modular counting gates. In contrast, several interesting computations exist with TC^0. Modifying the argument of Observation 2.8, one can show that every symmetric function can be computed in linear size by depth-2 $\mathrm{MAJ} \circ \mathrm{MAJ}$ circuits. With more care, one can sort n integers, each n-bit long in TC^0. One even can multiply n integers (n-bits long) and divide[6] two such integers [BCH86, Rei87]. Although most researchers believe that ACC^0 is a strict subclass of NC^1 (and even of TC^0), a substantial number of researchers believe that TC^0 and NC^1 are the same (see, for example, [AW93]). An interesting consequence of such a collapse is that TC^0 in that case can be simulated by polynomial size threshold circuits of some fixed depth k.

2.1.3 Polynomials and the Case of Prime Modulus

Although we do not know the power of $\mathrm{ACC}^0[m]$ in general, a beautiful argument due to Smolensky [Smo87], generalizing the earlier breakthrough work of Razborov [Raz87], pins down the weakness of such circuits when m contains only one prime factor, i.e. $m = p^k$ for some prime p. It shows that $\mathrm{ACC}^0[p^k]$ circuits cannot compute the MOD_q function in sub-exponential size if p, q are two distinct primes.

[6] More recently, in a breakthrough work [HAB02], it has been shown that division can be done by an 'extremely uniform' version of TC^0.

Theorem 2.10 (Razborov-Smolensky) *$ACC^0[p^k]$ circuits of depth d cannot compute the MOD_q function using $2^{n^{o(1/2d)}}$ AND and OR gates.*

The work in [Raz87, Smo87] introduced the powerful notion of approximating boolean functions by polynomials over finite fields for proving Theorem 2.10. In this thesis, the study of such polynomials plays an important role. We introduce this machinery below. Although [Smo87] worked with polynomials over a finite field $\mathbb{Z}_p$ for a prime p, we work with the more general setting of polynomials over the ring $\mathbb{Z}_m$ as in Barrington et.al.[BBR94], where m is an arbitrary but fixed positive composite integer.

Consider the space V_m of functions from $\{0,1\}^n \to \mathbb{Z}_m$. For each $w \in \{0,1\}^n$, define the function $\delta_w : \{0,1\}^n \to \mathbb{Z}_m$ as $\delta_w(x) = 1$ if $w = x$ and otherwise $\delta_w(x) = 0$. Consider the set of functions $\Delta = \{\delta_w \,|\, w \in \{0,1\}^n\}$. It is easy to see that every function $f \in V_m$ can be *uniquely* expressed as a $\mathbb{Z}_m$ linear combination of such functions. Indeed if m is a prime, then Δ forms a basis of the associated vector space.

Another useful set that spans V_m is the set $\mathcal{M}$ of all n-variate multilinear monomials, i.e. $\mathcal{M} = \{\chi_S = \prod_{i \in S} x_i \,|\, S \subseteq [n]\}$, where $[n] = \{1, \ldots, n\}$. To see that $\mathcal{M}$ spans V_m, it is enough to show that each element of Δ can be expressed as a $\mathbb{Z}_m$-linear combination of the monomials. Indeed, this gets verified by observing that

$$\delta_w(x) = \Big(\prod_{i:w_i=1} x_i\Big)\Big(\prod_{i:w_i=0} (1 - x_i)\Big)$$

and then expanding out the product as a sum over $\mathbb{Z}_m$. On the other hand, there are precisely m^{2^n} possible linear combinations of such monomials. This is exactly the number of functions in V_m. Thus, every $f \in V_m$ can be uniquely expressed as a sum of monomials. Any such linear combination of monomials is formally called a multilinear polynomial over $\mathbb{Z}_m$. Since in this thesis we exclusively deal with multilinear polynomials, the term 'multilinear' is henceforth omitted but is always implied. The *degree* of a polynomial is the cardinality of the largest subset S of $[n]$ such that the coefficient of χ_S is non-zero in the polynomial. The *exact* or *strong* MOD$_m$-degree of a boolean function is the degree of the unique polynomial over $\mathbb{Z}_m$ expressing it. For example,

$$AND(x) = x_1 x_2 \cdots x_n$$

$$OR(x) = 1 - \prod_{i=1}^{n}(1 - x_i)$$

showing that the strong MOD$_m$-degree of OR and AND is n, for each integer $m \geq 2$. In order to express MOD$_p$ function, when p is prime we recall the following simple but very useful fact:

Fact 2.11 (Fermat's Little Theorem) *For any prime p and any integer $a \not\equiv 0$ mod p, $a^{p-1} \equiv 1$ mod p.*

Using this fact, we get for a prime p

$$\mathrm{MOD}_p(x) \equiv (x_1 + \cdots + x_n)^{p-1} \pmod{p}$$

establishing that the strong MOD$_p$-degree of the boolean function MOD$_p$ is a constant, i.e. $p-1$. It is interesting to verify the following identity:

$$\mathrm{MOD}_{p^k}(x) \equiv \sum_{S \subset [n] : |S| \leq p^k - 1} (-1)^{|S|-1} \prod_{i \in S} x_i \pmod{p}.$$

This implies that the strong MOD_p-degree of MOD_{p^k} is $p^k - 1$, for any k. A slightly stronger statement is true. With each symmetric boolean function f, one naturally associates its spectrum function $\overline{f} : \{0, \ldots, n\} \to \{0, 1\}$, such that $f(x) = \overline{f}(x_1 + \cdots + x_n)$ for each $x \in \{0, 1\}^n$. A symmetric f is called *periodic* with period a precisely if $\overline{f}(t) = \overline{f}(t + a)$, for each $0 \le t \le n - a$. Then, the following useful fact appears implicitly in the work of Barrington et.al. [BBR94].

Lemma 2.12 *For any prime p and any integer $k \geq 1$, every symmetric boolean function f with period p^k has strong MOD_p-degree at most $p^k - 1$.*

The exact/strong degree of a boolean function is a natural algebraic complexity measure of a boolean function.

Based on the fact that OR and AND have very high degree (read complicated), it is reasonable to guess that modular counting with prime modulus alone should not help compute these high-degree functions. This notion gets verified by an elegant argument below. Before we state the argument, we recall a useful property of composition of polynomials.

Observation 2.13 *Let $P(y_1, \ldots, y_m)$ be a polynomial over $\mathbb{Z}_m$ of degree r and each $y_i = P_i(x_1, \ldots, x_n)$ be a polynomial of degree at most s. Then the composed polynomial $P(P_1(x_1, \ldots, x_n), \ldots, P_m(x_1, \ldots, x_n))$ is a polynomial of degree at most rs in x_i's.*

Theorem 2.14 (implicit in [Smo87]) *Constant-depth circuits using only MOD_{p^k} counting gates cannot compute the AND and OR function if p is a fixed prime and k is a fixed positive integer.*

Proof: The basic idea is to show that the function computed by a circuit of constant depth having only MOD$_p$ gates has constant MOD$_p$-degree. The theorem then follows immediately. We show by induction of depth that the function output by such a depth d circuit has MOD$_p$-degree at most $(p^k - 1)^d$. The base case of $d = 0$ is obvious. Let $y_1, \ldots, y_s$ be the inputs of the output MOD$_{p^k}$ gate in a circuit of depth d. Treating $y_1, \ldots, y_s$ as our input variables, we know that the output of the circuit is represented by a polynomial $P(y_1, \ldots, y_s)$ of degree at most $p^k - 1$. Since each y_i is the output of a depth $d - 1$ circuit, the inductive hypothesis yields that y_i is represented by a polynomial P_i over $\mathbb{Z}_p$ of degree at most $(p-1)^{d-1}$ in the input variables $x_1, \ldots, x_n$ of the circuit. Thus, using Observation 2.13, polynomial P has degree at most $(p-1)^d$ in $x_1, \ldots, x_n$. ∎

Theorem 2.14 is a nice dual to the fact that AND/OR gates cannot compute the MOD$_p$ function in sub-exponential size and constant depth. The dual we have proven happens to be much stronger as it is independent of the size of circuits. Circuits of constant depth composed of prime mod-counting gates are not even *universal*, i.e. they cannot compute all functions even when no restriction is imposed on their size.

The key ingredient in the above argument was the fact that MOD$_p$ function has constant MOD$_p$-degree when p is prime. We note this below:

Fact 2.15 *The MOD$_p$-degree of a function computed by a constant-depth circuit having only MOD$_{p^k}$ gates is constant.*

This fact is indeed very sensitive to the primality of p (or it being a prime power). As soon as m has two distinct prime factors, the MOD$_m$-degree of the

MOD_m function shoots up to linear. As we see later, one cannot even *approximate* the MOD_m function well anymore by low degree polynomials.

Let us relax the notion of exact representation of boolean functions to approximation of them by polynomials. A polynomial P over $\mathbb{Z}_m$ approximates a function f with error ϵ if $\Pr_x[P(x) \neq f(x)] \leq \epsilon$ where x is chosen at random, according to a given distribution. Note that under the uniform distribution over inputs, the constant zero polynomial is a good approximation of the OR function. On the other hand, tremendous savings is made in terms of degree when one moves from exact to approximate representations for any distribution over inputs as the following sequence of results from [Raz87, Smo87] show:

Proposition 2.16 *For every $x \in \{0,1\}^n$, if we pick a random linear polynomial P over $\mathbb{Z}_p$, then $\big(P(x)\big)^{p-1}$ is equal to $OR(x)$ with probability at least a half.*

Proof: Picking a random linear polynomial is the same as picking each of its n coefficients $c_1, \ldots, c_n$ independently at random from $\mathbb{Z}_p$ and then letting $P(x) = c_1x_1 + \cdots + c_nx_n$. If x is the all zero input, then $P(x) = 0$ with probability one and there is no error. Otherwise, there is some i for which $x_i = 1$. For every choice of all other coefficients, there is exactly one choice of c_i that is bad, i.e. makes $P(x) = 0$. Thus with probability $(1 - 1/p)$, polynomial $(P(x))^{p-1}$ evaluates to 1 and we are done. ∎

Lemma 2.17 *For each $0 < \epsilon < 1$ and for every circuit C in $ACC^0[p^k]$ of depth d and size s, there exists a distribution U_C over polynomials over $\mathbb{Z}_p$ of degree at most $\big((p^k - 1)(\log(s/\epsilon))\big)^d$, such that for each input x to C, $\Pr_{P \sim U_C}[P(x) \neq C(x)] \leq \epsilon$.*

Proof: For each gate G in the circuit, we do the following:

If G is an OR gate, pick $t = \log(s/\epsilon)$ random linear polynomials $P_1, \ldots, P_t$ independently. Let $y_i = (P_i(x))^{p-1}$. Let P_G be the polynomial that exactly computes $\mathrm{OR}(y_1, \ldots, y_t)$. Note that P_G is a random polynomial of degree at most $(p-1)t = (p-1)\log(\epsilon/s)$. If G outputs zero, then P_G outputs zero with probability one. If G outputs one, using Proposition 2.16, P_G outputs zero with probability at most $1/(2^t) = \epsilon/s$. Thus P_G disagrees with G with probability at most ϵ/S.

If G is an AND gate we think of it as the complement of an OR gate using de Morgan's law. We choose a random polynomial P'_G for this OR gate as prescribed before and then set $P_G = 1 - P'_G$. The same conclusions on the degree and error probability as before for a polynomial corresponding to an OR gate holds for P_G.

If G is a MOD_{p^k} gate we replace it by the unique polynomial of degree at most $p^k - 1$ that exactly computes it.

We combine polynomials for all gates by composing them, layer by layer, to obtain the polynomial P_C corresponding to circuit C. Using Observation 2.13, P_C has degree at most $(p^k-1)^d(\log(s/\epsilon))^d$. Using the union bound, P_C errs with probability at most ϵ and we are done. ∎

Corollary 2.18 *Let C be an $ACC^0[p^k]$ circuit of depth d and size s. For each distribution μ on $\{0,1\}^n$ and $0 < \epsilon < 1$, there exists a polynomial P of degree at most $\big((p^k-1)(\log(s/\epsilon))\big)^d$ such that $\Pr_{x \sim \mu}[P(x) \neq C(x)] \leq \epsilon$.*

Proof: Follows directly from Lemma 2.17 using an obvious counting argument. ∎

Corollary 2.18 shows the remarkable savings in degree that approximations can bring in.

Remark 2.19 *Even though the exact degree of an AND/OR gate is as high as it can get, functions computed by $ACC^0[p^k]$ circuits of quasi-polynomial size can be approximated with inverse-quasipolynomial error by polynomials over $\mathbb{Z}_p$ that have merely poly-logarithmic degree, if p is prime.*

However, there are some natural functions that are even hard to approximate. Based on the fact that modular counting over two different prime moduli are very different from each other, it is tempting to guess that low degree polynomials over $\mathbb{Z}_p$ do not approximate well the MOD_q function when p, q are two distinct primes. This was formally verified by Smolensky [Smo87]. We recall his neat argument.

We assume that p, q are two primes such that the field $\mathbb{Z}_p$ has a non-trivial q-th root of unity g i.e. $g \in \mathbb{Z}_p$, $g \neq 1$ and $g^q = 1 \bmod p$ (for instance $p = 3$ and $q = 2$ form such a pair of primes as $a = 2$ is a square-root of unity in $\mathbb{Z}_3$). The case when this is not satisfied can be handled like this case by using a simple algebraic trick that we describe later.

Consider the linear transformation $y_i = (g-1)x_i + 1$ for $1 \leq i \leq n$. This maps $0, 1$ to $1, g$ respectively. Using this map, we naturally identify the space V_p of functions from $\{0,1\}^n \to \mathbb{Z}_p$ with the space W_p of functions from $\{1, g\}^n \to \mathbb{Z}_p$. Note that

$$x_i = \frac{y_i - 1}{g - 1}$$

is well defined as $g \neq 1$ by assumption. Also, for $x_i \in \{0,1\}$,

$$y_i^{-1} = (g^{-1} - 1)x_i + 1 = \frac{g^{-1} - 1}{g - 1}(y_i - 1) + 1. \tag{2.2}$$

Using these identities, one can go back and forth between every polynomial P_x in the variables x_i's representing a function f in V_p and a polynomial P_y in y_i's representing the function corresponding to f in W_p. Further, it is simple to verify that the degrees of P_x and P_y are identical. Let $R \in W_p$ be the function given by $\prod_{i=1}^{n} y_i$.

Lemma 2.20 *Every polynomial P_f in variables $y_1, \ldots, y_n$ can be written as $P_f = P_g \cdot R + P_h$, such that each polynomial P_g, P_h has degree at most $n/2$.*

Proof: P_h is the sum of all monomial terms of P_f that have degree at most $n/2$. The Lemma follows by showing that each monomial of degree more than $n/2$ can be written as $P \cdot R$, where P is a polynomial of degree at most $n/2$. Consider any monomial $M = \prod_{i \in S} y_i$, where $S \subseteq [n]$ and $|S| > n/2$. Then, using the definition of R and (2.2), we see that

$$M = R\Big(\prod_{i \notin S} y_i^{-1}\Big) = R\Bigg[\prod_{i \notin S}\bigg(\frac{g^{-1}-1}{g-1}(y_i - 1) + 1\bigg)\Bigg] = R \cdot P$$

and clearly P has degree less than $n/2$. ∎

For any $0 \leq s \leq q-1$, (abusing notation) define MOD_q^s to be the function in V_p (W_p) that outputs 1 if the number of input bits set to 1 (g) is congruent to s modulo q and otherwise outputs zero. Then the following is obvious:

Observation 2.21

$$R = \sum_{i=0}^{q-1} g^i MOD_q^i$$

We are ready to prove the main result of this section.

Lemma 2.22 (Main Lemma, [Smo87]) *Every polynomial over $\mathbb{Z}_p$ of degree d disagrees with one of the boolean functions in $\{MOD_q^i | 0 \leq i \leq q-1\}$ in at least $2^n\,(1/2q - d/\Omega(q\sqrt{n}))$ input points.*

Proof: Recall that every polynomial in V_p of degree d has a polynomial P of degree d in W_p. Thus, using Observation 2.21, it will be sufficient to show every such P differs with R on at least $2^n(1/2 - d/\Omega(\sqrt{n}))$ points.

Let $A \subseteq \{1, g\}^n$ be the set of points on which P and R agree. Applying Lemma 2.20, every function $(\mathbb{Z}_p)^A$ is spanned by the set of monomials of degree at most $n/2+d$. The total number of such functions should therefore be at most the total number of polynomials of degree at most $n/2+d$. Hence,

$$p^{|A|} \leq p^{\sum_{i=0}^{n/2+d} \binom{n}{i}}$$

yielding (using Stirling's approximation)

$$|A| \leq 2^{n-1} + \frac{2^n}{\sqrt{n}} d.$$

Our result follows readily. ∎

Summarizing what we have seen so far will immediately yield Theorem 2.10 that claims an exponential lower bound on the size of ACC$^0[p^k]$ circuits computing MOD$_q$, if p, q are distinct primes.

Proof:[of Theorem 2.10] Recall that Corollary 2.18 showed us that every function computed by such a circuit of size s and depth d can be approximated by a polynomial of degree $O(\log(s/\epsilon))^d$ that errs at only ϵ fraction of inputs. Thus, if $\log s = o(n^{1/2d})$, then this says that the approximating polynomial has degree $o(\sqrt{n})$ and makes $o(1)$

errors. Combining this with Lemma 2.22, we see that one of the MOD_q^i functions cannot be approximated this well and therefore needs circuits of size $2^{\Omega(n^{1/2d})}$. On the other hand, observing that if circuits of size s and depth d can compute MOD_q, then in (almost) that size and depth they can compute MOD_q^i for all i gives us our theorem. ∎

The proof of Theorem 2.14 shows that MOD_{p^k} gates for a fixed prime p, are not universal. On the other hand, MOD_m gates are universal, if m has two distinct prime factors. In fact in depth-two, circuits comprising only such MOD_m gates can compute every function. However, it appears implausible that MOD_m gates, with m having two or more distinct prime factors, should give us significant advantage over the case when m has only a single prime factor in computing MOD_ℓ if m, ℓ are co-prime numbers. This motivated Smolensky to make the following outstanding conjecture:

Conjecture 2.23 (Smolensky's Conjecture [Smo87]) *$ACC^0[m]$ circuits cannot compute the MOD_ℓ function in size $2^{n^{o(1)}}$, if m, ℓ are relatively prime numbers.*

This beautiful conjecture drives our work on constant-depth circuits having modular gates. Recalling $\mathrm{MOD}_\ell \leq^{\mathrm{AC}^0}$ MAJORITY for any fixed ℓ, it is simple to verify that Smolensky's conjecture implies that MAJORITY $\notin \mathrm{ACC}^0$.

2.1.4 The Weakness of a Single MAJ Gate

Although we do not understand the computational power of even depth-three TC^0 circuits, we describe one weakness of MAJ gates that does provide traction in some interesting cases. Consider a circuit with a MAJ gate at the output computing a function f. Intuition suggests that at least one of the sub-circuits C_i must 'approximate' the function f well if the fan-in of the MAJ gate is small. The simple reason to expect this is that a MAJ gate decides what the majority of its sub-circuits decide on a given input.

We make this formal as follows: Let A and B be subsets of inputs on which f evaluates to 1 and 0 respectively i.e. $A \subseteq f^{-1}(1)$ and $B \subseteq f^{-1}(0)$. Let μ be a probability distribution with support $A \cup B$. Then, a function g is said to ϵ-discriminate f if the following holds:

$$\left| \Pr_{x \sim \mu} \left[g(x) = 1 \,\middle|\, x \in A \right] - \Pr_{x \sim \mu} \left[g(x) = 1 \,\middle|\, x \in B \right] \right| \geq \epsilon$$

.

This notion then highlights the weakness of a MAJ gate through the following lemma of Hajnal et.al. [HMP+93]

Lemma 2.24 (Discriminator Lemma) *Let f be a function computed by a MAJ gate that gets its inputs from t sub-circuits $C_1, \ldots, C_t$. Then, for every pair of subsets $A \subseteq f^{-1}(1)$ and $B \subseteq f^{-1}(0)$ and distribution μ on inputs, there exists a sub-circuit C_i that $1/t$-discriminates f.*

Proof: Let μ_A (μ_B) be the distribution induced on A (B) by μ conditioned on event $x \in A$ ($x \in B$). Then, from the definition of a MAJ gate,

$$\mathbb{E}_{x \sim \mu_A} \big[\sum_{i=1}^{t} C_i(x) \big] \geq \lceil \frac{t}{2} \rceil$$

and

$$\mathbb{E}_{x \sim \mu_B} \big[\sum_{i=1}^{t} C_i(x) \big] \leq \lfloor \frac{t}{2} \rfloor .$$

Subtracting the second inequality from the first, and using the triangle inequality, along with the linearity of expectation, we have

$$1 \leq \sum_{i=1}^{t} \left| \mathbb{E}_{x\sim\mu_A}\big[C_i(x)\big] - \mathbb{E}_{x\sim\mu_B}\big[C_i(x)\big] \right|$$

$$= \sum_{i=1}^{t} \left| \Pr_{x\sim\mu}\big[C_i(x) = 1 \big| x \in A\big] - \Pr_{x\sim\mu}\big[C_i(x) = 1 \big| x \in B\big] \right|.$$

Applying an averaging argument to the above yields the lemma. ∎

To illustrate the usefulness of the Discriminator Lemma, we show the following simple fact:

Fact 2.25 *Depth-two circuits with a MAJ gate at the output that is fed by AND gates of fan-in at most $n-1$ i.e. $MAJ \circ AND_{n-1}$ cannot compute the PARITY of n bits.*

Proof: Let A and B be set of inputs that have odd and even parity respectively. Let μ be simply the uniform distribution. It is not hard to verify that the probability of a given AND gate firing a 1 is unaffected by events $x \in A$ or $x \in B$. Consequently, each AND gate does not ϵ-discriminate PARITY for any non-zero ϵ. ∎

A combination of the Discriminator Lemma with Håstad's Switching Lemma results in a much more interesting fact that was first proved by Green [Gre91]:

Theorem 2.26 *Consider a circuit having a single MAJ gate at the output that is being fed by AC^0 sub-circuits of depth d, i.e. $MAJ \circ AC^0_d$. Any such circuit needs size $2^{\Omega(n^{1/d})}$ to compute PARITY.*

Proof: The idea of the proof is the following. We hit all AC^0 sub-circuits with random restrictions simultaneously just as we did to prove that PARITY requires exponential size AC^0 circuits to compute (Corollary 2.5). We show the following: if the size of the circuit is $2^{o(n^{1/d})}$, then with non-zero probability, each restricted AC^0 sub-circuit can be replaced by a few AND gates of fan-in *less* than the number of free variables. We choose one restriction that satisfies the above. Under this restriction, the restricted circuit still computes PARITY (or ¬PARITY) of the remaining free variables. Fact 2.25 provides a contradiction finishing the proof.

This idea is carried out by composing d random restrictions exactly like in the proof of Corollary 2.5. Hence, if S is the sum of the sizes of all the AC^0 circuits, there exists a restriction with the following property: the output of each restricted sub-circuit C_i has a decision tree T_i of depth at most $\log S$. The restriction leaves $n_d = n/(14(14 \log S)^{d-1})$ variables free.

We do the following surgery on each T_i. For each path P that leads T_i to output 1, we create an AND gate whose input variables are exactly the ones that T_i queries along P. Let there be k_i such paths in T_i which then results in k_i AND gates being created, each of fan-in at most $\log S$. The key observation is that, for a given input assignment, at most one of these k_i AND gates outputs 1. Thus, if we feed $(k_i - 1)$ constant 1's in addition to the k_i AND gates directly to the output MAJ gate, then we compute the same function as the restricted circuit. As argued before, Fact 2.25 implies that the fan-in of one of these AND gates is the number of free variables. Hence,

$$\log S \geq n_d = n/(14(14 \log S)^{d-1})$$

which provides the required bound on S, the size of the circuit. ∎

It remains a very interesting open question to determine whether super-polynomial lower bounds can be proven on the size of such circuits when the sub-circuits feeding into the MAJ gate are augmented with MOD_m gates for any odd m. This remains open even for prime m. In Chapter 3 and Chapter 7, we consider restricted circuits with MOD_m gates feeding into a MAJ gate and prove strong lower bounds for them.

CHAPTER 3
Lower Bounds for Circuits with Modular Gates

In the last chapter, we saw that random restrictions provide a powerful combinatorial tool for proving (optimal) lower bounds for AC^0 circuits. Unfortunately, there does not seem to be any way to apply restrictions to fix a modular counting gate without fixing almost all of its inputs. This renders the technique ineffective to deal even with circuits that contain only modular gates. This difficulty was overcome by the ingenious arguments of Razborov and Smolensky using the “polynomial method”. The second part of Smolensky’s argument shows that low degree polynomials over $\mathbb{Z}_p$ cannot even approximate well the MOD_q function, if p, q are distinct primes. Ironically, this result itself spells doom for the Razborov-Smolensky approach when modular gates involved have a modulus m that contains two such primes p, q. Indeed, it shows that the MOD_m function cannot be well approximated by a low degree polynomial over the ring $\mathbb{Z}_m$ when $m = pq$. This fails the first part of the Razborov-Smolensky approach to approximate functions computed by $\mathrm{ACC}^0[m]$ by low degree polynomials.

No satisfactory method is yet known for general constant-depth circuits with modular gates of composite modulus. In this chapter, we make progress, continuing a long line of intensive research (see for example [BS95, BS99, BST90, Gre04, Gro94b, Gro98, GT00, HM04, KW91, MPT91, Smo90, ST06, Thé94]). Our strategy is two-pronged. First we view ACC^0 circuits as AC^0 circuits augmented with modular gates. Besides being a natural point of view, this is inspired by a similar point of view on TC^0 being AC^0 circuits augmented with MAJ gates. This led to a series of interesting results [ABFR94, BRS91b, Bei94, BS94]. A natural question, with this point of view, then is the following: Can lower bounds be proved if we limit the amount of MOD_m gates used? We pursue this theme in Section 3.1 and prove that few MOD_m gates do not aid an AC^0 circuit significantly in computing MAJORITY and MOD_ℓ. More precisely, we show the following:

Theorem 3.1 *Let m be a positive integer with $r \geq 2$ distinct prime factors. Any AC^0 circuit augmented with s MOD_m gates requires size $n^{\Omega(\frac{1}{s}log^{\frac{1}{r-1}}n)}$ to compute MAJ or MOD_ℓ, if ℓ has a prime factor not dividing m.*

To get a feel for the meaning of this theorem, note that it implies that AC^0 circuits augmented with $o(\log n)$ MOD_6 gates, cannot compute MOD_5 or MAJORITY in polynomial size. It is interesting to note that our Theorem 3.1 complements the result obtained by [BS94] which shows that AC^0 augmented with polylogarithmic number of MAJ gates cannot compute MOD_ℓ efficiently. They deal with polylogarithmic number of MAJ gates using the result of Beigel [Bei94] which shows that every circuit with polylogarithmic number of MAJ gates can be simulated by one with a single MAJ gate, increasing the size of the original circuit by at most a quasipolynomial factor. No analogous simulation of circuits with a few MOD_m gates by a circuit with a single MOD_m gate is known.

We extend the machinery of polynomials over rings, introducing a new notion of polynomial representation of boolean functions. Our lower bounds on degrees of such representations in Section 3.1.1 are of independent interest. These bounds are then combined with random restrictions on AND/OR gates to yield Theorem 3.1. Results contained in this part appeared in joint work with K. A. Hansen in [CH05].

In the second part, we aim to understand the class of functions computable by circuits of polynomial size, comprising only MOD_m gates and having constant depth. We denote this class by $\mathrm{CC}^0[m]$. Define $\mathrm{CC}^0 = \cup_{m\geq 2}\mathrm{CC}^0[m]$. While developing techniques to prove lower bounds on the size of CC^0 circuits is a significant step towards understanding ACC^0, Caussinus [Cau96] points out that it is not even known if in depth-two and linear size CC^0 circuits can compute SATISFIABILITY when the modular gates are allowed to be *generalized*. A generalized MOD_m gate, denoted by MOD_m^S, has an associated accepting set $S \subset \mathbb{Z}_m$ and outputs 1 iff the sum of the input bits modulo m is an element of S.

Let the *support set* of a boolean function f be the set of inputs at which f is non-zero. Slightly abusing terminology, we call the size of the support set of f as *support*. One weakness of a MOD_m gate is that the size of its support set is large i.e. roughly $2^n/m$. It is tempting to postulate that constant-depth circuits of small size cannot quite overcome the weakness of its constituent gates. This intuition leads to the following conjecture:

Conjecture 3.2 (McKenzie, Peladeau and Thérien [MPT91]) *The AND of n bits cannot be computed in constant depth and polynomial size by circuits comprising only MOD_m gates , for any fixed modulus m, i.e. $AND \notin CC^0$.*

Observe that this conjecture is the dual of the classical result that MOD_m cannot be computed efficiently in constant depth using only AND and OR gates. The AND function has the smallest support that any non-constant function can have. On the other hand, it is not even known if a function with a sub-exponential size support is in CC^0. We dare conjecture the following:

Conjecture 3.3 (Small Support Set) *There exists a function $h : \mathbb{N} \to \mathbb{N}$, such that any non-constant function computed by a CC^0 circuit of size s and depth d has a support set of size at least $\frac{2^n}{2^{\Omega(\log s)^{h(d)}}}$.*

Recall that Fact 2.15 in Chapter 2 states that the MOD_p-degree of functions computed by $\mathrm{CC}^0[p^k]$ circuits of *arbitrary size* is a constant. It can be shown that functions represented by constant degree polynomials over $\mathbb{Z}_p$ have a support set[1] of exponential size. Thus, the Small Support Set Conjecture holds in a very strong sense for $\mathrm{CC}^0[p^k]$.

In Section 3.2, we make small but non-trivial progress on this conjecture. Specifically, we prove the following: let $\mathrm{CC}[m]$ denote the class of functions computable by polynomial size circuits having only MOD_m gates but *arbitrary* depth. Then,

Theorem 3.4 *For every positive integer m, there exists a positive constant c such that every non-constant boolean function with support size less than $2^n/c^s$ cannot be computed by any $CC[m]$ circuit whose Layer 1 has size less than s.*

Thérien [Thé94] gives a similar but weaker result that functions with support set of size less than $(\frac{\alpha(m)}{\alpha(m)-1})^n \frac{1}{\alpha(m)^s}$ require $\mathrm{CC}[m]$ circuits of size s, where $\alpha(m)$ is a growing function of m. In particular, such results imply that AND cannot be computed by sublinear size $\mathrm{CC}[m]$ circuits. In contrast to Thérien's technique of

[1] The result of Péladeau and Thérien [PT88] shows that this continues to hold even for polynomials over $\mathbb{Z}_m$ when m is an arbitrary composite number.

using Fourier Analysis over finite fields, we combine analysis over complex numbers with notions from additive number theory. As Smolensky [Smo87] remarked, analysis over characteristic zero may lead to further techniques being developed by making use of metric inequalities.

In the first part of this chapter, Theorem 3.1 makes progress towards Smolensky's Conjecture. In Section 3.2, we make progress on it from a different direction. Smolensky's conjecture implies that $\mathrm{CC}^0[m]$ circuits require exponential size to compute MOD_ℓ when m, ℓ are co-prime. Proving this will constitute significant advancement in our understanding of the limitations of modular counting. We report the following progress on this front: let $\mathrm{CC}_{o(n)}[m]$ denote the class of circuits, comprising only MOD_m gates, having sublinear size and arbitrary depth.

Theorem 3.5 *Any circuit of type $MAJ \circ CC_{o(n)}[m]$ computing MOD_ℓ requires the output gate to have fan-in $2^{\Omega(n)}$ if m, ℓ are co-prime.*

This result considerably improves the previous best lower bound due to Smolensky [Smo90] who showed an $\Omega(\log n)$ lower bound on the number of gates needed by $\mathrm{CC}^0[m]$ circuits to compute the MOD_ℓ function. We obtain Theorem 3.5, on the other hand, by showing that functions in $\mathrm{CC}_{o(n)}[m]$ have exponentially small correlation with MOD_ℓ. Results in this section appeared in the joint work with N. Goyal, P. Pudlák and D. Thérien [CGPT06].

3.1 Circuits with Few Modular Gates

3.1.1 Preliminaries of Polynomial Representation

Recall that Razborov and Smolensky [Raz87, Smo87] introduced polynomials over finite fields mainly as a tool to analyze circuits with modular gates. Their work was closely followed up by a number of other works (see for example [All89, Yao90, BRS91a, ABFR94, BT94]), where polynomials (over finite fields, finite rings and fields of characteristic zero) played a key role in obtaining strong lower bounds on various circuits. There is a nice (though somewhat outdated) survey of these works by Beigel [Bei93]. While these early works looked at polynomials mainly as a tool for obtaining lower bounds, the work of Barrington, Beigel and Rudich [BBR94] and that of Nisan and Szegedy [NS94] treat polynomials as an independent model of computation with degree being the most important resource. In this chapter, we focus on polynomials over the finite commutative ring $\mathbb{Z}_m$, for a fixed integer m. Interestingly, polynomials over reals show up as an invaluable tool in Chapter 6 to analyze the communication complexity of boolean functions.

A polynomial P over a ring is a *strong representation* of a boolean function f if $f(x) = P(x)$ for all $x \in \{0,1\}^n$. Note that this makes sense because rings, by definition, have 0 and 1 elements. Razborov and Smolensky, for instance, use the strong representation by polynomials over the special field $\mathbb{Z}_p$, where p is prime. As we saw in the last chapter, each boolean function has a unique strong representation by a polynomial over $\mathbb{Z}_m$ for any integer $m \geq 2$. In order to make use of the full power of the underlying ring $\mathbb{Z}_m$, this notion can be naturally relaxed in more than one way:

- P is a *one-sided* representation of f if $f(x) = 0 \Leftrightarrow \; P(x) \equiv 0 \pmod{m}$ for all $x \in \{0,1\}^n$.
- P is a *weak* representation of f if $P(x) \not\equiv 0 \pmod{m}$ for some $x \in \{0,1\}^n$, and $P(x) \not\equiv 0 \pmod{m} \Rightarrow f(x) = 1$ for all $x \in \{0,1\}^n$.

- P is a *generalized*[2] representation of f if there is an *accepting* set $S \subset \mathbb{Z}_m$ such that $f(x) = 1 \Leftrightarrow P(x) \in S$.

The minimal degree of a polynomial satisfying the above properties is called the strong, one-sided, weak and generalized MOD$_m$-degree, respectively. Note that a strong representation is also a one-sided representation. A one-sided representation is also a weak representation as well as a generalized representation (with accepting set $\mathbb{Z}_m - \{0\}$).

Tardos and Barrington [TB98] obtained the following lower bound on the generalized degree of the OR function.

Theorem 3.6 ([TB98]) *Let m be a positive integer with $r \geq 2$ distinct prime factors, and let q be the smallest maximal prime power divisor of m. The generalized MOD$_m$-degree of the OR function on n variables is at least $\left(\left(\frac{1}{q-1} - o(1)\right) \log n\right)^{\frac{1}{r-1}}$.*

Incidentally, this is the best lower bound on the generalized MOD$_m$-degree of the OR function for a composite m. The best upper bound is due to Barrington, Beigel and Rudich [BBR94]. They showed that there is a symmetric polynomial over $\mathbb{Z}_m$ of degree $O(n^{1/r})$ that one-sidedly represents the OR function, when m has r distinct prime factors. This is one of a few results that shows that composites have non-trivial advantage over primes in a reasonable model of computation. It is not known if the advantage in this case is exponential, but that is certainly not expected.

[2] This notion was actually called *weak* representation in [TB98], but we prefer to reserve this name for the representation introduced by Green [Gre00], which is analogous to the weak degree of a voting polynomial defined by Aspnes et al [ABFR94].

Improving the lower bound of Tardos and Barrington remains an outstanding open problem in the field of polynomial representation of boolean functions.

Although proving strong lower bounds on the generalized MOD_m-degree of explicit boolean functions has been hard, the situation is much better when one deals with one-sided and weak degree. Linear lower bounds on one-sided MOD_m-degree of the MOD_ℓ function is known when m, ℓ are relatively prime. This was first proved by Barrington et.al.[BBR94] and Tsai [Tsa96]. Finally these results were subsumed by the stronger result of Green [Gre00] on the weak MOD_m-degree of MOD_ℓ. Green's bound does not even require m to be fixed or a slowly growing number as needed by [BBR94, Tsa96]. We point out to the interested reader that Green's proof-method is also of independent interest as it uses novel algebraic arguments that could be of further use for proving degree lower bounds.

Theorem 3.7 (Green [Gre00]) *Let m and ℓ be positive relatively prime integers. The weak MOD_m-degree of the MOD_ℓ and $\neg MOD_\ell$ functions on n variables is at least $\lfloor \frac{n}{2(\ell-1)} \rfloor$.*

Finally, we need a technical Lemma that allows us to move from a polynomial over $\mathbb{Z}_{p^k}$ to a polynomial over $\mathbb{Z}_p$ with a small blow-up of degree, provided p is prime. This Lemma is derived from Lemma 2.12 in the last chapter that said that every periodic symmetric function of period p^k has strong MOD_p-degree at most $p^k - 1$. .

Lemma 3.8 (Tardos and Barrington [TB98]) *Let P be a polynomial of degree d in n variables over $\mathbb{Z}_{p^k}$ and let $S \subseteq \mathbb{Z}_{p^k}$ be any set. Then there exists another polynomial P' of degree at most $(p^k - 1)d$ in n variables over $\mathbb{Z}_p$ such that $P(x) \in S \Rightarrow P'(x) = 1$ and $P(x) \notin S \Rightarrow P'(x) = 0$ for all $x \in \{0, 1\}^n$.*

We include a proof for completeness, using ideas from [BBR94].

Proof: Let P have t monomials enumerated in some way. Let y_i be a boolean variable that takes the same value as the ith monomial of P. Despite the fact that the y_i's are not independent of each other, the boolean function represented by P naturally defines a partial function on $\{0,1\}^t$ that is symmetric and periodic[3] with period p^k. Applying Lemma 2.12, there exists a polynomial P' in variables $y_1, \ldots, y_t$ over $\mathbb{Z}_p$ of degree at most $p^k - 1$ that strongly represents the function represented by P with accepting set S. As each y_i is of degree at most d, composing P' with the monomials representing y_i results in degree at most $(p^k - 1)d$. ∎

Remark 3.9 *For a prime p, the strong MOD_p-degree of a boolean function f is at most $(p^k - 1)$ times the generalized MOD_{p^k}-degree of f.*

3.1.2 Weak Generalized Representation

We introduce a new representation of boolean functions over polynomials that is necessary to obtain our lower bounds on the size of circuits. We say P is a *weak generalized* representation of f if there is an accepting set $S \subset \mathbb{Z}_m$ and an $\bar{x} \in \{0,1\}^n$ such that $P(\bar{x}) \in S$ and that for all $x \in \{0,1\}^n$ we have $P(x) \in S \Rightarrow f(x) = 1$. The minimal degree of a polynomial satisfying the above property w.r.t. a function f, is called the weak generalized MOD$_m$-degree of f.

[3] Symmetricity in this context simply means that if x and y are two inputs of identical Hamming weight on which the function is defined, then the function evaluates identically on them. The notion of periodicity can be likewise extended to partial functions.

Observe that all three representations that we discussed in the last section are special cases of this new notion. Further, for a weak generalized representation we can assume that $|S| = 1$. In fact, if P is a weak generalized representation there exists $a \in \mathbb{Z}_m$ such that $P - a$ is a weak generalized representation with accepting set $\{0\}$ of the same boolean function.

We first show a simple consequence of the lower bound on the generalized degree of the OR function for the weak generalized degree of the MAJ and ¬MAJ functions.

Theorem 3.10 *Let m be a positive integer with $r \geq 2$ distinct prime factors, and let q be the smallest maximal prime power divisor of m. The weak generalized MOD_m-degree of the MAJ and ¬MAJ functions on n variables is at least $\left(\left(\frac{1}{q-1} - o(1)\right)\log n\right)^{\frac{1}{r-1}}$.*

Proof: We first observe that MAJ and ¬MAJ have almost the same degree. This is obvious from the following fact: if n is odd, $\mathrm{MAJ}(x) = \neg\mathrm{MAJ}(1-x_1, 1-x_2, \ldots, 1-x_n)$ and otherwise $\mathrm{MAJ}(x_1, \ldots, x_{n-1}) = \neg\mathrm{MAJ}(1 - x_1, \ldots, 1 - x_{n-1}, 0)$.

We now prove the lower bound on the degree of ¬MAJ by deriving a generalized representation of the OR function from a weak generalized representation of ¬MAJ. Let P be a polynomial over $\mathbb{Z}_m$ of degree d that is a weak generalized representation of ¬MAJ with accepting set S. Let $y \in \{0,1\}^n$ be an input with maximal Hamming weight such that $P(y) \in S$. Let $J \subset [n]$ be the set of indices where y has a 1. Clearly, $|J| < n/2$. For every $i \in J$ set $x_i = 1$ in P. Let P' be the resulting polynomial on variables having indices in $[n] - J$. Then, it is simple to verify that P' w.r.t accepting set $\mathbb{Z}_m - S$ is a generalized representation of the OR function over at least $n/2$ variables. The lower bound on d follows from Theorem 3.6. ∎

We combine techniques introduced in [TB98] and Green's lower bound on the weak MOD_m-degree of MOD_ℓ. This new combination proves the following result on the weak generalized MOD_m-degree of MOD_p.

Theorem 3.11 *Let m be a positive integer with $r \geq 2$ distinct prime factors, let p^k be the smallest maximal prime power factor of m. Let q be a prime not dividing m. For all $a \in \mathbb{Z}_q$, the weak generalized MOD_m-degree of the MOD_q^a and $\neg MOD_q^a$ functions on n variables is at least $\left(\left(\frac{1}{2(q-1)^2(p^k-1)} - o(1)\right)\log n\right)^{\frac{1}{r-1}}$.*

The general idea of proving this theorem is to successively convert a given representation over modulus m to another representation of a similar function of fewer variables over a new modulus m', where m' has one less prime factor than m. Applying this procedure a constant number of times, we are left with a representation over a modulus that has just one prime factor. At this point, we apply the following fact that follows from Theorem 3.7 and Lemma 3.8.

Fact 3.12 *The weak generalized MOD_{p^k}-degree of the MOD_q and $\neg MOD_q$ functions on n variables is at least $\frac{1}{(p^k-1)}\lfloor\frac{n}{2(q-1)}\rfloor$, if p is a prime that does not divide q.*

The scheme to move down from a given modulus to another simpler one without losing too many variables was first designed in [TB98] with respect to the OR function. We suitably modify this to work in our context. The main trick is the following: let $m = p^k m'$ for some prime p. Then any polynomial over $\mathbb{Z}_m$ can be decomposed, using Chinese Remaindering, into a polynomial over $\mathbb{Z}_{p^k}$ and a polynomial over $\mathbb{Z}_{m'}$. We switch off the contribution of the first polynomial towards the representation of the MOD_ℓ function in the following way: identify disjoint sets of variables $S^1, \ldots, S^t$

such that the polynomial over $\mathbb{Z}_{p^k}$ is reduced to a constant polynomial if variables in a given set S^i are restricted to take the same value. In this case, collapsing variables in each set S^i to a single variable y_i, forces the other polynomial over $\mathbb{Z}_{m'}$ to represent the MOD_ℓ function of the new auxiliary variables $y_1, \ldots, y_t$. This allows the induction step of our procedure to be carried out.

With the general idea of the argument described, let us state formally our result that allows us to switch off a polynomial over a modulus that is a prime power. For a subset $S \subseteq \{1, \ldots, n\}$, let $\chi(S) \in \{0,1\}^n$ denote its characteristic vector. Conversely for $x \in \{0,1\}^n$, let $\sigma(x) \subseteq \{1, \ldots, n\}$ be the set of indices where $x_i = 1$.

Lemma 3.13 *Let P be a polynomial of degree d in n variables over $\mathbb{Z}_{p^k}$ for a prime p, and let ℓ be a positive integer not divisible by p. Let t satisfy the condition $n \geq 2(\ell-1)\left(t + (p^k-1)\sum_{i=1}^{d}(d+1-i)\binom{t}{i}\right)$. Then, there exists pairwise disjoint non-empty sets $S^1, \ldots, S^t \subseteq \{1, \ldots, n\}$ such that for every $y \in \{0,1\}^t$ we have $P(\sum_{i=1}^{t} y_i \chi(S^i)) \equiv P(0) \pmod{p^k}$ and furthermore we have $|S^i| \not\equiv 0 \pmod{\ell}$ for all i.*

Proof: Assume without loss of generality that $P(0) = 0$. We will find sets S^i recursively with $|S^i| \leq s_i$, where $s_i = 2(\ell-1)\left(1 + (p^k-1)\sum_{j=0}^{d-1}\binom{i-1}{j}(d-j)\right)$. First pick a set S of $s_1 = 2(l-1)(d(p^k-1)+1)$ variables. Consider the polynomial obtained from P by substituting 0 for all variables not in S. Since the degree of this new polynomial is at most d, Fact 3.12 implies that it is not a weak generalized representation of $\neg\mathrm{MOD}_\ell$ with respect to the set $\{0\}$. Thus there is a subset $S^1 \subseteq S$ such that $P(\chi(S^1)) = 0 = P(0)$ and $\neg\mathrm{MOD}_\ell(\chi(S^1)) = 0$. Hence, $|S^1| \not\equiv 0 \pmod{\ell}$.

In the general case, assume that for $i < t$ we have found sets $S^1, \ldots, S^i$, where $|S^j| \leq s_j$ and $|S^j| \not\equiv 0 \pmod{\ell}$ for all $j \leq i$, such that $P(\sum_{j=1}^{i} y_j \chi(S^j)) = 0 = P(0)$ for all $y \in \{0,1\}^i$. Pick a set S of size s_{i+1} from the remaining variables. For any $y \in \{0,1\}^i$, let P_y be the polynomial obtained from P by substituting y_j for all variables in S^j for all j, and further substituting 0 for all remaining variables not in S.

We show below that there exists a subset S' of S such that $P_y(\chi(S')) \equiv 0 \pmod{p^k}$ for all y and $|S'| \not\equiv 0 \pmod{\ell}$. This finishes the argument as we set $S^{i+1} = S'$.

Let P'_y be the polynomial over $\mathbb{Z}_p$, obtained using Lemma 3.8, that is a strong representation of the boolean function of which P_y is a generalized representation with respect to $\{0\}$. That is $P'_y(x) \equiv 0 \pmod{p} \Leftrightarrow P_y(x) \not\equiv 0 \pmod{p^k}$ and $P'_y(x) \equiv 1 \pmod{p} \Leftrightarrow P_y(x) \equiv 0 \pmod{p^k}$.

Let $R = \prod_{y \in \{0,1\}^i} P'_y$. Note that R only takes values in $\{0,1\}$ modulo p, and that $R(x) \equiv 1 \pmod{p}$ iff $P'_y(x) \equiv 1 \pmod{p}$ for all y, that is, iff $P_y(x) \equiv 0 \pmod{p^k}$ for all y. Further, by construction $R(0^n) \equiv 1 \pmod{p}$. Hence, showing that R is not a weak representation of $\neg\text{MOD}_\ell$ for variables in S is sufficient for finding our desired set $S' \subset S$ such that $R(S') = R(0^n) = 1 \pmod{p}$. However, the degree of R is $2^i(p^k - 1)d$. This is unfortunately too big (when $i >> d$) compared to the size of S (that needs to be at most s_{i+1}, which grows roughly at the rate of i^d, to complete our induction). We overcome this problem below by using an idea of Tardos and Barrington [TB98].

We use inclusion-exclusion sums of P_y's to construct a set of new polynomials whose degrees are slightly less than that of P_y's, but have identical common zeroes as the P_y's. More precisely, for $z, y \in \{0,1\}^i$ say $z \leq y$ if $z_j \leq y_j$ for each $1 \leq j \leq i$. For any $y \in \{0,1\}^i$, define polynomial Q_y with variables in S over $\mathbb{Z}_{p^k}$ as follows:

$$Q_y \equiv \sum_{z \leq y} (-1)^{|z|} P_z.$$

The following claim is simple to verify.

Claim 3.14 *Any $x \in \{0,1\}^{s_{i+1}}$, is a common zero of polynomials Q_y's (over $\mathbb{Z}_{p^k}$) iff it is a common zero of polynomials P_y's (over $\mathbb{Z}_{p^k}$).*

We prove that the high-degree monomial terms of P vanish in Q_y.

Claim 3.15 *The degree of Q_y is at most $d - |y|$.*

Proof:[adapted from [TB98]] Consider any monomial M in P. Let y_j be 1 and assume that M does not depend on any variable in S^j. Consider $z_1, z_2 \in \{0,1\}^i$ such that they differ only in their jth bit. Clearly, the contribution of M to Q_y for z_1 and z_2 cancel each other out. Pairing up points below y in this fashion, it is not difficult to see that the total contribution of M to Q_y zeroes out. Thus, a monomial M has non-zero contribution to Q_y only if it contains a variable from each S^j such that $y_j = 1$. Hence, every monomial term of degree d in P is restricted to a polynomial of degree at most $d - |y|$ in Q_y. ■

As before, using Lemma 3.8, we replace Q_y over $\mathbb{Z}_{p^k}$ by Q'_y over $\mathbb{Z}_p$ such that $Q_y(z) \equiv 0 \pmod{p^k}$ iff $Q'_y(z) \equiv 1 \pmod{p}$ and Q'_y is 0/1 valued over $\mathbb{Z}_p$. We construct R as before replacing P'_y by Q'_y i.e. $R \equiv \prod_{y \in \{0,1\}^i} Q'_y$. Claim 3.15 yields the following bound on the degree of R:

$$\deg(R) \leq (p^k-1)\sum_{j=0}^{d-1}\binom{i}{j}(d-j).$$

From Fact 3.12 and the choice of s_{i+1} we have that R is not a weak representation of $\neg\mathrm{MOD}_\ell$. We can thus find $S^{j+1} \subseteq S$ such that $R(\chi(S^{i+1})) \not\equiv 0 \pmod p$ and $\neg\mathrm{MOD}_\ell(\chi(S^{j+1})) = 0$. It follows that $P_y(\chi(S^{i+1})) = 0$ for all y and $|S^{i+1}| \not\equiv 0 \pmod \ell$. To allow the induction to go through, we need that $n \geq \sum_{i=1}^{t} s_i$. Using the combinatorial identity $\sum_{i=0}^{t-1}\binom{i}{j} = \binom{t}{j+1}$, we see that the relationship between n and t is precisely what we need. ∎

We are ready to prove our bound of $\Omega((\log n)^{\frac{1}{r-1}})$ on the weak generalized MOD_m-degree of MOD_q, where m is a number having r distinct prime factors none of which is the prime q.

Proof:[of Theorem 3.11] Let us recall the idea of the proof: successively use Lemma 3.13 to convert a given representation into another representation on fewer (auxiliary) variables over a modulus that contains less prime factors. Finally use Fact 3.12 when there is just one prime factor left in the modulus.

Let $n = n(m,d)$ denote the maximal number of variables, for which there is a weak generalized representation over $\mathbb{Z}_m$ of degree d, for any of the $\mathrm{MOD}_q^{\{a\}}$ and $\neg\mathrm{MOD}_q^{\{a\}}$ functions. We need to prove that

$$\log n(m,d) \leq (2(q-1)^2(p^k-1)+o(1))d^{r-1}.$$

Let $m = p_1^{k_1}m_1$ where $p_1^{k_1}$ is a maximal prime power divisor of m different from p^k.

Assume that P is a polynomial in n variables of degree d over $\mathbb{Z}_m$ which is a weak generalized representation of f with respect to $\{0\}$, where f is either $\mathrm{MOD}_q^{\{a\}}$ or $\neg\mathrm{MOD}_q^{\{a\}}$ for some $a \in \mathbb{Z}_q$. In order to apply Lemma 3.13, we need to have $P(0) \equiv 0$ which may not be the case. But this is simple to deal with. By definition there exists $\bar{x} \in \{0,1\}^n$ such that $P(\bar{x}) \equiv 0 \pmod{m}$ and $f(\bar{x}) = 1$. If $|\sigma(\bar{x})| < \frac{n}{2}$ let P' be the polynomial obtained from P by setting the variables indexed by $\sigma(\bar{x})$ to 1. Otherwise, if $|\sigma(\bar{x})| \geq \frac{n}{2}$ we can let P' be the polynomial where variable x_i is substituted with $1 - x_i$ if $i \in \sigma(\bar{x})$ and otherwise set to 0.In either case, the number n' of unset variables in P' is at least $\frac{n}{2}$ and $P'(0) \equiv 0 \pmod{m}$.

For a given integer t, let $t' = (p-1)t$ and assume that the following holds:

$$n' \geq 2(q-1)\left(t' + (p_1^{k_1} - 1)\sum_{i=1}^{d}(d+1-i)\binom{t'}{i}\right).$$

Then using Lemma 3.13 we can find pairwise disjoint nonempty sets $S'^1, \ldots, S'^{t'} \subseteq \{1, \ldots, n'\}$ such that for every $y \in \{0,1\}^{t'}$ we have $P'(\Sigma_{i=1}^{t'} y_i \chi(S'^i)) \equiv P'(0) \equiv 0 \pmod{p_1^{k_1}}$ and furthermore we have $|S'^i| \not\equiv 0 \pmod{q}$ for all i. Choosing the most occurring residue $b \in \mathbb{Z}_q \setminus \{0\}$ among $|S'^i|$ modulo q and extending the sets to $\{1, \ldots, n\}$, we have pairwise disjoint nonempty sets $S^1, \ldots, S^t \subseteq \{1, \ldots, n\}$ such that $P(\bar{x} + \Sigma_{i=1}^t y_i \chi(S^i)) \equiv P(\bar{x}) \equiv 0 \pmod{p_1^{k_1}}$ for every $y \in \{0,1\}^t$, and $|S^i| \equiv b \pmod{q}$ for all i.

If f is $\neg\mathrm{MOD}_q^{\{a\}}$, then $P(\bar{x} + \Sigma_{i=1}^t y_i \chi(S^i)) \equiv 0 \pmod{m}$ implies that $|\sigma(\bar{x})| + \sum_{i=1}^t y_i |S^i| \not\equiv a \pmod{q}$. This further implies $\sum_{i=1}^t y_i \not\equiv b^{-1}(a - |\sigma(\bar{x})|) \pmod{q}$. On the other hand, if f is $\mathrm{MOD}_q^{\{a\}}$, then $P(\bar{x} + \Sigma_{i=1}^t y_i \chi(S^i)) \equiv 0 \pmod{m}$ implies that $|\sigma(\bar{x})| + \sum_{i=1}^t y_i |S^i| \equiv a \pmod{q}$. In this case, $|\sigma(\bar{x})| \equiv a \pmod{q}$ by definition.

Hence, $\sum_{i=1}^{t} y_i \equiv 0 \pmod{q}$. By our choice of sets $S^1, \ldots, S^t$, $P(\bar{x}+\Sigma_{i=1}^{t} y_i \chi(S^i)) \equiv 0$ $(\text{mod } m_1)$ iff $P(\bar{x} + \Sigma_{i=1}^{t} y_i \chi(S^i)) \equiv 0 \pmod{m}$. Let Q be the polynomial obtained from P by setting variables in $\sigma(\bar{x})$ to 1 and replacing every occurrence of a variable in the set S^i by the auxiliary variable y_i. Combining our observations, we conclude that Q is a weak generalized representation over $\mathbb{Z}_{m_1}$ of either $\mathrm{MOD}_q^{\{b^{-1}(a-|\sigma(\bar{x})|)\}}$ or $\neg\mathrm{MOD}_q^{\{0\}}$ on the auxiliary variables, w.r.t. the accepting set $\{0\}$.

Thus, setting $t = n(m_1, d) + 1$ (and recall $t' = (q-1)t$) we have the following recursion:

$$n(m,d)/2 \leq n' < 2(q-1)\left(t' + (p_1^{k_1} - 1)\sum_{i=1}^{d}(d+1-i)\binom{t'}{i}\right). \tag{3.1}$$

If $r = 2$, then $m_1 = p^k$ and from Fact 3.12 we have that

$$n(m_1, d) \leq 2(q-1)\left((p^k - 1)d + 1\right).$$

But (3.1) implies that $n(m,d) \leq O\left(d2^{(q-1)n(m_1,d)}\right)$. Hence,

$$\log n(m,d) \leq O(\log d) + (q-1)n(m_1,d) \leq \left(2(q-1)^2(p^k-1) + o(1)\right)d,$$

proving our result for $r = 2$.

If $r > 2$, we have by induction that

$$\log\left(n(m_1,d)\right) \leq (2(q-1)^2(p^k-1) + o(1))d^{r-2}.$$

On the other hand, (3.1) yields that $n(m,d) \leq O\left((q-1)^d n(m_1,d)^d\right)$. Taking logarithms on both sides,

$$\log n(m,d) \leq O(1) + d\left(\log(q-1) + \log(n(m_1,d)\right).$$

Plugging in our inductive estimate of $\log(n(m_1, d))$ from above, we get

$$\log(n(m, d)) \leq (2(q-1)^2(p^k - 1) + o(1))d^{r-1},$$

completing the induction. ∎

As said before, weak generalized representations are interesting in their own right. We show that lower bounds on the degree of such representations have interesting applications for boolean circuits. For ease in describing such applications, we consider the representation of a boolean function by more than one polynomial. Let f be as before and let $P_1, \ldots, P_s$ be polynomials in n variables over $\mathbb{Z}_m$. We say $P_1, \ldots, P_s$ is a *simultaneous weak* MOD_m-representation of f if there exits a $y \in \{0,1\}^n$ such that for each i, $P_i(y) \not\equiv 0 \pmod{m}$ and if it holds that whenever $P_i(x) \not\equiv 0 \pmod{m}$ for all i, we have that $f(x) = 1$. The degree of a simultaneous weak representation is simply the maximal degree of $P_1, \ldots, P_s$. The s-simultaneous weak MOD_m-degree of f is the degree of the simultaneous weak representation of f that has minimal degree.

The following lemma shows, that s-simultaneous weak degree and weak generalized degree are essentially the same, when s is a constant.

Lemma 3.16 *Let m be a positive integer and let $m = q_1 \cdots q_t$ be the factorization into prime powers with $q_i = p_i^{k_i}$. Further, let $m' = p_1 \cdots p_t$ and let f be a boolean function. The weak generalized $MOD_{m'}$-degree of f is at most $s(q-1)$ times the s-simultaneous weak MOD_m-degree of f, where q is the largest prime power factor of m.*

On the other hand, the $(m-1)$-simultaneous weak MOD$_m$-degree of f is at most as large as the weak generalized MOD$_m$ degree of f.

Proof: Let the s-simultaneous weak degree of f be d. Then, there exists a simultaneous weak representation of f by polynomials $P_1, \ldots, P_s$ over $\mathbb{Z}_m$, where $\deg(P_i) \leq d$ for each i. Let $y \in \{0,1\}^n$ be such that $P_i(y) \not\equiv 0 \pmod{m}$ for all i.

Using Chinese Remaindering, each P_i splits into t components $P_i^1, \ldots, P_i^t$ where P_i^j is over $\mathbb{Z}_{q_j}$ and $\deg(P_i^j) \leq \deg(P_i) \leq d$. From the definition of simultaneous representation, for each i, there exists an i_j such that $P_i^{i_j}(y) \not\equiv 0 \pmod{q_{i_j}}$. Applying Lemma 3.8, let Q_{i_j} be the polynomial over $\mathbb{Z}_{p_{i_j}}$ of degree at most $(q_{i_j}-1)d$ such that $P_i^{i_j}(x) \not\equiv 0 \pmod{q_{i_j}}$ iff $Q_{i_j} \not\equiv 0 \pmod{p_{i_j}}$. For each $1 \leq k \leq t$, consider the following polynomial over $\mathbb{Z}_{p_k}$

$$Q_k \equiv_{\text{def}} \prod_{i:i_j=k} Q_{i_j}$$

Let P' denote the polynomial over $m' = p_1 \cdots p_t$ that is obtained by combining, via Chinese Remaindering, the polynomials $Q_1, \ldots, Q_t$. Clearly, the degree of P' is at most $s(q-1)d$. Viewing each element of $\mathbb{Z}_{m'}$ to be a t-tuple with the ith co-ordinate being an element of $\mathbb{Z}_{p_i}$, define $S \equiv \{(a_1, \ldots, a_t) \, : \, a_i \in \mathbb{Z}_{p_i}, a_i \neq 0\} \subset \mathbb{Z}_{m'}$. Recalling that each p_i is prime, it is not hard to verify that P' w.r.t. accepting set S is a weak generalized representation of f. ∎

3.1.3 Application to Circuits

In this section, we combine machinery from the previous section with the Switching Lemma to derive lower bounds on AC^0 circuits augmented with few MOD$_m$ gates.

To illustrate how they may be combined, we consider the case of an AC^0 circuit feeding into a single MOD$_m$ gate at the output.

Theorem 3.17 (Hansen and Miltersen [HM04]) *An AC^0 circuit of depth d augmented with a single MOD_m gate at the output, i.e. a circuit of type $MOD_m \circ AC^0_d$ needs size $2^{\frac{1}{14}(cn)^{1/d}}$ to compute MOD_ℓ, for some constant $c = c(m, \ell)$ provided m, ℓ are relatively prime.*

Proof: The idea is to hit the AC^0 part with random restrictions just as we did in Chapter 2 to prove that AC^0 circuits cannot compute Parity. Let the size of the AC^0 part be S. As in the proof of Corollary 2.5, we choose a random restriction ρ that is a composition of d random restrictions $\rho_1, \ldots, \rho_d$. Each ρ_i is chosen randomly from the space of all restrictions, denoted by $R^{n_i}_{n_{i-1}}$, on n_{i-1} variables that leave exactly n_i free. Here, $n_i = p_i n_{i-1}$, where p_i is the probability with which each variable is left free and $n_0 = n$. Setting $p_1 = 1/14$, $p_i = 1/(14 \log S)$ for $i = 2, \ldots, d$, and using Beame's Switching Lemma, one observes that after applying $\rho_1 \circ \rho_2 \circ \cdots \rho_i$ the output of each gate at the ith layer is computed by a decision tree of height at most $\log S$. Thus, the output of each sub-circuit feeding into the MOD$_m$ gate can be computed by a decision tree of height $\log S$ under the effect of ρ. At this point, Hansen and Miltersen [HM04] make the following crucial observation, showing the utility of decision trees in this context:

Observation 3.18 *A function computed by a decision tree of height at most h has an exact/strong representation over $\mathbb{Z}_m$ of degree at most h, for every integer $m \geq 2$.*

Proof: The idea is quite simple. Consider a path in the tree that leads to a leaf labeled one. Let S be the set of indices of the variables queried along the path. Let

$i \in S$. If the path follows the edge labeled 0 coming out of node labeled x_i then set $y_i = 1 - x_i$, otherwise set $y_i = x_i$. Then, the polynomial $\prod_{i \in S} y_i$ evaluates to 1 (0) precisely if this path is followed (not followed) by the decision tree on a given assignment. Taking the sum of such terms over all paths in the decision tree that lead to a leaf labeled one, yields the desired polynomial of degree at most h. ∎

Applying Observation 3.18, with positive probability the restricted circuit has the following property: one can express exactly the output of each gate feeding into the single MOD_m gate by a 0/1 valued polynomial of degree at most $\log S$ over $\mathbb{Z}_m$. Summing up these polynomials yields a one-sided representation (of degree at most $\log S$) over $\mathbb{Z}_m$ of the restricted function on the remaining $n/(14(14\log S)^{d-1})$ free variables. Setting at most an additional $(\ell - 1)$ variables to 1, the restricted function becomes the MOD_ℓ function. Finally, applying Green's bound (Theorem 3.7) on the weak MOD_m-degree of MOD_ℓ, we get

$$\log S \geq \left\lfloor \frac{1}{2(\ell - 1)} \left(\frac{n}{14(14\log S)^{d-1}} - \ell + 1 \right) \right\rfloor$$

whence the desired bound on S follows. ∎

The reader may have noticed that using Green's lower bound on the weak MOD_m-degree is not strictly needed for the above argument. Indeed, it is sufficient to use lower bounds of [BBR94, Tsa96] on the one-sided MOD_m-degree of MOD_ℓ. However, Green's bound has its own advantage. Using it, Hansen and Miltersen [HM04] showed exponential lower bounds on the size of such circuits with a single MOD_m gate that is allowed to appear *anywhere* in the circuit. Our Theorem 3.1, significantly extends their result. For instance, it follows from Theorem 3.1 that super-polynomial size is still needed to compute the MOD_ℓ function even if we allow $o(\log n)^{1/r-1}$ many MOD_m gates, when m is a fixed composite number having at most r distinct prime factors. The key to this improvement is the use of our notion of weak generalized representation of boolean functions.

Proof:[of Theorem 3.1] We first assume that ℓ is a prime. The case of a composite ℓ is handled easily at the end by invoking the case of a prime ℓ.

Let C be a depth d AC^0 circuit of size $n^{\frac{\epsilon}{s}\log^{\frac{1}{r-1}} n}$ containing s MOD_m gates $g_1, \ldots, g_s$ computing a function f. Assume there is no path from the output of g_j to g_i if $i < j$. For each $\alpha \in \{0,1\}^s$ let C_i^α be the $\mathrm{MOD}_m \circ \mathrm{AC}^0$ subcircuit of C with g_i as output, where every g_j for $j < i$ is replaced by the constant α_j. Similarly, let C^α be the AC^0 circuit obtained from C by replacing every g_i with α_i. We choose a random restriction $\rho \in R_n^{\sqrt{n}}$. We show that for every $\delta > 0$, there exists an $\epsilon > 0$ sufficiently small such that with high probability, *for every* α there are polynomials p_i^α and q^α, of degree at most $\frac{\delta}{s}\log^{\frac{1}{r-1}} n$, such that $C_{i,\rho}^\alpha(x) = 1$ iff $p_i^\alpha(x) \not\equiv 0 \pmod{m}$ and $C_\rho^\alpha(x) = q^\alpha(x)$, for all x and for each $1 \le i \le s$.

Pick such a restriction ρ. We construct a simultaneous weak representation, using $s+1$ polynomials, of either f_ρ or $\neg f_\rho$ as shown next: Pick a *maximal* set G of the MOD_m gates that are 1 at the same time for some assignment x to the free variables of the restriction. Define α such that $\alpha_i = 1$ iff $g_i \in G$. If there exists $x \in \{0,1\}^{\sqrt{n}}$ such that all gates in G evaluate to 1 on x and $C_\rho(x) = 1$, then $\{p_i^\alpha \mid g_i \in G\} \cup \{q^\alpha\}$ is a simultaneous weak representation of f_ρ. Otherwise, $\{p_i^\alpha \mid g_i \in G\}$ is a simultaneous weak representation of $\neg f_\rho$.

Note that if f is MOD_ℓ, then f_ρ is $\mathrm{MOD}_\ell^{\mathbb{Z}_m - \{a\}}$ for some $a \in \{0, \dots, \ell - 1\}$. If f is MAJ and the number of 0 and 1 assigned by ρ differ by at most 1 (which happens with probability $\Omega(n^{-\frac{1}{2}})$), we fix at most one extra variable such that f_ρ computes MAJ. In both cases, we pick δ sufficiently small and obtain a contradiction to the degree lower bounds in Theorem 3.10 and Theorem 3.11, using Lemma 3.16.

It only remains to show that under the effect of ρ, with high probability, for each α one can find the polynomials p_i^α for every $i \leq s$ and q^α. To show this, we analyze the effect of ρ simultaneously on at most $2^s(s+1)$ different AC^0 circuits of depth d and size $S' = n^{\frac{\epsilon}{s}\log^{\frac{1}{r-1}} n}$ obtained by varying α and i. This analysis is carried out like in the proof of Theorem 3.17. We apply a series of random restrictions $\rho_1, \dots, \rho_d$, where $\rho_i \in R_{n_{i-1}}^{n_i}$, $n_i = p_i n_{i-1}$ and $n_0 = n$. Set $p_i = n^{-1/2d}$. Let us say that ρ *fails* if there is a MOD_m gate g such that the function computed by one of the subcircuits feeding into g does not have a decision tree of height $\frac{\delta}{s}(\log n)^{1/(r-1)}$ under ρ. Then, using Beame's Switching Lemma, as in the proof of Theorem 3.17, one concludes the following:

$$\Pr[\rho \text{ fails}] \leq 2^s(s+1)n^{\frac{\epsilon}{s}(\log n)^{\frac{1}{r-1}}} \times \left(7n^{-\frac{1}{2d}}\frac{\delta}{s}(\log n)^{\frac{1}{r-1}}\right)^{\frac{\delta}{s}(\log n)^{\frac{1}{r-1}}}.$$

This further simplifies, under the assumption $s = o(\log n)^{\frac{r}{r-1}}$, to the following:

$$\Pr[\rho \text{ fails}] \leq \exp\biggl(- \ln 2(\log n)^{\frac{r}{r-1}}\frac{1}{s}\left[\frac{\delta}{2d} - \epsilon - \frac{s + \log s + O(\log\log n)}{(\log n)^{\frac{r}{r-1}}}\right]\biggr)$$

$$= \exp\biggl(- \ln 2(\log n)^{\frac{r}{r-1}}\frac{1}{s}\left[\frac{\delta}{2d} - \epsilon - o(1)\right]\biggr).$$

Picking $\epsilon < \frac{\delta}{2d}$ and recalling $s = o(\log n)^{\frac{1}{r-1}}$, the probability above vanishes to zero, as δ is a constant. We fix the constant δ by combining Lemma 3.16 with either Theorem 3.10 or Theorem 3.11 depending on whether f is MAJ or MOD_ℓ.

Finally, we handle the case of a non-prime ℓ. Let p be a prime dividing ℓ. It is sufficient to show that a circuit C computing MOD_ℓ of n variables directly yields a circuit computing MOD_p of $\lfloor np/\ell \rfloor$ variables. This is done as follows: fix at most $\frac{\ell}{p} - 1$ variables to zero so that the number of remaining variables is a multiple of $\frac{\ell}{p}$. Form disjoint clusters of the unfixed variables, each of size ℓ/p. Consider only assignments in which every variable in a cluster is assigned the same way. Circuit C acting over such clustered assignments is precisely the circuit we need. ∎

3.2 Circuits with Only Modular Gates

In Section 3.1.1, we noted a connection between s-simultaneous weak representations and weak generalized representations of boolean functions via Lemma 3.16. Coupling this with our lower bounds of $\Omega(\log n)^{1/(r-1)}$ on the weak generalized MOD_m-degree of MOD_ℓ, one concludes that $\Omega(\log n)^{1/(r-1)}$ polynomials of constant degree d over $\mathbb{Z}_m$ are needed to form a simultaneous weak representation of MOD_ℓ. A similar argument, combining the lower bound for the weak generalized degree[4] of NOR and Lemma 3.16, yields identical conclusion about the simultaneous weak representability of NOR. These conclusions do not rule out the possibility of

[4] Tardos and Barrington prove a lower bound on the generalized degree of OR/AND (Theorem 3.6). The lower bounds translate to the NOR function as well. Note that for NOR/AND, the generalized degree and the weak generalized degree are identical.

AND/OR/MOD$_\ell$ having $(\log n)^{1/(r-1)}$-simultaneous weak degree of *one*. Our first technical result, in this section, rules this out for the case of AND/OR by showing that $o(n)$-simultaneous weak degree of OR/AND is more than one.

More precisely, let $\mathcal{L} = \{\theta_1, \ldots, \theta_s\}$ be a set of s n-variate linear forms over $\mathbb{Z}_m$. Such a set forms a linear map $\mathcal{L} : \mathbb{Z}_m^n \rightarrow \mathbb{Z}_m^s$. Conversely, given such a linear map, there exists a corresponding set of linear forms. For $v \in \mathbb{Z}_m^s$, let $K^{\mathcal{L}}(v)$ represent the set of points in $\{0,1\}^n$, that satisfy $\theta_i = v_i$ for all $1 \leq i \leq s$. Then, we show the following:

Theorem 3.19 *For every positive integer m, there exists a positive constant c such that the following holds. Let $\mathcal{L} : \mathbb{Z}_m^n \rightarrow \mathbb{Z}_m^s$ be a linear map. For any $v \in \mathbb{Z}_m^s$, if $K^{\mathcal{L}}(v)$ is non-empty, then*

$$|K^{\mathcal{L}}(v)| \geq \frac{2^n}{c^s}. \tag{3.2}$$

A simple averaging argument shows that for every $\mathcal{L} : \mathbb{Z}_m^n \rightarrow \mathbb{Z}_m^s$, there exists a $v \in \mathbb{Z}_m^s$ such that $K^{\mathcal{L}}(v)$ has size at least $2^n/m^s$. Theorem 3.19 is a kind of concentration result in the sense that it shows that every $K^{\mathcal{L}}(v)$ is of size close to the average size if it is non-empty. We note that the results in [Thé94], based on methods introduced in [BST90], imply a lower bound of $(\frac{\alpha}{\alpha-1})^n \cdot \frac{1}{\alpha^s}$ on the size of $K^{\mathcal{L}}(v)$ when it is non-empty, and α is an increasing function of m. This is still exponentially smaller than the average size.

We next rule out the possibility that $o(n)$-many linear polynomials over $\mathbb{Z}_m$ form a weak simultaneous representation of MOD$_\ell$. For any $b \in \{0, \ldots, q-1\}$, define the

bth MOD$_\ell$-residue class of $\{0,1\}^n$ by

$$M_{n,\ell}(b) = \{x = (x_1, \ldots, x_n) \in \{0,1\}^n \mid \sum_{i=1}^{n} x_i = b \pmod{\ell}\}$$

Lemma 3.20 (Linear Uniformity Lemma) *For all positive co-prime integers m, ℓ, there exists a positive constant $\gamma = \gamma(m, \ell) < 1$ such that for all n and linear mappings $\mathcal{L} : \mathbb{Z}_m^n \to \mathbb{Z}_m^s$,*

$$\left| \left|K^{\mathcal{L}}(v) \cap M_{n,\ell}(b)\right| - \left|K^{\mathcal{L}}(v)\right|/\ell \right| \leq (2\gamma)^n \tag{3.3}$$

for each $b \in \{0, \ldots, \ell - 1\}$ and $v \in \mathbb{Z}_m^s$.

The Linear Uniformity Lemma shows that if $|K^{\mathcal{L}}(v)|$ is large compared to $(2\gamma)^n$, then every MOD$_\ell$ residue class occurs with roughly the same frequency in $K^{\mathcal{L}}(v)$. In other words, intuitively speaking, $K^{\mathcal{L}}(v)$ looks random[5] to a MOD$_\ell$ counter. A combination of the Linear Uniformity Lemma and Theorem 3.19 yields the following:

Corollary 3.21 *There does not exist a set of linear polynomials over $\mathbb{Z}_m$ of size $o(n)$ that forms a simultaneous weak representation of the MOD$_\ell$ function over n variables, if m, ℓ are relatively prime to each other.*

[5] It is worthwhile to note that a set 'looking random' to a machine is an important notion in computational complexity. The machine considered here is weak: just a MOD$_\ell$ counter. However it is conjectured that 'efficient construction' of sets 'looking random' to polynomial size circuits, is possible. If true, such a conjecture has far reaching implications on derandomization of algorithms.

Proof: Assume that such a set $\mathcal{L} = \{\theta_1, \ldots, \theta_s\}$ exists, with $s = o(n)$. By the definition of weak simultaneous representation, there exists $x \in \{0,1\}^n$ such that $\mathcal{L}(x) = v \in \mathbb{Z}_m^s$ and $v \neq 0^s$. Applying Theorem 3.19, $|K^{\mathcal{L}}(v)|$ is at least $2^n/c^s$ for some constant c. The Linear Uniformity Lemma then implies that at least $\frac{2^n}{\ell c^s}(1 - o(1))$ elements of $M_{n,\ell}(b)$ are in $K^{\mathcal{L}}(v)$, for each b. As s is sublinear, choosing $b = 0$ yields a contradiction to the fact that $\mathcal{L}$ is a simultaneous weak representation of MOD_ℓ. ■

3.2.1 Fourier Analysis over Abelian Groups

Let G be a finite abelian group. We analyze the vector space of functions from G to the set of complex numbers $\mathbb{C}$, denoted by $\mathbb{C}^G$. As the boolean cube is the n-fold direct product of the two-element cyclic group $\mathbb{Z}_2$, analysis of boolean functions is a special case of this analysis. Of course, it is not necessary to view boolean functions sitting inside a vector space with an underlying field of characteristic zero. One can think of them sitting inside a space with the underlying field being finite (as done by Razborov-Smolensky and several authors later, for instance [BST90, ST06]) or even sitting inside a module, with fields replaced by commutative rings, as initiated by [BBR94] and further worked on in the first part of this chapter. In this section, we use complex numbers as it facilitates the powerful use of metric inequalities. With the seminal work of Kahn, Kalai and Linial [KKL88], complex Fourier analysis over the boolean cube has found numerous applications in computer science and discrete mathematics. An important difference between these works and what we do here is that our G in general will not be the boolean cube, but an m-ary cube i.e. $\mathbb{Z}_m^n$.

We equip $\mathbb{C}^G$ with the following inner product: let $\overline{z}$ denote the complex conjugate of $z \in \mathbb{C}$. For every $f, g \in \mathbb{C}^G$, define

$$\langle f, g \rangle = \frac{1}{|G|} \sum_{x \in G} \overline{f(x)} g(x).$$

Below, we find an interesting orthonormal basis for $\mathbb{C}^G$, called the Fourier basis. Let $\mathbb{C}^*$ represent the multiplicative group of complex numbers, i.e. $\mathbb{C} - \{0\}$. As G is abelian, we denote the group operation in G additively. A *character* χ of G is a homomorphism $\chi : G \to \mathbb{C}^*$, i.e. $\chi(a + b) = \chi(a)\chi(b)$, for every $a, b \in G$. Then, it is easy to verify that χ maps the identity of G, denoted by 0, to the identity of $\mathbb{C}^*$, denoted by 1. Further, if G has order m, then for any $a \in G$, $\chi(a)^m = \chi(ma) = \chi(0) = 1$. Thus, $\chi(a)$ is an mth root of unity, for each $a \in G$. This immediately shows that the set of characters of G, denoted by $\hat{G}$, is a finite set as G is finite.

Define the product of two characters $\chi_1, \chi_2 \in G$ as the following: $\chi_1 \circ \chi_2(x) = \chi_1(x)\chi_2(x)$. It is easy to verify that $\chi_1 \circ \chi_2$ is indeed a character. The trivial character, denoted by χ_0, that maps every element of G to 1 is called the *principal character* of G. Further, for each $\chi \in \hat{G}$, define the homomorphism χ^{-1} by imposing $\chi^{-1}(x) = \chi(x)^{-1}$. Then, clearly $\chi \circ \chi^{-1} = \chi_0$. Thus, $\hat{G}$ with the operation $\circ$ forms a finite abelian group with χ_0 serving as the identity. We state two basic properties of characters:

Proposition 3.22 *The following is true for any abelian group G:*

1. *$\sum_{x \in G} \chi(x)$ is equal to zero if $\chi \neq \chi_0$, otherwise is equal to $|G|$.*

2. *Dually, if x is a non-zero element of G, then $\sum_{\chi \in \hat{G}} \chi(x)$ is zero, otherwise it is $|G|$.*

Proof: We prove the second property and the first can be proved analogously. For any $x \neq 0$, we claim that there exists a $\chi' \in \hat{G}$ such that $\chi'(x) \neq 1$. Modulo this claim, we establish our property. Let $S = \sum_{\chi \in \hat{G}} \chi(x)$. Then,

$$\chi'(x)S = \sum_{\chi \in \hat{G}} \big(\chi' \circ \chi\big)(x) = S.$$

The last identity holds because the action of χ' is just a permutation of $\hat{G}$. Thus, $S(1 - \chi'(x)) = 0$. This implies $S = 0$ as $\chi'(x) \neq 0$. It remains to prove that indeed such a χ' exists.

Let the order of x in G be ℓ. Define $\chi'(x)$ to be any primitive ℓth root of unity. This naturally defines a homomorphism from the cyclic subgroup generated by x, denoted by G_x, to $\mathbb{C}^*$. This is extended to whole of G as follows. Let $G_x a_i$ for $i = 1, \ldots, k = |G|/\ell$ be the cosets of G_x. Set $\chi'(a_i) = 1$ for all i. This extends χ' naturally to all of G. ∎

For any $x \in G$, let δ_x be the function that maps x to 1 and every other element of G to 0. Clearly, $\Delta = \{\delta_x | x \in G\}$ forms a basis for $\mathbb{C}^G$. Using the second property in Proposition 3.22, one verifies that the following holds:

$$\delta_x \equiv \frac{1}{|G|} \sum_{\chi \in \hat{G}} \chi(-x)\chi$$

This immediately yields the following essential fact:

Fact 3.23 *The set of characters of a finite abelian group G spans the vector space $\mathbb{C}^G$.*

Further,

Lemma 3.24 *The set of characters forms an orthonormal basis for the vector space $\mathbb{C}^G$, i.e. the following holds:*

1. *Any two distinct characters χ_1, χ_2 are orthogonal to each other, i.e. $\langle \chi_1, \chi_2 \rangle = 0$.*
2. *$\langle \chi, \chi \rangle = 1$ for all $\chi \in \hat{G}$.*

Proof:

$$\langle \chi_1, \chi_2 \rangle \equiv \frac{1}{|G|} \sum_{x \in G} \overline{\chi_1(x)} \chi_2(x).$$

Observe that $\chi_1(x)$ lies on the unit circle. Hence, $\overline{\chi_1(x)} = \chi_1^{-1}(x)$. Thus,

$$\langle \chi_1, \chi_2 \rangle = \frac{1}{|G|} \sum_{x \in G} \left(\chi_1^{-1} \circ \chi_2 \right)(x).$$

Observe that $\chi_1 \neq \chi_2$ iff $\chi_1^{-1} \circ \chi_2$ is non-principal. Hence, applying the first property of Proposition 3.22, we are done. ∎

Combining Fact 3.23 and Lemma 3.24, we obtain the following fact that forms the basis of Fourier analysis:

Theorem 3.25 *If G is a finite abelian group, then every function $f \in \mathbb{C}^G$ can be uniquely expressed as a linear combination of the characters i.e. for every $x \in G$,*

$$f(x) = \sum_{\chi \in \hat{G}} \hat{f}(\chi) \chi(x) \tag{3.4}$$

where, for every $\chi \in \hat{G}$ the following holds:

$$\hat{f}(\chi) = \langle f, \chi \rangle = \frac{1}{|G|} \sum_{x \in G} \overline{f(x)} \chi(x). \tag{3.5}$$

In particular, this means that G and $\hat{G}$ have the same order. A more careful analysis shows that G and $\hat{G}$ are isomorphic to each other. Hence, (3.5) defines a linear invertible operator on $\mathbb{C}^G$, called the *Fourier transform.* The values $\hat{f}(\chi)$ are called *Fourier coefficients.* Interesting information about a function is revealed by inspecting its Fourier coefficients. The following very useful fact shows that the Euclidean norm of a function can be easily evaluated from its Fourier coefficients:

Theorem 3.26 (Parseval's Identity) *If G is an abelian group, the following holds for any $f \in \mathbb{C}^G$:*

$$\mathbb{E}_x \left| f(x) \right|^2 = \sum_{\chi \in \hat{G}} \left| \hat{f}(\chi) \right|^2. \tag{3.6}$$

Proof: Using (3.4), one writes

$$\mathbb{E}_x \left| f(x) \right|^2 = \mathbb{E}_x \left[\left(\sum_{\chi_1 \in \hat{G}} \hat{f}(\chi_1) \chi_1(x) \right) \left(\sum_{\chi_2 \in \hat{G}} \overline{\hat{f}(\chi_1)} \overline{\chi_1(x)} \right) \right].$$

This simplifies to the following:

$$\mathbb{E}_x \left| f(x) \right|^2 = \sum_{\chi_1, \chi_2 \in \hat{G}} \hat{f}(\chi_1) \overline{\hat{f}(\chi_1)} \langle \chi_1, \chi_2 \rangle.$$

Finally, (3.6) is established from the above by making use of the orthonormality of the set of characters as stated in Lemma 3.24. ∎

We recall below a beautiful and well-known trade-off, commonly referred to as the Uncertainty Principle, between the size of support set of a function and the size of the support set of its Fourier transform. Let the support set of a function f, denoted by $\mathrm{supp}(f)$, be the set of points at which the function evaluates to a non-zero value.

Theorem 3.27 (Uncertainty Principle) *For any $f \in \mathbb{C}^G$ that is not identically zero, the following holds:*

$$|supp(f)| \cdot |supp(\hat{f})| \geq |G|.$$

Proof: Let $||f||_\infty \equiv \max\{|f(x)| : x \in G\}$. Then,

$$\mathbb{E}_x \big|f(x)\big|^2 \leq \frac{\big|\mathrm{supp}(f)\big|}{|G|} ||f||_\infty^2.$$

Using the Fourier expansion of f given by (3.4), recalling that $\big|\chi(x)\big| \leq 1$ for any $\chi \in \hat{G}$, $x \in G$ and using the triangle inequality gives us the following:

$$||f||_\infty^2 \leq \left(\sum_{\chi \in \hat{G}} \big|\hat{f}(\chi)\big| \right)^2 \equiv \big|\big|\hat{f}\big|\big|_1^2$$

where $||\hat{f}||_1$ is the ℓ_1 norm of $\hat{f}$. Combining things we get

$$\mathbb{E}_x \big|f(x)\big|^2 \leq \frac{\big|\mathrm{supp}(f)\big|}{|G|} \big|\big|\hat{f}\big|\big|_1^2. \tag{3.7}$$

On the other hand, applying successively Parseval's identity and the Cauchy-Schwartz inequality yields the following:

$$\mathbb{E}_x\big|f(x)\big|^2 = \sum_{\chi\in\hat{G}}\big|\hat{f}(\chi)\big|^2 \geq \frac{1}{\big|\mathrm{supp}\big(\hat{f}\big)\big|}\big|\big|\hat{f}\big|\big|_1^2. \tag{3.8}$$

A combination of (3.7) and (3.8) easily proves the uncertainty principle.. ∎

3.2.2 Davenport constant

We draw on a notion from combinatorial group theory. Consider a fixed finite abelian group G. The *Davenport constant* of G, denoted by $s(G)$, is the smallest integer k such that every sequence of elements of G of length at least k, has a non-empty subsequence that sums to zero. The pigeon-hole-principle shows that $s(G)$ is finite if G is finite. This is because if we have a sequence of length larger than $|G|^2$, then some element a of G is repeated at least $|G|$ times. The sub-sequence formed by the first $|G|$ instances of a indeed sums to zero as the order of every element in G divides $|G|$. Thus, $s(G) \leq |G|^2$, which gives a quadratic upper bound on the Davenport constant w.r.t. the size of the group.

For specific groups, one can show much better bounds. For instance, if the group is $\mathbb{Z}_p$, then one can show, using the polynomial method, that $s(\mathbb{Z}_p)$ is p. Clearly, the lower bound follows by considering the sequence of $(p-1)$ occurrences of the identity element. Such a sequence has no non-empty subsequence summing to zero. The upper bound can be established as follows: Let $a_1, \ldots, a_p$ be a sequence of elements from $\mathbb{Z}_p$. Assume that no zero-sum subsequence of it exists. In other words, the polynomial $a_1x_1 + \cdots + a_px_p$ over $\mathbb{Z}_p$ evaluates to zero only at one point in the boolean cube $\{0,1\}^p$, which is the all zero point. Thus, applying Fermat's Little

Theorem, the polynomial $P \equiv 1 - (a_1x_1 + \cdots + a_px_p)^{p-1}$, strongly represents the OR function of p boolean variables over $\mathbb{Z}_p$. However, recall that in the last chapter we showed that the strong MOD$_m$-degree of OR is p. Hence, P of degree $p-1$ is a contradiction of the above and we are done.

Olson [Ols69a] showed a more general statement: Let G be an abelian p-group of the form $\mathbb{Z}_{p^{k_1}} \oplus \mathbb{Z}_{p^{k_2}} \oplus \cdots \oplus \mathbb{Z}_{p^{k_r}}$, where $\oplus$ denotes direct sum. Olson shows that $s(G) = 1 + \sum_{i=1}^{r} \left(p^{k_i} - 1\right)$ in this case. We show below that $s(\mathbb{Z}_m^r)$ is at most $c(m)r$, where $c(m)$ is a constant that just depends on m. Before doing that, we recall another result by Olson [Ols69b] that connects $s(G)$ with the set of boolean solutions to the equation $g_1x_1 + \ldots + g_nx_n = 0$, denoted by $K(G, n)$, where each $g_i \in G$.

Theorem 3.28 (Olson's Theorem) $|K(G, n)| \geq \max\{1, 2^{n+1-s(G)}\}$.

Proof:[adapted from [Ols69b]] We prove this by induction of n. For $n \leq s(G) - 1$, the theorem is vacuously true. Assuming it is true for n, we prove it for $n+1$. Let the equation be $g_1x_1 + \cdots + g_{n+1}x_{n+1} = 0$. By the definition of $s(G)$, there is a subsequence of $g_1, \ldots, g_{s(G)}$ that has a subsequence that sums to zero. W.l.o.g., assume this subsequence to be $g_1, \ldots, g_t$. Then consider the equation $(-g_2)x_2 + \cdots + (-g_t)x_t + g_{t+1}x_{t+1} + \cdots + g_{n+1}x_{n+1} = 0$. By our hypothesis, this equation on n variables has at least $2^{n+1-s(G)}$ solutions. For each such solution point u, we obtain a solution to the original equation over $n+1$ variables in which the value of x_1 is set to 1 in the following way: $x_1 = 1$, for $2 \leq i \leq t$, x_i is set to the value that is the complement of its value in u, and for $t < i \leq n+1$, x_i is set to its corresponding value in u. Finally, extend the solutions of $g_2x_2 + \cdots + g_{n+1}x_{n+1} = 0$ to our original equation by simply fixing $x_1 = 0$ to obtain at least another $2^{n+1-s(G)}$ solutions. Thus, we have at least $2^{n+2-s(G)}$ solutions in total, proving the theorem. ∎

3.2.3 Towards large support

The usefulness of Olson's Theorem for our purpose is evident from its following immediate corollary[6]:

Corollary 3.29 *Let $\mathcal{L} : \mathbb{Z}_m^n \to \mathbb{Z}_m^s$ be a linear map. Then, for all $v \in \mathbb{Z}_m^s$ such that $K^{\mathcal{L}}(v)$ is non-empty, we have $|K^{\mathcal{L}}(v)| \geq 2^{n+1-s(\mathbb{Z}_m^s)}$.*

Proof: Let $\mathcal{L} \equiv \{\theta_1, \ldots, \theta_s\}$ be the underlying linear forms, where $\theta_i = a_{i,1}x_1 + \cdots + a_{i,n}x_n$. As $K^{\mathcal{L}}(v)$ is non-empty, there exists $b \in \{0,1\}^n$ such that $\theta_i(b) = v_i$. Consider $\theta_i' = a_{i,1}'x_1 + \cdots + a_{i,n}'x_n$, where $a_{i,j}' = -a_{i,j}$ if $b_j = 1$ and otherwise $a_{i,j}' = a_{i,j}$, for each $1 \leq j \leq n$ and $1 \leq i \leq s$. Define $\mathcal{L}' \equiv \{\theta_1', \ldots, \theta_s'\}$. Then, it is straight-forward to verify that sets $K^{\mathcal{L}}(v)$ and $K^{\mathcal{L}'}(0^s)$ are in one-to-one correspondence with each other. The result follows by observing that Olson's Theorem implies $K^{\mathcal{L}'}(0^s)$ has size at least $2^{n+1-s(\mathbb{Z}_m^s)}$. ∎

In view of Corollary 3.29, it is sufficient to establish an $O(r)$ upper bound on $s(\mathbb{Z}_m^r)$ for proving Theorem 3.19. This is where Fourier analysis over groups of the form $\mathbb{Z}_m^s$ comes into play. Let $\mathrm{e}_m(t)$ denote the tth primitive m-th root of unity, i.e.

$$\mathrm{e}_m(t) \equiv \exp\Big(\frac{2\pi i t}{m}\Big)$$

[6] We have overloaded the symbol s in the statement of Corollary 3.29, but its meaning is clear from the context.

where i is the pure imaginary number, i.e. complex square-root of -1. Then, note that for each s-variate linear form[7] $\theta(x) \equiv a_1x_1 + \cdots + a_sx_s$ (with constants $a_i \in \mathbb{Z}_m$ and variable x_i taking value in $\mathbb{Z}_m$), $\mathrm{e}_m(\theta(x)) : \mathbb{Z}_m^s \to \mathbb{C}^*$ is a character of $\mathbb{Z}_m^s$. Hence, using the second property of characters from Proposition 3.22, we get[8]

Fact 3.30 *Let* $S(y) = \frac{1}{m}\sum_{j=0}^{m-1} \mathrm{e}_m(jy)$. *Then,* $S(y) = 0$ *if* $y \not\equiv 0 \pmod{m}$ *and* $S(y) = 1$ *otherwise.*

We are prepared to establish an upper bound on the Davenport constant of $\mathbb{Z}_m^r$ that is linear in r.

Theorem 3.31 *If* m *is even,* $s(\mathbb{Z}_m^r) \leq cr$, *where* $c = \frac{\log m}{\log m - \log(m-1)}$ *is a constant.*

Proof: Let $\mathcal{L} \equiv \{\theta_1, \ldots, \theta_r\}$ be a linear map from $\mathbb{Z}_m^s$ to $\mathbb{Z}_m^r$, such that $K^{\mathcal{L}}(0^r)$ is a singleton set, i.e. contains only the point 0^s. Let $\lambda_S : \mathbb{Z}_m^s \to \{0,1\}$ denote the characteristic function for any set $S \subseteq \mathbb{Z}_m^s$. Then, using Fact 3.30, one writes

$$\lambda_{\{0,1\}^s}(x) \equiv \frac{1}{m^s}\prod_{j=1}^{s}\left[\sum_{a=0}^{m-1}\mathrm{e}_m\big(ax_j\big)+\sum_{a=0}^{m-1}\mathrm{e}_m\big(a(x_j-1)\big)\right] = \frac{1}{m^s}\prod_{j=1}^{s}\left[\sum_{a=0}^{m-1}\big(1+\mathrm{e}_m(-a)\big)\mathrm{e}_m\big(ax_j\big)\right].$$

Let $m = 2\ell$. Then clearly for $a = \ell$, we have $(1 + \mathrm{e}_m(a)) = 1 + \mathrm{e}_m(\pi) = 0$ using a basic trigonometric identity. Thus, noting that $|\mathrm{supp}(\widehat{fg})| \leq |\mathrm{supp}(\hat{f})| \cdot |\mathrm{supp}(\hat{g})|$, we see that $|\mathrm{supp}(\widehat{\lambda_{\{0,1\}^s}})| \leq (m-1)^s$. Further,

$$\lambda_{K^{\mathcal{L}}(0^r)}(x) \equiv \left[\prod_{j=1}^{r}\left(\frac{1}{m}\sum_{a=0}^{m-1}\mathrm{e}_m\big(a\theta_j(x)\big)\right)\right]\lambda_{\{0,1\}^s}\big(x\big).$$

[7] There are precisely m^s such linear forms which is also the size of the group $\mathbb{Z}_m^s$.

[8] This has a direct proof using identities for summing geometric progressions.

Thus, one concludes

$$\left|\operatorname{supp}\left(\widehat{\lambda_{K^{\mathcal{L}}(0^r)}}\right)\right| \leq m^r \left|\operatorname{supp}(\widehat{\lambda_{\{0,1\}^s}})\right| \leq m^r(m-1)^s.$$

Applying the Uncertainty Principle, we get

$$m^r(m-1)^s \geq |\mathbb{Z}_m^s| = m^s$$

whence the result follows. ∎

The case of an odd m can be dealt with by the following simple trick. Multiply each linear form θ_i by 2. Viewing each modified linear form to be over $\mathbb{Z}_{2m}$ (instead of over $\mathbb{Z}_m$), we obtain a new map $\mathcal{L}' : \mathbb{Z}_{2m}^s \to \mathbb{Z}_{2m}^r$. It is easily verified that sets $K^{\mathcal{L}}(0^r)$ and $K^{\mathcal{L}'}(0^r)$ are in one-to-one correspondence with each other. Hence, applying Theorem 3.31 to $K^{\mathcal{L}'}(0^r)$ yields bounds on $K^{\mathcal{L}}(0^r)$ as well, though with a very slight worsening of the constant c.

Corollary 3.32 *For every m, $s(\mathbb{Z}_m^r) \leq cr$, where $c = \frac{\log(2m)}{\log(2m)-\log(2m-1)}$ is a constant that just depends on m.*

Combining Corollary 3.29 with bounds on $s(\mathbb{Z}_m^r)$ as given above, we immediately derive Theorem 3.19 which states that the size of each non-empty $K^{\mathcal{L}}(v)$ is at least $\frac{2^n}{c^s}$.

3.2.4 Uniformity

Our proof of the Uniformity Lemma uses an exponential sum argument. Use of exponential sums in circuit complexity was, as far as we know, introduced by Cai, Green and Thierauf [CGT96] and further pursued by Green [Gre99, Gre04]. Green's estimates were improved in a breakthrough work by Bourgain [Bou05] and further refined by Green, Roy and Straubing [GRS05]. The focus of these works is to show that the output of a restricted circuit with a single MOD_m gate at its output, is poorly correlated with the function MOD_ℓ, when m, ℓ are co-prime. The idea of using exponential sums to analyze the output of a circuit comprising several MOD_m gates is novel to our work.

Proof:[of the Linear Uniformity Lemma] We first write $|K^{\mathcal{L}}(v) \cap M_{n,\ell}(b)|$ as an exponential sum and then estimate this exponential sum by grouping the terms appropriately. The key to writing this out is the use of the basic identity from Fact 3.30, that we crucially used also while estimating the Davenport constant of $\mathbb{Z}_m^r$ in the proof of Theorem 3.31.

$$|K^{\mathcal{L}}(v) \cap M_{n,\ell}(b)| = \sum_{x \in \{0,1\}^n} \Big[\frac{1}{\ell} \sum_{c=0}^{\ell-1} \mathrm{e}_\ell(c(\sum_{k=1}^{n} x_k - b))\Big] \Big[\prod_{i=1}^{s} (\frac{1}{m} \sum_{j=0}^{m-1} \mathrm{e}_m\big(j\big(\theta_i(x) - v_i\big)\big))\Big]. \tag{3.9}$$

Separating out the $c = 0$ case, we rewrite the right hand side (RHS) of (3.9) as

$$\sum_{x \in \{0,1\}^n} \frac{1}{\ell} \prod_{i=1}^{s} (\frac{1}{m} \sum_{j=0}^{m-1} \mathrm{e}_m\big(j\big(\theta_i(x) - v_i\big)\big))$$
$$+ \sum_{x \in \{0,1\}^n} \Big[\frac{1}{\ell} \sum_{a=1}^{\ell-1} \mathrm{e}_\ell(a(\sum_{k=1}^{n} x_k - b))\Big] \Big[\prod_{i=1}^{s} \frac{1}{m} \sum_{j=0}^{m-1} \mathrm{e}_m\big(j\big(\theta_i(x) - v_i\big)\big)\Big].$$

The first term in the RHS is easily identified to be $|K^{\mathcal{L}}(v)|/\ell$. Hence we get

$$\begin{aligned}&\big||K^{\mathcal{L}}(v) \cap M_{n,\ell}(b)| - |K^{\mathcal{L}}(v)|/\ell\big| \\ &= \left| \sum_{x \in \{0,1\}^n} \left[\frac{1}{\ell} \sum_{a=1}^{\ell-1} \mathrm{e}_\ell(a(\sum_{k=1}^{n} x_k - b))\right] \left[\prod_{i=1}^{s} \frac{1}{m} \sum_{j=0}^{m-1} \mathrm{e}_m\big(j\big(\theta_i(x) - v_i\big)\big)\right] \right|. \qquad (3.10)\end{aligned}$$

We now estimate the RHS of (3.10). To do this, let us multiply out the terms in the summand inside the absolute value and then sum the resulting terms. We obtain $m^s(\ell - 1)$ terms after multiplying out the terms in the summand, each of which gives rise to a sum of the form

$$\frac{\mathrm{e}_\ell(-cb)\mathrm{e}_m(j)}{m^s \ell} \sum_{x \in \{0,1\}^n} \left[\mathrm{e}_m(j_1\theta_1(x) + \ldots + j_s\theta_s(x))\mathrm{e}_\ell(c \sum_{k=1}^{n} x_k)\right] \qquad (3.11)$$

where $(j_1, \ldots, j_s) \in \{0, \ldots, m-1\}^s$, $j = j_1v_1 + \cdots + j_sv_s$ and $c \in \{1, \ldots, \ell - 1\}$.

Bounding the absolute value of the expression in the previous equation is standard. We include it here for making our proof self-contained. Let the sum $j_1\theta_1(x) + \ldots + j_s\theta_s(x)$ give rise to a linear form that is denoted by $a_1x_1 + \ldots + a_nx_n$. Using the trigonometric identity $1 + \exp(i2\rho) = 2\exp(i\rho)\cos(\rho)$, and taking absolute values, we have

$$|(3.11)| = \left|\frac{1}{m^s\ell} \prod_{i=1}^{n} (1 + \mathrm{e}_m(a_i)\mathrm{e}_\ell(c))\right| = \left|\frac{2^n}{m^s\ell} \prod_{i=1}^{n} \cos\big(\pi(\frac{a_i}{m} + \frac{c}{\ell})\big)\right|. \qquad (3.12)$$

Let $\gamma = \max_{a_i \in \mathbb{Z}_m;\, c \in \mathbb{Z}_\ell} |\cos\big(\pi(\frac{a_i}{m} + \frac{c}{\ell})|$. Since, m and ℓ are co-prime and $c \neq 0$, it can be verified that $\gamma < 1$. Hence,

$$|(3.12)| \leq \frac{2^n\gamma^n}{m^s\ell}. \qquad (3.13)$$

Using the triangle inequality in the RHS of (3.10) and plugging in the bound of (3.13), we get

$$\big||K^{\mathcal{L}}(v) \cap M_{n,\ell}(b)| - |K^{\mathcal{L}}(v)|/\ell\big| \leq m^s(\ell-1)\frac{(2\gamma)^n}{m^s \ell}. \tag{3.14}$$

■

3.2.5 Lower Bounds for $\mathbf{CC}^0$

In this section, we show that our results on linear forms directly translate into lower bounds on the number of MOD_m gates in a $\mathrm{CC}[m]$ circuit computing the AND (or MOD_ℓ) function.

Consider a $\mathrm{CC}[m]$ circuit C having s MOD_m gates $g_1, \ldots, g_s$. For each gate g_i, we define the linear form $\theta_i = \sum_{j=1}^n c_{i,j} x_j$, where $c_{i,j}$ is the number (modulo m) of copies of input bit x_j feeding into g_i. We thus get at most s non-trivial linear forms that give rise to the linear map $\theta : \{0,1\}^n \to \mathbb{Z}_m^s$. One can easily verify that if $\theta(x) = \theta(y)$ for $x, y \in \{0,1\}^n$, then each gate of C outputs the same value on x and y. Consequently, C cannot distinguish x and y. Let $V \subseteq \mathbb{Z}_m^s$ be the set of those vectors which correspond to C outputting 1, i.e. for every y in V, $\theta(x) = y$ implies that $C(x) = 1$. If C is computing a non-constant function, then indeed there is a $y \in V$ such that $K^\theta(y)$ is non-empty. Applying Theorem 3.19, we immediately get $|K^\theta(y)| \geq 2^n/c^s$.

Theorem 3.33 (restatement of Theorem 3.4) *The support of a non-constant function computed by a $CC[m]$ circuit of size s has size at least $2^n/c^s$, where c is a constant for fixed m.*

Combining Theorem 3.4 with the Uniformity Lemma allows us to conclude that the support of C is almost equidistributed among the various residue classes of a MOD_ℓ counter. More precisely, one gets that for each $b \in \{0, \ldots, \ell - 1\}$,

$$\left|C^{-1}(1) \cap M_{n,\ell}(b)\right| \geq \frac{2^n}{c^s}(1 - \gamma^n c^s) = \frac{2^n}{c^s}(1 - o(1)).$$

This already shows that C cannot be computing the MOD_ℓ function. In fact, we show that C is very far from computing MOD_ℓ in a sense that is made precise below.

The first step in that direction is the following:

Lemma 3.34 *Consider any positive integers ℓ, m that are co-prime to each other and numbers $a, b \in \{0, \ldots, \ell - 1\}$. Then, for every $CC[m]$ circuit C of size $o(n)$, we have*

$$\left|\Pr_x[C(x) = 1 | x \in M_{n,\ell}(a)] - \Pr_x[C(x) = 1 | x \in M_{n,\ell}(b)]\right| \leq 2^{-\Omega(n)}. \qquad (3.15)$$

Proof: Let C have s gates. As before, we obtain a linear map $\theta : \{0,1\}^n \to \mathbb{Z}_m^s$ from C. Recall that V is the set of points in $\mathbb{Z}_m^s$ such that C outputs 1 on input x iff $\theta(x) \in V$. Thus, we obtain the following:

$$\begin{aligned}&\left|\Pr_x[C(x) = 1 \wedge x \in M_{n,\ell}(a)] \;-\; \Pr_x[C(x) = 1 \wedge x \in M_{n,\ell}(b)]\right| \\ &= \left|\sum_{y \in V}\left[\Pr_x[\theta(x) = y \wedge x \in M_{n,\ell}(a)] \;-\; \Pr_x[\theta(x) = y \wedge x \in M_{n,\ell}(b)]\right]\right|. \qquad (3.16)\end{aligned}$$

Using (3.3) from the Linear Uniformity Lemma and the triangle inequality, one can easily show that the summand in the RHS of (3.16), for every $y \in V$ is at most $2\gamma^n$, where the constant γ is defined in the Uniformity Lemma. Combining this with the fact that $|V| \leq m^s$ and $s = o(n)$, we obtain

$$(3.16) \leq |V| \cdot 2\gamma^n \leq m^s \cdot 2\gamma^n = 2^{-\Omega(n)}. \tag{3.17}$$

Since MOD_ℓ is an almost balanced function, i.e.

$$|\Pr_x[x \in M_{n,\ell}(a)] - \Pr_x[x \in M_{n,\ell}(b)]| \leq 2^{-\Omega(n)},$$

(3.17) implies Lemma 3.34. ∎

Recall, from Section 2.1.4 in Chapter 2, that Discriminator lemma of Hajnal et. al. states that if a circuit with a MAJ gate at the output computes a function f and the fan-in of the output MAJ gate is s, then for every $A \subseteq f^{-1}(1)$ and $B \subseteq f^{-1}(0)$ at least one of the sub-circuits feeding into the output gate $(1/s)$-discriminates f. Lemma 3.34 above implies that $\mathrm{CC}[m]$ circuits of sublinear size do not discriminate well the MOD_ℓ function. In particular, choose $A = M_{n,\ell}(1) \subset \mathrm{MOD}_q^{-1}(1)$ and $B = M_{n,q}(0) \subset \mathrm{MOD}_q^{-1}(0)$. Then it is easy to verify that Lemma 3.34 along with the Discriminator Lemma yields the following:

Theorem 3.35 (restatement of Theorem 3.5) *Any circuit of type* $\mathrm{MAJ}\circ CC_{o(n)}[m]$ *computing* MOD_ℓ *requires the output gate to have fan-in* $2^{\Omega(n)}$ *if* $(m, \ell) = 1$.

Thus, unless we take the majority vote of exponentially many $\mathrm{CC}_{o(n)}[m]$ circuits, we cannot compute MOD_ℓ. This is the sense in which $\mathrm{CC}_{o(n)}[m]$ circuits are far from computing MOD_ℓ.

3.3 Conclusion

In the first part of this chapter, we have demonstrated a new connection between the degree-complexity of a boolean function in a natural notion of representation by polynomials and its size-complexity in constant-depth boolean circuits with few MOD_m gates. Moreover, we have proved new lower bounds on the degree-complexity of MAJORITY and MOD_ℓ. These lower bounds on the degree-complexity are of independent interest, in addition to making progress on Smolensky's Conjecture via Theorem 3.1. Improving the lower bounds on the degree-complexity of OR is long overdue. Our work makes it an even more compelling research direction. For instance, a polylogarithmic lower-bound on the generalized MOD_m-degree of OR will result in a superpolynomial lower bound on the weak-generalized MOD_m-degree of MAJORITY (recall proof of Theorem 3.10). This will show that AC^0 circuits augmented with a polylogarithmic number of MOD_m gates, require superpolynomial size for computing MAJORITY (proof of Theorem 3.1). No such lower bounds are known.

In the second part of the chapter, we made progress towards Smolensky's Conjecture from another direction by considering circuits comprising only MOD_m gates. We proved that in sublinear size they cannot compute the AND and MOD_ℓ function if m and ℓ are co-prime. This involved the development of new techniques by novel combinations of Fourier analysis over complex numbers, exponential sums and additive number theory. We believe that these ingredients will be useful in making further progress. In particular, it is interesting to find out if these techniques can be combined to yield superlinear lower bounds on the size of depth-two circuits comprising only MOD_m gates. No such bound is known for any explicit function in NP if the output gate is a generalized gate.

Finally, we point out the following: subsequent to our work, Hansen [Han06a] has recently improved Theorem 3.1 w.r.t. computing MOD_ℓ. Hansen uses the breakthrough work of Bourgain [Bou05] on estimating the correlation between functions computed by low-degree polynomials over $\mathbb{Z}_m$ and MOD_ℓ. We remark that in the second part of Chapter 7, we simplify and improve Bourgain's work.

CHAPTER 4
Multiparty Communication with Input on the Forehead

Here, we formally define the model of computation that will occupy us in the next two chapters. Yao [Yao79] introduced the two party model of communication to investigate the mathematical structure and inherent complexity theoretic issues of distributed computing. He endowed his players with unlimited computational power in terms of time and space, in order to entirely focus on the communication needed among players as a resource. This model has inspired great research and too many beautiful results to cite. Indeed, the book by Kushilevitz and Nisan [KN97] provides an excellent exposition of this subject now known as Communication Complexity and surveys some of the diverse applications of this theory.

Our object of interest lies in a generalization of Yao's two player game to multiple players that was first defined by Chandra, Furst and Lipton in [CFL83]. In order to appreciate the subtleties of the multiparty model and its key differences from the two player version, we begin with the latter.

4.1 Two Player Games

In the basic model, there are two players often called Alice and Bob with unlimited computational power, who want to compute a certain function $f : \Sigma^n \to \{0,1\}$. The n input letters are partitioned into two sets X_A and X_B that are respectively assigned to Alice and Bob. The objective is that players devise a procedure beforehand so that given an arbitrary assignment to input letters, each player collaboratively determines the output of the function on the given assignment. They do so by communicating with each other according to a mutually agreed upon protocol. The protocol proceeds by players taking turns, as specified by the protocol, in communicating with each other. We assume that the players communicate with each other using the binary alphabet $\{0, 1\}$[1] . The cost of a protocol is the number of bits that the players communicate on the worst assignment of input letters. The communication complexity of a function f with respect to the above partition is the cost of the best protocol for computing it.

Notions of determinism, randomization and non-determinism manifest naturally in this setting. In a deterministic protocol Π, what Alice (Bob) communicates gets uniquely determined by the assignment to letters in X_A (X_B) and what has been communicated thus far by both players, called the communication history. The output of Π on any assignment is completely determined by the communication history at termination of Π. We say Π computes f precisely if $f(x, y) = \Pi(x, y)$ for each $x \in \Sigma^{X_A}$ and $y \in \Sigma^{X_B}$.

[1] This is w.l.o.g. as a protocol utilizing a fixed finite alphabet can be easily simulated by one with a binary alphabet with the cost blowing up by at most a constant factor.

In a randomized protocol, players are allowed to toss coins. In other words, players *jointly* select a random string r at the beginning and then follow a deterministic protocol that proceeds assuming Alice has input (x, r) and Bob has (y, r), where x, y are the original input assignments of Alice and Bob respectively. A randomized protocol is further allowed to err. Such a protocol Π computes f with advantage ϵ if $\Pr[f(x, y) = \Pi(x, y)] \geq 1/2 + \epsilon$ for every x, y, where the probability is taken over the random coin tosses r of Π. This is called the public coin model as the random string is accessible to each player without communication. In the private coin model, each player selects a random string that is not shared with the other player. As shown by Newman [New91], any protocol with public coin tosses can be simulated by a private coin protocol where the cost blows up by essentially an additive factor of at most $O(\log n)$. In this work, unless otherwise mentioned, protocols are assumed to toss coins publicly.

In a non-deterministic protocol, the prover, called 'God', furnishes a proof string s claiming that $f(x, y) = 1$. There is a deterministic verification protocol, denoted by Π, that players then use to verify the proof. More precisely, a non-deterministic protocol computes f if for every x, y such that $f(x, y) = 1$, there exists a proof string s such that $\Pi(x, y, s) = 1$. Further, if $f(x, y) = 0$, then $\Pi(x, y, s) = 0$ for all s . The cost of the protocol now includes the length of the proof string and the bits communicated by players to verify the proof.

Let $D(f), R^{\epsilon}(f)$ and $N(f)$ denote respectively the deterministic, randomized with advantage ϵ and non-deterministic communication complexity of the function f. Then, trivially for every $f : \Sigma^{X_A} \times \Sigma^{X_B}$, its deterministic, non-deterministic and randomized communication complexity is at most $\min\{|X_A|, |X_B|\}\log(\Sigma) + 1$ as the player with the minimum number of input letters communicates his/her input to the other, who just outputs the value of the function. Further, from the definitions above, we see that $N(f) \leq D(f)$ and $R^\epsilon(f) \leq D(f)$ for any f and ϵ. The example below shows that both non-determinism and randomization can offer huge savings in the cost of a protocol for computing some functions when compared with their deterministic counterparts.

Example. Define the Equality function $EQ : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ as $EQ(x, y) = 1$ iff $x = y$. The complement[2] of the equality function, called non-equality, is denoted by NEQ. It is not hard to verify that the best deterministic protocol essentially forces one player to communicate all its bits to the other i.e. $D(EQ) = D(NEQ) \geq n + 1$. On the other hand, the following simple non-deterministic protocol to compute NEQ provides exponential advantage in terms of cost: Let 'God' provide a $\log n$ bit string indicating an index i such that $x_i \neq y_i$. Alice just communicates the value of the bit x_i to Bob who can now verify if x_i and y_i are different. The cost incurred is $\log n + 2$, whence $N(NEQ) = O(\log n)$.

Randomization offers more dramatic cost savings for NEQ. Alice and Bob jointly choose a random n bit string r. Alice sends the bit representing the inner product modulo 2 of her input and the random string, i.e. $\langle x, r\rangle_2$, and Bob simply verifies if $\langle x, r\rangle_2 \neq \langle y, r\rangle_2$. The cost of this protocol is just two bits. Its correctness

[2] It is trivial to verify that the deterministic and randomized communication complexity of a function and its complement are the same.

follows from the fact that if x and y are different, then with probability exactly a half Alice and Bob detect it, i.e. $\Pr_r[\langle x, r\rangle_2 \neq \langle y, r\rangle_2] = 1/2$ for each $x \neq y$. Note that this protocol errs only on one side, i.e. if $x = y$, then Alice and Bob give the right answer with probability one. Further, the protocol can be repeated a constant number of times to reduce the error to any desired constant. Thus, $R^\epsilon(NEQ) = R^\epsilon(EQ) = O(1)$ for any fixed ϵ.

Before we move on further, let us make formal the last step of repeating a protocol enough number of times to boost its probability of success.

Observation 4.1 *Let Π be a randomized protocol that achieves advantage ϵ to compute a boolean function f. Then, the protocol Π' that runs $c_\epsilon \log(2/\delta)$ independent instances of Π and outputs the majority answer, achieves an advantage of at least δ to compute f.*

This implies that the cost of achieving any fixed advantage for computing a function is within a constant factor of the cost of achieving any other fixed advantage for computing the same function.

4.1.1 Lower Bound Techniques for Deterministic Protocols

A very convenient object for understanding the complexity of a function f is its *communication matrix* M_f. This is a boolean matrix that has $|\Sigma|^{|X_A|}$ many rows, one for each possible assignment to letters in X_A (the input letters of Alice), and $|\Sigma|^{|X_B|}$ many columns, one for each possible assignment to Bob's assignment. $M_f[x, y] = f(x, y)$.

For a deterministic protocol Π, we say that an input pair (x_1, y_1) is indistinguishable from the input pair (x_2, y_2) if Π generates the same communication history for both pairs. The basic weakness of deterministic protocols stems from the following simple observation on indistinguishability of input pairs.

Observation 4.2 *If a deterministic protocol* Π *does not distinguish* (x_1, y_1) *from* (x_2, y_2)*, then in fact it finds the following four pairs indistinguishable from each other:* (x_1, y_1), (x_2, y_2), (x_1, y_2) *and* (x_2, y_1).

This motivates the following definition: a set $R \subseteq \Sigma^{X_A} \times \Sigma^{X_B}$ is called a *rectangle* if for any two pairs $(x_1, y_1), (x_2, y_2) \in R$ we have that each of the four pairs (x_i, y_j) is in R for $i, j \in \{1, 2\}$. Further, a rectangle R is called monochromatic (w.r.t. a function f) if f evaluates to the same value at each element of R. Noting that a protocol of cost c can generate at most 2^c communication histories, Observation 4.2 immediately yields the following nice combinatorial fact:

Fact 4.3 *A deterministic protocol* Π *for* f *of cost* c *partitions the communication matrix* M_f *into at most* 2^c *many monochromatic rectangles.*

One convenient way of utilizing the above fact to prove lower bounds lies in the following idea: For obtaining a lower bound of c on the deterministic communication complexity of a target function f, we exhibit a set of input pairs of cardinality 2^c such that no two element from the set can lie in the same monochromatic rectangle. If they do, the protocol gets *fooled* to output a wrong answer on some input. Such a set is called a *fooling set*. The above method is called the *fooling set method* to prove lower bounds on the deterministic communication complexity of a function. The method is best illustrated by two text-book examples from [KN97]. For simplicity, let us assume that we have the binary alphabet i.e. $\Sigma = \{0, 1\}$.

Example 1. We show that the equality function EQ is hard for deterministic protocols by an application of the Fooling Set Method. Choose the set of pairs of equal strings along the diagonal of the matrix M_{EQ}. In any partition of M_{EQ} into monochromatic rectangles, no two such pairs can lie in the same rectangle. Thus, we need at least 2^n rectangles to partition M_{EQ}, one for each element of its diagonal. Additionally, we need at least one more rectangle to cover the zeroes of M_{EQ}, whence $D(EQ) = n + 1$.

Example 2. Define the function $DISJ$ by saying $DISJ(x, y) = 1$ iff there is no co-ordinate i such that both x, y have their ith bit set to 1. This is called the Disjointness function as one may view x, y to be characteristic vectors of subsets of $\{1, \ldots, n\}$. $DISJ$ then evaluates to 1 precisely if the two subsets are disjoint. It is a simple exercise to show that the set of pairs of the form (x, x^c) form a fooling set, where x^c is the characteristic vector corresponding the complement of the set represented by x. As the size of this set is 2^n and we additionally need at least one rectangle to cover the zeroes of M_{DISJ}, the Fooling Set Method yields $D(DISJ) = n + 1$.

4.1.2 Lower Bounds for Randomized Protocols

So far we have assumed that deterministic protocols are forced to give always the correct answer in contrast to their randomized counterparts that are allowed to err. Introducing errors of a different nature, this condition can be relaxed to allow error in deterministic protocols giving rise to the important notion of *distributional communication complexity* of a function. Given a probability distribution μ on the set of inputs $\Sigma^{X_A} \times \Sigma^{X_B}$, a deterministic protocol Π computes f with advantage ϵ

with respect to μ if $\Pr_{(x,y)\sim\mu}[f(x,y) = \Pi(x,y)] \geq 1/2 + \epsilon$. The (ϵ,μ)-distributional complexity of f, denoted by $D^{\epsilon,\mu}$, is then the cost of the best deterministic protocol computing f with advantage ϵ under distribution μ.

It turns out that the two notions of randomized and distributional communication complexity are not unrelated.

Fact 4.4 *For every distribution μ on $\Sigma^{X_A} \times \Sigma^{X_B}$, and for every $\epsilon > 0$, we have $D^{\epsilon,\mu}(f) \leq R^{\epsilon}(f)$.*

Proof: Consider a randomized protocol Π of cost c computing f with advantage ϵ. Notice that for each possible choice of its internal random string r, Π induces a deterministic protocol Π_r, where $\Pi_r(x,y) = \Pi(x,y,r)$. Now by the definition of Π,

$$\sum_r \Pr[r] \cdot \Pr_{(x,y)\sim\mu}[\Pi_r(x,y) = f(x,y)] \geq 1/2 + \epsilon.$$

This immediately yields that there exists at least one r for which Π_r has advantage at least ϵ and we are done. ∎

In fact, the relationship between randomized and distributional complexity of a function is more tight, as shown by Yao[3] [Yao83]:

$$R^{\epsilon}(f) = \max\{D^{\epsilon,\mu}(f) \mid \mu \text{ is a distribution}\}.$$

[3] Yao shows that such a relationship between the randomized and distributional complexity of a function holds much more generally and is not specific to the model of computation considered here. In particular, it is easy to verify that the proof of Fact 4.4 is a simple counting argument, not using any specifics of the model.

Fact 4.4 turns out to be quite helpful for proving lower bounds. The basic idea is that we find a convenient distribution μ and argue that all deterministic protocols with low cost will fail to attain the required advantage against our target function. Since we have to argue against deterministic protocols that are allowed to err, we define the following measure called *discrepancy*: given a rectangle $R \subseteq \Sigma^{X_A \times X_B}$, define its discrepancy under μ w.r.t. a function f, denoted by $\mathrm{disc}_{\mu}^{R}(f)$, to be the absolute value of the difference between the probability mass of inputs in R where f evaluates to 1 and the probability mass of inputs in R where f evaluates to 0. We recall the familiar algebraic trick[4] of mapping the boolean set $\{0, 1\}$ to the set $\{1, -1\}$. Under this mapping, discrepancy has the following nice expression:

$$\mathrm{disc}_{\mu}^{R}(f) = \left| \sum_{(x,y)\in R} \mu(x, y) f(x, y) \right|. \tag{4.1}$$

Hence, $0 \leq \mathrm{disc}_{\mu}^{R}(f) \leq \mu(R)$. Discrepancy thus measures how far a rectangle is from being monochromatic in the following sense: It attains the value of the probability mass $\mu(R)$ of the rectangle itself when R is monochromatic or completely unbalanced and is zero when the rectangle is perfectly mixed or balanced. The discrepancy of f under μ is simply the maximum over discrepancies of all rectangles, i.e. $\max\{\mathrm{disc}_{\mu}^{R}(f) \mid R \text{ is a rectangle}\}$. The reason we are interested in this quantity

[4] A more general form of this trick was used in the Razborov-Smolensky polynomial method described in Section 2.1.3.

is the following probabilistic variant of the fooling set method, widely known as the Discrepancy Method.

Lemma 4.5 (The Discrepancy Method) *For every distribution μ and any function f,*

$$R^{\epsilon}(f) \geq D^{\epsilon,\mu}(f) \geq \log\left(\frac{2\epsilon}{disc_{\mu}(f)}\right). \tag{4.2}$$

Proof: Consider any deterministic protocol Π that computes f with advantage at least ϵ and cost c. Let $\mathcal{R}$ be the set of rectangles into which Π partitions M_f. Clearly $|\mathcal{R}| \leq 2^c$. Assume, w.l.o.g, f and Π evaluate to 1/-1.

$$2\epsilon \leq \left|\sum_{(x,y)} f(x,y)\Pi(x,y)\mu(x,y)\right| \leq \sum_{R\in\mathcal{R}}\left|\sum_{(x,y)\in R} f(x,y)\Pi(x,y)\mu(x,y)\right|.$$

Noting that Π is constant-valued over every $R \in \mathcal{R}$ and recalling the definition of discrepancy in (4.1), we are done by the following:

$$2\epsilon \leq \sum_{R\in\mathcal{R}} \mathrm{disc}_{\mu}^{R}(f) \;\leq\; 2^c \mathrm{disc}_{\mu}(f). \tag{4.3}$$

■

The Discrepancy Method thus boils down to finding a convenient distribution μ such that that the discrepancy of the target function f is indeed very small. This yields good lower bounds on the communication complexity of f, using (4.2). Chor and Goldreich [CG85] used this method to obtain optimal lower bounds on the *Inner Product* (IP) function that is defined on the boolean alphabet as follows:

$IP(x, y) = \sum_{i=1}^{n} x_i y_i \pmod 2$. They showed that the discrepancy of IP was at most $1/2^n$ under the uniform distribution. This estimate along with (4.2) yields the following strong bound: any randomized protocol computing IP must have cost $\Omega(n)$ even if the advantage ϵ is an inverse sub-exponential function i.e. $\epsilon = 1/2^{o(n)}$.

However, the discrepancy method does not yield strong lower bounds (better than poly-logarithmic) for several natural functions including Disjointness. Razborov [Raz90], simplifying the earlier work of Kalyanasundaram and Schnitzer [KS92], developed a method proving linear lower bounds on the communication cost of protocols computing the Disjointness function with a *constant* advantage. Razborov's subtle calculations roughly show that under an appropriate distribution μ, every rectangle that assigns large weight to its set of disjoint points must also assign large weight to its set of non-disjoint points. As μ assigns constant weight to the set of all disjoint points, an averaging argument yields the desired bound.

Another interesting method based on tools from information theory was developed in Bar-Yossef et.al.[BYJKS04] that refined the earlier work of Chakrabarti et.al.[CSWY01]. This technique introduces a new measure called the *information cost* of a protocol. The idea is to measure the information that the communication history of a protocol reveals about inputs given to the player. Variants of this method have found more applications in both classical and quantum communication complexity (see for example [JKS03, JRS03]). We, however, do not delve more into Razborov's method or the information theoretic method as no generalization of them are yet known[5] that extend to the multiparty NOF model of communication which is our chief interest here. Extending either of these techniques to the NOF model remains an exciting challenge.

4.2 Number/Input in the Forehead model

One natural extension of Yao's two player model to k players is to partition the set of input letters into k sets and associate each such set with a player with every player having precisely one set. This results in the weakening of the model as k grows. For instance, in a partition where the size of each such set is equal, every player has no information about $(k-1)/k$ fraction of the input. However, this model, called the 'Number in the Hand' model, has important applications to other areas like data streams (see for example [CCM08, CKS03]).

On the other hand, we consider a model, introduced by Chandra, Furst and Lipton [CFL83], that is a significant strengthening of the two-player model. This is achieved by assigning inputs to the foreheads of players instead of assigning them to their hands. More precisely, let the sets $X_1, \ldots, X_k$ form a k-wise partition of the input letters as before. Player i's forehead is assigned X_i and i sees every other forehead except his own. Just as in the last section on two players, k-player communication protocols can be defined for computing functions $f : \Sigma^{X_1} \times \cdots \times \Sigma^{X_k} \rightarrow \{0, 1\}$. The cost of a protocol, as before, is the worst case cost. Generalizing the notions from the last section, we denote by $D_k(f)$, $N_k(f)$, $R_k^\epsilon(f)$, and $D_k^{\mu,\epsilon}(f)$ respectively the k-party

[5] Information theoretic techniques have had some success for restricted multiparty protocols (see for example [Cha07a, GP08]).

deterministic, non-deterministic, randomized and distributional (with advantage ϵ) communication complexity of f.

In order to illustrate the power of the new model, we consider the following:

Example. Recall the Equality function EQ from the last section. We generalize it: Let $EQ_k(x_1, \ldots, x_k) = 1$ precisely if all the k strings are equal i.e. $x_1 = x_2 = \cdots = x_k$. While EQ_2 was shown to be hard for two players, it is easy for k players to compute EQ_k i.e. $D_k(EQ_k) = 2$ for any $k \geq 3$. To see this, note that if two strings x_i and x_j differ, then Player k, for each $k \neq i, j$, spots this difference without communicating with others. Thus, the protocol simply boils down to the following: Player 1 announces if he/she spots any difference followed by Player 2 doing the same.

Remark 4.6 *The key feature of the 'Input on the Forehead' model that gets used in the protocol for k-wise Equality is that every* $(k-1)$*-tuple of input bits is accessible to some player.*

The multiparty model differs from the two party one in another key feature: the information available to two players *overlap* a lot. The following example illustrates the fact that this feature provides substantial computational power to multiparty protocols as opposed to two-player ones where there is no overlap of information.

Example. Generalize the Disjointness function defined earlier to k-wise Disjointness, denoted by $DISJ_k : \left(\{0,1\}^n\right)^k \rightarrow \{0,1\}$, in the following way: consider the $k \times n$ boolean matrix A formed from the k input strings $x_1, \ldots, x_k$ in the argument of $DISJ_k$ by placing x_i in the ith row of A. Thus, $A[i,j] = x_i[j]$ for $1 \leq i \leq k$ and $1 \leq j \leq n$. Then $DISJ_k(x_1, \ldots, x_k) \equiv_{\text{def}} 1$ iff there does not exist a j such that $A[i,j] = 1$ for all $1 \leq i \leq k$ i.e. A does not contain an all-one column. Recall that we showed, by a simple application of the Fooling Set Method, $DISJ_2$ requires $\Omega(n)$ bits to be communicated by two players employing the best deterministic protocol. Grolmusz [Gro94a] found a surprisingly powerful protocol for k-players that implies the complexity of $DISJ_k$ decreases exponentially with k. We describe this remarkable phenomenon using an elegant protocol due to Pudlák [Pud06].

Each boolean string that can appear in a column of A is called a *pattern*. Hence, the set of all patterns is the boolean cube $\{0,1\}^k$. Given an instance of A, we assign weights to the vertices (patterns) and edges of the cube in the following way: a pattern's weight is the number of times it occurs in the columns of A. The weight of an edge e connecting patterns u, v is the sum of the weights of u and v. Edge e is in the ith direction if patterns u, v differ only in their ith bit. Hence, the value of the bit held by the ith player of a column is irrelevant for determining if the column contributes to the weight of an edge in the ith direction. Thus, the following holds.

Observation 4.7 *The weight of each edge in the ith direction can be determined precisely by the ith player without any communication.*

Lemma 4.8 *Given the weight of pattern u, there is a deterministic k-player protocol, denoted by $\Pi_{u,v}$, of cost $O(k \log n)$ that outputs the weight of pattern v.*

Proof: Fix a path $P = e_1 e_2 \cdots e_t$ of length $t \leq k$ in the cube from u to v. Let the sequence of patterns visited along this path be $v_0 v_1 \cdots v_t$ with $u = v_0$ and $v = v_t$. The players compute the weight of v by successively computing weights of $v_1, v_2, \ldots, v_t$ along P in the following way: assume the weight of v_i is known. Let the edge e going out of v_i be along the jth direction in the cube. Then, by Observation 4.7, Player j

knows the weight of e and thus can compute the following: $\text{weight}(v_{i+1}) = \text{weight}(e) - \text{weight}(v_i)$. As $\text{weight}(v_{i+1}) \leq n$, he can announce this weight by communicating $\log n$ bits. Repeating this step t times, once for every vertex in the path P, we determine the weight of pattern v. ∎

Note that protocol $\Pi_{u,v}$ exploits one of the key features of the k-party model: every $(k-1)$-tuple of inputs is accessible to some player. Recall that the same feature was used by the constant cost protocol for EQ_k.

Lemma 4.8 shows that if players can somehow determine the weight of some pattern in low cost, then they can find the weight of the all-one pattern with little additional cost yielding a protocol to compute Disjointness efficiently. In order to do so, let us note the following:

Observation 4.9 *For any assignment of inputs, there is always one pattern whose weight is at most $n/2^k$.*

Observation 4.9 is utilized to yield a protocol that finds a pattern and its weight.

Lemma 4.10 *There is a deterministic protocol, denoted by Π_{start}, of cost at most $O(n/2^k + k + \log n)$, involving just the first two players, that for every assignment outputs a pattern and its weight.*

Proof: Both Player 1 and Player 2 see foreheads of other players that make up the sub-matrix of A, obtained by deleting the two rows occupied by the foreheads of Player 1 and 2, of size $(k-2) \times n$. Denote this sub-matrix by A'. Players 1 and 2 choose (without communicating among themselves) the pattern in A' with least weight[6] . Denote this pattern by u. Applying Observation 4.9 to A', we conclude that $\text{weight}(u) \leq n/2^{k-2}$. Player 1 communicates the bits held on the forehead of Player 2 at positions that correspond to the columns at which pattern u occurs in A'. This requires $\text{weight}(u)$ many bits of communication. Player 2, reading the forehead of Player 1, then determines the weight of the four patterns $00u$, $01u$, $10u$ and $11u$. He chooses one of these patterns and communicates both his choice and its weight using at most $(\log n + k)$ bits. ∎

Remark 4.11 *The overlap in information accessible to Player 1 and Player 2 is playing a key role in the protocol Π_{start}.*

The protocol for computing $DISJ_k$ is easily derived by running Π_{start} followed by $\Pi_{u,1^n}$, where u is the pattern whose weight is determined by Π_{start}. This yields $D_k(DISJ_k) = O(n/2^k + k \log n)$. Thus, $\log n$ players can compute Disjointness communicating only $O(\log^2 n)$ bits!

In fact, noting that the above protocol is easily modified to count the number of occurrences of any pattern in the input matrix A, one concludes the following slightly more general fact: Any function that just depends on the number of occurrences of a certain pattern in its input can be computed efficiently by $\log n$ players. More formally, let $D : \{0, \cdots, n\} \rightarrow \{0, 1\}$ be any predicate. For any pattern u of length k, define $G_k^{D,u} : (\{0,1\}^n)^k \rightarrow \{0,1\}$ by insisting $G_k^{D,u}(x_1, \ldots, x_k)$ be equal to $D(\text{weight}(u))$, where $\text{weight}(u)$ is the number of columns containing the pattern u

[6] In case of a tie, they resolve it according to a predetermined, mutually agreed-upon preference rule.

in the matrix A of size $k \times n$ induced from the k binary strings $x_1, \ldots, x_k$ as before. Then, $D_k(G_k^{D,u}) = O(n/2^k + k \log n)$.

The *Generalized Inner Product* function, which is the k-party analogue of Inner Product, is obtained by setting D as the parity predicate. The result of Babai, Nisan and Szegedy [BNS92] shows that the above upper bound on $G_k^{D,u}$ is nearly tight for the Generalized Inner Product (more generally for any mod-counting predicate D as shown by Grolmusz [Gro92]) by providing almost matching lower bounds of $\Omega(n/4^k)$. Techniques introduced in Chapter 6, provide alternative proofs of such lower bounds in addition to deriving lower bounds for predicates D for which earlier methods did not work.

The lower bounds cited above degrade exponentially fast with the number of players k. It is of significant interest to find bounds that do not degrade that fast. This is wide open and no explicit function is known for which we can prove non-trivial lower bounds for more than $\log n$ players. The difficulty of obtaining such bounds may be partly explained by the following surprising connection with ACC^0. Building upon the work of Yao [Yao90], Beigel and Tarui [BT94] showed the following strong result:

Theorem 4.12 ([BT94]) *For every function f computable by ACC^0 circuits of quasipolynomial size, there exists a multivariate polynomial P of degree at most polylogarithmic in n, over the ring of integers, that satisfies the following:*

- *There exists a constant c such that the absolute value of the coefficient of every monomial of P is at most $2^{(\log n)^c}$.*

- *There exists a symmetric function SYMM, such that for every $x \in \{0,1\}^n$, $f(x) = SYMM(P(x))$.*

The relevance of the above characterization of ACC^0 for multiparty communication complexity is evident from its following consequence:

Theorem 4.13 *For each function f in ACC^0, there exists a constant c such that f can be computed in polylogarithmic cost by $(\log n)^c$ players, under every partition of input bits, using a deterministic protocol.*

Proof: Consider the polynomial P over integers that computes f in the sense described in Theorem 4.12. Let d be the degree of P. Assume there are $d+1$ players. Then, for any partition of input bits, every monomial of P can be computed by some player without communicating with others. The players accordingly divide the monomials into $d+1$ classes so that Player i can compute every monomial in Class i. Each player announces the sum of the contribution made by monomials in his/her class, weighted by their coeffcients in P. Observe that there at most 2^{d+1} many monomials in a class and recall that each coefficient in P has absolute value at most $2^{(\log n)^c}$. Thus, each player communicates at most $\log(2^{d+1}2^{(\log n)^c}) = (d+1) + (\log n)^c$ bits. Hence, in total, $d\,(d+1+(\log n)^c)$ bits of communication suffice. As $d = (\log n)^{c'}$ and both c', c are constants independent of n, the cost of the protocol is merely polylogarithmic. ∎

4.3 Stars and Cylinders Intersections

The first thing to note is that the notion of a two-dimensional communication matrix from the two-party model naturally generalizes to a k dimensional array or tensor in the k-player model. More precisely, given $f : \Sigma^{X_1} \times \cdots \times \Sigma^{X_k} \rightarrow \{0,1\}$, M_f

is the boolean communication tensor, where $M_f[x_1, \ldots, x_k]$ is simply $f(x_1, \ldots, x_k)$, where $x_i \in \Sigma^{X_i}$.

We say that a set of k elements of $\Sigma^{X_1 \times \ldots \times X_k}$ forms a *star* if it is of the form:

$$(x'_1, x_2, \ldots, x_k), (x_1, x'_2, \ldots, x_k), \ldots, (x_1, x_2, \ldots, x'_k)$$

where the x_i are values for the input letters in X_i for each i with $x_i \neq x'_i$. In that case, we call $(x_1, x_2, \ldots, x_k)$ the *center* of this star. Further a set S is called *star-closed* if for every star in S, the center of the star is also in S. Then, the following observation, first made in [CFL83], explains the importance of star-closed sets for multiparty communication complexity.

Observation 4.14 *For any deterministic protocol* Π*, the set of inputs that lead* Π *to follow a given communication history is star-closed.*

Proof: Observe the following fact about deterministic communication protocol Π: at any point in the protocol, player i cannot distinguish between inputs $(x_1, \ldots, x_i, \ldots, x_k)$ and $(x_1, \ldots, x_i^{'}, \ldots, x_k)$ conditioned on the fact that the communication history generated by both inputs until that point in the protocol is the same. Thus, if the k inputs $(x_1^{'}, x_2, \ldots, x_k), \ldots, (x_1, \ldots, x_i^{'}, \ldots, x_k), \ldots, (x_1, \ldots, x_{k-1}, x_k^{'})$ share the same communication history τ, then Π communicates τ on the input $(x_1, \ldots, x_k)$ as well.
∎

An immediate but useful corollary of the above is the following generalization of Fact 4.3:

Corollary 4.15 *A deterministic k-party protocol Π computing a function f partitions the communication tensor M_f into at most 2^c f-monochromatic star-closed sets.*

However, unlike Fact 4.3 of two party protocols, Corollary 4.15 is much harder to use in practice. In particular, there are no known super-polylogarithmic bound (i.e. bounds of the form $(\log n)^{\omega(1)}$) for any explicit function using Corollary 4.15 directly, even for three players. Chandra et.al.[CFL83], introducing the method, used it in conjunction with Ramsey[7] Theory, to obtain a super-constant bound on the 'exactly-N' function, denoted by E_N^k. Let $E_N^k(x_1, \ldots, x_k)$ be 1 iff $\sum_{i=1}^k x_i = N$, where each x_i is a n-bit integer from the set $\{1, \ldots, N\}$. Chandra et.al. characterized the deterministic k-party communication complexity of E_N^k in terms of a combinatorial number $\chi_k(N)$ defined as follows: $\chi_k(N)$ is the smallest number of colours needed to colour the set $\{1, \ldots, N\}^{k-1}$ such that for each point $(x_1, \ldots, x_k)$ and each integer $\lambda \neq 0$ the following property holds: the k points $(x_1, \ldots, x_{k-1}), (x_1 + \lambda, x_2, \ldots, x_{k-1}), (x_1, x_2 + \lambda, \ldots, x_{k-1}), \ldots, (x_1, x_2, \ldots, x_{k-1} + \lambda)$ do not receive the same colour if they all lie in $\{1, \ldots, N\}^{k-1}$. While [CFL83] showed that $D_k\left(E_N^k\right) = \Theta\left(\log \chi_k(N)\right)$, determining good upper and lower bounds for $\chi_k(N)$ remain open problems. However, one knows that $\chi_k(N) = \omega(1)$, whence the super-constant lower bound on E_N^k follows.

[7] See the book by Graham et.al. [GRS90] for an excellent introduction to Ramsey Theory.

Before moving on, we note that we make further use of Ramsey Theory and Corollary 4.15 in the next chapter.

In the following discussion, we view star-closed sets in a different way. This point of view was introduced in the seminal work of Babai, Nisan and Szegedy [BNS92] and very effectively used along with the discrepancy method to obtain the first strong lower bounds on multiparty communication complexity of an explicit function.

A subset S_i of $\Sigma^{X_1 \times \ldots \times X_k}$ is a *cylinder in the ith dimension* if membership in S_i is independent of the ith coordinate, i.e. if for all $x_1, x_2, \ldots, x_k$ and any x_i' we have $(x_1, \ldots, x_i, \ldots, x_k) \in S_i$ iff $(x_1, \ldots, x_i', \ldots, x_k) \in S_i$. We say that S is a *cylinder intersection* if $S = \bigcap\limits_{1 \le i \le k} S_i$ where S_i is a cylinder in the ith dimension. A cylinder intersection is called *f-monochromatic* if the function f evaluates to the same value on every input instance in the intersection. The following lemma shows the equivalence of cylinder intersections and star-closed sets:

Lemma 4.16 *A set $S \subset \Sigma^{X_1 \times \ldots \times X_k}$ is a cylinder intersection iff it is star-closed.*

Proof: It is not hard to verify that every cylinder intersection is star-closed. Let us establish the other direction. Given a star-closed set S, define

$$\phi_S^i \equiv \big\{(x_1, \ldots, x_i, \ldots, x_k) \in \Sigma^{X_1 \times \cdots X_k} \mid \exists x_i^{'} : (x_1, \ldots, x_i^{'}, \ldots, x_k) \in S\big\}.$$

Then, one verifies that ϕ_S^i is a cylinder in the ith direction. Further, every element in S is in ϕ_S^i for each $1 \le i \le k$. Consider any $(x_1, \ldots, x_k)$ that lies in the intersection of all these cylinders. For each i, the definition of ϕ_S^i gives a point $(x_1, \ldots, x_i^{'}, \ldots, x_k)$ in S. The center of k such points is precisely $(x_1, \ldots, x_k)$ that must be in S as it is star-closed. Thus, we have established $S = \cap_{i=1}^k \phi_S^i$. ∎

Remark 4.17 *We can restate Corollary 4.15 in terms of cylinder intersections in the following manner: Let $f : \Sigma^{X_1 \times \ldots \times X_k} \to \{0,1\}$ be a function of k-inputs. Any k-party communication protocol of cost c computing f partitions the input space into at most 2^c f-monochromatic cylinder intersections.*

4.3.1 Discrepancy of Cylinder Intersections

The notion of discrepancy over rectangles generalizes to discrepancy over cylinder intersections in an obvious way: for a distribution μ over $\Sigma^{X_1 \times \cdots \times X_k}$ the discrepancy of function f over a cylinder intersection C, denoted by $\mathrm{disc}_{\mu}^{C}(f)$, as before is given by $\big|\sum_{(x_1,\ldots,x_k)\in C} f(x_1,\ldots,x_k)\mu(x_1,\ldots,x_k)\big|$. Here, we have assumed f to be 1/-1 valued. If ϕ is the 0-1 valued characteristic function of C then we can factorize it as $\prod_{i=1}^{k} \phi^i$, where ϕ^i is the characteristic function of the cylinder in the ith direction. It is straightforward to verify that one can rewrite things as follows:

$$\mathrm{disc}_{\mu,k}^{C}(f) = \left|\mathbb{E}_{x\sim\mu} f(x)\phi^1(x)\cdots\phi^k(x)\right|, \tag{4.4}$$

where x is a random k-tuple chosen according to μ from $\Sigma^{X_1 \times \cdots \times X_k}$. This way of expressing the discrepancy of a cylinder intersection is very convenient for the manipulations done to estimate discrepancy of concrete functions in Chapter 6.

Maximizing $\mathrm{disc}_{\mu,k}^{C}(f)$ over all cylinder intersections C yields the discrepancy of f over the distribution μ. An argument, identical to the two player case, immediately gives rise to the Discrepancy Method for multiple players:

Lemma 4.18 (The Multiparty Discrepancy Method) *For every function f and every distribution μ and every integer $k \geq 2$,*

$$R_k^{\epsilon}(f) \geq D_k^{\epsilon,\mu}(f) \geq \log\left(\frac{2\epsilon}{disc_{\mu,k}(f)}\right). \tag{4.5}$$

4.4 Communication Complexity Classes

Communication complexity is like a mini-world, existing independently inside the bigger world of computational complexity. Indeed, for each major complexity class, one can define its corresponding communication complexity analogue. This was first done by Babai, Frankl and Simon [BFS86] for the two-player model. This can be naturally extended to the k-player model. We define directly the multiparty complexity classes below.

The first thing to do is to fix our notion of "efficient" protocols. Noting that every function has communication complexity at most n, conventionally protocols of poly-logarithmic cost have been viewed as efficient. This naturally gives rise to the classes P_k^{cc}, NP_k^{cc} and BPP_k^{cc} as the class of those boolean functions that have efficient k-party deterministic, non-deterministic and randomized (bounded advantage) protocols respectively. The class coNP_k^{cc} is the class of functions whose complement have efficient non-deterministic protocols. While other complexity classes can be defined in the same spirit, we focus on these classes in this work.

We summarize some of the results stated earlier in terms of these complexity classes. The communication complexity of the function non-equality shows that P_2^{cc} is strictly contained in NP_2^{cc}. The same function also witnesses the separation of P_2^{cc}

from BPP_2^{cc}. Further, Equality separates BPP_2^{cc} from NP_2^{cc} by showing that the former is not a subset of the latter. On the other hand, the communication complexity of non-Disjointness (through the results of [BFS86, KS92, Raz90]) complements this by showing that NP_2^{cc} is not a subset of BPP_2^{cc}.

While for two players we have nice explicit separations of complexity classes, such separations, until recently, were not known for three or more players. In Chapter 6, we obtain explicit separation between BPP_k^{cc} and NP_k^{cc}, for every $k = o(\log \log n)$.

CHAPTER 5
Languages with Bounded Symmetric Multiparty Communication Complexity

5.1 Introduction

In the previous chapter, we discussed the communication complexity of a function with respect to a fixed partition of its input letters. In this chapter, we look at variable partitions of the input and restrict ourselves to deterministic protocols[1] . The *k-party symmetric communication complexity* of a function f, denoted by $D_k^{\text{sym}}(f)$, is defined to be the deterministic communication complexity of f with respect to the worst partition of its input. Variable partition models are mainly motivated from their applicability in proving lower bounds in other models of computation with no explicit mention of communication. Typically, such applications proceed by deriving an efficient communication protocol for f, with respect to every partition of its input letters, from the efficient algorithm for computing f in the given model. Efficient algorithms for f in the model are then ruled out by showing that f has large symmetric communication complexity.

We obtain new insight into the multiparty model by focusing on functions that have bounded k-party symmetric complexity, where $k \geq 3$ is an arbitrary constant.

[1] In this chapter, every protocol is deterministic unless stated explicitly otherwise.

A priori, there is no reason to guess that the communication complexity of a function has any bearing on its time-space complexity. Yet Szegedy [Sze93] shows that languages with two party bounded symmetric communication complexity can be computed economically by very shallow ACC^0 circuits. This surprising result is a consequence of the many beautiful characterizations of the class of such functions obtained in [Sze93]. A natural direction to pursue is to generalize these characterizations to the k-party model. Such an effort was initiated in the work of Tesson [Tes03].

We however show in Section 5.2 that there are languages with arbitrarily large uniform circuit complexity whose three-party communication complexity is bounded by a constant even for the worst-case partition of the input instances among the players. An analogous result for non-uniform circuit complexity is also derived. These languages are constructed using specially crafted *error-correcting codes*. Because of these results, we cannot expect to obtain characterizations of languages of bounded symmetric multiparty complexity that are as nice as those for the two-player case.

As remarked and exemplified in the previous chapter, the following key features of the multiparty model can be used to devise clever protocols: first, every input bit is seen by several players, second, every $(k-1)$-tuple of input positions is seen by at least one of the k players, and third, all players know the partitioning of the input, i.e., they know which positions they actually see. In the next section we show that this combination of features gives three-party protocols enough power to compute functions of arbitrarily high circuit-complexity in constant communication for every possible partition. If we remove the first two properties then we obtain essentially the multiparty "input in the hand" model which is computationally even weaker than the two-party communication model. To understand how crucial the last property is, we consider two restricted classes of languages/functions in which this advantage is in some sense taken away.

First, we consider in Section 5.3.3 languages with a *neutral letter* [BS95, BIL$^+$05], i.e. a letter which can be inserted or deleted at will in an input word without affecting its membership in the language. We show that every such language having bounded k-party communication complexity for some fixed k is regular. On the other hand, it is worth noting that the class of regular languages with a neutral letter that have constant k-party communication complexity has been nicely characterized by Tesson [Tes03] in terms of algebraic properties of their minimal automaton. Our results indicate that the presence of a neutral letter is a severe handicap in the multiparty game and suggests that it might be easier to prove communication complexity lower bounds under this assumption.

Finally, in Section 6.5.2, we use the Generalized Van der Warden Theorem to prove that for any fixed $k \geq 3$ the *symmetric* functions that can be computed in bounded k-party communication complexity by k-players are exactly the symmetric functions that have bounded 2-party complexity.

5.2 Functions with bounded multiparty complexity but high time/space complexity

In this section, we exhibit languages of arbitrarily large circuit complexity but with bounded multiparty communication complexity. For a language L and an *encoding* $C : \{0,1\}^* \rightarrow \{0,1\}^*$, we denote by $C(L)$ the set $\{C(x);\ x \in L\}$. We prove that for a suitably chosen error-correcting code C, any language L is such that its encoding $C(L)$ has bounded multiparty communication complexity. We will choose C such that the corresponding encoding and decoding function are efficiently computable and hence the time/space/circuit complexities of L and $C(L)$ will be closely related.

As a warm-up, we start with the *unary encoding* C_U defined as follows: for $x \in \{0,1\}^*$, $C_{\mathrm{U}}(x) = 0^x 1 0^{2^n - x - 1}$, where n is the length of x and x is interpreted as an integer between 0 and $2^n - 1$. Hence, C_{U} encodes bit strings of length n into strings of length 2^n having a single 1 in a one-to-one way.

Lemma 5.1 *For any language L and integer $k \geq 3$, $D_k^{sym}(C_{\mathrm{U}}(L)) \leq 3$.*

Proof: Without loss of generality $k = 3$. On an input w that is split among the three parties, the players need to verify two things: 1) whether w is a valid encoding of some string x, and 2) whether the corresponding string x is in L. To verify the first property, the players only need to check whether at least one of them sees a 1 and whether none of them sees two or more 1s. They can communicate their observations regarding this using six bits in total. Next, one of the players who sees the one, determines the unique string x with $C_{\mathrm{U}}(x) = w$. He can do this solely based on the position of the one since he knows how w is partitioned. This player can also determine whether $x \in L$ and hence $w \in C_{\mathrm{U}}(L)$. He communicates his conclusion to the other parties by sending one more bit. Hence in total players exchange at most seven bits. The protocol can be optimized so that each player simultaneously sends one bit of information for the total of three bits. ∎

The disadvantage of the unary encoding is its inefficiency: because codewords are exponentially longer than the words they encode, we cannot provide efficient reductions between L and $C(L)$. A better encoding can be obtained by concatenating Reed-Solomon codes with the unary encoding. In the 3-party scenario at least one of the parties has on its forehead at least a (1/3)-fraction of the input. Hence, if the chosen encoding has the property that from an arbitrary (1/3)-fraction of the input the whole word can be reconstructed (assuming the input is an encoding of some word, i.e., assuming that the input is a codeword) the other two parties can reconstruct the whole input and verify whether the parts on remaining foreheads are consistent with such an input. With the proper choice of parameters Reed-Solomon codes have this property.

Let n be a large enough integer, $m = \lceil \log_2 3n \rceil$ and $d = n/m$. Any string $x \in \{0,1\}^n$ can be interpreted as a sequence of d elements from $GF[2^m]$. Define p_x to be the degree $d-1$ polynomial over $GF[2^m]$ whose coefficients are given by x. Define the Reed-Solomon encoding by $C_{\mathrm{RS}}(x) = p_x(g_0)p_x(g_1)\cdots p_x(g_{3d-1})$, where $GF[2^m] = \{g_0, g_1, \ldots, g_{2^m-1}\}$, and we will encode each g_i as a binary string in $\{0,1\}^m$. Furthermore, define the concatenation of the Reed-Solomon encoding with the unary encoding by $C_{\mathrm{RS}\circ\mathrm{U}}(x) = C_{\mathrm{U}}(p_x(g_0))\cdots C_{\mathrm{U}}(p_x(g_{3d-1}))$. Codewords thus consist of $3d$ blocks of 2^m bits (corresponding to the $3d$ symbols of the Reed-Solomon encoding) with each block containing exactly one 1. Thus, $C_{\mathrm{RS}\circ\mathrm{U}}$ encodes strings of length n into strings of length $O(n^2)$. Furthermore, $C_{\mathrm{RS}\circ\mathrm{U}}$ can be encoded and decoded in polynomial time and so the languages L and $C_{\mathrm{RS}\circ\mathrm{U}}(L)$ are polynomial-time equivalent. Note that the decoding task at hand does not require us to perform error correction in the usual sense: we simply want to identify if an input is a codeword (since we reject all words that are not codewords) and we only care about decoding true codewords.

Lemma 5.2 *For any language L and any $k \geq 3$, $D_k^{sym}(C_{\mathrm{RS} \circ \mathrm{U}}(L)) \leq 6$.*

Proof: Without loss of generality $k = 3$ as all but the first two players can pretend they are the same party. Let $m = \lceil \log_2 3n \rceil$ and $d = n/m$. To check if an input is a codeword, the players can easily check that there are never two 1s in a single block of input bits. They cannot, however, verify at constant cost that each of the $3d$ blocks contains *at least* one 1 since this task is essentially the partition problem whose complexity we lower bound as superconstant through Lemma 5.11 in Section 5.3.2. We proceed differently: an input w of length $3d \cdot 2^m$ can only be a codeword if at least one player (say Player 1) has on its forehead at least d 1's and this player can be identified with three bits of communication. These d 1's determine d elements of $GF[2^m]$ hence players 2 and 3 can each privately reconstruct from them the unique degree $d - 1$ polynomial p that coincides with these elements. Players 2 and 3 now know that if the input is a codeword then it must be the one corresponding to p and player 2 can check that the bits on player 3's forehead are consistent with that hypothesis while player 3 can similarly cross-check the input bits on player 2's forehead. If this cross-checking procedure is successful, player 2 can determine the unique x such that $p_x = p$, verify $x \in L$ and send the result to all parties. Overall, only six bits of communication suffice to decide if the input is from $C_{\mathrm{RS} \circ \mathrm{U}}(L)$. ∎

As an immediate corollary to this lemma and the fact that the complexity of $C_{\mathrm{RS} \circ \mathrm{U}}(L)$ is polynomially related to the complexity of L we obtain:

Corollary 5.3 *The class of languages with bounded multi-party communication complexity contains languages with arbitrarily large uniform time and space complexity.*

In order to obtain also languages with essentially the largest possible non-uniform circuit complexity we need codes that map n bits into $O(n)$ bits. We can obtain such codes by concatenating codes provided by the following lemma with the unary code C_U.

Lemma 5.4 *For any integer $n \geq 1$, there exists a linear map $C_8 : \{0,1\}^n \rightarrow GF[8]^{39n}$ such that every $w \in C_8(\{0,1\}^n)$ is uniquely determined by any one-third of its coordinates.*

Proof:

To prove the existence of our code we only need to prove the following claim.

Claim 5.5 *For $c \geq 37$, with high probability a random matrix over $GF[8]$ of dimension $n \times cn$ has the property that each sub-matrix of dimension $n \times cn/3$ has rank n.*

For any $n' < n$, n' vectors over $GF[8]$ of length $cn/3$ span less than 8^n different vectors. Thus the probability that a new random vector of length $cn/3$ falls into the space spanned by these vectors is at most $8^{n-cn/3}$. Hence, the probability that a random matrix over $GF[8]$ of dimension n by $cn/3$ is of rank less than n is at most $n \cdot 8^{n-cn/3}$. (We pick the vectors step by step and at each step we fail to pick a linearly independent vector with probability at most $8^{n-cn/3}$.) Thus the expected number of singular n by $cn/3$ sub-matrices of a random matrix of dimension n by cn is at most $n \cdot 8^{n-cn/3} \cdot \binom{cn}{cn/3}$. Since $\binom{cn}{cn/3} \leq 2^{\mathrm{H}(1/3)cn}$, if $c \geq 37$ then $3 - c + \mathrm{H}(1/3)c < 0$ and the expected number of singular sub-matrices is $2^{-\epsilon n}$ for some $\epsilon > 0$. The claim follows.

∎

Consider the concatenation of the code C_8 and the unary code and denote it by $C_{8\circ\mathrm{U}}$. Recall the argument that showed that the three-party communication complexity of the language $C_{\mathrm{RS}\circ\mathrm{U}}(L)$ is constant, for every language L. Replacing Reed-Solomon codes in this argument by C_8 shows that $C_{8\circ\mathrm{U}}(L)$ has constant three-party communication complexity for any L. Further, notice that C_8 is over the fixed alphabet $GF[8]$. Thus, $C_{8\circ\mathrm{U}}(L)$ maps n bits to $O(n)$ bits. As a consequence, we obtain the following:

Corollary 5.6 *For any $k \geq 3$, the class of languages with bounded k-party communication complexity contains languages with $2^{\Omega(n)}$ circuit complexity.*

5.3 Two Special Classes of Languages

We consider two natural classes of functions for which the coding trick of the previous section fails. A letter $e \in \Sigma$ is said to be *neutral* with respect to a language L if for each word the addition or deletion of the letter e does not affect its membership in L i.e. for all $u, v \in \Sigma^*$ we have $uv \in L$ iff $uev \in L$. The neutral letter hypothesis was helpful in obtaining length lower bounds on bounded-width branching programs [BS95], was central to the Crane-Beach Conjecture [BIL$^+$05, LTT06], and the recent work of Roy and Straubing [RS07].

L is called a symmetric language if for each word w permuting its letters does not affect its membership in L i.e. the membership of w in L is completely determined by the count of the occurrences of each letter of the alphabet in w.

If a language is symmetric or has a neutral letter, then membership in L cannot depend, as in Lemma 5.1, on having specific value on a specific input position. Intuitively, the feature of the model [2] that each player knows the exact position in the input word of every letter that he sees, should not help deciding languages having a neutral letter and symmetric languages.

The rest of this section is devoted to proving the following two theorems that corroborate the above intuition:

Theorem 5.7 *If f is a function with a neutral letter such that $D_k^{sym}(f) = O(1)$ for some fixed k, then f is regular.*

Theorem 5.8 *If $f : \Sigma^n \to \{0, 1\}$ is symmetric and has bounded k-party symmetric communication complexity for some fixed k, then in fact f has bounded two-party symmetric communication complexity.*

Both proofs use notions from Ramsey theory that we quickly review.

5.3.1 A Primer on Ramsey Theory

"In any collection of six people, either three of them mutually know each other or three of them mutually do not know each other".

These are the opening lines of the excellent book by Graham, Rothschild and Spencer [GRS90] on Ramsey theory which is a classical branch of extremal combinatorics. These lines highlight the fact that there cannot be perfect chaos. Whenever a system is large enough, interesting structure emerges. Perhaps a little surprisingly, this conceptually simple principle has found powerful applications in diverse

[2] This feature is also present in the two-party model.

areas of mathematics like number theory, algebra and geometry and of course, computational complexity. The first application of Ramsey theory to communication complexity was made in the work of Chandra et. al. [CFL83] that introduced the very model of 'Number in the Forehead'.

Let C_t^n denote the n-dimensional cube over t elements, i.e.

$$C_t^n \equiv_{\text{def}} \big\{(x_1, \ldots, x_t) \,|\, x_i \in \{0, \ldots, t-1\}\big\}.$$

Such cubes are fundamental objects appearing in many different contexts. We however want to view cubes purely combinatorially. The t points $v^1, \ldots, v^t \in [t]^n$ are said to form a *combinatorial line* in C_t^n if the v^j's are distinct and for each $1 \leq i \leq n$ either all the v^j agree on co-ordinate i (i.e. $v_i^j = v_i^{j'}$ for all $1 \leq j \leq j' \leq t$) or we have $v_i^j = j$ for all $1 \leq j \leq t$. As an example, points $00, 01, 02$ form a line in C_3^2 and points $020, 121, 222$ form a line in C_3^3. Every function $\chi : C_t^n \rightarrow \{1, \ldots, c\}$ is called a c-colouring of C_t^n as each point of the cube receives one of c colours. A set of points P is rendered *monochromatic* by χ if every point in P is coloured the same by χ. The following result shows that any colouring of a sufficiently large dimensional cube has an interesting monochromatic set of points.

Theorem 5.9 (Hales-Jewett [GRS90]) *For any integers c, t there exists an integer $n = HJ(c, t)$ such that every c-colouring of $C_t^{n'}$ generates a monochromatic combinatorial line whenever $n' \geq n$.*

The Hales-Jewett Theorem is a central result of Ramsey theory from which several other results in the subject follow. It is not too difficult to derive from it the famous Van der Waerden's theorem[3] that says every finite colouring of the set of positive integers generates monochromatic arithmetic progressions of arbitrarily large length. Let $t, r > 0$ be any number such that we want to find a monochromatic arithmetic progression of length t when the positive integers are r-coloured. Map the first t^n integers bijectively into C_t^n where $n = HJ(r,t)$ by writing each integer in base t. Any r-colouring of the first t^n numbers thus induces a r-colouring of C_t^n. By the Hales-Jewett theorem there exists a monochromatic line. Van der Wareden's theorem follows by merely observing that any line in C_t^n, in our mapping, corresponds to an arithmetic progression of length t.

What we need is the following generalization of Van der Waerden's theorem to higher dimensions, whose short proof also follows from the Hales-Jewett theorem.

Theorem 5.10 (Generalized Van der Warden) *For any integers $c, k, m > 0$, there is an integer $R = GVW(c,k,m)$ such that for each c-coloring of $\{0, \ldots, R\}^k$, there exist $x_1^0, \ldots, x_k^0 < R$ and $1 \leq d < R$ such that all points of the set $\mathcal{P} = \{(x_1, \ldots, x_k) \ : \ x_i = x_i^0 + dy_i, 0 \leq y_i \leq m\}$ have the same color and $\mathcal{P} \subseteq \{0, \ldots, R\}^k$.*

Proof: Let $V \equiv [m]^k \equiv \{(y_1, \ldots, y_k) \,|\, 0 \leq y_i < m\}$. Order the elements of V in some arbitrary way so that $V = v_1, v_2, \ldots, v_t$ with $t = m^k$. Let $N = HJ(c,t)$ and let $R = m^N$. Consider the map $\psi : C_t^N \rightarrow [R]^k$ given by $\psi(x_1, \ldots, x_N) = \sum_{i=1}^N m^{N-i} x_i$ where x_i is viewed as a vector in V. Note that ψ is bijective. Thus a c-colouring

[3] Note that Van der Waerden's theorem is the starting point of such deep theorems as Szemeredi's theorem that has stimulated strong research recently.

of $[R]^k$ induces a c-colouring of C_t^N. Applying the Hales-Jewett Theorem 5.9, we see that a monochromatic line exists in the c-colouring of C_t^N. It is not difficult to verify that the image of this line under ψ is the monochromatic set P that we need to establish Theorem 5.10. ∎

5.3.2 Communication Complexity of Partition

We define the *k-wise partition problem*, denoted by $Part_k$. It takes as input a $k \times n$ Boolean matrix A and we think of the i^{th} row of A as representing a subset x_i of $[n] = \{1, \ldots n\}$. We define $Part_k(A) = 1$ iff each column of A contains exactly one 1 (i.e. the x_i form a partition of $[n]$). It is clear that for the k-party game the worst input partition for $Part_k$ is the one where player P_i holds the bits of row i on his forehead.

Below, we recall a super-constant lower bound, obtained by Pudlák and Tesson [Tes03], on the k-party communication complexity of $Part_k$ using the Hales-Jewett Theorem. This is interesting in its own right and useful for our analysis. We recall the argument below, that is reminiscent of the argument employed by Chandra et.al. to obtain super-constant lower bounds on the 'exactly-N' function (see Section 4.3 of Chapter 4).

Lemma 5.11 ([Tes03]) *For all k, $D_k(Part_k) = \omega(1)$.*

Proof: We identify a set of k-wise partitions of $[n]$ that form a star. Additionally, the communication history is the same on each of these partitions. Observation 4.14 from Chapter 4 then implies that the protocol generates the same communication history on the center of this star. The argument is finished by observing that the center point is not a partition of $[n]$.

For identifying the star, we use the Hales-Jewett Theorem. More precisely, consider the set of valid k-wise partitions of $[n]$. This set is in one-to-one correspondence with the cube C_k^n in the following way: we map a partition $\{S_1, \ldots, S_k\}$ to $(x_1, \ldots, x_n)$, where $x_i = j$ if $i \in S_j$ for each $1 \leq i \leq n$. This is a correct map because each $i \in [n]$ lies precisely in one S_j as the sets form a partition. Hence, a protocol of cost c for $Part_k^n$ induces a 2^c colouring of C_k^n, where each point of the cube is coloured by the communication history of the protocol on the corresponding partition. Set $n \geq HJ(2^c, k)$. Then, Theorem 5.9 guarantees the existence of a monochromatic line in C_k^n.

It is not hard to verify that a line in C_k^n corresponds to a set of k partitions of the following type: $\{S_1 \cup T, S_2, \ldots, S_k\}, \{S_1, S_2 \cup T, \ldots, S_k\}, \ldots, \{S_1, S_2, \ldots, S_k \cup T\}$ for some non-empty $T \subseteq [n]$. This forms a star. The fact that the line is monochromatic further means that the protocol generates the same communication history on each of these partitions. So it generates the same history on the center $\{S_1, \ldots, S_k\}$ which is not a star as T is non-empty. Hence, the protocol is incorrect. ∎

The proof of Lemma 5.11 only considers those instances of $Part_k$ in which any two subsets held by the k players are disjoint. Further, it is easily verified that the input instance (the center of the star) on which the players are forced to make an error, also has this disjointness property. These observations yield the following slightly stronger result : define the problem $RPart_k^n$ to be $Part_k$ with the restriction that the k sets given to players are pairwise disjoint and are subsets of $[n]$.

Corollary 5.12 *For each k, $RPart_k^n$ cannot be solved using c bits of communication whenever $n \geq HJ(2^c, k)$.*

Note that a $k \times n$ matrix A belongs to $Part_k$ iff none of its columns contains two 1 and the total number of 1 entries in A is n. If $k \geq 3$ then k players can check the first condition using k bits of communication since any pair of input bits is accessible to at least one player. They are then left with verifying that the sum of the input bits is n which can, surprisingly, be achieved with a communication cost much less than the trivial $O(\log n)$ [CFL83, GGK08].

5.3.3 Languages with a Neutral Letter

In this section, we show that languages with a neutral letter that have bounded k-party complexity for some fixed k are all regular. In order to prove this, we introduce a convenient notion of reduction among problems for the 'Number on the Forehead' model.

A k-rectangular reduction r from $L \subseteq \{0,1\}^{n \times k}$ to $K \subseteq \{0,1\}^{l(n) \times k}$ is a k-tuple of functions $(r_1, \ldots, r_k)$ with each $r_i : \{0,1\}^n \to \{0,1\}^{l(n)}$ such that $(x_1, \ldots, x_k) \in L$ iff $(r_1(x_1), \ldots, r_k(x_k)) \in K$. We call l the length of the reduction. The fact that in a k-player game, each r_i can be computed by every player individually except the ith, gives rise to the following useful observation:

Observation 5.13 *Let $L \subseteq \{0,1\}^{n \times k}$ and $K \subseteq \{0,1\}^{l(n) \times k}$ be languages such that there exists a rectangular reduction from L to K of length l. Then, $D_k(L)(n) \leq D_k(K)(l(n))$.*

Let $C \geq 0$ be an integer and let $\mathcal{G}$ be a family of functions over Σ^* with finite range R. We say that inputs with weight at most C *determine* the functions of $\mathcal{G}$ if every function $g : \Sigma^{\leq C} \to R$ has at most one extension to Σ^* in $\mathcal{G}$. Now, let $\mathcal{C}_{k,c}$ be the family of functions with a neutral letter and k-party communication complexity at most c. In order to show that every function f in $\mathcal{C}_{k,c}$ is regular, we first prove the following strong property of f:

Lemma 5.14 *Functions of $\mathcal{C}_{k,c}$ are determined by inputs of weight at most $C = HJ(k, 2^{2c})$, a constant.*

We obtain the above lemma as a consequence of the following one:

Lemma 5.15 *For any $C > 0$, if the functions of $\mathcal{C}_{k,c}$ are not determined by inputs of size C then there exists a $n > C$ such that $RPart_k^n$ can be solved by k parties communicating at most $2c$ bits.*

Observe that Lemma 5.15 and Corollary 5.12 together imply Lemma 5.14 immediately.

Proof:(Lemma 5.15) For any word $w \in \Sigma^*$, we shall denote by w_e the word obtained from w by deleting all occurrences of e in w. The ith letter of w will be denoted by w^i. Also, for k words $w_1, \ldots, w_k$, each of length ℓ, let $w = w_1 \Diamond \ldots \Diamond w_k$ denote the word obtained by interleaving the k words in the following way : $|w| = \ell k$ and for all $1 \leq i \leq \ell k$, $w^i = w_j^m$ if $i = (m-1)k + j$ with $0 < j < k+1$. Let us assume that f and g are in $\mathcal{C}_{k,c}$, such that they are not identical, but the minimal string $v \in \{\Sigma - e\}^*$ such that $f(v) \neq g(v)$ has length at least C. We show below a k party protocol that solves $RPart_k^{|v|}$ by communicating at most $2c$ bits.

Our protocol will work using a k-rectangular reduction r to language $H \subset \Sigma^{|v| \times k}$, where $(y_1, \ldots, y_k) \in H$ iff $v = (y_1 \Diamond \cdots \Diamond y_k)_e$. Consider an instance of $RPart_k^{|v|}$ in which player i's forehead holds a $|v|$ bit vector representing set I_i. Then, $I_i \cap I_j = \emptyset$ if $i \neq j$. We define r_i as follows : let $y_i = r_i(I_i)$. Then, $y_i^j = v^j$ if $j \in I_i$, otherwise $y_i^j = e$. Let $u = (y_1 \Diamond \cdots \Diamond y_k)_e$. The simple observation that is key to our argument, is that u is v if $\cup_{i=1}^{k} I_i = [|v|]$ and otherwise $|u| < |v|$. This shows that r is indeed a reduction from $RPart_k^{|v|}$ to H.

The observation above and the property of v (i.e. $f(u) = g(u)$, whenever $|u| < |v|$) imply the following : $y = y_1 \Diamond \ldots \Diamond y_k$ is in H iff $f(y) \neq g(y)$. The condition $f(y) \neq g(y)$ can be checked with $2c$ bits of communication by running the c-bit protocol on f and g separately. Thus, $2c$ bits of communication are enough to solve H and hence $RPart_k^{|v|}$. ∎

Remark 5.16 *It follows immediately that the number of languages in $\mathcal{C}_{k,c}$ over any fixed alphabet Σ is finite for a fixed k, c i.e. there are at most $2^{(|\Sigma|-1)^C}$ such languages where $C = HJ(k, 2^{2c})$.*

The first main theorem of the section is easily established below.

Proof: (*Theorem 5.7*) Let $f : \Sigma^* \to \{0,1\}$ be a function in $\mathcal{C}_{k,c}$: For a word $w \in \Sigma^*$, we define the function $f_w : \Sigma^* \to \{0,1\}$ by $f_w(z) = f(wz)$. It is easy to verify that for each w, f_w is also in $\mathcal{C}_{k,c}$. Define the equivalence relation $\sim_f$ on Σ^* by insisting $u \sim_f v$ iff $f(uz) = f(vz)$ for all $z \in \Sigma^*$ i.e. f_u and f_v are identical. Remark 5.16 ensures that $\sim_f$ has finite index. The classical Myhill-Nerode Theorem (see for example [HU79]) guarantees that if $\sim_f$ has finite index then f is regular and we are done. ∎

5.3.4 Symmetric Functions

For $w \in \Sigma^*$, we denote as $|w|_a$ the number of occurrences of a in w. The value of a symmetric function $f : \Sigma^* \to \{0,1\}$ on w thus is entirely determined by the values $|w|_a$ for each $a \in \Sigma$. We remind the reader of the intuition that $k \geq 3$ parties computing a symmetric function only get limited benefits from the features of the multiparty model since their protocol cannot significantly rely on the precise set of input positions accessible to each player or on the fact that any $(k-1)$-tuple of bits is seen by one party. This intuition is formalized by Theorem 5.8 and in this section we prove this theorem.

For simplicity, we first deal with functions with boolean inputs. To any symmetric function $f : \{0,1\}^n \to \{0,1\}$, we naturally associate the function $\overline{f} : \{0,\ldots,n\} \to \{0,1\}$ such that $f(x) = \overline{f}(|x|_1)$ for every $x \in \{0,1\}^n$. We say that f is $(\ell, r, p)-$*periodic* if $\overline{f}(a) = \overline{f}(a+p)$ for $\ell \leq a \leq n-r$.

We first observe that one can assume the protocol to be non-interactive in the following sense: a protocol is called *simultaneous* if each player sends a single message to an extra party, usually called the *referee*, who then computes the answer solely based on the messages he received. In particular, the message sent by a party does not depend on messages sent by other parties. It is easy to verify that a k-party protocol of communication cost c can be simulated by a k-party simultaneous protocol with cost at most $ck2^c$. This is done by making each player communicate all the eventualities (that he foresees) to the referee. Thus functions of bounded complexity in the simultaneous model are precisely those with bounded complexity in the standard model. This point of view turns out to be useful for the analysis.

Lemma 5.17 *For any constants k, c with $k \geq 1$ there exists an integer $N_{k+1} = N(k+1, c)$ such that every symmetric boolean function $f : \{0,1\}^n \to \{0,1\}$ that has a $k+1$-party simultaneous protocol of complexity c for the input partition in which players $X_1, \ldots, X_k$ each get N_{k+1} bits and player X_{k+1} gets the remaining $n - kN_{k+1}$ bits is (ℓ, r, p)-periodic for some $\ell, r \leq kN_{k+1}$ and some $p \leq N_{k+1}$.*

Theorem 5.8 then follows by observing that an (ℓ, r, p)-periodic function has 2-party simultaneous communication complexity roughly $2 \cdot \lceil \log(\ell + r + p) \rceil$. The proof of Lemma 5.17 proceeds by induction on k. Our induction step uses a non-trivial "player elimination" technique. More precisely, we use the generalization of Van-der Waerden's theorem as given by Theorem 5.10 to show that if f has a $(k+1)$-party protocol of bounded cost then there exists a large set of inputs $\mathcal{P}$ for the foreheads of the first k players on which player P_{k+1} always sends the same communication. This renders the $(k+1)$st player irrelevant if the input lies in $\mathcal{P}$. The special structure of $\mathcal{P}$ allows the use of the induction hypothesis.

We define $N(k, c)$ inductively. The base case of two players was first proved by Szegedy [Sze93]. We include the proof of this case below for the sake of self-containment.

Claim 5.18 $N(2, c) = 2^c$.

Proof: Consider the partition where the first player's forehead gets the first 2^c bits and the second player receives the remaining $n - 2^c$ bits. Consider the following $2^c + 1$ possible assignments: Player 1's forehead is assigned the string $1^i 0^{2^c - i}$ for $0 \leq i \leq 2^c$. As Player 2 sends out at most 2^c different messages, there are at least two such assignments to Player 1's forehead, for which Player 2 sends out the same message. Let these two assignments correspond to i being i_1 and i_2 respectively, with $i_1 < i_2$. We prove the claim by showing that $\overline{f}(j) = \overline{f}(j + i_2 - i_1)$, whenever $i_1 \leq j \leq n - 2^c + i_1$ i.e. f is (ℓ, r, p)-periodic with $\ell = i_1 \leq 2^c = N(2, c)$, $r = N(2, c) - i_1$ and $p = i_2 - i_1 \leq N(2, c)$.

Let j be in the required range. Fix the assignment $1^{j-i_1}0^{n-2^c-j+i_1}$ to the forehead of Player 2. The claim gets established by observing that the protocol outputs the same value for f when Player 1's forehead is assigned $1^{i_1}0^{2^c-i_1}$ or $1^{i_2}0^{2^c-i_2}$. ∎

Using the above as the base case, we prove our main lemma.

Proof: *(Lemma 5.17)* We show that $N(k+1,c) = GVW(2^c, k, N(k,c)! + (k-1)N(k,c))$ for $k \geq 2$, where GVW is the Generalized Van der Waerden number. The main idea is the following: given a constant cost $(k+1)$-party protocol for the symmetric function f, we use the Generalized Van-der-Waerden's Theorem to 'eliminate' the $(k+1)$st player by restricting f to a set of inputs on which that player's message is always the same. This enables us to construct a bounded cost k-party symmetric function f' closely related to f. Our inductive hypothesis applies to f' and we show that the periodicity of f' implies the periodicity of f.

Let Π be a simultaneous $(k+1)$-player protocol of cost c that computes f under a partition of the following form. Players $1, \ldots, k$ each have N_{k+1} bits assigned on his/her forehead, and Player $k+1$ gets the remaining $n - kN_{k+1}$ bits. Colour each point $(x_1, \ldots, x_k) \in \{0, \ldots, N_{k+1}\}^k$ by the message communicated by Player $k+1$ when $1^{x_i}0^{N_{k+1}-x_i}$ is on the forehead of Player i for $i \leq k$. By Generalized Van der Waerden's Theorem, there is a set $\mathcal{P}$ of points in $\{0, \ldots, N_{k+1}\}^k$, such that Player $k+1$ sends the same message for every assignment to the first $k-1$ foreheads that corresponds to a point in $\mathcal{P} = \{(x_1, \ldots, x_k) : x_i = x_i^0 + dy_i, 0 \leq y_i \leq N_k! + (k-1)N_k\}$, for some $d < N_{k+1}$.

Let $\ell = d(k-1)N_k + \sum_{i=1}^{k} x_i^0$, $r = kN_{k+1} - \ell$, and $p = d \cdot N_k!$. Clearly, they satisfy required bounds required by Lemma 5.17. We prove below the claim that f is (ℓ, r, p)-periodic.

For any positive integer α, define the spectrum function $\overline{f}'_\alpha : \{0, \ldots, N_k! + 2(k-1)N_k\} \to \{0,1\}$ by $\overline{f}'_\alpha(u) = \overline{f}(\alpha + \sum_i x_i^0 + du)$. This spectrum corresponds to a symmetric boolean function f'_α on $N_k! + 2(k-1)N_k$ bits. We verify that the function f'_α has a c bit k-party communication protocol for the partition where the first $k-1$ players get N_k bits on their foreheads and the remaining $N_k! + (k-1)N_k$ bits are on Player k's forehead. The reason is that when players 1 through k have on their foreheads strings of weights y_1 through y_k, they can simulate Π by doing the following: For $1 \leq i \leq k$, Player i's forehead is replaced by any string of weight $x_i^0 + dy_i$, and each of these k players assume that Player $k+1$ has a string of weight α on its forehead. They then communicate according to Π and the referee, knowing the constant message sent out by Player $k+1$, computes the correct answer.

The induction hypothesis implies the following Observation:

Observation 5.19 *For each $\alpha \leq n - kN_{k+1}$, there exists $\ell', r' \leq (k-1)N_k$ and $p' \leq N_k$ such that f'_α is (ℓ', r', p')-periodic i.e. $\overline{f}'_\alpha(u) = \overline{f}'_\alpha(u + p')$ for $\ell' \leq u \leq N_k! + 2(k-1)N_k - r'$.*

Let $x \geq \ell$. Note that $\overline{f}(x) = \overline{f}'_\alpha(u)$ with $\alpha = x - \ell$ and $u = (k-1)N_k$. Applying Observation 5.19, $\overline{f}'_\alpha(u) = \overline{f}'_\alpha(u + N_k!)$ as p' divides $N_k!$ and $u + N_k! \leq N_k! + 2(k-1)N_k - r'$. Thus $\overline{f}(x) = \overline{f}(x + d \cdot N_k!)$, when $\ell \leq x \leq n - kN_{k+1} + \ell$ establishing the (ℓ, r, p)-periodicity of f. ∎

We extend our result to any general finite alphabet $\Sigma = \{a_1, \ldots, a_t\}$, where $t \geq 2$. Consider three t-dimensional vectors $\overline{\ell} = (\ell_1, \ldots, \ell_t)$, $\overline{r} = (r_1, \ldots, r_t)$ and $\overline{p} = (p_1, \ldots, p_t)$ where ℓ_i, r_i and p_i are positive integers. Define an equivalence relation $\sim_{\overline{\ell},\overline{r},\overline{p}}$ over Σ^n by setting $x \sim_{\overline{\ell},\overline{r},\overline{p}} y$ precisely if for each i, either $|x|_{a_i} = |y|_{a_i}$ OR $\ell_i \leq |x|_{a_i}, |y|_{a_i} \leq n - r_i$ and $|x|_{a_i} \equiv |y|_{a_i} \bmod(p_i)$. We call a function $f : \Sigma^n \rightarrow \{0, 1\}$ to be $(\overline{\ell}, \overline{r}, \overline{p})$-periodic if $f(x) = f(y)$ whenever $x \sim_{\overline{\ell},\overline{r},\overline{p}} y$. We show the following:

Lemma 5.20 *If a symmetric function $f : \Sigma^n \rightarrow \{0, 1\}$ has bounded k-party symmetric communication complexity then f is $(\overline{\ell}, \overline{r}, \overline{p})$-periodic with $\ell_i = r_i = (k-1)N_k$ and $p_i = N_k!$ for each $i \leq t$, where t is the size of the alphabet Σ.*

Proof: Let $\Sigma = \{a_1, \ldots, a_t\}$. For any $\Sigma_0 \subseteq \Sigma$ and any word w in $(\Sigma - \Sigma_0)^*$, we define a symmetric function $f_w^{\Sigma_0} : \Sigma_0^{n-|w|}$ by letting $f_w^{\Sigma_0}(x) = f(wx)$. We now argue by induction of the cardinality t of Σ. It gets easily verified that our base case of $t = 2$ is guaranteed by Lemma 5.17. Let $t \geq 3$. Consider two string x, y with $x \sim_{\overline{\ell},\overline{r},\overline{p}} y$. If for each i $|x|_{a_i} = |y|_{a_i}$, then trivially $f(x) = f(y)$. Otherwise, pick $i \neq j$ such that $|x|_{a_i} < |y|_{a_i}$ and $|x|_{a_j} > |y|_{a_j}$. Assume w.l.o.g that $||x|_{a_i} - |y|_{a_i}| \leq ||x|_{a_j} - |y|_{a_j}|$. Let $s_i = |x|_{a_i}$ and $s_j = |x|_{a_j}$. Consider string $u = (a_i)^{s_i}(a_j)^{s_j}$ and a string $\alpha \in (\Sigma \backslash \{a_i, a_j\})^{n-s_i-s_j}$ that is obtained from x by deleting all the occurrences of letters a_i and a_j. Clearly, $f(x) = f_\alpha(u)$. The function $f_\alpha^{\{a_i,a_j\}}$ has constant k-party symmetric complexity over the binary alphabet $\{a_i, a_j\}$. Applying the base case of our induction to this function, f_α is periodic and there exists a string $v \sim_{\overline{\ell'},\overline{r'},\overline{p'}} u$ with $|v|_{a_i} = |y|_{a_i}$ and $f_\alpha(v) = f_\alpha(u) = f(x)$. Notice that $\alpha v \sim_{\overline{\ell},\overline{r},\overline{p}} y$ and $|\alpha v|_{a_i} = |y|_{a_i} =$ (say) r. Finally, let $\beta = (a_i)^r$. Then $f(x) = f_\beta(u')$ where $u' \in (\Sigma \setminus \{a_i\})^{n-r}$ is obtained by deleting all occurrences of letter a_i from αv. Applying the induction hypothesis on $\Sigma \setminus \{a_i\}$, we conclude that there exists a v' in $(\Sigma - \{a_i\})^{n-r}$ such that $|v'|_{a_m} = |y|_{a_m}$ for each $m \neq i$ and $f(x) = f_\beta(u') = f_\beta(v') = f(y)$.

$\blacksquare$

Theorem 5.8 now follows from Lemma 5.20 as two players can compute the count of each letter in Σ up to a constant threshold and a constant modulus in constant communication.

5.4 Consequences and Conclusion

There are interesting consequences of these results for low degree polynomials and constant depth circuits. For instance, it is already known by results of [PT88, TB98] that constant degree multivariate polynomials over a fixed modulus[4] m cannot compute MAJORITY by a generalized representation. Our results on the multiparty communication complexity of symmetric functions yields a new proof of this result as follows: consider any polynomial P over $\mathbb{Z}_m$ of degree d with accepting set $A \subseteq \mathbb{Z}_m$. It gets readily verified that the function f represented by P has constant $(d+1)$-party symmetric communication complexity as in any partition, each monomial of P can be evaluated by some player without communicating with others. Thus, the monomials of P are partitioned into at most $d+1$ classes with each player assigned one class. Given an assignment, each player computes a $\log m$ bit answer that is the sum of all the monomials of P in the player's class. Knowing all the answers (and the accepting set A) the referee can compute the value of f. Applying Lemma 5.20,

[4] The modulus m is not necessarily a prime power.

we know that f must be (ℓ, r, p)-periodic, for constants ℓ, r, p, and hence f cannot be MAJORITY.

We next describe an application of our result to constant-depth circuits. Although in this thesis we have focussed on models of computation that are non-uniform, it is known that uniformity conditions can ease the task of proving lower bounds. For instance, Allender and Gore [AG94] have shown that uniform ACC^0 circuits cannot compute the *Permanent* function efficiently[5] . Our results on the communication complexity of languages with neutral letters suggests that adding a neutral letter to functions might be a simple but effective way of containing the power of non-uniformity not only in the multiparty model, but also in the model of constant-depth circuits.

Corollary 5.21 *Every language with a neutral letter that can be computed by $CC^0[p^r]$ circuits of arbitrary size is regular, if p is a fixed prime and $r \geq 1$ is a fixed integer.*

Proof: Recall that the output of each gate of such a circuit can be exactly represented as a polynomial of degree at most $p^r - 1$ over $\mathbb{Z}_p$in the input variables of the gate. Thus, the output of the entire circuit is exactly represented by a polynomial of degree at most $k = (p^r - 1)^d$ in the input variables of the circuit over $\mathbb{Z}_p$. We conclude that the function computed has constant $k+1$-party symmetric communication complexity. Applying Theorem 5.7, we are done. ∎

[5] We cannot separate non-uniform ACC^0 from NEXP.

CHAPTER 6
Communication Complexity of Functions in AC^0

In the last chapter, our focus was on understanding the structure of the class of problems that admit constant communication protocols under every possible partition of the input letters. This investigation brought out further differences between the characteristics of the multiparty model and the two player model. In particular, we established that three players can compute functions of arbitrarily large circuit complexity in constant cost under the worst possible partition of input letters. In the first part of this chapter, we explore the multiparty model from the other direction. We want to answer the following question: "What is the lowest circuit complexity class which contains a function of very large[1] k-party communication complexity?". We explore this question by restricting ourselves to the binary alphabet. Further, for each function that we consider, the input bits are partitioned among players in some fixed way, unlike in the last chapter.

It is trivial to observe that if f lies in NC^0, then it has constant cost deterministic protocols for two players. This is because f depends on a constant number of letters and Alice can communicate to Bob the relevant letters from her partition in constant

[1] By 'very large', we typically mean $n^{\Omega(1)}$ complexity. We say that a function has 'large' complexity if it is superpolylogarithmic, i.e. it does not have *efficient* protocols.

cost. While it is well known that for $k = 2$ there are functions in shallow AC^0 (like Equality and Disjointness) that have linear deterministic complexity, no function in AC^0 was known, until recently, that had superlogarithmic three party deterministic communication complexity. The best that one could say was that ACC^0 contains functions of very large k-party complexity for every $k < \delta \log n$, where the input size is kn and δ is a constant. This followed from the work of Babai, Nisan and Szegedy [BNS92] who showed that the natural k-wise generalization of the Inner Product function, called *Generalized Inner Product*, has large k-party randomized complexity, for $k < \delta \log n$. This work introduced the powerful discrepancy method that has been the backbone of almost[2] all subsequent strong lower bound results (for example [Gro92, Raz00, FG05]) in the multiparty model. Unfortunately, it was not known if this method could be applied to a function in AC^0 even for two players.

Recently, Sherstov [She07] provided the first successful application of the discrepancy method for a function in AC^0 for two players. We extend this technique to multiple players yielding the following (first published in [Cha07b]):

Theorem 6.1 *For each k, there exists a function F_k^{MP} computable by depth-three AC^0 circuits of linear size that has the following randomized k-party communication complexity:*

$$R_k^{\epsilon}\big(F_k^{MP}\big) = \Omega\left(\frac{n^{\frac{1}{2k+1}}}{\left(2^{2^k/(2k+1)}2e(k-1)\right)^{k-1}} + \log \epsilon\right).$$

[2] In the few cases, like in [Cha07a, BPSW06, VW07a], where non-discrepancy based techniques have been applied, they are only known to apply to restricted communication protocols.

Consequently, for $k = o(\log \log n)$ there exist functions in linear depth-three AC^0 that have no efficient (i.e polylogarithmic cost) randomized k-party protocols computing them with an advantage ϵ that is better than any inverse-quasipolynomial function. This is in contrast to the easily verifiable fact that every function having polynomial size depth-two circuits has an efficient two-player randomized protocol computing it with advantage that is at least an inverse polynomial function of the length of its input. What happens if we demand more from our randomized protocols, i.e. we require them to have a fixed advantage over random guessing? Could we still compute every function in depth-two AC^0 efficiently?

It is not difficult to see that every function that is computable by a depth-two AC^0 circuit of size s has either $O(\log s)$ non-deterministic or co-non-deterministic communication complexity. As we point out later, the Discrepancy Method yields poor lower bounds on functions that have efficient non-deterministic or co-non-deterministic protocols. This makes the method unsuitable to work well for functions computable efficiently by depth-two circuits. A specific instance is the Disjointness function for which no superlogarithmic lower bounds were known for three or more players until recently. Fueled by two very recent and independent breakthroughs, made by Sherstov [She08b] and Shi and Zhu [SZ07] respectively in the context of quantum communication lower bounds for two players, we develop the Generalized Discrepancy Method for multiparty classical communication. This leads us to obtain the following strong bound on the communication complexity of Disjointness:

Theorem 6.2 *For any constant $\epsilon > 0$,*

$$R_k^\epsilon(DISJ_k) = \Omega\left(\frac{n^{\frac{1}{k+1}}}{2^{2^k}(k-1)2^{k-1}}\right).$$

A similar result was also obtained by Lee and Shraibman [LS08] independently.

6.0.1 Our Approach and Organization

Recall the k-wise generalization of the Inner Product function, called GIP_k from Section 4.2 in Chapter 4. There, we viewed functions like GIP_k and k-wise Disjointness to be generated by an underlying (base) symmetric predicate. We view things slightly more generally here by generating a function, to be computed by k-players, from a base function that is not necessarily symmetric.

Let $y^1, \dots, y^{k-1}$ be $(k-1)$ binary strings, each of length n. Define the $(k-1) \times n$ boolean matrix A obtained by placing y^i in the ith row of A. For $x \in \{0,1\}^n$, let $x \Leftarrow y^1, \dots, y^{k-1}$ be the n-bit string $x_{i_1}x_{i_2}\dots x_{i_t}0^{n-t}$, where $i_1, \dots, i_t$ are the indices of the all-one columns of A. Further, let $g : \{0,1\}^n \to \{-1,1\}$ be any function. We define $G_k^g : (\{0,1\}^n)^k \to \{-1,1\}$ by $G_k^g(x, y^1, \dots, y^{k-1}) := g(x \Leftarrow y^1, \dots, y^{k-1})$. We call g the base function of G_k^g. Observe that G_k^{PARITY} is the Generalized Inner Product function and G_k^{NOR} is the Disjointness function. While both the above examples use a symmetric base function, we use crucially a non-symmetric one to prove Theorem 6.1 in Section 6.4.

It is reasonable to expect that the communication complexity of a function is related to some intrinsic property of its base function. The result of Babai, Nisan and Szegedy can be interpreted as follows: if the base function is PARITY, then the generated function has low discrepancy under the uniform distribution. In this light, a natural question that emerges is “what happens if our base function is close to PARITY in an appropriately defined sense?”. Fourier analysis over $\mathbb{Z}_2^n$ provides a spontaneous measure of closeness to PARITY. Recall that this analysis decomposes every function as a linear combination of characters. It is easy to verify that each character of $\mathbb{Z}_2^n$ corresponds to the PARITY function defined over a subset of the set of n variables. The size of the subset is called the order of the parity. The function PARITY is orthogonal to every parity whose order is less than n. In this light, we say that a function is close to PARITY, if it can be expressed as a sum of high order parities or equivalently, is orthogonal to low order parities.

Our main technical ingredient, called the Orthogonality-Discrepancy Lemma, extending the Degree/Discrepancy Theorem of Sherstov [She07], generalizes the result of [BNS92]. Babai et.al. prove that G_k^{PARITY} has small discrepancy under the uniform distribution. For technical reasons, we look at a function F_k^g, generated by g employing another masking scheme, that is closely related to G_k^g. Roughly speaking, the Orthogonality-Discrepancy Lemma states that if g is orthogonal to low-order parities , then function F_k^g has low discrepancy under an appropriate probability distribution. The discrepancy method implies that F_k^g has large randomized communication complexity. As the communication tensor of F_k^g is a sub-tensor of the one for G_k^g, it follows that G_k^g has large communication complexity as well.

We prove Theorem 6.1 by finding a base function in AC^0 that has such nice orthogonality property. The key to finding it is to use the well-known notion of *voting representation* of boolean functions as introduced by Aspnes et.al. [ABFR94] (see Section 6.1.1). The use of a well-known duality principle, described in Section 6.3, allows passage from functions of high voting degree to functions with the above orthogonality property. This passage was invented in the context of two-player communication in the elegant work of Sherstov [She07]. Like Sherstov, we use the Minsky-Papert function, introduced in [MP88] and reviewed in Section 6.4 of this chapter, as our base function of high voting degree that can be computed by simple depth-two AC^0 circuits.

The base function generating k-wise Disjointness is the NOR function. Its voting degree is merely one and hence the Orthogonality-Discrepancy Lemma cannot be directly used for functions generated by NOR. However, the breakthrough work of Razborov [Raz03] and the earlier work of Buhrman et.al.[BCW98] established a tight relationship between the two-party quantum communication complexity of G_2^g and a well studied property of the symmetric function g. This property is the *approximate degree* of boolean functions, whose study was begun systematically in the work of Nisan and Szegedy [NS94]. In particular, they show that NOR has high approximation degree. While Razborov's lower bound, employing the "multidimensional discrepancy method", only worked for symmetric functions, the notion of approximate degree extends to all boolean functions.

Recently, Sherstov [She08b] and independently Shi and Zhu [SZ07] showed the following: a function g of high approximation degree (say d) correlates well with a function f under a distribution μ, where f has zero correlation with low-order parities (order less than d) under μ. Thus, the Orthogonality-Discrepancy Lemma, applied to f, shows that the function generated by f has high communication complexity. In order to reason about the communication complexity of G_k^g, an additional ingredient comes into play. This is an ingenious modification of the Discrepancy Method that originated in the work of Klauck [Kla01] and got further generalized by Razborov [Raz03]. This method, that we call the Generalized Discrepancy Method, is used to conclude that G_k^g has large (bounded error) randomized communication complexity based on the fact that g and f correlate well. We use this idea to prove Theorem 6.2 in Section 6.5.1. More generally, this leads us to obtaining lower bounds on the k-party communication complexity of every function of the form G_k^g, where g is a non-constant symmetric function (Corollary 6.22 in Section 6.5.2). Finally, we extend to the multiparty setting the work of [SZ07] in Section 6.6 for block-composed functions. Both these extensions yield exponential improvements for lower bounds on the k-party complexity of Disjointness. They also provide bounds on other interesting classes of functions.

6.1 Preliminaries

6.1.1 Voting and Approximation Degree

Recall that we reviewed Fourier analysis over abelian groups in Section 3.2.1 of Chapter 3. There, we specifically looked at the vector space of complex-valued functions over the group $\mathbb{Z}_m^n$. Here, we restrict ourselves to the space of real valued functions over the boolean cube $\mathbb{Z}_2^n$. The set of characters of the cube is given by $\widehat{\mathbb{Z}_2^n} = \{(-1)^{\sum_{i \in S} x_i} \mid S \subseteq [n]\}$. In this chapter, we map the cube $\{0,1\}^n$ bijectively to $\{1,-1\}^n$ by mapping the ith co-ordinate of a point as follows: $x_i \rightarrow (-1)^{x_i}$. Under this transformation, the set of characters becomes the familiar set of multilinear monomials[3] $\mathcal{M} = \{\chi_S = \prod_{i \in S} x_i \,|\, S \subseteq [n]\}$. Let us specialize the standard inner product, defined in Section 3.2.1, to the space of real-valued functions on the cube, i.e. for two functions $f, g : \{1, -1\}^n \to \mathbb{R}$,

$$\langle f, g \rangle = \mathbb{E}_x f(x) g(x).$$

Then, a basic fact of Fourier analysis is $\mathcal{M}$ forms an orthonormal basis w.r.t to the standard inner product. Thus, every boolean function[4] $f : \{1, -1\}^n \to \{1, -1\}$ is uniquely represented by a real linear combination of monomials from $\mathcal{M}$, i.e. a polynomial with real coefficients. The *exact degree* of f is the degree of this polynomial. However, in this section, we define two different representations.

Define $\text{sign}(z)$ to be -1 if $z < 0$ and 1 if $z > 0$, for every non-zero $z \in \mathbb{R}$. A polynomial $P = \sum_{S \subseteq [n]} a_S \chi_S$, with $a_S \in \mathbb{R}$, is a *voting representation* of a boolean function f if $f(x) = \text{sign}\big(P(x)\big)$ for each $x \in \{1, -1\}^n$. Note that this requires P not to evaluate to zero at any point of the cube. For example, polynomials $P_1(x) = x_1 + \cdots + x_n - 0.5$ and $P_2(x) = \prod_{i=1}^n x_i$ are voting representations of MAJORITY and PARITY respectively. It is not hard to verify that every boolean function f has a voting representation. In particular, the polynomial that exactly represents f is also a voting representation of f. However, it is not necessarily the most economical one

[3] Recall that a very similar basis set was used by Razborov and Smolensky to analyze the vector space of functions from the cube to a *finite field* (see Section 2.1.3 in Chapter 2).

[4] Note that we have changed the range of f from $\{0, 1\}$ to $\{1, -1\}$. Recall that we have encountered this change many times before.

in terms of degree. For instance, the exact degree of MAJORITY is $\Omega(n)$ whereas our representation uses a linear polynomial.

The degree of a voting representation is simply the degree of the polynomial P involved. Thus, in our example before, the representation of PARITY uses degree n. The *voting degree* of a function f, denoted by $\deg(f)$, is the minimum degree over all possible voting representations of f.

Fact 6.3 (from [ABFR94]) *The voting degree of PARITY is n.*

Proof: Let P be any polynomial that is a voting representation of PARITY. Then, by definition $P(x)\mathrm{PARITY}(x) > 0$, for each x. In other words, $\mathbb{E}_x P(x)\left(\prod_{i=1}^n x_i\right) > 0$. But if the degree of P is less than n, then by the orthonormality of monomials, $\mathbb{E}_x P(x)\left(\prod_{i=1}^n x_i\right) = 0$ and we get a contradiction. ■

In fact, using Fourier analysis over the cube $\mathbb{Z}_m^n$, one can, more generally, show that MOD_m has voting degree $\Omega(n)$, for any fixed integer $m \geq 2$ (see Barrington and Straubing [BS94]).

Is there any function in AC^0 that has high voting degree? It is easily verified that AND and OR have voting degree 1. On the other hand, a simple function in depth-two has high voting degree. Minsky and Papert [MP88] considered such a function that we call the Minsky-Papert function and denote by MP. For any t, we define MP over $m = 4t^3$ variables as follows: $\mathrm{MP}(x) = \vee_{i=1}^{t} \wedge_{j=1}^{4t^2} x_{i,j}$. There is a simple voting representation of MP having degree t. This is because of the following observation: rewrite MP as an AND of OR's by distributing the outer OR over the inner ANDs using basic boolean algebra. Each OR is over t variables and can be represented *exactly* by a degree t polynomial that has range $\{1, -1\}$. We are left to represent the AND of such t^{4t^2} polynomials, each of degree t. Treating each polynomial as a boolean variable and using the degree-one voting representation of AND, we get our desired representation of degree t of MP. Minsky and Papert showed that the degree of this representation is optimal.

Theorem 6.4 (from [MP88]) *The Minsky-Papert function defined on $m = 4t^3$ variables has voting degree t.*

Proof: We only need to show the lower bound of t for the voting degree. An important technique, called symmetrization, was introduced and used by [MP88] in the argument. It goes this way: let the clauses of MP be numbered $1, \ldots, t$, each having its own set of $4t^2$ variables disjoint from the others. Let $\mathcal{S}_i$ represent the set of permutations of the variables of the ith clause. Consider a set of t permutations $\sigma_1, \ldots, \sigma_t$, with $\sigma_i \in \mathcal{S}_i$. For any polynomial P, let $P_{\sigma_1,\ldots,\sigma_t}$ be the polynomial obtained from P by letting σ_i permute its variables from the ith clause. Then, observe that if P is a polynomial of minimal degree d that is a voting representation of MP, then so is $P_{\sigma_1,\ldots,\sigma_t}$. Hence,

$$P' \equiv_{\text{def}} \sum_{\sigma_i \in \mathcal{S}_i} P_{\sigma_1,\ldots,\sigma_t}$$

is a voting representation of degree t of MP. By construction, P' is symmetric w.r.t. variables in the same clause of MP. This passage from an arbitrary polynomial to a symmetric (w.r.t. clauses) polynomial is called symmetrization.

Let u_i represent the number of variables in Clause i of MP, set to 1. There exists a polynomial Q of degree d (same as that of P') on t variables $u_1, \ldots, u_t$, with $u_i \in \{0, \ldots, 4t^2\}$, such that Q outputs a negative number if at least one u_i is set to $4t^2$ and otherwise is positive. Obtain a univariate polynomial $R(v)$ from Q, by replacing the u_i in Q by $(4t^2 - (2i - v)^2)$. Clearly, the degree of R is at most twice the degree of Q, i.e. at most $2d$. On the other hand, consider the behavior of R on the set $\{0, \ldots, 2t\}$. It is easily verified that for odd values in this set R is positive and for even values it is negative. Thus, R has at least $2t$ zeroes and hence must have degree at least $2t$. Hence, $d \geq t$. ∎

We point out that the work of [ABFR94, OS03] are good sources for familiarizing oneself with further interesting properties of voting representations.

A voting polynomial just maintains the sign of the function. In principle, it could be very far from the value of the function at a given point of the cube. One could naturally tighten up this notion by demanding that a polynomial evaluate close to the value of the function represented, at each point of the cube. A polynomial P that is always within δ of the function f is a δ-approximation of f, i.e. $|f(x) - P(x)| \leq \delta$ for each $x \in \{1, -1\}^n$ and $\delta \geq 0$. The δ-approximation degree of f, denoted by $\deg_\delta(f)$, is the minimal degree such that there exists a polynomial of that degree which is a δ-approximation of f. Note that for any $\delta < 1$, a δ-approximation of a boolean function is a special voting representation of the function.

It follows that $\deg(f) \leq \deg_\delta(f)$ for any $\delta < 1$. The gap, between the two degrees, can be quite large. Nisan and Szegedy[5] [NS94] show that every boolean function that depends on each variable has δ-approximation degree $\Omega(\log n)$. Further,

[5] Nisan and Szegedy also related the approximation degree of a boolean function with its complexity in the model of decision trees.

they show that the AND and OR functions, each having voting degree 1, have (1/3)-approximation degree $\Theta(\sqrt{n})$. The work of [NS94] was followed by the work of Paturi [Pat92] who characterized the approximation degree of every symmetric function.

Paturi's characterization is quite helpful for our investigation and let us state his result. For any predicate $D : \{0, 1, \ldots, n\} \to \{1, -1\}$, define

$$\ell_0(D) \in \{0, 1, \ldots, \lfloor n/2 \rfloor\}$$

$$\ell_1(D) \in \{0, 1, \ldots, \lceil n/2 \rceil\}$$

such that D is constant over the interval $[\ell_0(D), n - \ell_1(D)]$ and $\ell_0(D)$ and $\ell_1(D)$ are the smallest possible values for which this happens. A symmetric function f induces a predicate D_f in the following natural way: $f(x) = D_f(x_1 + \cdots + x_n)$. For example, for the OR function $\ell_0(D_{\mathrm{OR}}) = 1$ and $\ell_1(D_{\mathrm{OR}}) = 0$.

Paturi's theorem provides bounds on the approximate degree of symmetric functions in terms of the properties of its underlying predicate.

Theorem 6.5 ([Pat92]) *Let $f : \{0,1\}^n \to \{1, -1\}$ be any symmetric function inducing the predicate $D_f : \{0, \ldots, n\} \to \{1, -1\}$. Then,*

$$deg_{1/3}(f) = \Theta\big(\sqrt{n(\ell_0(D_f) + \ell_1(D_f))}\big). \tag{6.1}$$

In particular, the (1/3)-approximate degree of NOR is $\Theta(\sqrt{n})$.

6.1.2 Discrepancy under Product Distributions

We recall a trick of repeatedly applying Cauchy-Schwartz to get an expression that upper bounds the cylindrical discrepancy of a function under product distributions. This trick to simplify the calculation of discrepancy appeared originally in the work of Babai et.al.[BNS92]. The explicit and convenient form in which we use it here is attributable to Raz [Raz00]. Our presentation below seems to be slightly simpler and more direct than Raz's.

Let $\mu^x, \mu^1, \cdots, \mu^{k-1}$ be probability distributions over finite sets $X, Y^1, \ldots, Y^{k-1}$ respectively. Let $\mu = \mu^x \times \mu^1 \times \cdots \times \mu^{k-1}$ be the product distribution generated and $f : \{X \times Y^1 \times \cdots \times Y^{k-1}\} \to \{-1, 1\}$ be any boolean function.

Lemma 6.6 (Raz [Raz00]) *For $1 \leq i \leq k-1$ and $j \in \{0,1\}$ let y_j^i be a random variable distributed according to μ^i and let x be distributed according to μ^x. Then,*

$$\left(disc_{k,\mu}(f)\right)^{2^{k-1}} \leq \mathbb{E}_{y_0^1, y_1^1, \ldots, y_0^{k-1}, y_1^{k-1}} \left| \mathbb{E}_x \prod_{u \in \{0,1\}^{k-1}} f\left(x, y_{u_1}^1, \ldots, y_{u_{k-1}}^{k-1}\right) \right|. \tag{6.2}$$

Proof: We prove (6.2) by induction of k. Thus, our Induction Hypothesis is that (6.2) is true for every function when $k = k-1$. Recall that for an arbitrary cylinder intersection ϕ,

$$\text{disc}_{k,\mu}^{\phi}(f) = \left| \mathbb{E}_{(x, y^1, \ldots, y^{k-1}) \sim \mu} f\left(x, y^1, \ldots, y^{k-1}\right) \phi\left(x, y^1, \ldots, y^{k-1}\right) \right|.$$

Let us factor the characteristic function ϕ in terms of the characteristic functions of the cylinders intersecting.

$$\phi\left(x, y^1, \ldots, y^k\right) = \phi^x\left(y^1, \ldots, y^k\right) \prod_{i=1}^{k-1} \phi^i\left(x, y^1, \ldots, y^{i-1}, y^{i+1}, \ldots, y^{k-1}\right)$$

where ϕ^x is the cylinder in the direction of X and ϕ^i is in the direction of Y^i. Then, using the triangle inequality, one gets

$$\begin{aligned}\mathrm{disc}^{\phi}_{k,\mu}(f) \quad &\leq \mathbb{E}_{x,y^1,\ldots,y^{k-2}}\Big|\phi^{k-1}\big(x,y^1,\ldots,y^{k-2}\big)\times \\ &\qquad \mathbb{E}_{y^{k-1}} f\big(x,y^1,\ldots,y^{k-1}\big)\phi^x\big(y^1,\ldots,y^{k-1}\big)\prod_{i=1}^{k-2}\phi^i\big(x,y^1,\ldots,y^{i-1},y^{i+1},\ldots,y^{k-1}\big)\Big|.\end{aligned}$$

Noting that characteristic functions are 0-1 valued, we further simplify:

$$\begin{aligned}&\mathrm{disc}^{\phi}_{k,\mu}(f) \leq \\ &\mathbb{E}_{x,y^1,\ldots,y^{k-2}}\Big|\mathbb{E}_{y^{k-1}} f\big(x,y^1,\ldots,y^{k-1}\big)\phi^x\big(y^1,\ldots,y^{k-1}\big)\prod_{i=1}^{k-2}\phi^i\big(x,y^1,\ldots,y^{i-1},y^{i+1},\ldots,y^{k-1}\big)\Big|.\end{aligned}$$

Squaring both sides and using the consequence $(\mathbb{E}z)^2 \leq \mathbb{E}z^2$ of Cauchy-Schwartz, one gets

$$\big(\mathrm{disc}^{\phi}_{k,\mu}(f)\big)^2 \leq \mathbb{E}_{y_0^{k-1},y_1^{k-1}}\; G\big(y_0^{k-1},y_1^{k-1}\big) \tag{6.3}$$

where

$$\begin{aligned}&G\big(y_0^{k-1},y_1^{k-1}\big) = \\ &\mathbb{E}_{x,y^1,\ldots,y^{k-1}}\prod_{u\in\{0,1\}} f\big(x,y^1,\ldots,y^{k-2},y_u^{k-1}\big)\phi^x\big(y^1,\ldots,y_u^{k-1}\big)\prod_{i=1}^{k-2}\phi^i\big(x,y^1,\ldots,y^{k-2},y_u^{k-2}\big).\end{aligned}$$

In order to apply our Inductive Hypothesis, we make the following definitions for every fixed y_0^{k-1} and y_1^{k-1}:

$$g\big(x,y^1,\ldots,y^{k-2}\big) =^{\mathrm{def}} f\big(x,y^1,\ldots,y^{k-2},y_0^{k-1}\big)f\big(x,y^1,\ldots,y^{k-2},y_1^{k-1}\big) \tag{6.4}$$

$$\gamma^x\big(y^1,\ldots,y^{k-2}\big) =^{\mathrm{def}} \phi^x\big(y^1,\ldots,y^{k-2},y_0^{k-1}\big)\phi^x\big(y^1,\ldots,y^{k-2},y_1^{k-1}\big)$$

and for $1 \leq i \leq k-2$,

$$\begin{aligned}&\gamma^i\big(x, y^1, \ldots, y^{i-1}, y^{i+1}, \ldots, y^{k-2}\big)\\ &\;=^{\text{def}} \phi^i\big(x, y^1, \ldots, y^{i-1}, y^{i+1}, \ldots, y^{k-2}, y_0^{k-1}\big)\phi^i\big(y^1, \ldots, y^{i-1}, y^{i+1}, \ldots, y^{k-2}, y_1^{k-1}\big).\end{aligned}$$

For each y_0^{k-1}, y_1^{k-1}, let γ denote the $(k-1)$-fold cylinder intersection formed by the cylinders $\gamma^x, \gamma^1, \ldots, \gamma^{k-2}$. Further, let ν be the $(k-1)$-fold product distribution $\mu^x \times \mu^1 \times \cdots \times \mu^{k-2}$. Then,

$$\left|G\big(y_0^{k-1}, y_1^{k-1}\big)\right| = \text{disc}_{k-1,\nu}^{\gamma}(g). \tag{6.5}$$

Noting that repeatedly applying Cauchy-Schwartz m times yields $(\mathbb{E}z)^{2^m} \leq \mathbb{E}z^{2^m}$ for any integer $m \geq 0$, plugging (6.5) into (6.3) yields,

$$\big(\text{disc}_{k,\mu}^{\phi}\big)^{2^{k-1}} \leq \big(\text{disc}_{k-1,\nu}^{\gamma}(g)\big)^{2^{k-2}}. \tag{6.6}$$

Applying the Inductive Hypothesis to the RHS of (6.6) further gives

$$\big(\text{disc}_{k,\mu}^{\phi}(f)\big)^{2^{k-1}} \leq \mathbb{E}_{y_0^{k-1}, y_1^{k-1}} \mathbb{E}_{y_0^1, y_1^1 \ldots, y_0^{k-2}, y_1^{k-2}} \left|\mathbb{E}_x \prod_{u \in \{0,1\}^{k-2}} g\big(x, y_{u_1}^1, \ldots, y_{u_{k-2}}^{k-2}\big)\right|.$$

Substituting the definition of g given in terms of f by (6.4), the above expression yields easily the RHS of (6.2). As ϕ is an arbitrary cylinder intersection, we are done. ∎

6.2 Generating functions with low discrepancy

6.2.1 Masking Schemes

We have already defined one masking scheme through the notation $x \Leftarrow y_1, \ldots, y_k$. This allowed us to define G_k^g for a base function g. Well-known functions such as GIP_k and DISJ_k are representable in this notation by G_k^{PARITY} and G_k^{NOR} respectively. We now define a second masking scheme which plays a crucial role in lower bounding the communication complexity of G_k^g. This masking scheme is obtained by first slightly simplifying the pattern matrices in [She08b] and then generalizing the simplified matrices to higher dimensions for dealing with multiple players.

Let $S^1, \ldots S^{k-1} \in [\ell]^m$ for some positive ℓ and m. Let $x \in \{0,1\}^n$ where $n = \ell^{k-1}m$. Here it is convenient to think of x to be divided into m equal blocks where each block is a $(k-1)$-dimensional array with each dimension having size ℓ. The array corresponding to the ith block of x is denoted by $x[i]$. Further, each S^i is a vector of length m with each co-ordinate being an element from $\{1, \ldots, \ell\}$. The $(k-1)$ vectors $S^1, \ldots, S^{k-1}$ jointly unmask m bits of x, denoted by $x \leftarrow S^1, \ldots, S^{k-1}$, precisely one from each block of x, i.e.

$$x[1]\left[S^1[1], S^2[1], ..., S^{k-1}[1]\right], \ldots, x[m]\left[S^1[m], S^2[m], \ldots, S^{k-1}[m]\right].$$

See Figure 6–1 for an illustration of this masking scheme.

For a given base function $f : \{0,1\}^m \to \{-1,1\}$, we define $F_k^f : \{0,1\}^n \times ([\ell]^m)^{k-1} \to \{-1,1\}$ as $F_k^f(x, S^1, \ldots, S^{k-1}) = f(x \leftarrow S^1, \ldots, S^{k-1})$, where $n = \ell^{k-1}m$.

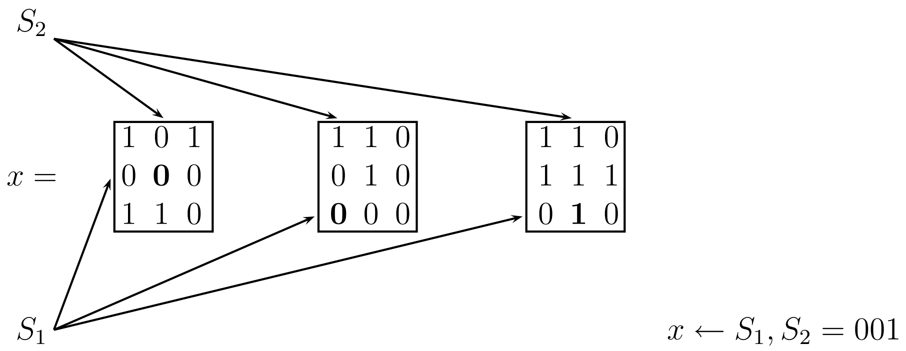

Figure 6–1: Illustration of the masking scheme $x \leftarrow S_1, S_2$. The parameters are $\ell = 3, m = 3, n = 27$.

Lemma 6.7 *Let $n = \ell^{k-1}m$. If $f : \{0,1\}^m \to \{-1,1\}$ and $f' : \{0,1\}^n \to \{-1,1\}$ are related by $f(z) = f'(z0^{n-m})$, then*

$$R_k^\epsilon(F_k^f) \le R_k^\epsilon(G_k^{f'}). \tag{6.7}$$

Proof:[Proof Sketch] Observe that there are functions $\Gamma_i : [\ell]^m \to \{0,1\}^n$ such that $F_k^f(x, S^1, \dots, S^{k-1}) = G_k^{f'}(x, \Gamma_1(S^1), \dots, \Gamma_{k-1}(S^{k-1}))$ for all $x, S^1, \dots, S^{k-1}$. Therefore the players can privately convert their inputs and apply the protocol for $G_k^{f'}$.

■

Note that the proof shows that (6.7) holds not just for randomized but any model of communication.

6.2.2 Orthogonality and Discrepancy

Here, we prove that if the base function f in our masking scheme has a certain nice property, then the masked function F_k^f has small discrepancy. To describe this property, let us define the following: for a distribution μ on inputs, equip the space with the biased inner product $\langle\cdot\rangle_\mu$, where for two functions f, g,

$$\langle f, g\rangle_\mu =^{\text{def}} \mathbb{E}_{x\sim\mu} f(x)g(x).$$

We say that f is (μ, d)-orthogonal if it is orthogonal, w.r.t. the above μ-biased inner product, to every parity/character of order less than d , i.e. $\langle f, \chi_S\rangle_\mu = 0$, for all $|S| < d$.

Lemma 6.8 (Orthogonality-Discrepancy Lemma) *Let $f : \{-1, 1\}^m \to \{-1, 1\}$ be any (μ, d)-orthogonal function for some distribution μ on $\{-1, 1\}^m$ and some integer $d > 0$. Derive the probability distribution λ on $\{-1, 1\}^n \times \big([\ell]^m\big)^{k-1}$ from μ as follows: $\lambda(x, S^1, \ldots, S^{k-1}) = \frac{\mu(x\leftarrow S^1,\ldots,S^{k-1})}{\ell^{m(k-1)}2^{n-m}}$. Then,*

$$\Big(disc_{k,\lambda}\big(F_k^f\big)\Big)^{2^{k-1}} \leq \sum_{j=d}^{(k-1)m} \binom{(k-1)m}{j}\left(\frac{2^{2^{k-1}-1}}{\ell-1}\right)^j. \tag{6.8}$$

Hence, for $\ell - 1 \geq \frac{2^{2^k}(k-1)em}{d}$ and $d > 2$,

$$disc_{k,\lambda}\big(F_k^f\big) \leq \frac{1}{2^{d/2^{k-1}}}. \tag{6.9}$$

Proof: The starting point is the expression for discrepancy w.r.t. an arbitrary cylinder intersection ϕ,

$$\mathrm{disc}_k^\phi(F_k^f) = \Bigg|\sum_{x,S^1,\ldots,S^{k-1}} F_k^f(x, S^1, \ldots, S^{k-1})\phi(x, S^1, \ldots, S^{k-1}) \cdot \lambda(x, S^1, \ldots, S^{k-1})\Bigg|. \tag{6.10}$$

This changes to the more convenient expected value notation as follows:

$$\mathrm{disc}_k^{\phi}(F_k^f) = 2^m \left| \mathbb{E}_{x,S^1,\ldots,S^{k-1}} F_k^f(x,S^1,\ldots,S^{k-1}) \times \ \phi(x,S^1,\ldots,S^{k-1})\mu\big(x \leftarrow S^1,\ldots,S^{k-1}\big) \right| \tag{6.11}$$

where, $(x,S^1,\ldots,S^{k-1})$ is now uniformly distributed over $\{0,1\}^{\ell^{k-1}m} \times \big([\ell]^m\big)^{k-1}$. Thus, defining G_k^f as $G_k^f(x,S^1,\ldots,S^{k-1}) =^{\mathrm{def}} F_k^f(x,S^1,\ldots,S^{k-1})\mu(x \leftarrow S^1,\ldots,S^{k-1})$, we have

$$\mathrm{disc}_{k,\lambda}(F_k^f) = 2^m \mathrm{disc}_{k,\mathcal{U}}(G_k^f)$$

where $\mathcal{U}$ is the uniform distribution.

Application of Equation (6.2) of Lemma 6.6 to the function G_k^f easily yields

$$\begin{aligned}\big(\mathrm{disc}_{k,\lambda}(F_k^f)\big)^{2^{k-1}} &= 2^{m2^{k-1}}\big(\mathrm{disc}_{k,\mathcal{U}}(G_k^f)\big)^{2^{k-1}} \\ &\le 2^{m2^{k-1}} \mathbb{E}_{S_0^1,S_1^1,\ldots,S_0^{k-1},S_1^{k-1}} H_k^f\big(S_0^1,S_1^1,\ldots,S_0^{k-1},S_1^{k-1}\big) \qquad (6.12)\end{aligned}$$

where,

$$\begin{aligned}&H_k^f\big(S_0^1,S_1^1,\ldots,S_0^{k-1},S_1^{k-1}\big) \\ &= \left| \mathbb{E}_{x\in\{0,1\}^{\ell^{k-1}m}} \prod_{u\in\{0,1\}^{k-1}} \left(F_k^f(x,S_{u_1}^1,\ldots,S_{u_{k-1}}^{k-1})\mu(x \leftarrow S_{u_1}^1,\ldots,S_{u_{k-1}}^{k-1}) \right) \right|. \qquad (6.13)\end{aligned}$$

We look at a fixed S_0^i, S_1^i, for $i = 1,\ldots,k-1$. Let $r_i = \big|S_0^i \cap S_1^i\big|$ and $r = \sum_i r_i$ for $1 \le i \le k-1$. We make two claims below.

Claim 6.9

$$H_k^f\big(S_0^1, S_1^1, \ldots, S_0^{k-1}, S_1^{k-1}\big) \leq \frac{2^{(2^{k-1}-1)r}}{2^{2^{k-1}m}}. \tag{6.14}$$

Claim 6.10 *Let $r < d$. Then,*

$$H_k^f\big(S_0^1, S_1^1, \ldots, S_0^{k-1}, S_1^{k-1}\big) = 0. \tag{6.15}$$

We prove these claims in the next section. Claim 6.9 simply follows from the fact that μ is a probability distribution and f is 1/-1 valued while Claim 6.10 uses the (μ, d)-orthogonality of f. We now continue with the proof of the Orthogonality-Discrepancy Lemma assuming these claims. Applying them, we obtain

$$\begin{aligned}&(\mathrm{disc}_k^{\phi}(F_k^f))^{2^{k-1}}\\ &\leq \sum_{j=d}^{(k-1)m} 2^{(2^{k-1}-1)j} \sum_{j_1+\cdots+j_{k-1}=j} \Pr\big[r_1 = j_1 \wedge \cdots \wedge r_{k-1} = j_{k-1}\big].\end{aligned} \tag{6.16}$$

Since the random variables $r_1, \ldots, r_{k-1}$ are independent, $\Pr[r_1 = j_1 \wedge \cdots \wedge r_{k-1} = j_{k-1}] = \Pr[r_1 = j_1] \cdots \Pr[r_{k-1} = j_{k-1}]$. Noting that $\Pr[r_i = j_i] = \binom{m}{j_i}\frac{(\ell-1)^{m-j_i}}{\ell^m}$, we further obtain:

$$\begin{aligned}&(\mathrm{disc}_k^{\phi}(F_k^f))^{2^{k-1}}\\ &\leq \sum_{j=d}^{(k-1)m} 2^{(2^{k-1}-1)j} \sum_{j_1+\cdots+j_{k-1}=j} \binom{m}{j_1} \cdots \binom{m}{j_{k-1}} \frac{(\ell-1)^{m-j_1} \cdots (\ell-1)^{m-j_{k-1}}}{\ell^{(k-1)m}}.\end{aligned} \tag{6.17}$$

The following simple combinatorial identity is well known:

$$\sum_{j_1+\cdots+j_{k-1}=j} \binom{m}{j_1}\cdots\binom{m}{j_{k-1}} = \binom{(k-1)m}{j}.$$

Plugging this identity into (6.17) immediately yields (6.8) of the Orthogonality-Discrepancy Lemma. Recalling $\binom{(k-1)m}{j} \leq \left(\frac{e(k-1)m}{j}\right)^j$, and choosing $\ell - 1 \geq 2^{2^k}(k-1)em/d$, we get (6.9).

∎

6.2.3 Proofs of Claims

We identify the set of all assignments to boolean variables in $X = \{x_1, \ldots, x_n\}$ with the n-ary boolean cube $\{0,1\}^n$. For any $u \in \{0,1\}^{k-1}$, let Z_u represent the set of m variables indexed jointly by $S^1_{u_1}, \ldots, S^{k-1}_{u_{k-1}}$. There is precisely one variable chosen from each block of X. Denote by $Z_u[\alpha]$ the unique variable in Z_u that is in the αth block of X, for each $1 \leq \alpha \leq m$. Let $Z = \cup_u Z_u$. Abusing notation, we use Z_u in the context of expected value calculations to also mean a uniformly chosen random assignment to the variables in the set Z_u.

Proof:[Proof of Claim 6.10]

$$\begin{aligned} & H^f_k\left(S^1_0, S^1_1, \ldots, S^{k-1}_0, S^{k-1}_1\right) \\ &= \left| \mathbb{E}_{Z_{0^{k-1}}} f(Z_{0^{k-1}})\mu(Z_u)\, \mathbb{E}_{X - Z_{0^{k-1}}} \prod_{\substack{u \in \{0,1\}^{k-1} \\ u \neq 0}} f(Z_u)\mu(Z_u) \right|. \end{aligned} \tag{6.18}$$

Observe that for any block α and any $u \neq 0^{k-1}$, $Z_u[\alpha] = Z_{0^{k-1}}[\alpha]$ iff for each i such that $u_i = 1$, $S^i_0[\alpha] = S^i_1[\alpha]$. Recall that r_i is the number of indices α such that $S^i_0[\alpha] = S^i_1[\alpha]$. Therefore, there are at most $r = \sum_{i=1}^{k-1} r_i$ many indices α such that

$Z_u[\alpha] = Z_{0^{k-1}}[\alpha]$ for some $u \neq 0^{k-1}$. This means the inner expectation in (6.18) is a function that depends on at most r variables. Since f is orthogonal under μ with every polynomial of degree less than d and $r < d$, we get the desired result. ■

Proof:[Proof of Claim 6.9] Observe that since F_k^f is 1/-1 valued, we get the following:

$$\begin{aligned} H_k^f\big(S_0^1, S_1^1, \ldots, S_0^{k-1}, S_1^{k-1}\big) &\leq \mathbb{E}_x \prod_{u \in \{0,1\}^{k-1}} \mu(x \leftarrow S_{u_1}^1, \ldots, S_{u_{k-1}}^{k-1}) \\ &= \mathbb{E}_{X-Z}\, \mathbb{E}_Z \prod_{u \in \{0,1\}^{k-1}} \mu(Z_u) \\ &= \mathbb{E}_{X-Z}\, \frac{1}{2^{|Z|}} \sum_{Z \in \{0,1\}^{|Z|}} \prod_{u \in \{0,1\}^{k-1}} \mu(Z_u) \qquad (6.19) \\ &\leq \mathbb{E}_{X-Z}\, \frac{1}{2^{|Z|}} \sum_{\substack{y^1, \ldots, y^{2^{k-1}} \\ \in \{0,1\}^m}} \prod_{i=1}^{2^{k-1}} \mu(y^i) \qquad (6.20) \end{aligned}$$

where the last inequality holds because every product in the inner sum of (6.19) appears in the inner sum of (6.20). Using the fact that μ is a probability distribution, we get:

$$\begin{aligned} \text{RHS of (6.20)} &= \mathbb{E}_{X-Z}\, \frac{1}{2^{|Z|}} \prod_{i=1}^{2^{k-1}} \sum_{y^i \in \{0,1\}^m} \mu(y^i) \\ &= \mathbb{E}_{X-Z}\, \frac{1}{2^{|Z|}} \\ &= \frac{1}{2^{|Z|}}. \end{aligned}$$

We find a lower bound on $|Z|$. Let t_u denote the Hamming weight of the string u and $\{j_1, \ldots, j_{t_u}\}$ denote the set of indices in $[k-1]$ at which u has a 1. Define

$$Y_u = \Big\{ Z_u[\alpha] \mid S_1^{j_s}[\alpha] \neq S_0^{j_s}[\alpha];\ 1 \leq s \leq t_u;\ 1 \leq \alpha \leq m \Big\}. \tag{6.21}$$

The following follow from the above definition.

- $|Y_{0^{k-1}}| = m$ and $|Y_u| \geq m - \sum_{1 \leq s \leq t_i} r_{j_s} \geq m - r$ for all $u \neq 0^{k-1}$.
- $Y_u \cap Y_v = \emptyset$, for $u \neq v$. This follows from the following argument: W.l.o.g. assume there is an index β where u has a one but v has a zero. Consider any block α such that $Z_u[\alpha]$ is in Y_u. It must be true that $S_1^{\beta}[\alpha] \neq S_0^{\beta}[\alpha]$. This means that $Z_u[\alpha] \neq Z_v[\alpha]$. Therefore $Z_u[\alpha]$ is not in Y_v and we are done.
- $Y := \cup_{u \in \{0,1\}^{k-1}} Y_u = Z$. This is because if $Z_u[\alpha]$ is not in Y_u then there are indices $j_1, \ldots, j_s$ where u contains a one and $S_0^{j_i}[\alpha] = S_1^{j_i}[\alpha]$. Let v be the string that contains a zero at positions $j_1, \ldots, j_s$ and at other positions, corresponds to u. Then by definition, $Z_u[\alpha] = Z_v[\alpha] \in Y_v$.

Thus, $|Z| = |Y| = \sum_u |Y_u| \geq m + \sum_{u \neq 0}(m - r) = 2^{k-1}m - (2^{k-1} - 1)r$ and the result follows. ∎

6.3 Masking functions of high voting degree

The theorem below shows that (μ, d)-orthogonality of a function f, that is key to using the Orthogonality-Discrepancy Lemma, follows from the fact that the voting degree of f is more than d.

Theorem 6.11 (see [She07]) *For any boolean function $f : \{-1,1\}^n \to \{-1,1\}$, precisely one of the following holds:*

- *$deg(f) \leq d$.*

- *there exists a distribution μ over $\{-1,1\}^n$, such that f is (μ, d)-orthogonal, i.e. for all $|S| \leq d$, $\langle f(x), \chi_S(x)\rangle_\mu = 0$.*

In particular, this means that if $\deg(f) \geq d$, then for any function g that depends on at most $d-1$ variables, $\langle f(x), g(x)\rangle_\mu = 0$.

As an immediate consequence of Theorem 6.11 and the Orthogonality-Discrepancy Lemma, we obtain the following:

Corollary 6.12 (Multiparty Degree-Discrepancy Lemma) *Let $f : \{-1,1\}^m \to \{-1,1\}$ have voting polynomial degree d. Then for any $k \geq 2$, there exists a probability distribution λ such that for $\ell \geq m$,*

$$\left(disc_{k,\lambda}\big(F_k^f\big)\right)^{2^{k-1}} \leq \sum_{j=d}^{m} \binom{(k-1)m}{j} \left(\frac{2^{2^{k-1}-1}}{\ell-1}\right)^j.$$

Hence, for $\ell - 1 \geq \frac{2^{2^k}(k-1)em}{d}$ and $d > 2$,

$$disc_{k,\lambda}\big(F_k^f\big) \leq \frac{1}{2^{d/2^{k-1}}}.$$

The above lemma, using a slightly different masking function and with a quadratic dependence of ℓ on m (instead of the linear dependence above), appeared in our work [Cha07b] as an extension of the two player Degree/Discrepancy Theorem of Sherstov [She07].

Combining Corollary 6.12 with the Discrepancy Method (i.e. Lemma 4.18) directly yields a method to obtain lower bounds on a masked function whose base function has high voting degree.

Theorem 6.13 *Let f, defined on inputs of length m, have voting degree d. For any $k \geq 2$, define F_k^f as before on inputs of length $n = m\ell^{k-1}$, where $\ell \geq \frac{2^{2^k}(k-1)em}{d}$ and $d > 2$. Then,*

$$R_k^{\epsilon}\big(F_k^f\big) = \Omega\big(\frac{d}{2^{k-1}} + \log \epsilon\big). \tag{6.22}$$

6.4 Communication complexity of functions in AC^0

Given Theorem 6.13, the natural question is "how difficult is it to find a function of high voting degree?". We recall from Section 6.1.1 that the Minsky-Papert function, $\mathrm{MP}(x) = \vee_{i=1}^{t} \wedge_{j=1}^{4t^2} x_{i,j}$, is in depth-two AC^0 and has a high voting degree of t.

Corollary 6.14 *Consider the Minsky-Papert function MP on m variables. Let $d = \Omega(m^{1/3})$ denote its voting degree. If $n = m\ell^{k-1}$ and $\ell = 2^{2^k}(k-1)em/d$, then*

$$R_k^{\epsilon}\big(F_k^{MP}\big) = \Omega\Bigg(\frac{n^{\frac{1}{2k+1}}}{\big(2^{2^k/(2k+1)}2e(k-1)\big)^{k-1}} + \log \epsilon\Bigg).$$

Proof: The result follows by a short and straightforward calculation, starting from Theorem 6.13. We include it for the sake of completeness. Noting that $m = d^3$

$$n = m\ell^{k-1} = \big(2^{2^k}(k-1)e\big)^{k-1} d^{2k+1}.$$

Hence,

$$d = \frac{n^{\frac{1}{2k+1}}}{\big(2^{2^k/(2k+1)}e(k-1)\big)^{k-1}}.$$

Application of (6.22) to the above completes the calculation. ∎

This shows that there exists simple base functions computable by small depth-two AC^0 circuits that give rise to masked functions of large randomized communication complexity. The following observation shows that our masking scheme does not significantly increase the circuit complexity of the base function.

Observation 6.15 *Let $f : \{0,1\}^m \to \{0,1\}$ be any boolean function and F_k^f be the corresponding masked function on $n = m\ell^{k-1}$ bits. If f can be computed by a circuit C of size $s(m)$ and depth $d(m)$, then F_k^f can be computed by a circuit $C \circ AND \circ OR$ of size $n + s(m)$ and depth $d(m) + 2$.*

Proof: We view the domain of F_k^f as $\{0,1\}^{m\ell^{k-1}} \times \left(\left(\{0,1\}^{\log \ell}\right)^m\right)^{k-1}$, encoding each index/pointer by $(\log \ell)$ bits.

Consider the decoding function $U : \{0,1\}^{\ell^{k-1}} \times \{0,1\}^{(k-1)\log \ell}$ that on input (α, β) interprets β to be a set of $k-1$ indices from $[\ell]$ and then outputs the bit of the block α (of size ℓ^{k-1}) corresponding to this set of indices. It is not hard to verify that U is computed by a depth-two $\mathrm{OR} \circ \mathrm{AND}$ circuit of size ℓ^{k-1}. It also gets easily verified that if we replace each bit of the block α by its complement in the $\mathrm{OR} \circ \mathrm{AND}$ circuit for U, we compute the complement of U, i.e. $\neg U$. Applying de Morgan's law to this circuit for $\neg U$ (i.e. negating the circuit and propagating the negations using de Morgan's laws to the bottom) yields the required $\mathrm{AND} \circ \mathrm{OR}$ circuit of size ℓ^{k-1} for U. Thus, $F_k^f(x, S^1, \ldots, S^{k-1}) = f\left(U(x[1], y_1), \ldots, U(x[m], y_m)\right)$, where $x[i]$ is the ith block of x and each y_i is the binary string of length $(k-1)\log \ell$ obtained by concatenating the encodings of the ith co-ordinate of each vector $S^1, \ldots, S^{k-1}$.

Thus, computing each of the m instances of U by a corresponding AND $\circ$ OR circuit of size ℓ^{k-1} and f by the circuit C we derive the observation. ∎

Fact 6.16 (follows from [She07]) *The function F_k^{MP} has a linear size depth-three AC^0 circuit.*

Proof: One derives a depth-four circuit for F_k^{MP} by applying Observation 6.15 and the fact that MP on m variables can be computed by a depth-two AND $\circ$ OR circuit. Note that the two middle layers of this circuit consist only of AND gates and can thus be collapsed into a single layer. This yields the required depth-three circuit for F_k^{MP}. ∎

The above fact and the lower bound on the randomized communication complexity of F_k^{MP} shows that there are functions that can be computed very efficiently by depth-three AC^0 circuits that have no efficient multiparty randomized protocols as long as the number of players is $o(\log\log n)$, even when a mere inverse-quasipolynomial advantage over random guessing is required. This, in some sense, complements the result from the last chapter where we saw that just three players can compute deterministically functions of arbitrarily large circuit complexity in *constant* cost.

6.5 The Generalized Discrepancy Method

At the heart of the technique introduced in the last section is the Discrepancy Method (Lemma 4.18). Unfortunately, its applicability is limited to those functions that have small discrepancy. However, there are several important and simple functions that have large cylindrical discrepancy. Disjointness is a classical example of such a function.

Lemma 6.17 (Folklore) *Under every distribution μ over the inputs,*

$$disc_{k,\mu}(DISJ_k) \geq \frac{1}{2n} - \frac{1}{2n^2}.$$

Proof: Let X^+ and X^- be the set of disjoint and non-disjoint inputs respectively. The first thing to observe is that if $|\mu(X^+) - \mu(X^-)| \geq (1/n)$, then we are done immediately by considering the discrepancy over the intersection corresponding to the entire set of inputs. Hence, we may assume $|\mu(X^+) - \mu(X^-)| < (1/n)$. Thus, $\mu(X^-) \geq 1/2 - (1/2n)$. However, X^- can be covered by the following n *monochromatic* cylinder intersections: let C_i be the set of inputs in which the ith column is an all-one column. Then $X^- = \cup_{i=1}^{n} C_i$. By averaging, there exists an i such that $\mu(C_i) \geq 1/2n - (1/2n^2)$. Taking the discrepancy of this C_i, we are done. ∎

It is therefore impossible to obtain better than $\Omega(\log n)$ bounds on the communication complexity of Disjointness by a direct application of the discrepancy method. In fact, the above argument shows that this method fails to give better than polylogarithmic lower bounds for any function that is in NP_k^{cc} or co-NP_k^{cc}. In other words, the Discrepancy Method is too strong, i.e. not only does it yield bounds for the randomized model, but it also yields bounds on non-deterministic communication complexity. This makes it unsuitable as a method for separating the power of randomness from non-determinism, i.e. classes BPP_k^{cc} and NP_k^{cc} (or co-NP_k^{CC}).

Fortunately, there is a simple generalization of the Discrepancy Method that is somewhat surprisingly effective for dealing with several functions that have large discrepancy. Curiously, this method grew out of research on quantum communication complexity. To the best of our knowledge, it remained unknown among several researchers whose primary focus was on classical communication.

The origins of the idea of generalizing the discrepancy method can be found in the work of Klauck [Kla01][6] . Klauck considered, in the setting of two players, functions of the form $f(x, y) = g(x \wedge y)$ where the $\wedge$ operation is naturally applied bitwise to the bits of x and y. He observed that if g correlates well with a parity function on some large subset S of $\{1, \ldots, n\}$ under the uniform distribution[7] , then f correlates well with the inner-product function of the columns indexed by elements of S, denoted by IP_S, under a simple product distribution μ. The ingenuity in Klauck's argument is that he shows IP_S having small discrepancy under μ implies that f has large distributional complexity under μ. This, as he correctly adds, follows despite the possibility that f itself has large discrepancy. Indeed, Klauck proves that IP has very small rectangular discrepancy under μ. Klauck goes on to show that this "generalized form of the discrepancy method" can be used to obtain a lower bound of $\Omega(n/\log n)$ on the quantum (and hence classical randomized) communication complexity of $\mathrm{MAJ}(x \wedge y)$ despite the fact that it has large discrepancy.

The main idea in Klauck's work can be abstracted in following terms: A function f may have high discrepancy and still correlate well under some distribution μ with a function h that has small discrepancy under μ. Exhibiting such a h, yields lower bounds on the bounded-error communication complexity of f.

[6] The full version of Klauck's work appears in [Kla07].

[7] In other words, g has a large high-order Fourier coefficient, i.e. $\hat{f}(S)$ is large.

This principle was re-expressed, in a more general fashion, in matrix theoretic terms for the two player quantum communication model by Razborov [Raz03], where he called it the "Discrepancy Method". One may dare say, that this matrix theoretic formulation may have hindered the recognition of the wider applicability of the underlying principle. Sherstov [She08b, Sec. 2.4] provides a nice reinterpretation of Razborov's formulation of the Discrepancy method and points out the fact that the general principle at play is independent of the precise communication model for two players. Based on this observation by Sherstov, we specialize the Klauck-Razborov Principle to the multi-party model in [CA08] as follows:

Lemma 6.18 (Generalized Discrepancy Method) *Denote* $X = Y_1 \times ... \times Y_k$. *Let* $f : X \to \{-1, 1\}$ *and* $g : X \to \{-1, 1\}$ *be such that under some distribution* μ *we have* $Corr_\mu(f, g) \geq \delta$. *Then*

$$R_k^\epsilon(f) \geq \log\left(\frac{\delta + 2\epsilon - 1}{disc_{k,\mu}(g)}\right). \tag{6.23}$$

Proof: Let $\mathcal{P}$ be a k-party randomized protocol that computes f with advantage ϵ and cost c. Then for every distribution μ over the inputs, we can derive a deterministic k-player protocol $\mathcal{P}'$ for f that errs only on at most a $(1/2 - \epsilon)$ fraction of the inputs (w.r.t. μ) and has cost c. Take μ to be a distribution satisfying the correlation inequality. We know that $\mathcal{P}'$ partitions the input space into at most 2^c monochromatic (w.r.t. $\mathcal{P}'$) cylinder intersections. Let $\mathcal{C}$ denote this set of cylinder intersections. Then,

$$\begin{aligned}\delta &\leq \big|\mathbb{E}_{x\sim\mu} f(x)g(x)\big| \\ &= \big|\sum_x f(x)g(x)\mu(x)\big| \\ &\leq \big|\sum_x \mathcal{P}'(x)g(x)\mu(x)\big| + \big|\sum_x (f(x)-\mathcal{P}'(x))g(x)\mu(x)\big|.\end{aligned}$$

Since $\mathcal{P}'$ is a constant over every cylinder intersection S in $\mathcal{C}$, we have

$$\begin{aligned}\delta &\leq \sum_{S\in\mathcal{C}} \big|\sum_{x\in S} \mathcal{P}'(x)g(x)\mu(x)\big| + \sum_x \big|g(x)\big|\cdot\big|f(x)-\mathcal{P}'(x)\big|\mu(x) \\ &\leq \sum_{S\in\mathcal{C}} \big|\sum_{x\in S} g(x)\mu(x)\big| + \sum_x \big|f(x)-\mathcal{P}'(x)\big|\mu(x) \\ &\leq 2^c \mathrm{disc}_{k,\mu}(g) + 2(1/2-\epsilon).\end{aligned}$$

This gives us immediately (6.23). ∎

Observe that when $f = g$, i.e. $\mathrm{Corr}_\mu(f,g) = 1$, we recover the classical discrepancy method (Lemma 4.18).

6.5.1 Applications to Disjointness

Although the "generalized form of the discrepancy method" was known to researchers in quantum communication complexity since the work of Klauck [Kla01], it was not known if this method could be applied to Disjointness. In fact, Razborov [Raz03] remarks that even this generalized principle is not applicable to the Disjointness function. Sherstov [She08b] disproves this remark by designing a novel strategy that allows the application of this Generalized Discrepancy Method to yield strong lower bounds on the 2-party bounded-error quantum communication complexity of

Disjointness. A key ingredient in this strategy is a beautiful duality between approximability and orthogonality. The intuition is that if a function is at a large distance from the linear space spanned by characters of degree less than d, then its projection on the dual space spanned by characters of degree at least d is large. More precisely, recall from Section 6.1.1 that the δ-approximation degree of a boolean function f, denoted by $\deg_\delta(f)$, is the degree of the smallest degree real polynomial that approximates f point-wise within δ.

Lemma 6.19 (Sherstov [She08b], Shi and Zhu [SZ07]) *Let $f : \{-1,1\}^m \to \mathbb{R}$ be given with $deg_\delta(f) = d \geq 1$. Then there exists $g : \{-1,1\}^m \to \{-1,1\}$ and a distribution μ on $\{-1,1\}^m$ such that g is (μ, d)-orthogonal and $Corr_\mu(f, g) > \delta$.*

This Approximation/Orthogonality Principle is a classical result in functional analysis. It has been of interest to researchers in computational complexity before[8] in other contexts. But to the best of our knowledge, its use in communication complexity first appears in the independent works of Sherstov [She08b] and Shi and Zhu [SZ07]. We do not prove this lemma but the interested reader can look up its short proof in [She08b, SZ07, Spa08] which is based on an application of linear programming duality. In this section, we extend Sherstov's strategy to the multiparty setting using the Orthogonality-Discrepancy Lemma.

[8] For instance, in his work [Spa08] Spalek credits Buhrman and Szegedy to have discovered this principle independently.

Theorem 6.20 *Let $f : \{0,1\}^m \to \{-1,1\}$ have δ-approximate degree d. Let $n \geq \big(\frac{2^{2^k}(k-1)e}{d}\big)^{k-1} m^k$, and $f' : \{0,1\}^n \to \{-1,1\}$ be such that $f(z) = f'(z0^{n-m})$. Then*

$$R_k^{\epsilon}(G_k^{f'}) \geq \frac{d}{2^{k-1}} + \log(\delta + 2\epsilon - 1). \tag{6.24}$$

Proof: Applying Lemma 6.19 we obtain a function g and a distribution μ such that $\mathrm{Corr}_{\mu}(f,g) > \delta$ and g is (μ, d)-orthogonal. Thus, applying the Orthogonality-Discrepancy Lemma 6.8, we get

$$\mathrm{disc}_{k,\lambda}\big(F_k^g\big) \leq \frac{1}{2^{d/2^{k-1}}} \tag{6.25}$$

where λ is precisely obtained from μ as stated in Lemma 6.8 and $\ell \geq 2^{2^k}(k-1)em/d$. Since $n = \ell^{k-1}m$, (6.25) holds for $n \geq \big(\frac{2^{2^k}(k-1)e}{d}\big)^{k-1} m^k$.

It can be easily verified that $\mathrm{Corr}_{\lambda}(F_k^f, F_k^g) = \mathrm{Corr}_{\mu}(f,g) > \delta$. Thus, by plugging the value of $\mathrm{disc}_{k,\lambda}\big(F_k^g\big)$ in (6.23) of the Generalized Discrepancy Method, we get

$$R_k^{\epsilon}(F_k^f) \geq \frac{d}{2^{k-1}} + \log(\delta + 2\epsilon - 1).$$

We observe that the communication matrix of F_k^f embeds as a submatrix in the communication matrix of $G_k^{f'}$. The proof is finished by noting that a protocol for solving $G_k^{f'}$ yields one for G_k^f. ■

In particular, strong lower bounds on the bounded-error randomized multiparty communication complexity of Disjointness follows readily from Theorem 6.20. This significantly improves the best earlier lower bound of $\Omega(\log n)$ due to Tesson [Tes03] and Beame et.al. [BPSW06] for three or more players.

Corollary 6.21

$$R_k^{\epsilon}(DISJ_k) = \Omega\left(\frac{n^{\frac{1}{k+1}}}{2^{2^k}(k-1)2^{k-1}}\right)$$

for any constant $\epsilon > 0$.

Proof: Let $f = \mathrm{NOR}_m$ and $f' = \mathrm{NOR}_n$. We know $\deg_{1/3}(\mathrm{NOR}_m) = \Theta(\sqrt{m})$ by Theorem 6.5. Setting $n = \left(\frac{2^{2^k}(k-1)e}{\deg_{1/3}(\mathrm{NOR}_m)}\right)^{k-1} m^k$, and writing (6.24) in terms of n gives the result for any constant $\epsilon > 1/6$. The bound can be made to work for every constant ϵ by a standard boosting argument. ∎

Recall that there is a simple non-deterministic protocol of cost $O(\log n)$ computing non-Disjointness, i.e. G_k^{OR}. Thus, Corollary 6.21 provides an explicit separation of the class of functions having efficient randomized protocols with bounded error from the class of functions having efficient non-deterministic protocols, i.e. $\mathrm{NP}_k^{cc} \nsubseteq \mathrm{BPP}_k^{cc}$ for $k < \log\log n - \log\log\log n$. Such a separation first appeared in the joint work with A. Ada [CA08] and independently in the work of Lee and Shraibman [LS08]. David, Pitassi and Viola [DPV08] have recently pushed our argument further, making elegant use of the probabilistic method, to show that such a separation continues to exist for $\delta \log n$ players for every constant $\delta < 1$. They also provide an explicit function witnessing their separation by derandomizing their argument.

6.5.2 Other Symmetric Functions

Theorem 6.20 does not immediately provide strong bounds on the communication complexity of G_k^f for every symmetric f. For instance, if f is the MAJORITY function then one has to work a little more to derive strong lower bounds[9].

In this section, using Theorem 6.20 and Paturi's Theorem (Theorem 6.5), we obtain a lower bound on the communication complexity of G_k^f for each non-trivial symmetric f. Let $f : \{0,1\}^n \to \{1,-1\}$ be the symmetric function induced from a predicate $D : \{0,1,\ldots,n\} \to \{1,-1\}$. We denote by G_k^D the function G_k^f. For $t \in \{0,1,\ldots,n-1\}$, define $D_t : \{0,1,\ldots,n-t\} \to \{1,-1\}$ by $D_t(i) = D(i+t)$. Observe that the communication complexity of G_k^D is at least the communication complexity of $G_k^{D_t}$.

Corollary 6.22 *Let $D : \{0,1,\ldots,n\} \to \{1,-1\}$ be any predicate with the $(1/3)$-approximate degree of D, denoted by $deg_{1/3}(D)$, equal to d. Let $\ell_0 = \ell_0(D)$ and $\ell_1 = \ell_1(D)$. Define $T : \mathbb{N} \to \mathbb{N}$ by*

$$T(n) = \left(\frac{n}{(2^{2^k}(k-1)e/d)^{k-1}} \right)^{\frac{1}{k}}.$$

Then for any constant $\epsilon > 0$,

$$R_k^\epsilon(G_k^D) = \Omega\left(\Psi(\ell_0) + \frac{T(\ell_1)}{2^{k-1}} \right)$$

[9] Lower bounds for G_k^{MAJ} can be obtained in another way. It is not too difficult to see that a k-party protocol for G_k^{MAJ} can be used to derive a protocol for G_k^{PARITY} with a small blow-up in cost. Thus, G_k^{MAJ} is as hard as GIP.

where,

$$\Psi(\ell_0) = \min\left\{\frac{\sqrt{T(n)\ell_0}}{2^{k-1}}, \frac{T(n-\ell_0)}{2^{k-1}}\right\}.$$

Proof: The first thing to note is that the relationship between $T(n)$ and n is exactly the relationship between m and n in Theorem 6.20. This is not accidental. Indeed, the general idea of our proof is to show that the predicate D 'embeds' another predicate D'_1 with the following property: D'_1 is defined over the set $\{0,\ldots,n_1\}$ and there is a predicate D_1 defined over $\{0,\ldots,T(n_1)\}$. Further, we show that $T(n_1), n_1$ and $\deg_{1/3}(D_1)$ can be made to correspond to m, n and d of Theroem 6.20 respectively. Here, D_1 plays the role of D in Theorem 6.20 and D'_1 that of D'. This allows us to conclude that the communication complexity of $G_k^{D'_1}$ is high. Thus, G_k^D has high communication complexity as well.

We implement the above idea by considering the following three cases. In each case, let $\ell_0 = \ell_0(D)$ and $\ell_1 = \ell_1(D)$. Further, let $c = \log(1/3 + 2\epsilon - 1)$. W.l.o.g., we assume[10] that $\epsilon > 1/3$, so that c is a well defined constant.

Case 1: Suppose $\ell_0 \leq T(n)/2$. In this case D'_1 is the same as D. Let $D_1 : \{0,1,\ldots,T(n)\} \rightarrow \{1,-1\}$ be such that for any $z \in \{0,1\}^{T(n)}$, we have $D_1(|z|) = D'_1(|z|)$. By Theorem 6.20, the complexity of G_k^D is $\Omega(d/2^{k-1})$ where $d = \deg_{1/3}(D_1)$. By Paturi's Theorem, $deg_{1/3}(D_1) \geq \sqrt{T(n)\ell_0(D_1)} = \sqrt{T(n)\ell_0}$ and so

$$R_k^\epsilon(G_k^D) \geq R_k^\epsilon(G_k^{D'_1}) = \frac{\sqrt{T(n)\ell_0}}{2^{k-1}} + c.$$

[10] We can always apply boosting later to lift our bounds to any constant ϵ, using Observation 4.1 in Chapter 4.

Case 2: Suppose $T(n)/2 < \ell_0 \leq n/2$. In this case D'_1 is D_t where $t = \ell_0 - T(n-\ell_0)/2$. Let $D_1 : \{0, 1, \ldots, T(n-\ell_0)\} \rightarrow \{1, -1\}$ be such that $D_1(|z|) = D'_1(|z|)$. So by Theorem 6.20, the complexity of $G_k^{D'_1}$ is $\Omega(d/2^{k-1})$ where d is the approximation degree of D_1. We know

$$\begin{aligned}
D_1(T(n-\ell_0)/2) &= D'_1(T(n-\ell_0)/2) \\
&= D_t(T(n-\ell_0)/2) \\
&= D(T(n-\ell_0)/2 + \ell_0 - T(n-\ell_0)/2) \\
&= D(\ell_0) \\
&\neq D(\ell_0 - 1) \text{ by defn. of } \ell_0 \\
&= D_1(T(n-\ell_0)/2 - 1).
\end{aligned}$$

Hence, $\ell_0(D_1) = T(n-\ell_0)/2$. Thus by Paturi's Theorem, $\deg_{1/3}(D_1) \geq \sqrt{T(n-\ell_0)^2/2}$. This implies, as before,

$$R_k^{\epsilon}(G_k^D) = \frac{T(n-\ell_0)}{2^{k-1}} + c.$$

Case 3: Suppose $\ell_0 = 0$ and $\ell_1 \neq 0$. Unlike in the first two cases, we bound the approximate degree of D_1 by estimating $\ell_1(D_1)$ in terms of ℓ_1. The rest of the argument is similar to the one for Case 2. Consider $D'_1 = D_t$ where $t = n - \ell_1 - T(\ell_1)/2$. Let $D_1 : \{0, 1, \ldots, T(\ell_1)\} \rightarrow \{1, -1\}$ be such that $D_1(|z|) = D'_1(|z|) = D_t(|z|)$. As in Case 2, one verifies that $D_1(T(\ell_1)/2) \neq D_1(T(\ell_1)/2 + 1)$. Thus $\ell_1(D_1) = T(\ell_1)/2$. So, $deg_{1/3}(D_1) \geq \sqrt{T(\ell_1)^2/2}$. Therefore,

$$R_k^{\epsilon}(G_k^D) = \frac{T(\ell_1)}{2^{k-1}} + c.$$

Combining these three cases, we get the desired result. ∎

6.6 Lower Bounds by Block-Composition

In this section, we develop a new lower bound technique extending the recent work of Shi and Zhu [SZ07]. We call this the multiparty block-composition method. It also yields strong lower bounds of $n^{\Omega(1)}$ on the k-party communication complexity of Disjointness, when k is a constant. But the new bound decays much faster with k and therefore provides considerably weaker bounds for non-constant k as compared to the one derived earlier, in Section 6.5.1. The reason we present this technique is two-fold. First, it is of independent interest as it yields a new proof of strong bounds on the Disjointness result. In particular, recently Sherstov [She08a] remarked that it is not clear how to extend the method of Shi and Zhu to the multiparty setting. Our extension shows that indeed their method can be modified and extended for multiple parties in a simple fashion. Second, the technique also appears slightly more general than the one presented earlier. It is not clear whether in some context, the second technique may be more convenient to apply.

Both techniques make use of the same duality between the notions of approximability and orthogonality (Lemma 6.19) and the Generalized Discrepancy Method. In fact, they are closely related and we further discuss this relationship in Section 6.6.2.

We start with the formal description of the block-composition method. Consider a real valued function $h : \{1, -1\}^m \to \mathbb{R}$ and a boolean function $q : (\{1, -1\}^s)^{k+1} \to \{1, -1\}$. We naturally view the input space of q as a two dimensional block with $(k+1)$ rows and s columns and we call q the *block function* in the ensuing discussion. Consider a boolean matrix A of dimension $(k+1) \times (ms)$ that we view as made up of m

contiguous blocks, each of dimension $(k+1)\times s$. We define a function $(h\Box q)$ over such boolean matrices that evaluates on its input in the following way: it first applies q to each of the m blocks of the matrix to obtain an m-bit boolean string and then applies h to this string to output its value, i.e. $(h\Box q)(z_1,\ldots,z_m) = h\big(q(z_1),\ldots,q(z_m)\big)$, with each $z_i \in \{1,-1\}^{(k+1)s}$. In this language, functions like GIP and Disjointness are rewritten as (PARITY□AND) and (NOR□AND) respectively, where the inner function AND acts on blocks of dimension $(k+1)\times 1$.

Equivalently, in the context of the k-party communication problem of evaluating $(h\Box q)$, we partition the input matrix A in the obvious way: the $(k+1)$ rows of the matrix are denoted by $x, y^1, \ldots, y^k$ respectively and Player 1 gets x on the forehead, and for $1 \le i \le k$, Player $(i+1)$ gets y^i on the forehead.

We are interested in the question "For a boolean h, what properties of h and q are sufficient to make $(h\Box q)$ have large communication complexity?". This question, in the context of two-player quantum communication, was introduced and investigated in the recent work of Shi and Zhu [SZ07]. They derive tight lower bounds on the two-party quantum communication complexity by using the sophisticated machinery of Hahn polynomials and spectral analysis. However, we do not use these tools in extending the method to the multiparty setting.

6.6.1 Hardness Amplification

Let ν_x, $\nu_1, \ldots, \nu_k$ be probability distributions over sets $I_x, I_1, \ldots, I_k \subset \{1,-1\}^s$. Let ν be the product of these distributions and consider a boolean function q defined over blocks of dimension $(k+1)\times s$. Then, define the $(k+1)$-dimensional *cube measure* of q w.r.t. ν, denoted by $\mathcal{E}_{\nu,k+1}(q)$, as follows:

$$\mathcal{E}_{\nu,k+1}(q) \ \equiv_{\text{def}} \ \mathbb{E}_{y_0^i,y_1^i\sim\nu_i}\left|\mathbb{E}_{x\sim\nu_x}\left(\prod_{u\in\{0,1\}^k} q(x,y_{u_1}^1,\ldots,y_{u_k}^k)\right)\right|.$$

We say that q is *balanced* under ν if

$$\mathbb{E}_{x\sim\nu_x;y^i\sim\nu_i} q(x,y^1\ldots,y^k) = 0.$$

Before we proceed further, let us derive a probability distribution λ over the inputs of a block-composed function $(h\Box g)$, given any distribution for inputs of h and a distribution ν that leaves q balanced.

Proposition 6.23 *Let μ be any distribution over $\{0,1\}^m$. Let $q : \{1,-1\}^{(k+1)s} \to \{1,-1\}$ be a block function that is balanced by a distribution ν over its inputs. Then, the function*

$$\lambda(z_1,\ldots,z_m) = 2^m \times \big((\mu\Box q)(z_1,\ldots,z_m)\big) \times \prod_{i=1}^{m} \nu(z_i)$$

is a probability distribution over the set of $(k+1)\times(ms)$ boolean matrices, where each z_i is a block of dimension $(k+1)\times s$.

Proof: This is true because q is balanced under ν. More precisely,

$$\sum_{z_i\in\{1,-1\}^{(k+1)s}:i\leq m} \lambda(z_1,\ldots,z_m) = \sum_{z_i\in\{1,-1\}^{(k+1)s}:i\leq m} 2^m\times\big((\mu\Box q)(z_1,\ldots,z_m)\big)\times\prod_{i=1}^{m}\nu(z_i).$$

This can be re-written in the following manner. For any $x\in\{1,-1\}^m$, let x_i denote its ith coordinate.

$$\sum_{x\in\{1,-1\}^m} 2^m \sum_{z_i\in\{1,-1\}^{(k+1)s}:q(z_i)=x_i} \mu(x_1,\dots,x_m)\prod_{i=1}^m \nu(z_i)$$

$$= \sum_{x\in\{1,-1\}^m} \left[\mu(x_1,\dots,x_m)2^m \prod_{i=1}^m \left(\sum_{z_i\in\{1,-1\}^{(k+1)s}:q(z_i)=x_i} \nu(z_i)\right)\right]. \qquad (6.26)$$

Since q is balanced under ν, for each i and x, we have

$$\sum_{z_i\in\{1,-1\}^{(k+1)s}:q(z_i)=x_i} \nu(z_i) = \frac{1}{2}.$$

Substituting this in (6.26), and recalling that μ is a distribution on $\{1,-1\}^m$, we get

$$\sum_{z_i\in\{1,-1\}^{(k+1)s}:i\le m} \lambda(z_1,\dots,z_m) = \sum_{x\in\{1,-1\}^m} \mu(x_1,\dots,x_m) = 1.$$

■

Let h be (μ,d)-orthogonal for some distribution μ and integer $d>0$. Further, let q be balanced under a distribution ν such that the cube measure of q w.r.t ν is not too large. The following lemma shows that the discrepancy of $(h\Box q)$ is exponentially small w.r.t. the distribution λ that is generated out of μ and ν according to Proposition 6.23.

Lemma 6.24 (Discrepancy Amplification) *Let $h:\{1,-1\}^m\to\{1,-1\}$ be a (μ,d)-orthogonal function and $q:\{1,-1\}^{(k+1)s}\to\{1,-1\}$ be a block function that is balanced under a product distribution ν. If $\left(\mathcal{E}_{\nu,k+1}(q)\right)^{1/2^k}\le\frac{d}{8em}$, then*

$$disc_{\lambda,k+1}(h\Box q)\le\frac{1}{2^d} \qquad (6.27)$$

where, λ is the probability distribution defined in the following manner:

$$\lambda(z_1, \ldots, z_m) = 2^m \times \big((\mu \Box q)(z_1, \ldots, z_m)\big) \times \prod_{i=1}^{m} \nu(z_i). \tag{6.28}$$

Here, each z_i is a block of dimension $(k+1) \times s$.

Proof: Recall from Section 6.1.1 that every real valued-function over $\{1, -1\}^m$ can be decomposed, via the Fourier transform, in terms of the monomials χ_S, with $S \subseteq [m]$. The main idea in the proof is the following: Define $h\mu(z_1, \ldots, z_m) = h(z_1, \ldots, z_m)\mu(z_1, \ldots, z_m)$. Use the Fourier expansion of the function $h\mu$ to decompose the function $(h\mu \Box q)$ in terms of functions of the form $(\chi_S \Box q)$. Use this decomposition to upper bound the discrepancy of $(h \Box q)$, w.r.t. λ, as the sum of discrepancies of each $(\chi_S \Box q)$, w.r.t. to the distribution that is an m-fold product of ν. Finally, using the cube measure, we show that the discrepancy of each $(\chi_S \Box q)$ decays rapidly with the size of the set S.

Forthwith are the details. Let τ be the characteristic function of any $(k+1)$-wise cylinder intersection. Then, using the definition of λ and discrepancy one gets

$$\mathrm{disc}_{\lambda,\tau}\big(h \Box q\big) = 2^m \Bigg| \sum_{z=(z_1,\ldots,z_m)} \big((h\mu) \Box q\big)(z)\tau(z) \prod_{i=1}^{m} \nu(z_i) \Bigg|.$$

Applying the (μ, d)-orthogonality of h and the triangle inequality, the RHS above simplifies to

$$2^m \Bigg| \sum_{|S| \geq d : S \subseteq [m]} \widehat{h\mu}(S) \sum_{z=(z_1,\ldots,z_m)} \big(\chi_S \Box q\big)(z)\tau(z) \prod_{i=1}^{m} \nu(z_i) \Bigg| \leq$$

$$2^m \sum_{|S| \geq d} \big|\widehat{h\mu}(S)\big| \Bigg| \sum_{z=(z_1,\ldots,z_m)} \big(\chi_S \Box q\big)(z)\tau(z) \prod_{i=1}^{m} \nu(z_i) \Bigg|.$$

It is not hard to verify that, as h is boolean valued and μ is a probability distribution, $\widehat{h\mu}(S) \leq 1/2^m$ for any S. Using this, the above further simplifies to

$$\mathrm{disc}_{\lambda,\tau}\big(h\square q\big) \leq \sum_{|S|\geq d} \left| \mathbb{E}_{z_i\sim\nu}\big(\chi_S\square q\big)(z)\tau(z)\right|.$$

As τ is an arbitrary cylinder intersection,

$$\mathrm{disc}_{\lambda}\big(h\square q\big) \leq \sum_{|S|\geq d} \mathrm{disc}_{\overline{\nu}}\big(\chi_S\square q\big), \tag{6.29}$$

where $\overline{\nu}$ is simply the m-fold product of ν. This completes the first part of the proof. For the second part, we readily estimate the discrepancy of $(\chi_S\square q)$ below in terms of $\mathcal{E}_{\nu,k+1}(q)$. Henceforth, we abuse notation and overload S to also mean the characteristic vector of the set S.

Proposition 6.25 *For any $S \in \{0,1\}^m$,*

$$disc_{\overline{\nu},k+1}\big(\chi_S\square q\big) \leq \big(\mathcal{E}_{\nu,k+1}(q)\big)^{|S|/2^k}. \tag{6.30}$$

Proof: Before we plunge into the calculations, we set some notation. Recall that $x, y^1, \ldots, y^k$ represent the $(k+1)$ rows of the input matrix of $(h\square q)$. Let $x[i]$ and $y^1[i], \ldots, y^k[i]$ represent respectively the portion of these rows that belongs to the ith block z_i of the input matrix, for $1 \leq i \leq m$. In other words, denoting the jth row of the ith block naturally by $z_i[j]$, $x[i] = z_i[1]$ and $y^j[i] = z_i[j+1]$, for $1 \leq i \leq m$ and $1 \leq j \leq k$.

Recall the upper bound on discrepancy provided by the cube measure through Lemma 6.6 in Section 6.1.2. Using the definition of ν and $\overline{\nu}$, and applying equation

(6.2), we proceed as follows:

$$\left(\mathrm{disc}_{\overline{\nu},k+1}\big(\chi_S\Box q\big)\right)^{2^k} \le \mathbb{E}_{y_0^i,y_1^i\sim(\nu_i)^m:1\le i\le k}\left|\mathbb{E}_{x\sim(\nu_x)^m}\prod_{u\in\{0,1\}^k}\big(\chi_S\Box q\big)\big(x,y^1_{u_1},\dots,y^k_{u_k}\big)\right|$$

$$= \mathbb{E}_{y_0^i,y_1^i\sim\nu_i:1\le i\le k}\left|\prod_{j:S_j=1}\mathbb{E}_{x[j]\sim\nu_x}\left[\prod_{u\in\{0,1\}^k} q\big(x[j],y^1_{u_1}[j],\dots,y^k_{u_k}[j]\big)\right]\right|$$

$$= \mathbb{E}_{y_0^i,y_1^i\sim\nu_i}\prod_{j:S_j=1}\left|\mathbb{E}_{x[j]\sim\nu_x}\left[\prod_{u\in\{0,1\}^k} q\big(x[j],y^1_{u_1}[j],\dots,y^k_{u_k}[j]\big)\right]\right|$$

$$= \prod_{j:S_j=1}\mathbb{E}_{y_0^i[j],y_1^i[j]\sim\nu_i:1\le i\le k}\left|\mathbb{E}_{x[j]\sim\nu_x}\left[\prod_{u\in\{0,1\}^k} q\big(x[j],y^1_{u_1}[j],\dots,y^k_{u_k}[j]\big)\right]\right|. \tag{6.31}$$

Applying the definition of $\mathcal{E}_{\nu,k+1}$ to equation (6.31) immediately yields equation (6.30). ∎

Below, we combine Proposition 6.25 and Equation (6.29). Further we drop the subscript of $k+1$ from $\mathcal{E}_{\nu,k+1}$ to avoid clutter.

$$\mathrm{disc}_\lambda\big(h\Box q\big) \le \sum_{j=d}^{m}\binom{m}{j}\big(\mathcal{E}_\nu(q)\big)^{j/2^k}.$$

Substituting the identity $\binom{m}{j} \le \big(\frac{em}{j}\big)^j$, we get

$$\mathrm{disc}_\lambda\big(h\Box q\big) \le \sum_{j=d}^{m}\left(\frac{em}{j}\big(\mathcal{E}_\nu(q)\big)^{1/2^k}\right)^j,$$

whence equation (6.27) readily follows under the condition $\big(\mathcal{E}_\nu(q)\big)^{1/2^k} \le \frac{d}{8em}$ imposed by Lemma 6.24. ∎

6.6.2 Application to Disjointness

We show that the masking scheme that we created in Section 6.2.1 can be viewed as a special case of Block Composition. Consider the following k-wise indexing function: $\mathrm{IN}_k : X \times Y^1 \times \cdots \times Y^k \to \{1, -1\}$ where $X = \{1, -1\}^{\ell^k}$ is the space of k-dimensional boolean arrays with each dimension of size ℓ i.e. an instance of X contains ℓ^k boolean elements. Each $Y^i = [\ell]$ is the space of pointers in the ith dimension of X. On a given input instance $(x, y^1, \ldots, y^k)$, IN_k outputs the value of the element of x jointly indexed by the k pointer variables. The starting point is to observe that the communication tensor of $(\mathrm{NOR}\Box\mathrm{IN}_k)$ is embedded as a sub-tensor of the $(k+1)$-wise Disjointness function. Thus, lower bounding the communication complexity of $(\mathrm{NOR}\Box\mathrm{IN}_k)$ is sufficient for our application. Here, we show that the Discrepancy Amplification Lemma yields interesting lower bounds for $(\mathrm{NOR}\Box\mathrm{IN}_k)$ by choosing the right block size.

As before, we use the Generalized Discrepancy Method. From Paturi's Theorem, we recall that $\deg_{1/3}(NOR) = \theta(\sqrt{m})$. We use the Approximation/Orthogonality Principle of Lemma 6.19 to derive a function g and a distribution μ such that g is (μ, d)-orthogonal. Further, $\mathrm{Corr}_\mu(OR, g)$ is at least 1/3. The Generalized Discrepancy Method prescribes us to upper bound the discrepancy of $(g\Box IN_k)$ to lower bound the communication complexity of $(\mathrm{NOR}\Box\mathrm{IN}_k)$. To that effect, let $\mathcal{U}$ be the uniform distribution over the space of inputs to IN_k. Define λ just as given by equation (6.28) in the Discrepancy Amplification Lemma with $\nu = \mathcal{U}$. Note that $\mathcal{U}$ renders IN_k balanced. In order to apply our Amplification Lemma, we estimate

$\mathcal{E}_{\mathcal{U},k+1}\big(\mathrm{IN}_k\big)$.

$$\mathcal{E}_{\mathcal{U},k+1}(\mathrm{IN}) = \mathbb{E}_{y_0^i, y_1^i \sim \mu_i}\left|\mathbb{E}_x\bigg(\prod_{u\in\{0,1\}^k} \mathrm{IN}_k(x, y_{u_1}^1, \ldots, y_{u_k}^k)\bigg)\right|.$$

It is not hard to verify that the inner expectation over x is one whenever $y_0^i = y_1^i$ for some i and is zero otherwise. Thus, applying the union bound,

$$\mathcal{E}_{\mathcal{U},k+1}(\mathrm{IN}_k) \le \frac{k}{\ell}.$$

Observe that the parameter s, which is the length of a block in the Amplification Lemma, is set to ℓ^k for IN_k. Substituting $s = \ell^k$, one gets $\mathcal{E}_{\mathcal{U},k+1}(\mathrm{IN}) = \frac{k}{s^{1/k}}$. In order to apply the Discrepancy Amplification Lemma, we require

$$\left(\frac{k}{s^{1/k}}\right)^{1/2^k} \le \frac{d}{8em},$$

where d is the approximation degree of the outer function g. The above is satisfied by setting

$$s = k^k\left(\frac{8em}{d}\right)^{k2^k}.$$

Plugging in $d = \theta(\sqrt{m})$ for g and noting that $(g\square\mathrm{IN}_k)$ in this case is over $n = sm$ columns, gives us the bound below:

$$\mathrm{disc}_{\lambda,k+1}\big(g\square\mathrm{IN}_k\big) = O\bigg(2^{-n^{\frac{1}{k2^k+2}}}\bigg).$$

It can be easily verified that $\mathrm{Corr}_\lambda\big(\mathrm{NOR}\square\mathrm{IN}, g\square\mathrm{IN}\big) = \mathrm{Corr}_\mu(NOR, g) \ge 1/3$. Hence, equation (6.23) of the Generalized Discrepancy Method finally yields:

$$R_{k+1}^{1/3}\big(\mathrm{DISJ}\big) \ge R_{k+1}^{1/3}\big(\mathrm{OR}\square\mathrm{IN}\big) = \Omega\bigg(n^{\frac{1}{k2^k+2}}\bigg). \tag{6.32}$$

Note that for constant k, we obtain a bound of $n^{\Omega(1)}$, that is exponentially better than the $\log n$ bound that was the best known bound for Disjointness until very recently. However, it is much weaker than the bound obtained earlier applying the Orthogonality-Discrepancy Lemma. This is despite the fact that in both cases we use identical indexing function over blocks. The reason for it is that in establishing the Discrepancy Amplification Lemma, we are heavily using the triangle inequality without assuming anything about our inside function on blocks. The calculation in Orthogonality-Discrepancy Lemma, on the other hand, proceeds much more carefully taking into account the very special structure of the indexing function.

6.7 Conclusion

We have shown that depth-three AC^0 circuits contain functions that are hard for k-player randomized protocols in a very strong sense. They need to communicate superpolylogarithmic number of bits even when they are required to gain a mere inverse-quasipolynomial advantage over random guessing and $k = o(\log\log n)$. This result, building on the work of Sherstov [She07], exploits a connection between voting degree of a boolean function f and the discrepancy of another function F_k^f that masks f. In the next chapter, we derive important applications of this result to circuit complexity.

Further, we have shown that multiparty randomized protocols cannot compute efficiently functions in depth-two AC^0, when they are required to achieve bounded advantage over random guessing. This has settled a major open question by showing that Disjointness has $n^{\Omega(1)}$ k-party complexity in the bounded error model, if k is a constant. We prove this result in two ways. The first is by extending the pattern-matrix method of Sherstov [She08b] for two-player quantum protocols. The second is by extending the block-composition method of Shi and Zhu [SZ07], also designed for two-player quantum protocols. Both our extensions use the beautiful duality between the notion of approximability of boolean functions by polynomials over reals and the notion of a polynomial being orthogonal to low-order parities. This duality was introduced in the setting of communication complexity by [She08b, SZ07]. Finally, we remark that our extension of the block-composition method to the multiparty setting, answers a recent question raised by Sherstov [She08a].

Beame, Pitassi and Segerlind [BPS05] have shown that strong lower bounds on the randomized multiparty communication complexity of Disjointness results in new separation of proof systems. In this regard, our bounds yield such separations that are not yet known to follow from other techniques. Our bounds on Disjointness also results in the first explicit separation of communication complexity classes BPP_k^{cc} and NP_k^{cc} for $k = o(\log\log n)$. This separation has been recently improved by David, Pitassi and Viola [DPV08], building upon our work.

An interesting direction for future research is to answer the following two questions: (a)Can we find a function in AC^0 that has no efficient protocol of bounded advantage for $\delta \log n$ players for some constant δ? (b)Can we find such a function if we require only inverse-quasipolynomial advantage from protocols? The last question if answered in the positive will have important consequences for depth-three circuits as the discussion in the next chapter shows. Very recently, Beame and Huynh-Ngoc [BHN08] have made progress towards answering the first question. They show a function in AC^0 that has no efficient randomized bounded-error protocols for $\delta\sqrt{\log n}$ players, where δ is a constant less than 1.

CHAPTER 7
Some Consequences for Depth-Three Circuits

In this chapter, we derive some results on depth-three circuits that follow either directly from results in the last chapter on multiparty communication or use very similar ideas.

We recall that understanding the computational power of depth-three circuits made of MAJORITY and MOD counting gates remains open. In particular, we do not know if linear size depth-three circuits comprising only MOD_m gates or comprising only MAJ gates can compute every function in NP. Indeed, proving superlinear lower bounds on the size of such circuits for computing any explicit function is one of the frontiers in the theory of lower bounds. Given this situation, it is pertinent to ask what functions are computable by depth-three circuits in a non-trivial way.

A classical result of Allender [All89] shows that all functions computable by quasipolynomial size AC^0 circuits can be computed by circuits of depth-three and quasipolynomial size and of the following kind: $\mathrm{MAJ} \circ \mathrm{MAJ} \circ \mathrm{MAJ}_{(\log n)^{O(1)}}$, i.e. circuits of depth three having only MAJORITY gates in which the gates at the base layer are restricted to have polylogarithmic fan-in. This result follows almost directly[1] from the result, by Razborov and Smolensky, on the approximability of

[1] Allender showed a uniform version of this theorem, i.e. every uniform AC^0 circuit of quasipolynomial size can be simulated by a uniform depth-three circuit with MAJ

AC^0 circuits by low degree polynomials over finite fields. More surprisingly, the work of Yao [Yao90] and Beigel-Tarui [BT94], making use of ideas in the proof of Toda's Theorem [Tod91], show that such circuits are powerful enough to simulate the strictly bigger class $qACC^0$, i.e. the class of functions that are computable by circuits of constant depth and quasipolynomial size that use MOD_m gates in addition to AND and OR gates, for some fixed integer $m > 1$. The following is intriguing: although the simulation requires these bottom fan-in restricted circuits to be quasipolynomial size, we cannot rule out the much stronger (and stranger) possibility that linear size suffices to simulate the whole of NP.

Håstad and Goldmann [HG91] showed that if such depth three circuits were further restricted to have sub-logarithmic fan-in at the bottom layer, then they cannot simulate ACC^0 in sub-exponential size. This left open the question whether such restricted circuits, even when they have constant fan-in at the bottom, could simulate AC^0 in quasi-polynomial size. In fact until very recently, no super-polynomial lower bounds were known on the size of depth-two circuits of type $\mathrm{MAJ} \circ \mathrm{MAJ}$ for simulating AC^0. Sherstov [She07] recently resolved the depth two question in the negative by analyzing the two-party randomized communication complexity of an appropriately chosen function in AC^0. Håstad and Goldman, on the other hand, invoked the result of Babai, Nisan and Szegedy [BNS92] for the stronger multiparty

gates of quasipolynomial size. This uniform version is not known to follow from the Razborov-Smolensky argument. Allender used ideas from Toda's theorem to obtain his result.

model to show their lower bound on the size of depth three circuits computing the generalized inner product function.

We extended Sherstov's [She07] work in the last chapter to the multiparty model. As a simple consequence of that extension, we prove the following result in this chapter:

Theorem 7.1 *Circuits having a MAJ gate at the output, a middle layer of gates computing arbitrary symmetric functions and a base layer of gates computing any functions of k input variables, i.e. of type $MAJ \circ SYMM \circ ANY_k$, need size at least $exp\left(\Omega\left(\frac{n^{1/(2k+3)}}{(2^{2^k}ek)^k}\right)\right)$ to simulate depth-three AC^0 of linear size. Specifically, if k is a constant (resp. $o(\log \log n)$) then such circuits cannot simulate AC^0 if the top fan-in is subexponential (resp. quasipolynomial).*

In particular, the above shows that Allender's classic construction to simulate AC^0 is reasonably close to being optimal. In fact, Allender's original construction shows that qpoly size circuits of type $\text{MAJ} \circ \text{MOD}_m \circ \text{AND}_{(\log n)^{O(1)}}$ can simulate ACC$^0[p^r]$ (i.e. circuits with MOD$_{p^r}$ gates in addition to AND/OR gates), for every prime p that divides m and any fixed r. A long line of research (see for example [CGT96, Gre99, Gre04, AB01]) seeks to show that such depth-three circuits cannot simulate ACC^0 in quasipoly size. On the other hand, it is commonly believed that such circuits cannot even compute MOD$_q$, if m, q are co-prime.

Recall, from Section 2.1.4, that the Discriminator Lemma implies that obtaining an exponentially small upper bound on the correlation between a function f and any boolean function that is represented by a polynomial of poly-logarithmic degree over $\mathbb{Z}_m$, is enough to prove that f cannot be computed in subexponential size by such depth-three circuits. It is widely conjectured that MOD_q has small correlation with functions represented by low degree polynomials over $\mathbb{Z}_m$, if m and q are co-prime. However for a long time, no good estimates were available even for the correlation between general quadratic polynomials over $\mathbb{Z}_m$ and MOD_q. This state of affairs has been significantly improved by the breakthrough work of Bourgain [Bou05] and Green et.al. [GRS05], although the original problem of separating the class of functions computed by circuits $\mathrm{MAJ} \circ \mathrm{MOD}_m \circ \mathrm{AND}_{(\log n)^{O(1)}}$ of polynomial size from ACC^0 remains wide open. Note that this is unresolved even when m is a prime and the depth-three circuits are of linear size.

In the second part of this chapter, we simplify Bourgain's method [Bou05, GRS05] of estimating the correlation between polynomials of degree d over $\mathbb{Z}_m$ and MOD_q when $(m, q) = 1$. We argue that the notion of discrepancy, suitably modified, can be used conveniently to obtain this estimate. This approach also points out the similarities between the techniques used for estimating cylindrical discrepancy in the communication setting and the techniques used for obtaining bounds on correlation. Additionally, our estimates for correlation are slightly better than previous estimates of [Bou05, GRS05].

Applying the Discriminator Lemma from Section 2.1.4, we obtain the following:

Theorem 7.2 *Any depth-three circuit of type $MAJ \circ MOD_m \circ ANY_k$ requires size $exp\big(\Omega(n/(m2^{m-1})^d)\big)$ to compute MOD_q function, if m, q are co-prime.*

For the special case of $m = 2$, this matches the recent bounds obtained by Viola and Wigderson [VW07a]. It is not known if techniques of [VW07a], based on Gowers norm, can be extended to all m.

7.1 Simulating AC^0 by Depth-Three Circuits

Razborov and Smolensky showed that $\mathrm{ACC}^0[p^k]$ circuits can be well-approximated by low degree polynomials over $\mathbb{Z}_{p^k}$. Let us recall, from Section 2.1.3, their characterization of these circuits:

Lemma 7.3 (Restatement of Theorem 2.17) *Let p be any fixed prime. For each $0 < \epsilon < 1$ and for every circuit C in $ACC^0[p^k]$ of depth d and size s, there exists a distribution U_C over polynomials over $\mathbb{Z}_p$ of degree at most $\left((p^k - 1)(\log(s/\epsilon))\right)^d$, such that for each input x to C, $\Pr_{P \sim U_C}[P(x) \neq C(x)] \leq \epsilon$.*

Fix ϵ in the above characterization to be sufficiently smaller than $1/2$. If we pick t polynomials independently and according to distribution U_C, then we expect ϵt of them to evaluate differently than the circuit C on any fixed input $x \in \{0,1\}^n$. The probability that the number of such erring polynomials exceeds $\frac{1}{2}t$ (in this case they deviate by a lot from the expected number) is very small if the number of polynomials t is suitably large. Indeed, it is not hard to verify, using the Chernoff bound, that there exists a constant c_ϵ depending on ϵ alone such that if we pick $t = c_\epsilon n$ polynomials at random, then for any given x the probability that more than half of them err on a fixed input is less that 2^{-n}. Taking a union bound, the probability that they err on at least one input is less than one. Noting that every polynomial of degree d over $\mathbb{Z}_p$ can be evaluated by a depth-two circuit of type $\mathrm{MOD}_p \circ \mathrm{AND}_d$ of size $O(n^d)$, the probabilistic method implies the following:

Theorem 7.4 *A function computed by any $ACC^0[p^k]$ circuit of size s and depth d can be also computed by a depth-three circuit $MAJ \circ MOD_p \circ AND_t$ of size $O(n^{t+1})$, where $t = O((\log s)^d)$. The top fan-in of such a depth-three circuit is merely linear.*

Note that, by contrast, Theorem 7.1 says that if the bottom fan-in is restricted to $o(\log\log n)$, then the top fan-in of depth-three circuits itself needs to be super-quasipolynomial to simulate AC^0.

7.2 From Communication to Circuits

In this section, we derive Theorem 7.1 from our results on multiparty communication in the last chapter. In order to do so, we recall an established connection between randomized communication complexity of a function f and the size of depth-three circuits needed to compute f.

Fact 7.5 (see [HG91]) *If f is computed by a circuit of type $MAJ \circ SYMM \circ ANY_k$, of size s, then $R_{k+1}^{1/2s}(f) \leq k \log s$.*

Proof: Let $C_1, \ldots, C_t$, $t \leq s$, be the subcircuits feeding into the output MAJ gate in the circuit C for computing f. The $(k+1)$-player protocol first flips a set of coins to randomly select $i \in \{1, \ldots, s\}$. Then it outputs the value of C_i on the input instance. By the definition of a MAJ gate, it is easy to verify that the error probability is bounded by $(1/2 - 1/2s)$.

The proof is completed by showing that each C_i can be evaluated by communicating at most $k. \log s$ bits. The key thing to note is that every ANY_k gate at the base of C_i can be evaluated by at least one of the $k+1$ players with no communication. The players agree beforehand on the set of base gates that each player evaluates. Since the output gate of C_i computes a symmetric function, the $(k+1)$-th player can determine the value of C_i, once the remaining players send the number of base gates that they respectively see evaluating to 1. This clearly takes at most $k \log s$ bits of communication[2]. ■

Armed with this observation, we are ready to prove our main theorem showing that AC^0 does not have efficient simulation by depth-three circuits with restricted bottom fan-in. For this, recall the Minsky-Papert function, denoted by MP, defined as $\mathrm{MP}(x) = \vee_{i=1}^{t} \wedge_{j=1}^{4t^2} x_{i,j}$. This is easily seen to be computable by linear depth-two AC^0. Using the masking scheme defined in Section 6.2.1, we consider the $(k+1)$-wise masked Minsky-Papert function F_{k+1}^{MP}. This masked function, using Fact 6.16, can be computed in depth-three and linear size by AC^0 circuits. On the other hand, Corollary 6.14 (which is a corollary to the Multiparty Degree-Discrepancy Lemma), says that it has large randomized communication complexity even when the advantage over random guessing is small. We have recalled all the necessary facts to finish off the short formal argument proving our main theorem below.

Proof:[Of Theorem 7.1] Let s be the size of any depth-three circuit of bottom fan-in k computing F_{k+1}^{MP}. Then applying Fact 7.5 and Corollary 6.14, we get

$$k \log s \geq R_{k+1}^{1/2s}\big(F_{k+1}^{\mathrm{MP}}\big) \geq \Omega\left(\frac{n^{\frac{1}{2k+3}}}{\big(2^{2^{k+1}/(2k+3)} 2ek\big)^{k}} + \log \frac{1}{2s}\right).$$

This immediately yields our theorem. ■

[2] It is worthwhile to note that this protocol is simultaneous.

7.3 Polynomial Discrepancy

In this section, we show that boolean functions represented by low degree polynomials over $\mathbb{Z}_m$ do not correlate well with MOD_q, if m, q are co-prime. For this, we define the notion of *polynomial discrepancy* of a function.

Let P be any multilinear polynomial of degree d over $\mathbb{Z}_m$ in n variables. Let L_q be the linear polynomial $x_1 + \cdots + x_n$ evaluated over $\mathbb{Z}_q$. Recall that $\mathrm{e}_q(k)$ denotes $\exp\big(2\pi ik/q\big)$, where i is the square-root of -1. Further, let $f : \{0,1\}^n \rightarrow \mathbb{Z}_q$. Consider a distribution μ such that f is almost balanced under μ, i.e. $\Pr_x[f(x) = b] = 1/q + 2^{-\Omega(n)}$. For example, L_q is almost balanced under the uniform distribution for every q. Let $1_{P(x)\equiv a}$ denote the characteristic vector of the set of those points of the cube where polynomial P evaluates to a in $\mathbb{Z}_m$. We define the *mod-m polynomial discrepancy* of f w.r.t. P and $a \in \mathbb{Z}_m$, $b \in \mathbb{Z}_q - \{0\}$ under μ, denoted by $\mathrm{Pdisc}_{\mu,m}^{P,a,b}(f)$, to be the following:

$$\mathrm{Pdisc}_{\mu,m}^{P,a,b}(f) \;=\; \Big|\mathbb{E}_{x\sim\mu}\mathrm{e}_q\big(bf(x)\big)\cdot 1_{P(x)\equiv a}\Big|. \tag{7.1}$$

Note that if f has zero discrepancy, then f evaluates to each element of $\mathbb{Z}_q$ with equal probability over the set of points where P evaluates to a. Intuitively, the higher the discrepancy of f, the more skewed is the behavior of f over the set $1_{P(x)\equiv a}$.

It is interesting to compare the above notion of polynomial discrepancy and discrepancy of cylinder intersections as defined by (4.4) in Chapter 4. Note that in (4.4) f is assumed to be $1/-1$ valued. Noting this, we remark that the two notions are extremely similar and this similarity becomes even clearer if we assume $q = 2$ in

(7.1). In this regard, the degree d of the polynomial has the same role as that of the parameter k in a k-wise cylinder intersection. Further, the role played by polynomial discrepancy in bounding the correlation of a polynomial with a boolean function is very similar to the role played by discrepancy of cylinder intersections in bounding the distributional communication complexity of a boolean function.

The Mod-m, degree-d Polynomial Discrepancy of f under μ, denoted by $\mathrm{Pdisc}_{d,\mu,m}(f)$, is simply $\max\{\mathrm{Pdisc}^{P,a,b}_{\mu,m}(f) | \deg(P) = d, a \in \mathbb{Z}_m \; b \in \mathbb{Z}_q\}$. In this chapter, the default distribution is uniform. Henceforth, we drop the subscript denoting the distribution explicitly.

Our main technical lemma, in this section, is the following :

Lemma 7.6 (Polynomial Discrepancy Lemma) *Let $m, q > 1$ be integers that are co-prime and $d \geq 1$. Then, there exists a constant $\alpha = \alpha(m, q)$, such that the following holds:*

$$Pdisc_{d,m}(L_q) \leq exp\left(- \frac{\alpha n}{(m2^{m-1})^d} \right). \tag{7.2}$$

In words, (7.2) shows that $P^{-1}(a)$, for each a, looks *uniform* to a MOD_q counter i.e. every $L_q^{-1}(b)$ is almost equally represented in the set, provided the size of the set is large compared to the size of the cube. We identify the similarities between the calculation of polynomial discrepancy of the L_q function and the method used by [BNS92] to estimate the cylindrical discrepancy for the generalized inner product function. In both estimates, the key technical ingredient is to raise the sum in question to its appropriate power.

This easily leads to an upper bound of $\exp(-\Omega(n/(m2^{m-1})^d))$ on correlation between the MOD_q function and functions represented by polynomials of degree d over $\mathbb{Z}_m$. In particular, this implies the bound of $\exp(-\Omega(n/4^d))$ for the special case of $m = 2$ that was first reported in the recent work of [VW07a]. Recall the elementary identity for roots of unity: $\sum_{a=0}^{m-1} \mathrm{e}_m(ay) = 1$ if y is a multiple of m and is zero otherwise. We start by re-writing, using complex roots of unity, the quantity $\mathrm{Pdisc}_m^{P,a,b}(L_q)$ for any polynomial P over $\mathbb{Z}_m$ and for any $a \in \mathbb{Z}_m, b \in \mathbb{Z}_q$ as follows:

$$\mathrm{Pdisc}_m^{P,a,b}(L_q) \;=\; \left|\mathbb{E}_x\left[\left(\frac{1}{m}\sum_{\alpha=0}^{m-1} \mathrm{e}_m\big(\alpha(P(x)-a)\big)\right)\mathrm{e}_q\big(b(x_1+\cdots+x_n)\big)\right]\right|. \quad (7.3)$$

Let,

$$S_n^{m,q}(\alpha,b,P) = \mathbb{E}_x\left[\mathrm{e}_m(\alpha P(x))\cdot \mathrm{e}_q\big(b(x_1+\cdots+x_n)\big)\right]. \quad (7.4)$$

Then,

$$\mathrm{Pdisc}_m^{P,a,b}(L_q) \le \frac{1}{m}\sum_{\alpha\in[m]} \big|S_n^{m,q}(\alpha,b,P)\big|. \quad (7.5)$$

It is simple to verify that the Polynomial Discrepancy Lemma gets established by the bound on $|S_n^{m,q}(\alpha,b,P)|$ provided below.

Lemma 7.7 *For each pair of co-prime integers $m, q > 1$ there exists a constant $\beta = \beta(q)$ such that for every polynomial P of degree $d > 0$ over $\mathbb{Z}_m$ and numbers*

$\alpha \in [m]$, $b \in [q] - \{0\}$, *the following holds :*

$$|S_n^{m,q}(\alpha, b, P)| \leq exp\bigg(- \frac{\beta n}{(m2^{m-1})^d}\bigg). \tag{7.6}$$

Before we begin our formal calculations, we remind the reader that a slightly weaker estimate of $|S_n^{m,q}(\alpha, b, P)|$ was first obtained in [Bou05, GRS05]. The case when P is a linear polynomial was essentially dealt with in [CGT96].

Observe that the quantity $S_n^{m,q}$, defined in (7.4), looks very similar to the sum that was obtained in Babai, Nisan and Szegedy [BNS92] to calculate the discrepancy of GIP. There, they were interested in bounding discrepancy of GIP w.r.t. k-cylinder intersections. Here, we are interested in bounding the discrepancy of L_q w.r.t. to a set that is the image of a polynomial. The key idea, introduced in [BNS92], is that squaring the sum is effective in dealing with cylinder intersections. This is something that we adapted to our proof of the Degree-Discrepancy Lemma in the previous chapter. Here, the analogue of the BNS trick will be to raise the sum in (7.4) to its mth power.

In order to further explain the intuition behind our proof of Lemma 7.7, we introduce some definitions and notations. Let $f : \{0,1\}^n \to \mathbb{Z}_m$ be any function. Consider any set $I \subseteq [n]$. Note that each binary vector v of length $|I|$ can be thought of as a partial assignment to the input variables of f by assigning v to the variables in I in a natural way. Let $f^{I(v)}$ be the subfunction of f on variables not indexed in I induced by the partial assignment v to variables indexed in I. For any sequence $Y = \{y_1, \ldots, y_t\}$ having t boolean vectors from $\{0,1\}^n$, let f_Y be the function defined by $f_Y(x) = f(x) + \sum_{i=1}^t f(x \oplus y_i)$, where the sum is taken in $\mathbb{Z}_m$.

Let $I[Y] \subseteq [n]$ be the set of those indices on which every vector in Y is zero and $J[Y]$ be just the complement of $I[Y]$. Then, the following observation will be very useful in our calculation :

Observation 7.8 *Let P be a polynomial of degree d in n variables over Z_m for any $m > 1$. Then, for each sequence Y of $(m-1)$ boolean vectors in $\{0,1\}^n$, the polynomial $P_Y^{J[Y](v)}$ is a polynomial of degree $(d-1)$ in variables from $I[Y]$, for each vector $v \in \{0,1\}^{|J[Y]|}$.*

A point worth mentioning is that, P_Y behaves almost like a *discrete derivative* of the polynomial[3] P.

Proof Sketch:[of Lemma 7.7] We drop the superscript from $S_n^{m,q}$ to avoid clutter in the following discussion. We induce on the degree d of the polynomial. Our Inductive Hypothesis is that there exists a positive real constant $\mu_{d-1} < 1$ such that for all polynomials R of degree at most $d-1$ and for all $n \geq 0$ we have $|S_n(\alpha, b, R)| \leq 2^n \mu_{d-1}^n$. The base case of $d = 0$ is essentially dealt with in Chapter 3, Section 3.2.4. Note that μ_0 depends only on q. Our inductive step yields a relationship between μ_{d-1} and μ_d that also gives us our desired explicit bound of (7.6). As in [Bou05, GRS05], we raise S_n to its mth power. Our point of departure from the earlier techniques, is to write $(S_n)^m$ in a different way.

[3] In the case of $m = 2$, the notion of a discrete derivative appears in several works (see for example [GT05, Sam07]).

$$(S_n)^m \; = \mathbb{E}_{y^1,\dots,y^{m-1}}\mathbb{E}_x\Bigg[\mathrm{e}_m\Bigg(P(x)+\sum_{j=1}^{m-1}P(x\oplus y^j)\Bigg) \times \mathrm{e}_q\Bigg(\sum_{i=1}^{n}x_i+\sum_{k=1}^{m-1}\sum_{i=1}^{n}(x_i\oplus y_i^k)\Bigg)\Bigg] \tag{7.7}$$

Let Y be the sequence of length $m-1$ formed by a given set of vectors $y^1,\dots,y^{m-1}$. We denote by u and v respectively the projection of x to $I[Y]$ and $J[Y]$. Let n_I and n_J be the cardinality of $I[Y]$ and $J[Y]$ (note that $n_I+n_J=n$) . Then, one can verify

$$(7.7)=\mathbb{E}_{y^1,\dots,y^{m-1}}\mathbb{E}_v\Bigg[\mathrm{e}_m\big(Q^{y^1,\dots,y^{m-1}}(v)\big)\mathrm{e}_q(n_J)\mathbb{E}_u\mathrm{e}_m\big(P_Y^{I[Y](v)}(u)\big)\mathrm{e}_q\big(m\sum_{i=1}^{n_I}u_i\big)\Bigg] \tag{7.8}$$

where $Q^{y^1,\dots,y^{m-1}}$ is some polynomial that is determined by $y^1,\dots,y^{m-1}$ and polynomial P.

The key thing to note is that Observation 7.8 implies that $P_Y^{I[Y](v)}$ is a polynomial of degree at most $d-1$ over u for every sequence $Y=y^1,\dots,y^{m-1}$ and every vector v. Hence, the inside sum of (7.8) over the variable u can be estimated using our inductive hypothesis. Note that raising to the mth power in (7.7) has achieved a degree reduction of the polynomial in a manner that is very reminiscent of how [BNS92] does dimension reduction of cylinder intersections in the proof of their Lemma 2.5.

The rest of the calculation proceeds exactly as in Green et. al. [GRS05], which again is very similar to the series of final steps in the proof of Lemma 2.5 in [BNS92]. We repeat them for the sake of self-containment.

Using the triangle inequality, the binomial theorem and noting that the number of sequences Y for which $|I_Y| = k$ is exactly $\binom{n}{k}(2^{m-1}-1)^{n-k}$, we get

$$|S_n|^m \leq \sum_{k=0}^{n} \binom{n}{k} (2^{m-1}-1)^{n-k} 2^{n-k} 2^k \mu_{d-1}^k \; = \; 2^{nm}\left(1 - \frac{1-\mu_{d-1}}{2^{m-1}}\right)^n. \tag{7.9}$$

Taking the mth root of both sides of (7.9), using the inequality $(1-x)^{1/m} \leq 1 - x/m$ if $0 \leq x < 1$ and $m > 1$ after rearranging, we obtain

$$1 - \mu_d \geq \frac{1-\mu_{d-1}}{m2^{m-1}} \geq \frac{1-\mu_0}{\left(m2^{m-1}\right)^d}. \tag{7.10}$$

Substituting $\beta = 1 - \mu_0$, one gets $\mu_d \leq \exp\left(-\frac{\beta}{(m2^{m-1})^d}\right)$. This immediately yields (7.6) in Lemma 7.7. ■

Consider $A = L_q^{-1}(1)$ and $B = L_q^{-1}(0)$. For any $a \in \mathbb{Z}_m$ and any polynomial P over $\mathbb{Z}_m$, let $P^{-1}(a)$ be the subset of the cube where P evaluates to a. Then using the estimate on the mod-m polynomial discrepancy of L_q, the following uniformity Lemma gets easily established.

Lemma 7.9 (Polynomial Uniformity Lemma) *For any polynomial P of degree d over $\mathbb{Z}_m$, $a \in \mathbb{Z}_m$ and $b \in \{0, \ldots, q-1\}$, the following holds:*

$$\left| \Pr_x \left[P(x) \equiv a \wedge x \in M_{n,q}(b)\right] - \frac{1}{q} \Pr_x \left[P(x) \equiv a\right] \right| \leq \frac{q-1}{q} exp\left(- \frac{\beta n}{(m2^{m-1})^d}\right).$$

Proof:

$$\Pr_x \left[P(x) \equiv a \wedge x \in M_{n,q}(b)\right] = \mathbb{E}_x \left[\left(\frac{1}{q} \sum_{\beta=0}^{q-1} \mathrm{e}_q\big(\beta(x_1 + \cdots + x_n - b)\big) \right) \cdot 1_{P(x)\equiv a} \right].$$

Expanding the sum inside the first multiplicand and treating the case of $\beta = 0$ separately, the RHS above simplifies to the following:

$$\frac{1}{q}\mathbb{E}_x\left[1_{P(x)\equiv a}\right] + \frac{1}{q}\sum_{\beta\neq 0} \mathrm{e}_q(-\beta b)\mathbb{E}_x\left[\mathrm{e}_q\left(\beta L_q\right)\cdot 1_{P(x)\equiv a}\right].$$

Identifying the first term above as just $\frac{1}{q}\Pr_x[P(x)\equiv a]$, we get the following

$$\left|\Pr_x\left[P(x)\equiv a \wedge x\in M_{n,q}(b)\right] - \frac{1}{q}\Pr_x\left[P(x)\equiv a\right]\right| \leq \frac{1}{q}\sum_{\beta\neq 0}\mathrm{Pdisc}_m^{P,a,\beta}\left(L_q\right).$$

Plugging in the estimate from the Polynomial Discrepancy Lemma finishes the proof.

■

Choose $A = L_q^{-1}(1)$ and $B = L_q^{-1}(0)$. The proof of Theorem 7.2 follows quite easily now using the Discriminator Lemma and the Polynomial Uniformity Lemma in exactly the same fashion as we derived Theorem 3.5 from the Linear Uniformity Lemma in Section 3.2.5 of Chapter 3.

CHAPTER 8
Conclusion

We have explored the models of constant-depth boolean circuits, ‘Number on the Forehead’ multiparty communication protocols and representation of boolean functions by multivariate polynomials over commutative rings. While each of them is fascinating in its own right, the three models are not unrelated. Indeed, it has been known for more than fifteen years that there are deep connections between them. In this thesis, we unravel fresh connections that we exploit crucially to make progress on questions that naturally arise in each model. For instance, in Chapter 3, our bounds on the size of ACC^0 circuits directly results from bounds on degree in a new model of polynomial representation of boolean functions. In Chapter 6, we utilize the notion of threshold and approximation degree of boolean functions to make significant progress in multiparty communication complexity. Finally in Chapter 7, we find a new kind of interplay between polynomials and communication: *ideas* (as opposed to concrete results) used in analyzing the communication complexity of a function are re-usable for obtaining lower bounds on the degree needed by polynomials to approximate boolean functions.

The depth and richness of these models are further suggested by the diversity of the mathematical tools employed to analyze them. For example, Chapter 3 makes heavy use of ideas from algebraic combinatorics, probabilistic method, Fourier analysis and exponential sums. Chapter 5 uses tools from error-correcting codes and

Ramsey theory. Chapter 6 draws on approximation theory and linear programming duality. Dually, the computational view on classical objects like polynomials raises new questions that are of independent mathematical interest: "how much degree is needed to represent a simple function like AND/MAJORITY/MOD_ℓ in a natural model of representation by polynomials?". Such questions are fundamental and the fact that polynomials have been under investigation for a long time, makes one feel that they ought to have been answered. Yet, not only have they not been answered, making progress on them have required sophisticated arguments. In the first part of Chapter 3, we explored this theme. We defined a notion of representation by polynomials that generalizes earlier notions described in the literature. Proving lower bounds on the degree of such representations entailed a combination of arguments from the combinatorial work of Tardos and Barrington [TB98] and the more algebraic work of Green [Gre00]. Further strong progress about these questions is very likely to result in progress in mainstream mathematics.

In this context, it is worthwhile to note that a new theory of low degree polynomials over finite fields is being developed, among others, by mathematicians Gowers [Gow01], Green[1] and Tao [GT05, GT07]. It is quite interesting to study the relationship between the point of view on polynomials used in this thesis and the above works which draw motivation from additive combinatorics. There already has been exchange of ideas among the two points of view. For instance, Lovett, Meshulam

[1] Earlier, we referred to works by the computer scientist Fred Green [Gre00, Gre99]. The Green referred to here, is the combinatorial number theorist Ben Green.

and Samorodnitsky [LMS08] and independently Green and Tao [GT07], disproved recently an important conjecture in additive combinatorics, called the Gowers Inverse Conjecture, using ideas from the work of Alon and Beigel [AB01]. The work of Alon and Beigel, on the other hand, was motivated by the question of determining the correlation between low degree polynomials over $\mathbb{Z}_m$ and MOD_ℓ. Recall that this question is explored in our work (in continuation of a long line of research) in the second part of Chapter 7. Indeed, the interaction between the theory of computation and pure mathematics is truly a two-way process. The theory of low degree polynomials is a key area where further meaningful exchange between the two disciplines is very likely to continue.

While reaching the goal of proving strong lower bounds in the model of constant-depth circuits with modular gates is still distant, our work suggests some intermediate steps that should be attainable more easily. Let us outline a few such steps. Analyzing a single layer of MOD_m gates is an obvious direction to pursue. In Chapter 3, we proved that a sublinear number of them at the base is too weak to compute the MOD_ℓ or AND function. This weakness is essentially information theoretic. In other words, $C \circ \mathrm{MOD}_m$ cannot compute such functions, no matter how powerful the circuit C is, if the MOD_m layer is sublinear in size. What bounds on the size of MOD_m layer can be proved if we limit the power of C? If C is a single AND,OR or MAJ gate, then our results (this is also known from the work of [KP94, Gre99]) imply that the MOD_m layer must have exponential size for the circuit to compute MOD_ℓ. On the other hand, if C is a generalized MOD_m gate or an AC^0 circuit of polynomial size, no non-linear lower bounds are known on the size of the MOD_m layer. Making progress on this frontier should be within reach and is likely to shed new light on how to approach more general circuits. We believe that the use of exponential sums in analyzing circuits should be of further use here. While we have used exponential sums on their own, an interesting direction to pursue is to see if they can be combined with existing tools to approximate AC^0 circuits, for proving new lower bounds.

Several areas in theoretical computer science, the theory of constant-depth circuits in particular, have immensely benefitted from the study of the 'Number on the Forehead' model of multiparty communication. Starting with the work of Håstad and Goldmann [HG91], other works like [Gro92, RW93] have used the strong lower bounds of Babai, Nisan and Szegedy [BNS92] on the multiparty communication complexity of a function to make progress in circuit complexity. The technique of Babai et.al. was the only known method for proving such strong lower bounds. Before our work, it only yielded lower bounds for those functions whose computation involved modular counting in one form or the other. Consequently, it could not be directly applied to yield bounds for a function in AC^0. Building on the work of Sherstov [She07], we have rectified this problem in Chapter 6 to yield strong lower bounds on the communication complexity of functions in AC^0. This has resulted in a new application to circuit complexity: depth-three circuits comprising MAJ gates with small bottom fan-in cannot efficiently compute even functions in AC^0. This makes important progress in understanding the limitations of a natural subclass of TC^0.

The most powerful known application of the multiparty model to circuit complexity comes from proving lower bounds in the presence of a polylogarithmic number of players. Although this seems a distant goal, it is worth noting that analyzing simultaneous protocols is enough for this application. Our work suggests that new structure can be discovered even analyzing such protocols for constant number of players. We initiated such a study in Chapter 5 and discovered a surprising phenomenon. The presence of a neutral letter in a language takes away a lot of the power of the multiparty model if the players are allowed to communicate constant number of bits. This has been crucially used further in the work of Lautemann, Tesson and Thérien [LTT06]. Does a similar phenomenon still occur when more communication is allowed? What can be said about the structure of languages that can be recognized by randomized protocols in constant communication? Investigations of such questions are likely to yield further insight into the model.

In the second part of Chapter 6, we made substantial progress in understanding the communication complexity of the Disjointness function for a constant number of players. Apart from its application to other areas, this generated an important new technique for the multiparty model: the Generalized Discrepancy Method. Our technique has been improved very recently by the interesting work of Beame and Huynh-Ngoc [BHN08]. However, even their improvement, does not yield better bounds for Disjointness for constant number of players. Our bound for Disjointness is not known to be tight even for three players. It remains interesting to determine if linear lower bounds continue to hold for Disjointness with a constant number of players. On a different note, Disjointness is an example of a function with low non-deterministic communication complexity but high randomized communication complexity. Can we exhibit an explicit function that has the reverse property? This is a natural question regarding the relationship between randomness and non-determinism. Further, making progress on the question, almost surely, will generate new techniques as all known ones for the multiparty model end up proving lower bounds for randomized protocols.

References

[AB01] N. Alon and R. Beigel. Lower bounds for approximations by low degree polynomials over $\mathbb{Z}_m$. In *16th Annual IEEE Conference on Computational Complexity*, pages 184–187. IEEE Computer Society, 2001.

[AB09] S. Arora and B. Barak. *Computational complexity: A modern approach*. Cambridge University Press, 2009. draft avail.at: http://www.cs.princeton.edu/theory/index.php/Compbook/Draft.

[ABFR94] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):1–14, 1994.

[AG94] E. Allender and V. Gore. A uniform circuit lower bound for the permanent. *SIAM J. Computing*, 23:1026–1049, 1994.

[AHT07] M. Agrawal, T. M. Hoang, and T. Thierauf. The polynomially bounded perfect matching is in NC^2. In *24th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 489–499, Aachen, 2007.

[AIK04] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. In *45th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 166–175, 2004.

[Ajt83] M. Ajtai. $\sum_1^1$ formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, 1983.

[AKS04] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[All89] E. Allender. A note on the power of threshold circuits. In *30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 580–584. IEEE Computer Society, 1989.

[Amb96] A. Ambainis. Upper bounds on multiparty communication complexity of shifts. In *13th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, number 1046 in LNCS, pages 631–642. Springer, 1996.

[AS04] S. Aaronson and Y. Shi. Quantum lower bound for the collision problem. *J.ACM*, 51(4):595–605, 2004.

[AW93] E. Allender and K. W. Wagner. Counting hierarchies: polynomial time and constant depth circuits. In *Current Trends in Theoretical Computer Science*, volume 40, pages 469–483. World Scientific Press, 1993.

[AW08] S. Aaronson and A. Wigderson. Algebrization: a new barrier in complexity theory. In *40th Annual Symposium on Theory of Computing (STOC)*, 2008.

[Bar86] D. A. M. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In *18th Annual Symposium on Theory of Computing (STOC)*, pages 1–5, 1986.

[Bar92] D. A. M. Barrington. Some problems involving Razborov-Smolensky polynomials. In *Boolean function complexity*, volume 169 of *London Math.Soc.Lec.Note.*, pages 109–128. Cambridge University Press, Durham, 1990, 1992.

[BBR94] D. A. M. Barrington, R. Beigel, and S. Rudich. Representing boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:367–382, 1994.

[BCH86] P. Beame, S. Cook, and J. Hoover. Log depth circuits for division and related problems. *SIAM J. Computing*, 15:994–1003, 1986.

[BCW98] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *30th ACM Annual Symposium on Theory of Computing (STOC)*, pages 63–68, 1998.

[BDPW07] P. Beame, M. David, T. Pitassi, and P. Woelfel. Separating deterministic from non-deterministic NOF multiparty communication complexity. In *34th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 134–145, 2007.

[Bea94] P. Beame. A switching lemma primer. Technical Report UW-CSE-95-07-01, Department of computer science and engineering, University of Washington, November, 1994.

[Bei93] R. Beigel. The polynomial method in circuit complexity. In *Structure in Complexity Theory*, pages 82–95, 1993.

[Bei94] R. Beigel. When do extra majority gates help? Polylog(n) majority gates are equivalent to one. *Computational Complexity*, 4:314–324, 1994.

[BFS86] László Babai, Péter Frankl, and János Simon. Complexity classes in communication complexity theory. In *Proc. 27th IEEE Symp. on Foundations of Comp. Sci. (FOCS)*, pages 337–347, 1986.

[BGS75] T. Baker, J. Gill, and R. Solovay. Relativizations of the $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$ question. *SIAM J. Computing*, 4:431–442, 1975.

[BHN08] P. Beame and D. Huynh-Ngoc. Multiparty communication complexity of AC^0. Technical Report TR08-061, Electronic Colloquium on Computational Complexity, 2008.

[BIL$^+$05] D. A. M. Barrington, N. Immerman, C. Lautemann, N. Schweikardt, and D. Thérien. First order expressibility of languages with neutral letters or: The Crane Beach conjecture. *J.Computer.System.Sciences*, 70(2):101–127, 2005.

[BNS92] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Computer and System Sciences.*, 45(2):204–232, 1992.

[Bou05] J. Bourgain. Estimates of certain exponential sums arising in complexity theory. *C.R.Acad.Sci.Paris*, Ser I 340(9):627–631, 2005.

[BPS] P. Beame, T. Pitassi, and N. Segerlind. Lower bounds for Lovász-Schrijver systems and beyond follow from multiparty communication complexity.

[BPS05] P. Beame, T. Pitassi, and N. Segerlind. Lower bounds for Lovász-Schrijver systems and beyond follow from multiparty communication complexity. In *32nd International Colloquium on Automata, Languages and Programming (ICALP)*, number 3580 in LNCS, pages 1176–1188. Springer, 2005.

[BPSW06] P. Beame, T. Pitassi, N. Segerlind, and A. Wigderson. A strong direct product theorem for corruption and the multiparty communication complexity of Disjointness. *J. Computational Complexity*, 15(4):391–432, 2006.

[BRS91a] R. Beigel, N. Reingold, and D. Spielman. The perceptron strikes back. In *Sructure in Complexity Theory*, pages 286–291, 1991.

[BRS91b] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. In *Twenty-third Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–9, 1991.

[BS94] D. A. M. Barrington and H. Straubing. Complex polynomials and circuit lower bounds for modular counting. *Computational Complexity*, 4:325–338, 1994.

[BS95] D. A. M. Barrington and H. Straubing. Superlinear lower bounds for bounded-width branching programs. *J.Computer.System.Sciences*, 50(3):374–381, 1995.

[BS99] D. A. M. Barrington and H. Straubing. Lower bounds for modular counting by circuits with modular gates. *Computational Complexity*, 8(3):258–272, 1999.

[BSSV00] P. Beame, M. Saks, X. Sun, and E. Vee. Super-linear time-space tradeoff lower bounds for randomized computation. In *41st IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000.

[BST90] D. A. M. Barrington, H. Straubing, and D. Thérien. Non-uniform automata over groups. *Information and Computation*, 89(2):109–132, 1990.

[BT94] R. Beigel and J. Tarui. On ACC. *Computational Complexity*, 4:350–356, 1994.

[BT06] A. Bogdanov and L. Trevisan. Average-case complexity. In *Foundations and Trends in Theoretical Computer Science*, volume 2. Now, 2006.

[BV02] P. Beame and E. Vee. Time-space tradeoffs multiparty communication complexity and nearest neighbor problems. In *34th Annual Symposium on Theory of Computing (STOC)*, pages 688–697. ACM, 2002.

[BYJKS04] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *J.Comput.Syst.Sci.*, volume 68, pages 702–732, 2004.

[CA08] A. Chattopadhyay and A. Ada. Multiparty communication complexity of Disjointness. *Electronic Colloquium on Computational Complexity (ECCC)*, TR08-002, 2008.

[Cau96] H. Caussinus. A note on a theorem of Barrington, Straubing and Thérien. *Inf.Process.Lett.*, 58(1):31–33, 1996.

[CCM08] A. Chakrabarti, G. Cormode, and A. McGregor. Robust lower bounds for communication and stream computation. In *40th ACM Annual Symposium on Theory of Computing (STOC)*, pages 641–650, 2008.

[CFL83] A. Chandra, M. Furst, and R. Lipton. Multi-party protocols. In *15th Annual Symposium on Theory of Computing (STOC)*, pages 94–99. ACM, 1983.

[CG85] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. In *26th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 429–442, 1985.

[CGPT06] A. Chattopadhyay, N. Goyal, P. Pudlák, and D. Thérien. Lower bounds for circuits with MOD_m gates. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 709–718, 2006.

[CGT96] J.Y. Cai, F. Green, and T. Thierauf. On the correlation of symmetric functions. *Mathematical systems theory*, 29(3):245–258, 1996.

[CH05] A. Chattopadhyay and K. A. Hansen. Lower bounds for circuits with few modular and symmetric gates. In *32nd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 994–1005, 2005.

[Cha07a] A. Chakrabarti. Lower bounds for multi-player pointer jumping. In *22nd Annual IEEE Conference on Computational Complexity*, pages 33–45. IEEE Computer Society, 2007.

[Cha07b] A. Chattopadhyay. Discrepancy and the power of bottom fan-in in depth-three circuits. In *48th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 449–458, 2007.

[Cho08] T. Chow. Almost-natural proofs. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2008. to appear.

[CKK$^+$07] A. Chattopadhyay, A. Krebs, M. Koucký, M. Szegedy, P. Tesson, and D. Thérien. Languages with bounded multiparty communication complexity. In *24th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 500–511, 2007.

[CKS03] A. Chakrabarti, S. Khot, and X. Sun. Near-optimal lower bounds on the multi-party communication complexity of Set-Disjointness. In *IEEE Conference on Computational Complexity*, pages 107–117, 2003.

[CKSU05] H. Cohn, R. Kleinberg, B. Szegedy, and C. Umans. Group theoretic algorithms for matrix multiplication. In *46th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 379–388, 2005.

[CSWY01] A. Chakrabarti, Y. Shi, A. Wirth, and A. C. C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 270–278, 2001.

[DPV08] M. David, T. Pitassi, and E. Viola. Improved separations between nondeterministic and randomized multiparty communication. In *12th International Workshop on Randomization and Computation (RANDOM)*, 2008. to appear.

[FG05] J. Ford and A. Gál. Hadamard tensors and lower bounds on multiparty communication complexity. In *32nd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1163–1175, 2005.

[FKPS85] R. Fagin, M. Klawe, N. Pippenger, and L. Stockmeyer. Bounded depth, polynomial size circuits for symmetric functions. *Theor. Comput. Sci.*, 36:239–250, 1985.

[FSS84] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits and the polynomial time hierarchy. *Math. Systems Theory*, 17:13–27, 1984.

[GGK08] W. I. Gasarch, J. Glenn, and C. P. Kruskal. Finding large 3-free sets I: The small n case. *J.Comput.Syst.Sci.*, 74(4):628–655, 2008.

[Gow01] T. Gowers. A new proof of Szemeredi's theorem. *Geometric and Functional Analysis*, 11(3):465–558, 2001.

[GP08] D. Gavinsky and P. Pudlák. Exponential separation of quantum and classical non-interactive mulit-party communication complexity. In *IEEE Conference on Computational Complexity*, pages 332–339, 2008.

[Gre91] F. Green. An oracle separating ⊕P from $\mathrm{PP}^{\mathrm{ph}}$. *Information Processing Letters*, 37(3):149–153, 1991.

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

[Gre00] F. Green. A complex-number fourier technique for lower bounds on the $\mathrm{MOD}-m$ degree. *Computational Complexity*, 9:16–38, 2000.

[Gre04] F. Green. The correlation between parity and quadratic polynomials mod 3. *J.Computer. Systems. Sciences*, 69(1):28–44, 2004.

[Gro92] V. Grolmusz. Separating the communication complexities of MOD m and MOD p circuits. In *33rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 278–287. IEEE Computer Society, 1992.

[Gro94a] V. Grolmusz. The BNS lower bound for multi-party protocols in nearly optimal. *Information and Computation*, 112(1):51–54, 1994.

[Gro94b] V. Grolmusz. A weight-size trade-off for circuits and MOD m gates. In *26th Annual Symposium on Theory of Computing (STOC)*, pages 68–74. ACM, 1994.

[Gro98] V. Grolmusz. A degree-decreasing lemma for MOD_p-MOD_m circuits. In *25th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1443 of *Lecture Notes in Computer Science*, pages 215–222. Springer, 1998.

[GRS90] R. L. Graham, B. Rotschild, and J. H. Spencer. *Ramsey Theory.* Discrete Mathematics. Wiley Interscience, New York, second edition, 1990.

[GRS05] F. Green, A. Roy, and H. Straubing. Bounds on an exponential sum arising in boolean circuit complexity. *C.R.Acad.Sci.Paris*, Ser I 341:279–282, 2005.

[GT00] V. Grolmusz and G. Tardos. Lower bounds for (MOD-p-MOD-m) circuits. *SIAM J. Computing*, 29(4):1209–1222, 2000.

[GT05] B. Green and T. Tao. An inverse theorem for the Gower's $\mathcal{U}^3$ norm. Technical report, 2005. arXiv.org:math/0503014.

[GT07] B. Green and T. Tao. The distribution of polynomials over finite fields, with applications to the Gower's Norm. Submitted, 2007.

[HAB02] W. Hesse, E. Allender, and D. A. M. Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *J. Computer and System Sciences*, 65:695–716, 2002.

[Han06a] K. A. Hansen. Lower bounds for circuits with few modular gates using exponential sums. Technical Report TR06-079, Electronic Colloquium on Computational Complexity, 2006.

[Han06b] K. A. Hansen. On modular counting with polynomials. In *IEEE Conference on Computational Complexity*, pages 202–212, 2006.

[Hås86] J. Håstad. *Computational limitations on small depth circuits.* PhD thesis, MIT, 1986.

[HG91] J. Håstad and M. Goldmann. On the power of small depth threshold circuits. *Computational Complexity*, 1:113–129, 1991.

[HM04] K. A. Hansen and P. B. Miltersen. Some meet-in-the-middle circuit lower bounds. In *29th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 334–345, 2004.

[HMP+93] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *J.Computer.System.Sciences*, 46(2):129–154, 1993.

[HU79] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison Wesley, 1979.

[JKS03] T. S. Jayram, Ravi Kumar, and D. Sivakumar. Two applications of information complexity. In *35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 673–682, 2003.

[JRS03] R. Jain, J. Radhakrishnan, and P. Sen. A lower bound for the bounded round quantum communication complexity of Set Disjointness. In *44th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 220–229, 2003.

[KKL88] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *29th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 68–80, 1988.

[Kla01] H. Klauck. Lower bounds for quantum communication complexity. In *42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 288–297, 2001.

[Kla07] H. Klauck. Lower bounds for quantum communication complexity. *SIAM J. Computing*, 37(1):20–46, 2007.

[Kli02] A. R. Klivans. *A complexity theoretic approach to learning.* PhD thesis, MIT, 2002.

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[KP94] M. Krause and P. Pudlák. On the computational power of depth 2 circuits with threshold and modulo gates. In *26th Annual Symposium on Theory of Computing (STOC)*, pages 48–57. ACM, 1994.

[KRW95] M. Karchmer, R. Raz, and A. Wigderson. On proving super-logarithmic depth lower bounds via the direct sum in communication complexity. *J. Computational Complexity*, 5:191–204, 1995.

[KS92] B. Kalyanasundaram and G. Schnitger. The probabilistic communication complexity of set intersection. *SIAM J.Discrete Math*, 5(4):545–557, 1992.

[KW88] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proc. 20^{th} ACM Annual Symposium on Theory of Computing (STOC)*, pages 539–550, 1988.

[KW91] M. Krause and S. Waack. Variation ranks of communication matrices and lower bounds of depth-two circuits having symmetric gates with unbounded fan-in. In *32nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 777–782, 1991.

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *J. ACM*, 40(3):607–620, 1993.

[LMS08] S. Lovett, R. Meshulam, and A. Samorodnitsky. Inverse conjecture for the Gowers Norm is false. In *40th ACM Annual Symposium on the Theory of Computing (STOC)*, pages 547–556. ACM, 2008.

[LR01] O. Lachisch and R. Raz. Explicit lower bound of $4.5n - o(n)$ for boolean circuits. In *33rd ACM Symposium on Theory of Computing (STOC)*, pages 399–408, 2001.

[LS08] T. Lee and A. Shraibman. Disjointness is hard in the multi-party number-on-the-forehead model. In *23rd Annual IEEE Conference on Computational Complexity*, 2008.

[LTT06] C. Lautemann, P. Tesson, and D. Thérien. An algebraic point of view on the Crane Beach Property. In *20th International Workshop on Computer Science Logic (CSL)*, pages 426–440, 2006.

[LU97] W. Lenhart and C. Umans. Hamiltonian cycles in solid grid graphs. In *38th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 496–505, 1997.

[MP88] M. Minsky and S. Papert. *Perceptrons:expanded edition.* MIT Press, Cambridge, MA, USA, 1988.

[MPT91] P. McKenzie, P. Péladeau, and D. Thérien. NC^1: The automata-theoretic viewpoint. *Computational Complexity*, 1:330–359, 1991.

[New91] I. Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67–71, 1991.

[NS94] N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994.

[Ols69a] J. E. Olson. A combinatorial problem on finite abelian groups, I. *J. Number Theory*, 1:8–10, 1969.

[Ols69b] J. E. Olson. A combinatorial problem on finite abelian groups, II. *J. Number Theory*, 1:195–199, 1969.

[OS03] R. O'Donnell and R. Servedio. Extremal properties of polynomial threshold functions. In *18th Annual IEEE Conference on Computational Complexity*, pages 3–12. IEEE, 2003.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing, 1994.

[Pat92] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *24th Annual Symposium on the Theory of Computing (STOC)*, pages 468–474, 1992.

[PF77] N. Pippenger and M.J. Fischer. Relationships among complexity measures. Technical Report RC-6569, IBM, Yorktown Heights, 1977.

[PT88] P. Péladeau and D. Thérien. Sur les langages reconnus par des groupes nilpotents. *C.R. Acad. des Sci. Paris Sér. I Math.*, 306(2):93–95, 1988. English translation by A. Russell and S. Russell appears as TR01-040 of ECCC.

[Pud06] P. Pudlák. 2006.

[Raz86] A. A. Razborov. Lower bounds for the monotone complexity of boolean functions. In *Proceeding of the ICM*, pages 1478–1487, Berkeley, California, 1986.

[Raz87] A. A. Razborov. Lower bounds on the size of bounded-depth networks over a complete basis with logical addition. In *Math. Notes of the Acad. of Sci. of USSR*, volume 41, pages 333–338. 1987.

[Raz89] A. A. Razborov. On the method of approximations. In *21st Annual Symposium on Theory of Computing (STOC)*, pages 167–176, 1989.

[Raz90] A. A. Razborov. On the distributional complexity of Disjointness. In *17th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 249–253, 1990.

[Raz00] R. Raz. The BNS-Chung criterion for multi-party communication complexity. *J. Computational Complexity*, 9(2):113–122, 2000.

[Raz03] A. A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya:Mathematics*, 67(1):145–159, 2003.

[Rei87] J. Reif. On Threshold circuits and polynomial computation. In *2nd Annual IEEE Structure in Complexity Theory Conference*, pages 118–123, 1987.

[Rei05] O. Reingold. Undirected ST-connectivity in Logspace. In *37th ACM Annual Symposium on Theory of Computing (STOC)*, pages 376–385, 2005.

[RM97] R. Raz and P. McKenzie. Separation of the monotone *NC* hierarchy. In *Proc. 38^{th}th IEEE FOCS*, 1997.

[RR97] A. A. Razborov and S. Rudich. Natural proofs. *J.Comput.Syst.Sci.*, 55(1):24–35, 1997.

[RS07] A. Roy and H. Straubing. Definability of languages by generalized first-order formulas over n^+. *SIAM J. Computing*, 37(2):502–521, 2007.

[RW91] P. Ragde and A. Wigderson. Linear-size constant-depth polylog-threshold circuits. *Information Processing Letters*, 39(3):143–146, 1991.

[RW92] R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. *J.ACM*, 39:736–744, 1992.

[RW93] A. A. Razborov and A. Wigderson. $n^{\Omega(\log n)}$ lower bounds on the size of depth-three Threshold circuits with AND gates at the bottom. *Information Processing Letters*, 45(6):303–307, 1993.

[Sam07] A. Samorodnitsky. Low degree tests at large distances. In *39th Annual Symposium on Theory of Computing (STOC)*, pages 506–515. ACM, 2007.

[She07] A. Sherstov. Separating AC^0 from depth-2 Majority circuits. In *39th Annual Symposium on Theory of Computing (STOC)*, pages 294–301, 2007.

[She08a] A. Sherstov. Communication lower bounds using dual polynomials. *Bulletin of the European Association of Theoretical Computer Science*, 95:59–93, 2008.

[She08b] A. Sherstov. The pattern matrix method for lower bounds on quantum communication. In *40th Annual Symposium on Theory of Computing (STOC)*, 2008.

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *19th Symposium on Theory of Computing (STOC)*, pages 77–82, 1987.

[Smo90] R. Smolensky. On interpolation by analytic functions with special properties and some weak lower bounds on the size of circuits with symmetric gates. In *31st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 628–631, 1990.

[Spa08] R. Spalek. A dual polynomial for OR. *arXiv:0803.4516v1 [cs.CC]*, 2008.

[ST06] H. Straubing and D. Thérien. A note on MOD_p-MOD_m-circuits. *Theory of Computing Systems*, 39(5):699–706, 2006.

[SZ07] Y. Shi and Y. Zhu. The quantum communication complexity of block-composed functions. 2007.

[Sze93] M. Szegedy. Functions with bounded symmetric communication complexity, programs over commutative monoids and ACC. *J. Comput. Syst. Sci.*, 47(3):405–423, 1993.

[TB98] G. Tardos and D. A. M. Barrington. A lower bound on the MOD 6 degree of the OR function. *Computational Complexity*, 7(2):99–108, 1998.

[Tes03] P. Tesson. *Computational complexity questions relating to finite monoids and semigroups.* PhD thesis, McGill University, 2003.

[Thé94] D. Thérien. Circuits constructed with MOD_q gates cannot compute "And" in sublinear size. *Computational Complexity*, 4:383–388, 1994.

[Tod91] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Computing*, 20(5):865–877, 1991.

[Tsa96] S. C. Tsai. Lower bounds on representing boolean functions as polynomials in $\mathbb{Z}_m$. *SIAM J. Discrete Math*, 9:55–62, 1996.

[Vaz85] U. V. Vazirani. Towards a strong communication complexity, or generating quasi-random sequences from two communicating slightly random sources. In *17th Annual Symposium on Theory of Computing (STOC)*, pages 366–378. ACM, 1985.

[Vol99] H. Vollmer. *Introduction to circuit complexity.* Springer-Verlag New York, 1999.

[VW07a] E. Viola and A. Wigderson. Norms, XOR Lemmas, and lower bounds for GF(2) polynomials and multiparty protocols. In *22nd Annual IEEE Conference on Computational Complexity*, pages 141–154. IEEE Computer Society, 2007.

[VW07b] E. Viola and A. Wigderson. One-way multi-party communication lower bound for pointer jumping with applications. In *48th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 427–437, 2007.

[WWY90] I. Wegener, N. Wurm, and S.Z. Yi. Symmetric functions in AC^0 can be computed in constant depth with very small size. In *15th Mathematical Foundations of Computer Science (MFCS)*, pages 523–529, 1990.

[Yao79] A. C. C. Yao. Some complexity questions related to distributive computing. In *11th Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.

[Yao83] A. C. C. Yao. Lower bounds by probabilistic arguments. In *24th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 420–428, 1983.

[Yao85] A. C. C. Yao. Separating the polynomial-time hierarchy by oracles. In *26th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1–10, 1985.

[Yao90] A. C. C. Yao. On ACC and Threshold circuits. In *37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 619–627. IEEE Computer Society, 1990.

Index