Computing with Polynomials over Composites

A Thesis
Presented to
The Academic Faculty

by

Parikshit Gopalan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Algorithms, Combinatorics and Optimization
Georgia Institute of Technology
August 2006

Computing with Polynomials over Composites

Approved by:

Professor Richard J. Lipton,
College of Computing,
Georgia Institute of Technology., Advisor

Professor Robin Thomas,
School of Mathematics,
Georgia Institute of Technology.

Professor Subhash Khot,
College of Computing,
Georgia Institute of Technology.

Professor Saugata Basu,
School of Mathematics,
Georgia Institute of Technology.

Professor Dana Randall,
College of Computing,
Georgia Institute of Technology.

Professor Prasad Tetali,
School of Mathematics,
Georgia Institute of Technology.

Date Approved: 26^{th} June 2006

To my parents.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dick Lipton for all his encouragement and support over the years. Dick was a tremendous source of new problems and ideas. His extensive knowledge of several areas of computer science was a great asset. He always encouraged me to work on any topic I found interesting, and I think this has made me a better researcher.

I was fortunate to have the chance to work with Subhash Khot. Subhash has a deep understanding of several areas, a great taste in problems, and the amazing ability to make everything look crystal clear. He is an inspiring person to work with and has helped me broaden my reserach horizons.

In the course of my stay at Georgia Tech, I have had the chance to interact with almost all the faculty in the theory group, and I would like to thank them all. I would especially like to thank Dana Randall and Eric Vigoda, for their advice on matters outside theory and their help during the application process. A special thank you to Dana for sitting through endless practice talks. I benifitted greatly from interacting with people from the School of Mathematics and the ACO program. I would especially like to thank Saugata Basu and Ernie Croot, who were always ready to help me with all things mathematical.

I would like to thank all my friends at Georgia Tech for making this a memorable stay, especially Aranyak, Vangelis, Amin and Nikhil. I also thank my friends and collaborators at IBM Almaden where I spent a very enjoyable summer: T.S. Jayram, Phokion Kolaitis, Ravi Kumar, Robi Krauthgamer and Elitza Maneva. As an undergraduate at IIT Bombay, I was fortunate to have many truly inspirational teachers who were an important factor in my deciding to take up a research career. I would especially like to thank Milind Sohoni and Sundar Vishwanathan.

I would like to thank my parents for encouraging me to continue the *family business*. It is no surpise to me that they are the people who are most excited about my becoming a Doctor. I cannot adequately express my gratitude to them for their unwavering support

and confidence in me over the years. I also thank my sister Preeti, who encouragement and advice, especially the latter, were always abundantly available.

Finally, I want to thank Nayantara for our five memorable years together in Atlanta, for all the fun times we had and the great theorems that we have proved. This is undoubtedly the part I will treasure most about my Ph.D experience.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

In the last twenty years, algebraic techniques have been applied with great success to several areas in theoretical computer science. However, for many problems involving modular counting, there is a huge gap in our understanding depending on whether the modulus is prime or composite. A prime example is the problem of showing lower bounds for circuits with Mod gates in circuit complexity. Proof techniques that work well for primes break down over composites. Moreover, in some cases, the problem for composites turns out to be very different from the prime case. Making progress on these problems seems to require a better understanding of polynomials over composites. In this thesis, we address some such *prime versus composite* problems from computational complexity, algorithms and combinatorics, and the surprising connections between them.

We consider the complexity-theoretic problem of computing Boolean functions using polynomials modulo composites. We show that symmetric polynomials can viewed as simultaneous communication protocols. This equivalence allows us to use techniques from communication complexity and number theory to prove degree bounds. We use these techniques to give the first tight degree bounds for a number of Boolean functions.

We consider the combinatorial problem of explicit construction of Ramsey graphs. We present a simple construction of such graphs using polynomials modulo composites. This approach gives a unifying view of many known constructions, and explains why they all achieve the same bound. We show that certain approaches to this problem cannot give better bounds.

Finally, we consider the algorithmic problem of interpolation for polynomials modulo composites. We present the first query-efficient algorithms for interpolation and learning under a distribution. These results rely on some new structural results about such polynomials.

CHAPTER I

INTRODUCTION

The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic ...

C.F. Gauss.

Disquisitiones Arithmeticae (1801)

Computational problems regarding prime and composite numbers are among the oldest and most well-studied problems in mathematics and computer science. In addition to being fundamental in nature, primality-testing and factoring have assumed even greater significance in the light of modern cryptography, where several commonly used cryptosystems rely on assumptions about the complexity of the latter problem. However, the importance of these problems was understood long before the notion of efficient computation was even formalized as the above quotation by Gauss illustrates. Gauss was perhaps the first to point out that one might be able to identify composites without explicitly factoring them, or in the language of computer science, to point out that primality and factoring might have different computational complexity.

One of the great algorithmic advancements of the 1970s was the discovery of randomized algorithms for primality-testing due to Solovay-Strassen [72] and Rabin [67], which ushered in the systematic use of randomization in computer science. More recently, a major algorithmic milestone of this decade was the deterministic primality testing algorithm of Agrawal, Kayal and Saxena that proved that primality-testing is in fact in P [2]. Thus, while the search for more efficient algorithms continues, one can regard primality-testing as a solved problem from a complexity-theoretic viewpoint.

The complexity of factoring however, is far from resolved. The fastest known algorithm

for factoring numbers, the Number-Field-Sieve runs in time $2^{O(\log^{\frac{1}{3}} n)}$ [28]. Many modern-day public-key cryptosystems including the RSA and Rabin cryptosystems rely on the assumption that factoring is intractable. Given our current knowledge of lower bounds however, proving this statement seems to be out of our reach.

In addition to these fundamental questions, the differences between primes and composites have surfaced in several areas of computer science and discrete mathematics, sometimes in rather unexpected contexts. We begin with a couple of simple examples of such problems. The first is from Boolean function complexity.

Problem 1.1 *A polynomial $P(X_1, \ldots, X_n)$ over $\mathbb{Z}_m$ represents the* OR *function on n variables modulo m if for $(x_1, \ldots, x_n) \in \{0,1\}^n$, it satisfies*

$$\begin{aligned} P(0, \ldots, 0) &\equiv 0 \bmod m \\ P(x_1, \ldots, x_n) &\not\equiv 0 \bmod m \quad \textit{if } (x_1, \ldots, x_n) \neq (0, \ldots, 0) \end{aligned}$$

What is the minimum degree polynomial representing the OR *function modulo m?*

For this problem, we regard m as a constant and are interested in the minimum degree as a function of n. It is fairly easy to show by using linear algebra that when m is a prime, any polynomial satisfying these conditions must have degree $\Omega(n)$. With some additional effort, one can prove a similar bound when m is a prime power. A natural question is what happens for m composite. There was a conjecture due to Barrington that an $\Omega(n)$ lower bound should hold in this case too [14]. Surprisingly, however this conjecture was disproved by Barrington, Beigel and Rudich who showed that the OR function can be represented by polynomials having degree $O(n^{\frac{1}{t}})$ where t is the number of prime divisors of m [15].

Their construction gives a symmetric polynomial representing the OR function. They show a matching $\Omega(n^{\frac{1}{t}})$ lower bound for representations by symmetric polynomials. This raises a natural question: can asymmetric polynomials help us represent the OR function with low degree? This question is wide open, the best lower bound till date is $\Omega(\log n)$ due to Barrington and Tardos [74].

Our second example is a problem from combinatorics.

Problem 1.2 *OddTown is a town of n people. The residents of OddTown like to form clubs (which are subsets of $[n]$ for our purposes). But the laws of OddTown state that the size of each club must be odd, whereas the intersection of any two clubs must be even. How many such clubs exist in OddTown?*

An easy bound of n follows by observing that each resident can form a club of one, and this collection is consistent with the laws of the town. A more surprising result is that this is the best that they can do. This is indeed surprising since if the law were changed to restrict both the club sizes and their intersections to be even, then the number of possible clubs jumps to $2^{\frac{n}{2}}$.

A natural question is what happens if we insist that all club sizes are 0 mod m but their intersections are non-zero mod m. We will refer to such families of clubs as set systems with restricted intersections mod m. When $m = p$ is prime, a classical result of Deza, Frankl and Singhi, referred to as the *modular RCW theorem* (RCW stands for Ray-Chaudhuri and Wilson) implies that there can be no more than $O(n^p)$ clubs [10]. For fixed p, this gives a polynomial upper bound in terms of n. This is non-trivial given that in all, there are 2^n possible clubs in OddTown. We will show that a similar bound holds for prime powers. A polynomial upper bound was conjectured for composite m and this problem was open for a while [10]. Surprisingly, Grolmusz disproved the conjecture by constructing a super-polynomial sized set system with restricted intersection modulo m, for all m having at least two prime divisors [43]. The main ingredients in his construction were the low-degree polynomials representing the OR function modulo m discovered by Barrington *et al.* [15].

Let us point out some common features shared by the two problems above. The techniques that work in the prime case, which rely on linear algebra and dimension arguments no longer work over composites. Moreover in both problems, the composite case is very different from the prime case. Thus it is not merely a failure of proof techniques that is keeping us from making progress; it is not clear what the right answer is modulo composites.

In this thesis we address a variety of such prime versus composite questions from computational complexity, algorithms and combinatorics and explore the connections between them. We begin our study of these problems by framing them in the right context.

1.1 Prime versus Composite Problems

One of the major successes in the last two decades has been the successful application of algebraic techniques to a variety of different areas across computer science and discrete mathematics. Major breakthroughs in areas such as circuit complexity [68, 70], algorithmic coding theory [45, 46], algorithmic derandomization [2] and probabilistic proof checking [8, 9] were achieved via extensive use of algebra and polynomial based methods. Similarly in combinatorics, the linear-algebra method and the polynomial method have been applied with great success to several problems [4, 10].

At a high level, many of these results utilize properties of low degree polynomials over fields (usually $\mathbb{Z}_p$) and dimension based arguments. As an illustration, we consider the OddTown problem discussed earlier. We view the incidence vector for each club as a vector in n dimensions over $\mathbb{Z}_2$. A moment's thought reveals that the vectors for various clubs must be linearly independent, which implies the desired bound.

This hints at some of the problems one faces in trying to prove such a bound for composites, the machinery of finite fields and linear algebra is no longer at our disposal. Similarly, arguments that involve properties of low-degree polynomials typically use the fact that over a field, such a polynomial cannot have too many zeroes. This is again something that fails over composites (take $X^k \equiv 0 \bmod 2^k$ for example). This difference often results in a huge gap in our understanding of problems that in some way involve *modular counting*, where we wish to solve some problem for general moduli m, but are typically not able to progress beyond the prime or prime-power case.

Perhaps the central problem of this kind, which suggests limitations to our understanding of modular counting is proving lower bounds for circuits with AND, OR and MOD gates. When all the MOD gates involved are MOD-p gates for a fixed prime p, we have excellent lower bounds via the algebraic techniques of Razborov and Smolensky [68, 70]. The simplest class of circuits that we our lower-bound techniques fail, is when all the gates involved are MOD-6 gates. Indeed, we do not know how to show a super-linear lower bound on the circuit size, even for NP-complete problems. This is one of the frontier open problems in the circuit lower bounds approach to the P $\neq$ NP problem.

The circuit lower bounds of Razborov and Smolensky are obtained by considering a simple algebraic computational model: computing Boolean functions by multivariate polynomials over $\mathbb{Z}_p$. They show an upper bound proving that any function which can be computed using small circuits with MOD-p gates can be computed with low-degree polynomials over $\mathbb{Z}_p$. They then find functions that cannot be computed by low-degree polynomials. This suggests a natural step towards circuit lower bounds: prove degree lower bounds for computing functions using polynomials over $\mathbb{Z}_m$. While we do not know if every circuit can be computed by a low-degree polynomial, it is definitely true that low-degree polynomials are small circuits. Even this problem is still wide open. Surprisingly, a stumbling block appears to be the fact that low-degree polynomials over $\mathbb{Z}_m$ are *more powerful* than their counterparts over $\mathbb{Z}_p$. We now know non-trivial upper bounds for many functions which one might have believed are hard (meaning that they require high degree) based on the prime power case. Indeed, while one certainly expects NP-complete problems to require large circuit-size with MOD-6 gates, it is entirely possible that the simple candidate functions which work for the prime case are no longer hard.

The OddTown theorem and its extensions to general moduli are well studied problems in extremal set theory [10]. They have connections to many other combinatorial problems, one of which is the construction of *low-rank co-diagonal matrices*. Here the question is, given an $n \times n$ matrix where all the diagonal entries are 0 modulo m but the off-diagonal entries are non-zero modulo m, how low can its rank be? Again we think of m as a fixed constant and wish to show bounds as a function of n. One can obtain a large matrix with low rank from a set system with restricted intersection modulo m.

Both these problems are closely linked to a problem of Erdős from 1947, that of explicit Ramsey graph construction. Here the problem is to construct a large graph with no clique or independent set of size k. In his seminal paper introducing the probabilistic method in combinatorics, Erdős showed that there exist such graphs with as many as $2^{\frac{k}{2}}$ vertices [29]. Indeed, he shows that a *random* graph has this property. He posed the problem of explicitly constructing a graph i.e giving a polynomial time algorithm to compute its adjacency matrix. Known constructions fall well short of the probabilistic bound. This is

one of the central open problems in the area of explicit combinatorial constructions. It is not clear at first sight, what constructing Ramsey graphs has to do with primes and composites. However, it is known that set systems with restricted intersections and low rank co-diagonal matrices modulo composites can be used to construct Ramsey graphs [10, 42]. Further, we will see in this thesis that many of the known Ramsey constructions can be unified under the umbrella of polynomials over $\mathbb{Z}_m$ computing Boolean functions.

There are several well-studied algorithmic prime versus composite problems in algebra and number theory, where the goal is to understand the computational complexity of a certain task for various moduli. These include fundamental questions in algorithmic algebra such as root-finding, polynomial factorization and interpolation over $\mathbb{Z}_m$. The complexity of root-finding and polynomial factorization is well understood for general moduli m. Both problems are tractable for prime powers using the famous Hensel lifting algorithm [28], and as hard as integer-factoring for general m [66, 69]. The fact that finding square-roots modulo m is as hard as factoring underlies the security of the Rabin cryptosystem [66].

The problem of polynomial interpolation is to reconstruct a polynomial based on its evaluations. The problem of interpolation over $\mathbb{Z}_m$ is implicit in many of the complexity-theoretic and combinatorial problems discussed above , which are concerned with the lowest degree polynomial that satisfies certain conditions. We will address the problem of interpolation modulo m for general m, with goal of designing algorithms that require only a few evaluations to reconstruct the polynomial correctly. This problem is thoroughly understood over $\mathbb{Z}_p$ for p prime, going back to the work of Newton and Lagrange. Not much is known about the composite case. The main difference between primes and composites in this setting is that for composites, one can hope to reconstruct the polynomial correctly without knowing its evaluations at every point in $\mathbb{Z}_m$, unlike the prime case. Thus, it seems that polynomial interpolation should be easier for composite m, in contrast to the problems of root-finding and polynomial factorization.

1.2 Polynomials over Composites

For many of the prime versus composite problems discussed in the last section, the reason why proof techniques from the prime case do not carry over to the composite case is that polynomials over composites behave differently. Familiar properties like a degree d polynomial can have only d roots cease to hold. Further, the machinery of finite fields and dimension-based arguments is no longer at our disposal. However, the goal of this thesis is to show that polynomials over composites are rich in structure, and that this structure can be exploited to gain new insight into many of these problems. This structure however is very different from the prime case and exploiting it requires techniques from several diverse areas including algebra, number theory, combinatorics and communication complexity.

Further, polynomials over composites serve as a bridge between the various prime versus composite problems arising in computational complexity, algorithms and combinatorics and reveal some surprising connections between them. As an example, let us take the problem of explicit construction of Ramsey graphs. Grolmusz's construction of a super-polynomial size set system with restricted intersections modulo m uses low degree polynomials over $\mathbb{Z}_m$ for computing the OR function. These polynomials were discovered by Barrington, Beigel and Rudich in the context of computing Boolean functions using polynomials [15]. Grolmusz's construction in turn immediately gives a Ramsey graph construction with parameters that were the best known at the time of discovery. In this thesis, we will build on this connection and show that many of the known Ramsey constructions can be viewed as coming from polynomials over composites. This view allows us to take insights from complexity theory and use them to shed light on problems in combinatorics and vice versa. We will show that Ramsey graphs based on symmetric polynomials cannot improve on already known bounds. This *lower-bound* result is suggested by similar results that are known in computational complexity. The techniques used in proving this however are very different, and they build on the structural properties of polynomials modulo prime powers. These properties were in turn discovered in the context of designing efficient algorithms for polynomial interpolation modulo prime powers. Also, by viewing the well-known Ramsey construction of Frankl and Wilson [31] in this framework, we discover new low-degree polynomials computing the OR

function.

1.3 Our Contributions

In this thesis, we explore some prime versus composite questions arising from computational complexity, combinatorics and algorithmic algebra and the connections between them.

1.3.1 Computational Complexity

We study the problem of computing or representing Boolean functions using polynomials over $\mathbb{Z}_m$. We focus on representing symmetric Boolean functions by symmetric polynomials over $\mathbb{Z}_m$. Our main conceptual contribution is an equivalence between computing Boolean functions by symmetric polynomials modulo m and computing the functions by certain one-round simultaneous communication protocols.

For polynomials over $\mathbb{Z}_6$, these protocols involve two players and a referee who are trying to compute a certain Boolean-valued function f of some number $w \in \{0, \ldots, n\}$ (for instance is $w \geq 2$?). The first player is given a few low-order digits of the w in base 2 and the other is given a few low-order digits of the weight in base 3. Each player sends a message to the referee: the first player sends a number in $\mathbb{Z}_2$, and the second sends a number in $\mathbb{Z}_3$. The referee tries to compute f based on these messages. The number of bits that the players each have to read for the protocol to succeed corresponds to the minimum degree symmetric polynomial that represents a Boolean function on $\{0,1\}^n$ derived from f.

This equivalence allows us to show degree lower bounds by using techniques from communication complexity. We show lower bounds of $\Omega(n)$ on symmetric polynomials weakly representing classes of Mod and Threshold functions. Previously the best known lower bound for such representations of any function modulo m when m has t prime factors was $\Omega(n^{\frac{1}{t}})$ [15]. The equivalence also allows us to use results from number theory to prove degree bounds. We show that proving bounds on the degree of symmetric polynomials strongly representing the Threshold functions is equivalent to counting the number of solutions to certain Diophantine equations. We use this to show an upper bound of $O(nk)^{\frac{1}{2}+\varepsilon}$ for the Threshold-k function assuming the *abc* conjecture from number theory. We show the same bound unconditionally for k constant. Prior to this, non-trivial upper bounds were

known only for the OR function and its shifts [15]. We show an almost tight lower bound of $\Omega(nk)^{\frac{1}{2}}$, improving the previously known bound of $\Omega(\max(k, \sqrt{n}))$ [76].

Based on these results and subsequent work on this problem by Hansen [47], we now have a good understanding of representations by symmetric polynomials and fairly tight degree bounds for such representations of most natural Boolean functions. The outstanding open problem in this area is whether asymmetric polynomials can give better (i.e lower degree) representations of symmetric Boolean functions than symmetric polynomials. While there are no known examples of symmetric Boolean functions where asymmetry does help, there are no degree lower bounds better than $\Omega(\log n)$ known for any Boolean function.

1.3.2 Algorithms

The problem of polynomial interpolation is to reconstruct a polynomial based on its valuations on a set of inputs I. We consider the problem over $\mathbb{Z}_m$ when m is composite. We ask the question: *Given $I \subseteq \mathbb{Z}_m$, how many evaluations of a polynomial at points in I are required to compute its value at every point in I?* Surprisingly for composite m, this number can vary exponentially between $\log |I|$ and $|I|$ in contrast to the prime case where $|I|$ evaluations are necessary. While this minimization problem is NP-hard, we give an efficient algorithm of query complexity within a factor t of the optimum where t is the number of prime factors of m. In fact the guarantee is slightly stronger. When the algorithm terminates, it produces a factorization of m into $t' \leq t$ relatively prime factors. The approximation factor is in fact bounded by t'. Thus input sets I which force the algorithm to make several queries must also reveal the factorization of m. We use our interpolation algorithm to design algorithms for zero-testing and distributional learning of polynomials over $\mathbb{Z}_m$. In some cases, we get an exponential improvement over known algorithms in query complexity and running time.

Our main technical contribution is the notion of an interpolating set for I which is a subset S of I such that a polynomial which is 0 over S must be 0 at every point in I. Any interpolation algorithm needs to query an interpolating set for I. Our query-efficient algorithms are obtained by constructing interpolating sets whose size is close to optimal.

Interpolating sets modulo prime powers have rich algebraic and combinatorial structure which we study in detail, these properties are also useful in analyzing our algorithm. In proving these properties, we make crucial use of the fact that the underlying space is in fact an ultrametric space (metrics where the following strengthening of the triangle inequality holds: $d(x,y) \leq \max(d(x,z), d(y,z))$. We show that many algebraic properties of polynomials can be reinterpreted as geometric properties of ultrametric spaces. Further, the proof of these properties for general ultrametric spaces follows directly from the proof for polynomials modulo prime powers.

1.3.3 Combinatorics

We consider the problem of explicit construction of Ramsey graphs or graphs with no large clique or independent set. Constructing Ramsey graphs was known to be related to polynomial representations of Boolean functions; Grolmusz showed that a low degree representation for the OR function modulo m can be used to construct set systems with restricted intersections modulo m, from which one can construct explicit Ramsey graphs [43].

We generalize the above relation by proposing a new framework. We propose a new definition of OR representations: a pair of polynomials represent the OR function if the union of their zero sets contains all points in $\{0,1\}^n$ except the origin. We give a simple construction of a Ramsey graph using such polynomials. Furthermore, we show that all the known algebraic constructions, ones to due to Frankl-Wilson [31], Grolmusz [43] and Alon [5] are captured by this framework; they can all be derived from various OR representations of degree $O(\sqrt{n})$ based on symmetric polynomials. This gives a simple explanation for why all these constructions achieve the same bound. It simplifies the construction of Grolmusz, and relates it to those of Frankl-Wilson and Alon, which look very different at first. It places the constructions of Alon and Frankl-Wilson, which were originally derived using different techniques, in the context of Boolean function representations, and raises the possibility of getting better constructions from low degree representations.

By our view, all the constructions naturally extend to the problem of constructing

multicolor Ramsey graphs, where the goal is to t-color the edges of the complete graph so that there are no large monochromatic cliques. Such an extension was not known previously for the Frankl-Wilson construction. Another consequence of our construction is improved bounds for set systems with restricted intersections modulo prime powers.

Thus the barrier to better Ramsey constructions through such algebraic methods appears to be the construction of lower degree representations. We show that the question of symmetry versus asymmetry from Boolean function complexity applies to Ramsey graph constructions as well. We show that better Ramsey graphs cannot be obtained using symmetric polynomials. Thus to obtain better graphs using such algebraic techniques, one has to use asymmetric polynomials. Proving this bound for our new definition of OR representations calls for some new algebraic techniques, which build on structural properties of interpolating sets that were discovered in the context of polynomial interpolation.

Organization of This Thesis

The results in this thesis on polynomials representations of Boolean functions are based on joint work with Nayantara Bhatnagar and Richard J. Lipton. Extended abstracts of this work appeared as *Symmetric Polynomials over $\mathbb{Z}_m$ and Simultaneous Communication Protocols* in FOCS'03 [20], and *The Degree of Threshold Mod 6 and Diophantine Equations* which appeared as a technical report on ECCC in 2004 [21]. A combined full version of both these papers appeared in JCSS [22]. An extended abstract of the work on polynomial interpolation appeared in SODA'06 under the title *Query-Efficient Algorithms for Polynomial Interpolation over Composites* [36], the full version of this paper is currently under review. An extended abstract of the work on Ramsey graph constructions will appear in CCC'06 under the title *Constructing Ramsey Graphs from Boolean Function Representations* [35].

This thesis is organized as follows. We begin with the problem of computing Boolean functions by polynomials from computational complexity. In Chapter 2, we re-cap some basic results about polynomials computing Boolean functions, and present applications to Computational Learning and Circuit Complexity. Most of these results are well known, except Section 2.2. We consider representations over $\mathbb{Z}_m$ in Chapter 3. We prove the

equivalence between polynomials over $\mathbb{Z}_m$ and communication protocols. We then consider Threshold functions and prove tight degree bounds using techniques from number theory. We turn to the Algorithmic problem of polynomial interpolation over $\mathbb{Z}_m$ in Chapter 4, We consider the Combinatorial problem of explicit Ramsey graph construction in Chapter 5. We have attempted to keep each chapter fairly self-contained.

CHAPTER II

AN ALGEBRAIC VIEW OF BOOLEAN FUNCTIONS

Representations of Boolean functions as polynomials over various rings such as $\mathbb{R}, \mathbb{Z}_p$ and $\mathbb{Z}_m$ have been well studied in computer science starting with the work of Minsky and Papert [60]. Polynomials provide a simple and natural algebraic model of computation, with the degree serving as a natural complexity measure. In addition, this study has proved useful for applications in computational complexity, computational learning and combinatorics.

Representations of Boolean functions as polynomials over $\mathbb{Z}_p$ were used by Razborov and Smolensky [68, 70] in proving lower bounds for constant-depth circuits with MOD-p gates. Such representations are also useful in learning-theoretic scenarios, an example which we will discuss later in this chapter is an algorithm due to Mossel, O'Donnell and Servedio for learning a class of Boolean functions known as Juntas [61]. Representations of polynomials over $\mathbb{Z}_m$ for m composite are studied as a first step towards showing lower bounds for circuits with MOD-m gates, this study was initiated by the work of Barrington, Beigel and Rudich [15]. Such representations have surprising applications to problems in combinatorics [35, 42, 43] which we will explore further in chapter 5. Our main focus in this thesis will be on such representations.

The Fourier representation of Boolean functions is essentially a representation of Boolean functions by real polynomials. Fourier analysis has proved to be a very powerful tool for several applications in computer science, including hardness of approximation and computational learning. Two examples of important breakthroughs in these areas that rely on Fourier analytic methods are Håstad's optimal inapproximability result for Max-3-SAT [49], and the low-degree algorithm of Linial, Mansour and Nisan for learning constant depth circuits under the uniform distribution in quasi-polynomial time [59]. We will cover the basic facts about Fourier analysis in this chapter, for a comprehensive study of Fourier analytic methods and their applications, we refer the reader to O'Donnell's thesis [65].

Another kind of representation of Boolean functions by real polynomials, which were first studied by Minsky and Papert [60] are sign-representations, which are also called Polynomial Threshold Functions (PTFs) or Perceptrons. Here the value of the Boolean function at a point is decided by the sign of the representing polynomial evaluated at that point. Such representations are widely used in learning theory, and the best known algorithms for learning several classes of Boolean functions including DNFs use such representations [54, 55]. However, we will not focus on such representations in this thesis.

In this chapter, we will survey the basic concepts about exact representations of Boolean functions as polynomials. Most of these results were known previously, unless otherwise indicated. Throughout, we will try to present results in the most general setting possible, such as a general commutative ring R or a field $\mathbb{F}$. We first introduce polynomial representations and prove their existence and uniqueness. We then consider symmetric Boolean functions, which are the focus of much of this thesis. We develop tools for proving degree lower bounds for such functions over $\mathbb{Z}_p$, using Lucas' Theorem about binomial coefficients modulo p. Our treatment follows that of Bhatnagar *et al.* [20]. We then introduce the Fourier representations of Boolean functions. We prove a basic result due to Nisan and Szegedy which gives a lower bound on the degree of arbitrary Boolean functions over $\mathbb{R}$ using Fourier analysis.

We then present a couple of sample applications of polynomial representations from computational learning and circuit complexity. We consider the problem of learning k-juntas (functions that depend on only k out of their n input variables) under the uniform distribution. We present an algorithm due to Mossel *et al.* [61] which is the best algorithm known for this problem. The analysis presented here is somewhat simpler than the original analysis. We consider the problem of proving lower bounds for $\mathrm{AC}^0[3]$, the class of polynomial-size, constant-depth circuits with AND, OR and MOD-3 gates. We prove a result due to Smolensky showing that the PARITY function cannot be computed by such circuits.

2.1 Computing Boolean functions by Polynomials

We use $\mathbf{X} = (X_1, \ldots, X_n)$ to denote a vector of variables and $\mathbf{x} = (x_1, \ldots, x_n)$ denote a vector of constants. Let $\mathbb{F}$ be a field (containing $0, 1$). We will study polynomials in $\mathbb{F}[\mathbf{X}]$ and the functions that they compute on $\{0,1\}^n$. Given a monomial $\prod_i X_i^{d_i}$, we define its total degree to be $\sum_i d_i$ and its degree in the variable X_i to be d_i. We define the degree of the 0 polynomial to be ∞.

Definition 2.1 *A polynomial $P(\mathbf{X}) \in \mathbb{F}[X_1, \ldots, X_n]$ is a multilinear polynomial if the degree of each variable in every monomial is at most* 1.

One can equivalently define multilinear polynomials as all polynomials of the form

$$P(\mathbf{X}) = \sum_{S \subseteq [n]} c_S \prod_{i \in S} X_i \qquad c_S \in F.$$

Fact 2.1 *The space of a multilinear polynomials in $\mathbf{X}$ forms a vector space of dimension 2^n over $\mathbb{F}$. A basis for this vector space is given by the monomials $\prod_{i \in S} X_i$ for all sets $S \subseteq [n]$.*

Over the Boolean hypercube, we can restrict our attention to multilinear functions without loss of generality.

Proposition 2.2 *Given a polynomial $Q(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$, there is a multilinear polynomial $P(\mathbf{X})$ so that $P(\mathbf{x}) = Q(\mathbf{x})$ for every point $\mathbf{x} \in \{0,1\}^n$.*

PROOF: We are interested in the values taken by the polynomial $Q(\mathbf{X})$ at points in $\{0,1\}^n$ where the relation $X_i^d = X_i$ holds for $d \geq 2$. Hence, we can replace X_i^d by X_i for $d \geq 2$ without changing the value of the polynomial over $\{0,1\}^n$. This replacement results in a multilinear polynomial $P(\mathbf{X})$ that agrees with $Q(\mathbf{X})$ at every point in $\{0,1\}^n$. □

The above replacement procedure is sometimes referred to as *multilinearization* [10].

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function defined on the n-dimensional Boolean hypercube. There is a natural way to associate a multilinear polynomial over $\mathbb{F}(\mathbf{X})$ to every Boolean function f.

Definition 2.2 *A multilinear polynomial $P(\mathbf{X})$ exactly represents the Boolean function f if for all $\mathbf{x} \in \{0,1\}^n$, $P(\mathbf{x}) = f(\mathbf{x})$.*

We also say that the polynomial $P(\mathbf{X})$ *computes* the function $f(\mathbf{X})$. We will show that such a polynomial representing the function f exists and it is unique. In fact, this holds not just for Boolean functions but more generally for functions taking values in the field $\mathbb{F}$.

Theorem 2.3 *For every function $f : \{0,1\}^n \to \mathbb{F}$, there is a unique multilinear polynomial in $\mathbb{F}[\mathbf{X}]$ such that for all $\mathbf{x} \in \{0,1\}^n$, $P(\mathbf{x}) = f(\mathbf{x})$.*

PROOF: We will first prove the existence part. Consider a Boolean vector $\mathbf{x} \in \{0,1\}^n$. We define the Boolean indicator function $I_{\mathbf{x}}$ as

$$I_{\mathbf{x}}(\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{y} \\ 0 & \text{if } \mathbf{x} \neq \mathbf{y} \end{cases}$$

It is easy to see that this function is computed by the polynomial

$$P_{\mathbf{x}}(\mathbf{X}) = \prod_{i|x_i=0} (1 - X_i) \prod_{i|x_i=1} X_i.$$

Further the polynomial $P_{\mathbf{x}}(\mathbf{X})$ is multilinear. The indicator functions form a basis for the space of all functions $f : \{0,1\}^n \to \mathbb{F}$. Hence we can write every such function as a suitable linear combination. More precisely, given a function f, the polynomial

$$P_f(\mathbf{X}) = \sum_{\mathbf{x} \in \{0,1\}^n} f(\mathbf{x}) \prod_{i|x_i=0} (1 - X_i) \prod_{i|x_i=1} X_i \tag{1}$$

computes the function f exactly.

An alternate proof of existence is via the Moebius inversion formula. For the purposes of this proof, it will be convenient to identify the subset $S \subseteq [n]$ with its incidence vector $\mathbf{s} = (s_1, \ldots, s_n)$ in $\{0,1\}^n$ where

$$s_i = \begin{cases} 1 \text{ if } x \in S \\ 0 \text{ otherwise} \end{cases}$$

Given a function $f : \{0,1\}^n \to \mathbb{F}$, we claim that it is computed by the polynomial

$$P_f(\mathbf{X}) = \sum_{S \subseteq [n]} c_S \prod_{i \in S} X_i$$

where c_S is given by the *Moebius inversion formula*:

$$c_S = \sum_{T \subseteq S} (-1)^{|S|-|T|} f(\mathbf{t}) \tag{2}$$

The prove the correctness, we check that

$$\begin{aligned}
f(\mathbf{r}) &= \sum_{S \subseteq [n]} c_S \prod_{i \in S} r_i \\
&= \sum_{S \subseteq R} c_S \\
&= \sum_{S \subseteq R} \sum_{T \subseteq S} (-1)^{|S|-|T|} f(\mathbf{t}) \\
&= \sum_{T \subseteq R} f(\mathbf{t}) \sum_{S:\ T \subseteq S \subseteq R} (-1)^{|S|-|T|} \qquad \text{By Equation (2)} \\
&= f(\mathbf{r})
\end{aligned}$$

The last equality holds since

$$\sum_{S:\ T \subseteq S \subseteq R} (-1)^{|S|-|T|} = \begin{cases} 1 \text{ if } T = R \\ 0 \text{ otherwise} \end{cases}$$

To prove the uniqueness of the multilinear polynomial computing the function f, note that if two distinct multilinear polynomials $P(\mathbf{X})$ and $P'(\mathbf{X})$ compute the same function on $\{0,1\}^n$, then their difference is a non-zero polynomial computing the 0 function. Hence it suffices to show that a non-zero multilinear polynomial $Q(\mathbf{X})$ cannot be 0 at every point in $\{0,1\}^n$. We prove this by induction on n.

The base case when $n = 1$ is simple, since a univariate polynomial of degree 1 cannot have two roots. For the inductive case, assume that the variable X_n appears in some monomial of Q, since if not, we are done by induction on n. Grouping together the monomials where X_n appears,

$$Q(\mathbf{X}) = X_n R(X_1, \ldots, X_{n-1}) + S(X_1, \ldots, X_{n-1})$$

where R and S are also multilinear polynomials in $n-1$ variables. Further $R(X_1, \cdots, X_{n-1})$ is not the 0 polynomial, since X_n appears in some monomial of Q. Hence, by the induction hypothesis, we can find $x_1, \ldots, x_{n-1}$ so that

$$Q(x_1, \ldots, x_{n-1}) = a \neq 0.$$

Setting $S(x_1, \ldots, x_{n-1}) = b$, we get

$$Q(x_1, \ldots, x_{n-1}X_n) = aX_n + b \qquad a \neq 0.$$

Using the base case, this polynomial is non-zero at for $X_n = 0$ or $X_n = 1$. □

A shorter proof of uniqueness would be to note that the map $f \to P_f$ given by Equation (1) is a linear map that gives an injection from the space of functions $f : \{0,1\}^n \to \mathbb{F}$ to the space of multilinear polynomials in $\mathbb{F}[X_1, \ldots, X_n]$. Since both are vector spaces of dimension 2^n over $\mathbb{F}$, the map must be invertible.

However, an advantage of our proofs is that, somewhat surprisingly, they do not use the field structure of $\mathbb{F}$, they work over any commutative ring. This generalization will be helpful to us when we consider Boolean function representations by polynomials over $\mathbb{Z}_m$. Let R be a commutative ring containing the additive identity O such that for all $a \in R$, $a + 0 = a$ and the multiplicative identity 1 so that $a \cdot 1 = a$ and $0 \neq 1$.

Corollary 2.4 *For every function $f : \{0,1\}^n \to R$, there is a unique multilinear polynomial in $R[\mathbf{X}]$ such that for all $\mathbf{x} \in \{0,1\}^n$, $P(\mathbf{x}) = f(\mathbf{x})$.*

PROOF: Both our proofs of existence in Theorem 2.3 hold for any ring. In particular, given a function f, the polynomial

$$P_f(\mathbf{X}) = \sum_{\mathbf{x} \in \{0,1\}^n} f(\mathbf{x}) \prod_{i|x_i=0} (1 - X_i) \prod_{i|x_i=1} X_i$$

computes the function f exactly.

For the uniqueness part, it suffices to show that a linear polynomial in one variable of the form

$$Q(X_1) = aX_1 + b \qquad a, b \in R$$

cannot satisfy $Q(0) = Q(1) = 0$ unless $a = 0$ and $b = 0$. Plugging in $X_1 = 0$,

$$Q(0) \;=\; b \;=\; 0$$

hence $Q(X_1) = aX_1$. Now plugging in $X_1 = 1$,

$$Q(1) \;=\; a \;=\; 0.$$

One can now repeat the same inductive argument. □

We have defined a notion of computing a Boolean function using a polynomial. This gives rise to a natural complexity measure for a function, namely its degree.

Definition 2.3 *The $\mathbb{F}$-degree of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is the degree of the unique multilinear polynomial in $\mathbb{F}[\mathbf{X}]$ computing the function f and is denoted by* $\deg_{\mathbb{F}}(f)$.

Of particular interest to us are the cases when $\mathbb{F} = \mathbb{R}$ and $\mathbb{F} = \mathbb{Z}_p$. We denote these degrees by $\deg(f)$ and $\deg_p(f)$ respectively. In fact, this suffices to cover all possible fields $\mathbb{F}$.

Lemma 2.5 *For any field $\mathbb{F}$ of characteristic 0,*

$$\deg_{\mathbb{F}}(f) = \deg(f).$$

For any field $\mathbb{F}$ of characteristic p,

$$\deg_{\mathbb{F}}(f) = \deg_p(f).$$

PROOF: By the Moebius inversion formula, the coefficients of the polynomial computing f are given by the equation

$$c_S \;=\; \sum_{T \subseteq S} (-1)^{|S|-|T|} f(\mathbf{t}).$$

From this it follows that the values of c_S, and hence $\deg_{\mathbb{F}}(f)$ depends only on the characteristic of the field $\mathbb{F}$. □

A natural question is to ask how these various degrees are related.

Lemma 2.6 *[61] For any prime p:*

$$\deg_p(f) \;\leq\; \deg(f)$$

PROOF: Let $P_f(\mathbf{X})$ be the unique multilinear polynomial computing f over $\mathbb{Q}$. Note that by the Moebius inversion formula, the coefficients of P_f are all integers. Hence if

$$P_f(\mathbf{X}) = \sum_S c_S \prod_{i \in S} X_i$$

then we can define the polynomial $P'(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ as

$$P'_f(\mathbf{X}) = \sum_S c'_S \prod_{i \in S} X_i \qquad c'_i \stackrel{\triangle}{=} c_i \bmod p$$

It is easy to see that for any $\mathbf{x} \in \{0,1\}^n$,

$$P'_f(\mathbf{x}) \equiv P(\mathbf{x}) \bmod p.$$

It follows that $P'(\mathbf{X})$ computes the function f over $\mathbb{Z}_p$. Also it's degree can only be less than the degree of $P(\mathbf{X})$. □

One cannot hope for a non-trivial inequality in the other direction, since for instance PARITY can be represented by $\sum X_i$ over $\mathbb{Z}_2$, hence $\deg_2(\text{PARITY}) = 1$. However over $\mathbb{R}$, it is represented by the polynomial

$$\frac{1}{2} - \frac{1}{2} \prod_i (1 - 2X_i)$$

hence $\deg(\text{PARITY}) = n$.

Note that we have identified the Boolean values $\{0,1\}$ with the elements 0 and 1 of the field $\mathbb{F}$. This choice is arbitrary, we can associate them with any two distinct field elements a_0 and a_1. In fact, over $\mathbb{Q}$, it is often convenient to associate the Boolean hypercube with $\{\pm 1\}^n$. A natural question is whether the choice of field elements changes the degree of a function. We will show that this does not matter.

We need to introduce some notation for this. We will refer to the pair $\{a_0, a_1\}$ as the input basis. Let us define the map $A : \{0,1\} \to \{a_0, a_1\}$ as

$$A(0) = a_0, \qquad A(1) = a_1.$$

Similarly, for a Boolean vector $\mathbf{x}$, let

$$A(\mathbf{x}) = (A(x_1), \ldots, A(x_n)).$$

Definition 2.4 *A polynomial $P(\mathbf{X})$ computes a function $f : \{0,1\}^n \to \{0,1\}$ over the input basis (a_0, a_1) if for $\mathbf{x} \in \{0,1\}^n$,*

$$P(A(\mathbf{x})) = A(f(x)).$$

In this definition, we could ask that the polynomial output $f(x)$ rather than $A(f(x))$. Once can show that such a polynomial exists and is unique. The proof is similar to that of Theorem 2.3, and is omitted.

Theorem 2.7 *[61] The degree of the polynomial computing a function $f : \{0,1\}^n \to \{0,1\}$ is independent of the choice of input basis.*

PROOF: Assume that the polynomial $P(\mathbf{X})$ of degree d computes the function f over the input basis $\{0,1\}$. The linear transformation

$$T(x) = \frac{X - a_0}{a_1 - a_0}$$

maps $\{a_0, a_1\}$ to $\{0,1\}$. Hence the polynomial

$$P'(\mathbf{X}) \stackrel{\triangle}{=} P(T(X_1), \cdots, T(x_n))$$

compute the function f over the input basis (a_0, a_1). More precisely, it satfifies the condition that for $\mathbf{x} \in \{0,1\}^n$,

$$P'(A(\mathbf{x})) = P(\mathbf{x}) = f(\mathbf{x}).$$

Thus P' is the unique polynomial computing f over the $\{a_0, a_1\}$ basis.

To show that it must have degree d, assume that the monomial $X_1 \cdots X_d$ occurs in $P(\mathbf{X})$. Then we claim that it must also occur in $P'(\mathbf{X})$. When we apply the transformation to the variables $X_1, \ldots, X_d$, we get

$$\prod_{i=1}^{d} \frac{X_i - a_0}{a_1 - a_0}.$$

On expanding this term, we get the monomial $X_1 \cdots X_d$. This term cannot cancel out with terms coming from the expansion of any other monomial, since $\deg(P) = d$. Hence $\deg(P') = d$.

Finally, if we want the polynomial to output $A(f(\mathbf{x}))$ rather than $f(x)$, we take the polynomial

$$Q(\mathbf{X}) = (a_1 - a_0)P'(\mathbf{X}) + a_0.$$

It is easy to see that $\deg(Q) = \deg(P') = d$. □

Finally, we show an analogue of the Schwartz-Zippel lemma for low degree multilinear polynomials, saying that low degree polynomials cannot have too many zeroes.

Lemma 2.8 *Given a polynomial $P[\mathbf{X}] \in \mathbb{F}[\mathbf{X}]$ of degree d,*

$$\Pr_{\mathbf{x} \in \{0,1\}^n}[P(\mathbf{x}) \neq 0] \geq 2^{-d}.$$

PROOF: The proof is by induction on the number of variables n. The base case is trivial.

For the inductive case, assume that X_n occurs in some monomial (else we are done by induction). We can write

$$P(\mathbf{X}) = X_n Q(X_1, \ldots, X_{n-1}) + R(X_1, \ldots, X_{n-1})$$

where $\deg(Q) \leq d-1$. By the induction hypothesis, over the random choice of $x_1, \ldots, x_{n-1}$,

$$\Pr_{(x_1,\ldots,x_{n-1}) \in \{0,1\}^{n-1}}[Q(x_1, \ldots, x_{n-1}) \neq 0] \geq 2^{-(d-1)} \tag{3}$$

Conditioning on this event, we are left with

$$P(x_1, \ldots, x_{n-1}, X_n) = aX_n + b \qquad a \neq 0$$

It is easy to see that

$$\Pr_{x_n \in \{0,1\}}[P(x_1, \ldots, x_n) \neq 0 \mid Q(x_1, \ldots, x_{n-1}) \neq 0] \geq \frac{1}{2}. \tag{4}$$

Hence by Equations 3 and 4,

$$\Pr_{\mathbf{x}\{0,1\}^n}[P(\mathbf{x}) \neq 0] \geq \frac{1}{2^d}$$

Hence, the probability hat $P(\mathbf{x}) \neq 0$ is at least 2^{-d}. □

An immediate consequence of this Lemma is a bound on the number of zeroes of a low degree polynomial.

Corollary 2.9 *A degree d multilinear polynomial in $\mathbb{F}[\mathbf{X}]$ is non-zero at at least 2^{n-d} points.*

This Corollary is a strengthening of the statement (see Theorem 2.3) that a non-zero multilinear polynomial cannot be 0 everywhere in $\{0,1\}^n$. Since $d \leq n$, it follows that the

polynomial is non-zero at some point. The bound of 2^{n-d} is tight, as can be seen from the polynomial

$$P(\mathbf{X}) = \prod_{i=1}^{d} X_i.$$

2.1.1 Degree bounds for Threshold functions

Definition 2.5 *The Threshold-k function* $T_k : \{0,1\}^n \to \{0,1\}$ *is defined to be* 1 *if the weight of the input is at least* k.

Note that T_1 is the OR function whereas T_n is the AND function. Our goal is to show tight degree bounds on all threshold functions.

Corollary 2.10 *The* OR *and* AND *functions have degree* n.

PROOF: We present two proofs of this fact.

One can write down the polynomials computing the OR and AND functions explicitly, and observe that they have degree n.

$$\text{AND}(\mathbf{X}) = \prod_{i=1}^{n} X_i$$

$$\text{OR}(\mathbf{X}) = 1 - \prod_{i=1}^{n} (1 - X_i)$$

The AND function is non-zero at exactly 1 point: $(1, \ldots, 1)$. Hence by Corollary 2.9, the polynomial computing it has degree n. Similarly, assume that $P(\mathbf{X})$ computes the OR function. Then

$$Q(\mathbf{X}) \stackrel{\triangle}{=} 1 - P(\mathbf{X})$$

is non-zero at a single point:$(0, \ldots, 0)$. Hence $\deg(Q) = \deg(P) = n$. □

Theorem 2.11 *For the Threshold-k function:*

$$\deg(T_k) \;\geq\; \max(k, n-k+1)$$

PROOF: Let $P(\mathbf{X})$ be the polynomial computing the Threshold-k function. Set the last $k-1$ variables of $P(\mathbf{X})$ to 1. Denote the resulting polynomial by $Q(X_1, \ldots, X_{n-k+1})$. It is

easy to see that Q computes the OR function on $k-1$ variables. Hence

$$\deg(P) \ \geq \ \deg(Q) \ = \ n-k+1.$$

Set the last $n-k$ variables of $P(\mathbf{X})$ to 0. Call the resulting polynomial $R(X_1, \cdots, X_k)$. It is easy to see that R computes the AND function on k variables. Hence

$$\deg(P) \ \geq \ \deg(R) \ = \ k.$$

Hence we have

$$\deg(T_k) \ \geq \ \max(k, n-k+1) \ > \ \frac{n}{2}$$

□

We point out that this bound holds over any field, irrespective of the characteristic. For other symmetric functions like the Mod function whose degree depends on the characteristic of the field, we need stronger tools which are developed in the next Section.

2.2 Symmetric Boolean Functions

Given a permutation $\sigma \in \mathbb{S}_n$, and a vector $\mathbf{x}$ of length n, let

$$\sigma(\mathbf{x}) \stackrel{\triangle}{=} (x_{\sigma(1)}, \ldots, x_{\sigma(n)}).$$

Definition 2.6 *A function $f : \{0,1\}^n \to \mathbb{F}$ is symmetric if for every $\mathbf{x} \in \{0,1\}^n$ and every $\sigma \in \mathbb{S}_n$, $f(\mathbf{x}) = f(\sigma(\mathbf{x}))$.*

For $\mathbf{x} \in \{0,1\}^n$ let the weight be $w(\mathbf{x}) = \sum x_i$. It is clear that the value of a Boolean function only depends on the weight of its input. Thus one can equivelently think of a symmetric Boolean function on n Boolean variables as a function defined on the integers $[0, \ldots, n]$, taking values in $\mathbb{F}$. We will use these two views of symmetric functions interchangeably.

Definition 2.7 *A polynomial $P(\mathbf{X}) \in \mathbb{F}[X_1, \ldots, X_n]$ is symmetric if for every $\sigma \in \mathbb{S}_n$, $P(X_1, \ldots, X_n) = P(X_{\sigma(1)}, \ldots, X_{\sigma(n)})$.*

When f is symmetric, it follows from the Moebius inversion formula that the polynomial $P(\mathbf{X})$ computing f is also symmetric. Define the elementary symmetric polynomials $S_0, \ldots, S_n$ as

$$S_0(\mathbf{X}) = 1$$
$$S_k(\mathbf{X}) = \sum_{i_1 < i_2 \cdots < i_k} X_{i_1} X_{i_2} \cdots X_{i_k}$$

It is well-known [58] that the elementry symmetric polynomials generate the ring of symmetric polynomials over $\mathbb{F}$. We show that the elementry symmetric polynomials in fact generate symmetric multilinear polynomials as a *vector space* over $\mathbb{F}$. In other words, we do not need to multiply elementry symmetric polynomial together in order to generate symmetric multilinear polynomials.

Proposition 2.12 *A symmetric multilinear polynomial can be written as a linear combination of the elementry symmetric polynomials over* $\mathbb{F}$.

PROOF: The proof is by induction on the degree d of the symmetric multilinear polynomial $P(\mathbf{X})$. The base case is simple.

Since $\deg(P) = d$, we can assume that the monomial $\prod_{i=1}^{d} X_1$ occurs in $P(\mathbf{X})$. Assume that its coefficient is c. By symmetry, every monomial of degree d must occur with exactly the same coeffecient, if not then the polynomial is not symmetric. Hence

$$\begin{aligned} P(\mathbf{X}) = &\ Q(\mathbf{X}) + c \cdot \sum_{i_1 < i_2 \ldots < i_d} X_{i_1} X_{i_2} \ldots X_{i_d} \\ = &\ Q(\mathbf{X}) + c \cdot S_d(\mathbf{X}) \end{aligned}$$

where $\deg(Q) = d - 1$. By induction, $Q(\mathbf{X})$ is a linear combination of elementry symmetric polynomials, which implies the same for $P(\mathbf{X})$. □

Note that on an input $\mathbf{x} \in \{0,1\}^n$ of weight w,

$$S_k(\mathbf{x}) = \binom{w}{k}.$$

Thus the symmetric polynomial

$$P(\mathbf{X}) = \sum_k c_k \cdot S_k(\mathbf{X})$$

computes the same function $f : w \to \mathbb{F}$ as the polynomial

$$Q(w) = \sum_k c_k \binom{w}{k}.$$

Note that this does not imply that symmetric polynomials can be treated as univariate polybomials over $\mathbb{F}$. For instance, if $\mathbb{F} = \mathbb{Z}_p$, then the co-efficients of the polynomial $Q(w)$ do not lie in $\mathbb{Z}_p$. We will treat the $\mathbb{Z}_p$ case spearately in section 2.2.1. For the case $\mathbb{F} = \mathbb{R}$ however, one can think of the polynomial $Q(w)$ as a univariate polynomial in $\mathbb{R}$. This gives the following easy bound on the degree of any symmetric function.

Proposition 2.13 *Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant symmetric function. Then* $\deg(f) \geq \frac{n}{2}$.

PROOF: Let $Q(w)$ be the univariate polynomial described above such that $Q(w(\mathbf{x})) = f(\mathbf{x})$. Note that Q maps $\{0, \ldots, n\} \to \{0,1\}$. Thus either the polynomial $Q(w)$ or $Q(w) - 1$ has at least $\frac{n}{2}$ zeroes. □

Von zur Gathen and Roche improve this to show that for symmetric functions f, $\deg(f) \geq n - n^{0.548}$ [77]. They conjecture that in fact $\deg(f) \geq n - O(1)$.

2.2.1 Symmetric Polynomials over $\mathbb{Z}_p$

In thi section, we focus on symmetric functions and polynomials over $\mathbb{Z}_p$. As observed earlier, these results hold for all fields of characteristic p.

Since elementry symmetric polynomials compute binomial coefficients, it is natural to ask what functions are computed by binomial coefficients mod p. For this we turn to a classical result about binomial coefficients modulo p called Lucas' Theorem [37].

Theorem 2.14 Lucas' Theorem. *Let*

$$w = \sum_{i \geq 0} w_i p^i, \ 0 \leq w_i < p$$

$$k = \sum_{i \geq 0} k_i p^i, \ 0 \leq k_i < p.$$

Then

$$\binom{w}{k} \equiv \prod_i \binom{w_i}{k_i} \bmod p$$

PROOF: We use the fact that over $\mathbb{Z}_p$, for all i,

$$(1+X)^{p^i} \;=\; 1+X^{p^i}.$$

We have

$$(1+X)^w = \sum_{k=0}^{w} \binom{w}{j} X^j \tag{5}$$

But since $w = \sum_i w_i p^i$,

$$\begin{aligned}
(1+X)^w &= \prod_i (1+X)^{w_i p^i} \\
&= \prod_i (1+X^{p^i})^{w_i} \\
&= \prod_i \sum_{k_i=0}^{w_i} \binom{w_i}{k_i} X^{k_i p^i} \\
&= \sum_{k_0,k_1,\ldots} X^{\sum_i k_i p^i} \prod_i \binom{w_i}{k_i}
\end{aligned}$$

Setting $k = \sum_i k_i p^i$, we get

$$(1+X)^w = \sum_{k=0}^{w} X^k \prod_i \binom{w_i}{k_i} \tag{6}$$

Comparing Equations 5 and 6, over $\mathbb{Z}_p$ we have the identity

$$\binom{w}{k} = \prod_i \binom{w_i}{k_i}$$

□

Theorem 2.15 *The function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a symmetric polynomial $P(\mathbf{X})$ over $\mathbb{Z}_p$ where $\deg(P) < p^\ell$ iff f is a function of only the ℓ least significant digits $w_0, \cdots, w_{\ell-1}$ of the weight $w(\mathbf{x})$ in base p.*

PROOF: Assume that $P(\mathbf{X})$ is a symmetric polynomial and that $\deg(P) < p^\ell$. We can write $P(\mathbf{X})$ as a linear combination of $S_k(\mathbf{X})$ for $1 \le k < p^\ell$. We show that $S_k(\mathbf{x})$ depends only on $w_0, \cdots, w_{\ell-1}$, which will imply that $P(\mathbf{x})$ depends only on $w_0, \cdots, w_{\ell-1}$.

By Lucas' Theorem, on an input $\mathbf{x} \in \{0,1\}^n$ of weight w,

$$S_k(\mathbf{x}) \;=\; \binom{w}{k} \;\equiv\; \prod_i \binom{w_i}{k_i} \bmod p$$

However $k < p^l$, so $k_i = 0$ for $i \geq l$. Hence

$$\binom{w}{k} \equiv \prod_{i=0}^{l-1} \binom{w_i}{k_i} \prod_{i \geq l} \binom{w_i}{0} \equiv \prod_{i=0}^{l-1} \binom{w_i}{k_i} \bmod p$$

For the other direction, we need to write every function f which depends only on $w_0, \ldots, w_{\ell-1}$ as a polynomial of degree less than p^ℓ. We first show that on input $\mathbf{x} \in \{0,1\}^n$ of weight w, $S_{p^\ell}(\mathbf{x}) \equiv w_\ell \bmod p$. By Lucas' theorem,

$$S_{p^\ell}(\mathbf{x}) = \binom{w}{p^\ell} \equiv \binom{w_\ell}{1} \prod_{i \neq \ell} \binom{w_i}{0} \equiv w_\ell \bmod p.$$

Now consider any function f which depends only on $w_0, \cdots, w_{\ell-1}$, the ℓ least significant digits of the weight. Using the fact that every function from $\mathbb{Z}_p^\ell \to \mathbb{Z}_p$ is computed by some polynomial, f can be written as a polynomial $Q(w_0, \cdots, w_{\ell-1})$ over $\mathbb{Z}_p$ with the degree of each $w_i \leq p-1$. But $S_{p^i}(a) \equiv w_i \bmod p$. Hence the polynomial

$$P(\mathbf{X}) = Q(S_1(\mathbf{X}), \ldots, S_{p^{\ell-1}}(\mathbf{X}))$$

computes the function f on 0-1 inputs. It is a symmetric polynomial whose degree is bounded by by $\sum_{i=0}^{\ell-1} p^i(p-1) = p^\ell - 1$. □

Saying that f depends only on $w_0, \ldots, w_{\ell-1}$ is equivalent to saying that f is a function of $w \bmod p^\ell$. Hence, we can restate Theorem 2.15 as follows:

Theorem 2.16 *The symmetric functions $f : \{0,1\}^n \to \mathbb{Z}_p$ that can be computed by polynomials of degree $k < p^\ell$ are exactly the functions which depend only on $w \bmod p^\ell$.*

These theorems give us tools for showing lower bounds on $\deg_p(f)$ for symmetric functions f. If we can prove that f must depend on the digit w_k, then Theorem 2.15 implies that $\deg_p(f) \geq p^k$. Such arguments will give us bounds that are tight to within a factor of p. While this is good enough for all our complexity-theoretic applications where p will be a fixed constant, the combinatorial applications in Chapter 5 require sharper degree bounds. A refined bound that gives an exact characterization of the degree is given by Theorem 5.14 in Chapter 5. One can extend the above results to the prime power case, we defer this to Appendix A.

It is known that the polynomials $S_j(\mathbf{X})$ for $1 \leq j \leq n$ generate the symmetric polynomials in $\mathbb{Z}_p[\mathbf{X}]$ and further they are algebraically independent. We can think of symmetric multilinear polynomials as symmetric polynomials in the quotient ring $\mathbb{Z}_p[\mathbf{X}]/(X_1^2 - X_1, \ldots, X_n^2 - X_n)$. We have just proved that the symmetric polynomials in this quotient ring are generated by $S_p(\mathbf{X}), \ldots, S_{p^\ell}(\mathbf{X})$ where $\ell = \lfloor \log_p n \rfloor$.

2.2.2 Degree Bounds for Mod functions

Definition 2.8 *The Mod-k function $M_k : \{0,1\}^n \to \{0,1\}$ is 1 if the weight of the input is divisible by k and 0 otherwise.*

Unlike the Threshold-k function, where we proved a lower bound of $n/2$ irrespective of the field, the degree of the Mod-k function depends crucuially on k and the characteristsic of the field $\mathbb{F}$. Proposition 2.13 implies that over $\mathbb{R}$, $\deg(f) \geq n/2$. Thus we can focus on the $\mathbb{Z}_p$ case. In what follows, we will think of k and p as constants, and we are interested in the asymptotic behaviour of $\deg(f)$ in terms of the number of variables n.

Theorem 2.17 *For the Mod-k function over $\mathbb{Z}_p$, if $k = p^a$, then $\deg_p(M_k) = O(1)$. Else $\deg_p(M_k) = \Omega(n)$.*

PROOF: Let $k = p^a$. The Mod-k function is 1 iff $w \mod k \equiv 0$. This happens iff $w_0 = w_1 \cdots = w_{a-1} = 0$. Hence the Mod-$k$ function depends only on the first a digits of the weight. So $\deg_p(M_k) < p^a = k$ by Theorem 2.15.

Assume that k is not a power of p. We will show that the Mod-k function depends on the most significant digit of the weight $w_{\ell-1}$ where $\ell = \lfloor \log_p n \rfloor$. Set $w_0, \ldots, w_{\ell-2} = 0$. Now if $w_{\ell=0}$, then

$$w = 0 \;\Rightarrow\; w \equiv 0 \bmod k \;\Rightarrow\; M_k(w) = 1.$$

On the other hand, if $w_{\ell-1} = 1$, then since k is not a power of p,

$$w = p^\ell \;\Rightarrow\; w \not\equiv 0 \mod k \;\Rightarrow\; M_k(w) = 0.$$

By Theorem 2.15, this implies that $\deg_p(M_k) = \Omega(n)$. □

2.3 The Fourier Representation

For this section, we assume that we are working over a field $\mathbb{F}$ whose characteristic does not equal 2.

In the Fourier representation, we regard Boolean functions as functions $f : \{\pm 1\}^n \to \{\pm 1\}$, where -1 is associated with True and $+1$ with False. Note that this does not make sense over charcteristic 2, since we would have $-1 = +1$. One can check that in this basis, OR and AND are represented as:

$$OR(\mathbf{X}) = -1 + \frac{\prod_i (1 + X_i)}{2^{n-1}}.$$

$$\text{AND}(\mathbf{X}) = 1 - \frac{\prod_i (1 - X_i)}{2^{n-1}}.$$

The parity functions or characters are defined for every $S \subseteq [n]$ as

$$\chi_S(\mathbf{X}) = \prod_{i \in S} X_i.$$

In the $\{0, 1\}$ basis, these functions can be written as

$$\chi_S(\mathbf{X}) = \oplus_{i \in S} X_i$$

where $\oplus$ denotes addition modulo 2, hence the name parity functions. Also, these are the characters of the Abelian group $\mathbb{Z}_2^n$ [58]. We use $\mathbf{x} \odot \mathbf{y}$ to denote coordinate-wise multiplication. The *linearity* of the characters implies that:

$$\chi_S(\mathbf{x} \odot \mathbf{y}) = \chi_S(\mathbf{x}) \cdot \chi_S(\mathbf{y})$$

We will show that the characters give a natural basis for the space of functions $f : \{\pm 1\}^n \to \mathbb{F}$ i.e. $\mathbb{F}$-valued functions on the Boolean hypercube. To do so, we define the following inner-product on such functions.

$$\langle f, g \rangle \stackrel{\triangle}{=} \underset{\mathbf{x} \in \{\pm 1\}^n}{E} [f(\mathbf{x}) g(\mathbf{x})] = 2^{-n} \sum_{\mathbf{x} \in \{\pm 1\}^n} f(\mathbf{x}) g(\mathbf{x})$$

Observe that for any Boolean function,

$$\langle f, f \rangle = 2^{-n} \sum_{\mathbf{x} \in \{\pm 1\}^n} f(\mathbf{x})^2 = 2^{-n} \sum_{\mathbf{x} \in \{\pm 1\}^n} 1 = 1.$$

Lemma 2.18 *Given $S, T \subseteq [n]$,*

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

PROOF: We denote the empty set by ϕ. We observe that if $S \neq 0$, then

$$\sum_{\mathbf{x} \in \{\pm 1\}^n} \chi_S(\mathbf{x}) \;=\; 0$$

whereas if $S = \phi$, then

$$\sum_{\mathbf{x} \in \{\pm 1\}^n} \chi_\phi(\mathbf{x}) \;=\; \sum_{\mathbf{x} \in \{\pm 1\}^n} 1 \;=\; 2^n.$$

Let $S \Delta T$ denote the symmetric difference of S and T.

$$\begin{aligned} \chi_S(\mathbf{x}) \cdot \chi_T(\mathbf{x}) &= \prod_{i \in S} x_i \prod_{j \in T} x_j \\ &= \prod_{i \in S \Delta T} x_i \qquad \text{since } x_i^2 = 1 \\ &= \chi_{S \Delta T}(\mathbf{x}) \end{aligned}$$

Hence,

$$\langle \chi_S, \chi_T \rangle \;=\; 2^{-n} \sum_{\mathbf{x} \in \{\pm 1\}^n} \chi_{S \Delta T}(\mathbf{x}).$$

If $S = T$ hence $S \Delta T = \phi$, this sum is 1, else it is 0. □

An immediate consequence of this Lemma is the following Corollary:

Corollary 2.19 *The characters χ_S for $S \subseteq [n]$ form an orthonormal basis for the space of all functions $f : \{\pm 1\}^n \to \mathbb{F}$.*

PROOF: By Lemma 2.18, the characters are set of orthonormal vectors. To see that they give a basis, note that the dimension of the space of functions $f : \{\pm 1\}^n \to \mathbb{F}$ is 2^n, and there are 2^n characters. □

The function f can be written in this basis in terms of its Fourier representation:

$$f \;= \sum_{S \in [n]} \hat{f}(S) \chi_S \qquad \hat{f}(S) = \langle f, \chi_S \rangle.$$

The coeffecients $\hat{f}(S)$ are called the Fourier coeffecients of the function f, each coeffecient measures the correlation between f and the character χ_S. Finally, we note that

$$\begin{aligned}\langle f, g\rangle &= \left\langle \sum_S \hat{f}(S)\chi_S, \sum_T \hat{g}(T)\chi_T \right\rangle \\ &= \sum_S \hat{f}(S)\hat{g}(S) \qquad \text{since } \langle \chi_S, \chi_T\rangle = 0\end{aligned}$$

In the case of Boolean functions, we know that $\langle f, f\rangle = 1$. So we derive the following identity known as Parseval's identity:

$$\langle f, f\rangle \;=\; \sum_S \hat{f}^2(S) \;=\; 1.$$

In the Fourier representation, we have

$$\deg_{\mathbb{F}}(f) = \max\{|S| \mid S \subseteq [n], \hat{f}(S) \neq 0\}.$$

Note that by Theorem 2.7, the degree of a function is independent of the choice of basis.

It is easy to show the analogue of the Schwarz-Zippel lemma for low degree multilinear polynomials, saying that low degree polynomials cannot have too many zeroes on $\{\pm 1\}^n$. The proof of this Lemma is idetical to the proof of Lemma 2.8 and is omitted.

Lemma 2.20 *Given a function $f : \{\pm 1\}^n \to \mathbb{F}$ of degree d,*

$$\Pr_{\mathbf{x}\in\{\pm 1\}^n}[P(\mathbf{x}) \neq 0] \geq 2^{-d}.$$

2.3.1 Degree Lower Bounds via Influence of Variables

In this section, we consider representing Boolean functions as polynomial over $\mathbb{R}$. Our goal is to prove a result of Nisan and Szegedy, saying that any function that depends on all n variables must have degree $\Omega(\log n)$. The condition that f depends on all n variables is necessary, consider for instance the function X_1 which depends only on 1 variable and has degree 1. In contrast, over $\mathbb{Z}_p$ there are functions depending on all n variables that have degree $O(1)$, the Parity functions over $\mathbb{Z}_2$ being an example.

The proof of this theorem uses the notion of sensitivity of a Boolean function. We use $\mathbf{e}_i$ to denote the vector which is $-$ in the i^{th} coordinate and $+1$ elsewhere. Note that

$$\mathbf{x} \odot \mathbf{e}_i = (x_1, \ldots, -x_i, \ldots x_n)$$

Definition 2.9 *The influence of the variable X_i on the Boolean function f is defined as*

$$\mathrm{Inf}_i(f) = \Pr_{\mathbf{x}\in\{\pm 1\}^n}[f(\mathbf{x}) \neq f(\mathbf{x} \odot \mathbf{e}_i)]$$

The influence measures the probability that the value of f changes on flipping the i^{th} bit for a random point $\mathbf{x} \in \{\pm 1\}^n$. One can derive a closed from for $\mathrm{Inf}_i(f)$ in terms of the Fourier spectrum of f.

Lemma 2.21 *[50] Let $f = \sum_S \hat{f}(S)\chi_S$. Then*

$$\mathrm{Inf}_i(f) = \sum_{S \ni i} \hat{f}^2(S).$$

PROOF: Note that

$$\frac{1 - f(\mathbf{x}) \cdot f(\mathbf{x} \odot \mathbf{e}_i)}{2} = \begin{cases} 1 \text{ if } f(\mathbf{x}) \neq f(\mathbf{x} \odot \mathbf{e}_i) \\ 0 \text{ otherwise.} \end{cases}$$

So we can write

$$\mathrm{Inf}_i(f) = \mathop{E}_{\mathbf{x}\in\{\pm 1\}^n}\left[\frac{1 - f(\mathbf{x}) \cdot f(\mathbf{x} \odot \mathbf{e}_i)}{2}\right] \tag{7}$$

Note that

$$\begin{aligned} f(\mathbf{x} \odot \mathbf{e}_i) &= \sum_S \hat{f}(S)\chi_S(\mathbf{x} \odot \mathbf{e}_i) \\ &= \sum_S \hat{f}(S)\chi_S(\mathbf{x}) \cdot \chi_S(\mathbf{e}_i) \qquad \text{By linearity of characters} \\ &= \sum_{S \not\ni i} \hat{f}(S)\chi_S(\mathbf{x}) - \sum_{S \ni i} \hat{f}(S)\chi_S(\mathbf{x}) \end{aligned}$$

Plugging this into Equation 7,

$$\begin{aligned} \mathrm{Inf}_i(f) &= \mathop{E}_{\mathbf{x}\in\{\pm 1\}^n}\left[\frac{1}{2} - \frac{1}{2}\left(\sum_S \hat{f}(S)\chi_S(\mathbf{x})\right) \cdot \left(\sum_{S \not\ni i} \hat{f}(S)\chi_S(\mathbf{x}) - \sum_{S \ni i} \hat{f}(S)\chi_S(\mathbf{x})\right)\right] \\ &= \frac{1}{2} + \frac{1}{2}\sum_{S \ni i} \hat{f}^2(S) - \frac{1}{2}\sum_{S \not\ni i} \hat{f}^2(S) \qquad \text{By orthogonality of characters} \end{aligned} \tag{8}$$

By Parseval's identity, $\sum_S \hat{f}^2(S) = 1$. Plugging this into Equation 8,

$$\begin{aligned} \mathrm{Inf}_i(f) &= \frac{1}{2}\sum_S \hat{f}^2(S) + \frac{1}{2}\sum_{S \ni i} \hat{f}^2(S) - \frac{1}{2}\sum_{S \not\ni i} \hat{f}^2(S) \\ &= \sum_{S \ni i} \hat{f}^2(S) \end{aligned}$$

□

We can also relate the Influence of a variable to $\deg(f)$.

Lemma 2.22 *Let* $\deg(f) = d$. *Then* $\mathrm{Inf}_i(f) \geq 2^{-d}$.

PROOF: We prove the theorem for $i = n$. Let us define th polynomial

$$Q(X_1, \ldots, X_{n-1}) = f(X_1, \ldots, X_n) - f(X_1, \ldots, -X_n).$$

Since $\deg(f) = d$, it follows that $\deg(Q) \leq d$. It is easy to see that

$$\mathrm{Inf}_n(f) = \Pr_{x_1,\ldots,x_{n-1} \in \{\pm 1\}^{n-1}} [Q(x_1, \ldots, x_{n-1}) \neq 0]$$

This quantity is at least 2^{-d} by Lemma 2.20. □

We can now prove the theorem of Nisan and Szegedy.

Theorem 2.23 *[64] Let* $f(\mathbf{X})$ *be a function that depends on all* n *variables* $X_1, \ldots, X_n$. *Then*

$$\deg(f) \geq \log n - O(\log \log n)$$

PROOF: Let $\deg(f) = d$. Consider the quantity $\sum_i \mathrm{Inf}_i(f)$. This is also called the *average sensitivity* of f. From Lemma 2.21, we have

$$\begin{aligned}
\sum_{i=1}^{n} \mathrm{Inf}_i(f) &= \sum_i \sum_{S \ni i} \hat{f}^2(S) \\
&= \sum_{S \subseteq [n]} |S| \hat{f}^2(S) \\
&\leq \max\{|S| \mid \hat{f}^S \neq 0\} \qquad \text{Since } \sum_S \hat{f}^2(S) = 1 \\
&= d.
\end{aligned}$$

On the other hand, by Lemma 2.22, we get that

$$\sum_{i=1}^{n} \mathrm{Inf}_i(f) \geq \frac{n}{2^d} \tag{9}$$

Thus we get

$$d \geq \frac{n}{2^d} \Rightarrow d2^d \geq n.$$

From this it follows that

$$d \geq \log n - O(\log \log n).$$

□

This bound is in fact tight [64].

2.4 Application to Learning Juntas

We consider the problem of learning a k-junta under the uniform distribution on $\{0,1\}^n$. This problem was proposed by Blum and Langley [24], as a clean formulation of the problem of efficient learning in the presence of irrelevant information.

Problem 2.1 Learning Juntas: *Let f be an unknown function on n bits which actually depends only on a subset of the input of size k. Given random examples $\langle x, f(x) \rangle$ where x is chosen uniformly at random from $\{0,1\}^n$, learn the function f.*

Such a Boolean function which depends only on k inputs is called a k-junta. For this problem, we assume that $k \leq \log n$, so that the truth table of the function is polynomial in n.

For $k \leq \log n$, a k-junta can be expressed as a decision tree or a DNF of size n. Hence, a polynomial time algorithm for DNFs or decision trees under the uniform distribution would imply an algorithm for the k-junta problem. Thus, learning juntas is a first step towards learning polynomial size decision trees and DNFs under the uniform distribution. A brute force approach would be to take $O(k \log n)$ samples, and then run through all n^k subsets of possible relevant variables. The first non-trivial algorithm was given only recently by Mossel *et al.* [61], and runs in time roughly $O(n^{0.7k})$. However, even the question of whether one can learn k-juntas in polynomial time for $k = \omega(1)$ still remains open.

We present a slight simplification of the algorithm dues to Mossel *et al.*. The crux of both algorithms is structural results that relate the representations of a Boolean function over $\mathbb{Z}_2$ and $\mathbb{R}$. We first present these structural results (Lemmas 2.24 and 2.25). The first is from Mossel *et al.*, the second is new.

Lemma 2.24 *[61] Let f be a Boolean function on k variables such that $\hat{f}(S) = 0$ for all $S \subseteq [k]$ such that $|S| \leq t$. Then $\deg_2(f) < k - t$.*

PROOF: We consider the function g where

$$g(X_1, \ldots, X_k) \stackrel{\triangle}{=} f(X_1, \ldots, X_k) \cdot \chi_{[k]}(X_1, \ldots, X_k)$$

If we denote the complement of the set S by $\bar{S}$, then $\hat{g}(S) = \hat{f}(\bar{S})$. Hence if $|S| \geq k - t$, then $|\bar{S}| \leq t$ so $\hat{g}(S) = \hat{f}(\bar{S}) = 0$. Thus $\deg(g) < k - t$. Hence

$$\deg_2(g) \ \leq \ \deg(g) \ < \ k - t.$$

But over $\mathbb{Z}_2$,

$$g(X_1, \ldots, X_k) = f(X_1, \ldots, X_k) \oplus X_1 \cdots \oplus X_k$$

hence $\deg_2(f) = \deg_2(g) < k - t$. □

For the second Lemma, we only assume that $\hat{f}(S) = 0$ for all small sets $S \neq \phi$. In this case, while $\deg(g) = k$, we will still show that $\deg_2(g)$ is small.

Lemma 2.25 *Let f be a Boolean function on k variables such that $\hat{f}(S) = 0$ for all $S \subseteq [k]$ such that $1 \leq |S| \leq t$. Then $\deg_2(f) \leq k - t$.*

PROOF: Define the function $g(X_1, \ldots, X_k)$ as before. Once again $\hat{g}(S) = \hat{f}(\bar{S})$, hence

$$\begin{aligned} g(\mathbf{x}) &= \sum_S \hat{f}(\bar{S}) \chi_S(\mathbf{x}) \\ &= \hat{f}(\phi)\chi_{[k]}(\mathbf{x}) + \sum_{|S|<k-t} \hat{f}(\bar{S})\chi_S(\mathbf{x}) \end{aligned}$$

We now write $g(\mathbf{x})$ in the $\{0, 1\}$ basis. This is done be replacing X_i with $(1 - 2X_i)$.

$$g(\mathbf{X}) = \hat{f}(\phi) \prod_{i=1}^{k} (1 - 2X_i) + \sum_{|S|<k-t} \hat{f}(\bar{S}) \prod_{i \in S} (1 - 2X_i)$$

Note that the coefficients of the monomials of degree $k-t+d$ are of the form $\hat{f}(\phi) \cdot (-2)^{k-t+d}$ for $0 \leq d \leq t$. But since all the coefficients have to be integers, we know that $\hat{f}(\phi) \cdot (-2)^{k-t}$ is an integer. Hence for $d \geq 1$,

$$\hat{f}(\phi) \cdot (-2)^{k-t+d} \equiv 0 \bmod 2$$

Note that the $\mathbb{Z}_2$ representation of g is obtained from the representation over $\mathbb{R}$ in the $\{0,1\}$ input basis by taking every coefficient modulo 2. Hence all coefficients of degree $k-t+1$ and higher vanish modulo 2. Thus

$$\deg_2(f) = \deg_2(g) \leq k - t.$$

□

We are now ready to state the algorithm for learning k-juntas. We will need some facts from Mossel *et al.*The first states that being able to find a single relevant variable efficiently suffices to solve the junta problem. This is proved used a divide and conquer argument.

Fact 2.26 *Suppose $\mathcal{A}$ is an algorithm running in time $n^{\alpha}\text{poly}(2^k, n)$ that identifies a single variable relevant to f. Then there is an algorithm to learn f that runs in time $n^{\alpha}\text{poly}(2^k, n)$.*

One approach to finding relevant variables is to compute $\hat{f}(S)$ for all sets S where $1 \leq |S| \leq t$. If we find a set $S \neq \phi$ such that $\hat{f}(S) \neq 0$, then all the variables in S are relevant. Note that $\hat{f}(\phi)$ being non-zero does not allow us to identify any relevant variables. A simple Chernoff-bound argument shows that we can in fact compute any Fourier coefficient exactly.

Fact 2.27 *There is an algorithm to calculate any Fourier coefficient $\hat{f}(S)$ exactly, that succeeds with probability $1-\delta$ and runs in time* $\text{poly}(2^k, n, \log\frac{1}{\delta})$.

Finally, we need the fact that one can learn low-degree polynomials over $\mathbb{Z}_2$ efficiently via Gaussian elimination. This is proved via a standard Occam's razor argument. Let $\omega < 2.376$ be the exponent of matrix-multiplication. Note that if we learn the polynomial exactly, we have also identified all its relevant variables.

Fact 2.28 *There is an algorithm that can learn degree d polynomials in n variables over $\mathbb{Z}_2$ exactly, that succeeds with probability $1-\delta$ and runs in time $n^{\omega d}\text{poly}(2^k, n, \log\frac{1}{\delta})$.*

Algorithm 2.1 Identifying a Set of Relevant Variables

Compute $\hat{f}(S)$ for $1 \leq |S| \leq \frac{\omega}{\omega+1}k$.

If $\hat{f}(S) \neq 0$ for some set S, output S.

Else

   Compute f as a $\mathbb{Z}_2$ polynomial of degree $\frac{k}{\omega+1}$.

   Output all its relevant variables.

Theorem 2.29 *Algorithm 2.1 runs in time* $n^{\frac{\omega}{\omega+1}k}\text{poly}(2^k, n, \log\frac{1}{\delta})$ *and outputs a set of relevant variables with probability* $1-\delta$.

PROOF: Assume that $\hat{f}(S)$ is non-zero for some S where $1 \leq |S| \leq \frac{\omega}{\omega+1}k$. Then the algorithm will correctly identify this set with probability $1-\delta$. Since this requires computing no more than $n^{\frac{\omega}{\omega+1}k}$ coefficients, this step runs in time $n^{\frac{\omega}{\omega+1}k}\text{poly}(2^k, n, \log\frac{1}{\delta})$.

On the other hand, if all these coefficients are 0, then by Lemma 2.25

$$\deg_2(f) \leq k - \frac{\omega}{\omega+1}k = \frac{k}{\omega+1}.$$

In this case, the $\mathbb{Z}_2$ algorithm will learn the function exactly with probability $1-\delta$ and run in time $n^{\frac{\omega}{\omega+1}k}\text{poly}(2^k, n, \log\frac{1}{\delta})$. □

Our analysis is different from that of Mossel *et al.*They use Lemma 2.24 to handle the case when $\hat{f}(S)$ for $1 \leq |S| \leq \frac{\omega}{\omega+1}k$ and $\hat{f}(\phi) = 0$. To deal with the case when $\hat{f}(\phi) \neq 0$, they show that in this case, f has a non-zero Fourier coefficient of size at most $\frac{2k}{3}$. Our analysis does not improve the running time, but it gives a more unified approach.

2.5 Application to Circuit Lower Bounds

Definition 2.10 *The class* $\text{AC}^0[m]$ *denotes the class of polynomial size, bounded depth circuits on* n *inputs with* AND, OR, NOT *and* MOD-m *gates.*

The depth of the circuit will be denoted by d and the size by s. Note that $d = O(1)$ while $s = n^{O(1)}$. We have strong lower bounds for such circuits in the case when $m = p$ is a

prime. These results are primarily due to Razborov [68] and Smolensky [70] and crucially use polynomial representations of Boolean functions. A frontier open problem in circuit complexity is to show similar lower bounds for the case when m is composite.

We will prove the following result due to Smolensky: that PARITY $\notin$ AC$^0[3]$.

This proof is in two stages: we first show that every function in AC$^0[3]$ can be *approximated* in some sense by a low degree polynomial over $\mathbb{Z}_3$. We then show that PARITY cannot have such a low degree approximation.

We start by defining the notion of computing a function by a randomized polynomial.

Definition 2.11 *[75] A Boolean function f is computed by a randomized polynomial over $\mathbb{F}$ with degree d and error probability δ, if there is a sample space $\mathcal{P}$ of degree d polynomials in $\mathbb{F}[\mathbf{X}]$ such that for every $\mathbf{x} \in \{0,1\}^n$,*

$$\Pr_{P(\mathbf{X}) \in \mathcal{P}}[P(\mathbf{x}) = f(\mathbf{x})] \geq 1 - \delta.$$

Note that we require that for every $\mathbf{x}$, $f(\mathbf{x})$ be computed with good probability.

Lemma 2.30 *The* AND *and* OR *functions can be computed by a randomized polynomial over $\mathbb{Z}_3$ with degree $2k$ and error probability 3^{-k}.*

PROOF: We prove the result for the OR function, a similar argument holds for the AND function. Pick $\mathbf{c} \in \mathbb{Z}_3^n$ at random. For any $\mathbf{x} \in \{0,1\}^n$, such that $\mathbf{x} \neq 0$,

$$\Pr_{\mathbf{c}}[\mathbf{c} \cdot \mathbf{x} \neq 0] \ \geq \ \frac{2}{3}$$

whereas if $\mathbf{x} = \mathbf{0} = (0, \ldots, 0)$ then $\mathbf{c} \cdot \mathbf{0} = 0$. We pick $\mathbf{c}^1, \ldots, \mathbf{c}^k$ at random, and define

$$P(\mathbf{x}) = \mathrm{OR}((\mathbf{c}^1 \cdot \mathbf{x})^2, \ldots, (\mathbf{c}^k \cdot \mathbf{x})^2)$$
$$= 1 - \prod_{i=1}^{k} (1 - (\mathbf{c}^i \cdot \mathbf{x})^2)$$

Taking $(\mathbf{c}^i \cdot \mathbf{x})^2$ maps non-zero values of $\mathbf{c}^i \cdot \mathbf{x}$ to 1 by Fermat's theorem. It is easy to see that $P(\mathbf{X})$ has degree $2k$ and it computes the OR function with error probability 3^{-k}. $\square$

It is easy to see that the NOT and MOD-p functions can be computed exactly using polynomials of degree 1 and $p-1$ respectively.

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

$$\mathrm{MOD}(\mathbf{X}) = 1 - \left(\sum_i X_i\right)^{p-1}$$

Putting these together, we can show the following upper bound on $\mathrm{AC}^0[3]$.

Theorem 2.31 *Every function $f \in \mathrm{AC}^0[3]$ can be computed by a randomized polynomial over $\mathbb{Z}_3$ with degree $O((\log n)^d)$ and error probability n^{-c} for any constant c.*

PROOF: By taking $k = C \log s$ in Lemma 2.30 for some constant C, we can replace every gate of the circuit by a polynomial of degree $O(\log s)$ whose error probability for every input to the gate is $s^{-C/2}$. By composing these polynomials, we get a randomized polynomial $P(\mathbf{X})$ of degree $(C \log s)^d$ that computes the function f. Using the union bound, we can bound the error probability of $P(\mathbf{X})$ by $s^{-C/2+1}$. Since $s = n^{O(1)}$, an appropriate choice of C gives error probability n^{-c}. The degree of the polynomial is $(C \log s)^d = O((\log n)^d)$ since $\log s = O(\log n)$. □

A easy consequence is that there is a polynomial $P(\mathbf{X})$ that agrees with the function f on a large subset S of the hypercube.

Corollary 2.32 *Given function $f \in \mathrm{AC}^0[3]$, for any $c > 0$, there exists a polynomial $P(\mathbf{X})$ of degree $O((\log n)^d)$ and a set $S \subseteq \{0,1\}^n$ such that*

- $|S| \geq 2^n \left(1 - \frac{1}{n^c}\right)$.
- *For every $\mathbf{x} \in S$, $P(\mathbf{x}) = f(\mathbf{x})$.*

PROOF: It is clear from Theorem 2.31 that a randomly chosen polynomial from $\mathcal{P}$ satisfies

$$\Pr_{\mathbf{x} \in \{0,1\}^n}[P(\mathbf{x}) \neq f(\mathbf{x})] \;\leq\; n^{-c}.$$

We simply choose $P(\mathbf{X})$ to be the polynomial in $\mathcal{P}$ that has maximum agreement with f. We define S to be the set of points where $P(\mathbf{x}) = f(\mathbf{x})$. It is clear by an averaging argument that $|S|$ satisfies the desired condition. □

Thus we have showed that functions in $\mathrm{AC}^0[3]$ are can be approximated by low degree polynomials. So the task of proving lower bounds reduces to finding explicit functions that

cannot have such approximations. We show that PARITY is such a function. For this part of the proof it is more convenient to work with the $\{\pm 1\}$ basis. We map $S \subseteq \{0,1\}^n$ to $T \subseteq \{\pm 1\}^n$ by replacing 0 with 1 and 1 with -1. If the polynomial $P(\mathbf{X})$ computes PARITY over S, then the polynomial $Q(\mathbf{X})$ computes PARITY over T where

$$Q(\mathbf{X}) = 1 - 2 \cdot P\left(\frac{1-X_1}{2}, \ldots, \frac{1-X_n}{2}\right).$$

It is easy to see that $\deg(P) = \deg(Q)$.

Lemma 2.33 *Assume that there is a degree k polynomial computing* PARITY *over T. Then every function $f : T \to \mathbb{Z}_3$ can be computed by a polynomial of degree $\frac{n+k}{2}$.*

PROOF: Note that on $\{\pm 1\}^n$, the parity function is computed over by $\chi_{[n]}(\mathbf{X})$. Thus over the set T, we have the identity:

$$\chi_{[n]}(\mathbf{X}) \;=\; Q(\mathbf{X}) \tag{10}$$

The monomials χ_S form a basis for all functions $f : T \to \{\pm 1\}$. Thus it suffices to write each χ_S as a polynomial of degree at most $\frac{n+k}{2}$ over T. If $|S| \leq \frac{n+k}{2}$, we are done. So assume $|S| \geq \frac{n+k}{2}$, so that $|\bar{S}| \leq \frac{n-k}{2}$.

$$\begin{aligned} \chi_S(\mathbf{X}) &= \chi_{[n]}(\mathbf{X}) \cdot \chi_{\bar{S}}(\mathbf{X}) \\ &= Q(\mathbf{X}) \cdot \chi_{\bar{S}}(\mathbf{X}) \qquad \text{By Equation 10} \end{aligned} \tag{11}$$

Thus we can write χ_S as a polynomial of degree $\frac{n-k}{2} + k = \frac{n+k}{2}$. □

Theorem 2.34 PARITY *cannot be computed over a set T of size $2^n\left(1 - \frac{1}{n^c}\right)$ by a polynomial of degree $k = o(\sqrt{n})$.*

PROOF: Assume for contradiction that $k = o(\sqrt{n})$. By Lemma 2.33 function $f : T \to \{\pm 1\}$ can be written as a polynomial of degree at most $\frac{n+k}{2}$. Thus the space of all such functions is spanned by monomials of degree at most $\frac{n+k}{2}$. Hence the dimension of this space is bounded by

$$\sum_{j \leq \frac{n+k}{2}} \binom{n}{j}$$

By the law of large numbers, since $k = o(\sqrt{n})$, this sum is bounded by $2^n\left(\frac{1}{2} + o(1)\right)$. On the other hand, the dimension must be at least $|T| \geq 2^n\left(1 - \frac{1}{n^c}\right)$, which is a contradicton. □

By putting Theorems 2.31 and 2.34 together, we conclude that PARITY $\notin$ AC^0[3].

Theorem 2.35 *[70]* PARITY $\notin$ AC^0[3]

We note that there is some considerable slack in this proof. Our approximating polynomials had degree $O((\log n)^d)$ whereas we only require the degree to be $o(\sqrt{n})$ for a contradiction. One can use this observation to improve the lower bound and show that circuits of depth d computing PARITY must have size $2^{n^{\Omega(\frac{1}{d})}}$. Further, once can prove the following general lower bound using these techniques.

Theorem 2.36 *[70] If m is not a power of p, then* MOD-m $\notin$ AC^0[p]

Thus for instance, MOD-6 $\notin$ AC^0[2]. For these improvements, we refer the reader to Smolensky's original paper [70].

CHAPTER III

POLYNOMIAL REPRESENTATIONS OVER COMPOSITES

3.1 Beyond Exact Representations

In this chapter, we study two weaker notions of computing a Boolean function using a polynomial, namely *strong* and *weak* representations. These definitions are motivated by the failure of current techniques to prove lower bounds for $\mathrm{AC}^0[6]$.

To begin with, let us examine why the proof techniques that work modulo p do not work for composite m. Let us consider the case when $m = 6$. The problem lies in showing an upper bound for circuits in $AC^0[6]$ using low-degree polynomials. Lemma 2.30 saying that OR and AND can be approximated by low-degree polynomials still holds. The problem however is that the MOD-6 function can no longer be computed, even approximately by low degree polynomials over $\mathbb{Z}_6$. Our construction modulo p uses Fermat's theorem stating $x^{p-1} \equiv x \bmod p$. This no longer works modulo 6. In fact, if we could approximate the MOD-6 function by a low-degree polynomial over $\mathbb{Z}_6$, by the Chinese Remainder Theorem, we would also get low-degree approximations for MOD-6 over $\mathbb{Z}_2$ and $\mathbb{Z}_3$. But Smolensky shows that this is not possible [70].

However, this appears to be more of a problem with our definition of polynomial representations. It is natural to say that the polynomial

$$P(\mathbf{X}) = \sum_i X_i$$

computes the MOD-6 function over $\mathbb{Z}_6$. This motivates the following definitions due to Barrington [14] and Barrington, Beigel and Rudich [15].

Definition 3.1 *Polynomial* $P(\mathbf{X}) \in \mathbb{Z}_m[\mathbf{X}]$ *strongly represents function* $f : \{0,1\}^n \to$

$\{0,1\}$ *if for* $\mathbf{x} \in \{0,1\}^n$,

$$f(\mathbf{x}) = 0 \quad \Rightarrow P(\mathbf{x}) \equiv 0 \bmod m$$

$$f(\mathbf{x}) = 1 \quad \Rightarrow P(\mathbf{x}) \not\equiv 0 \bmod m.$$

It is clear that the MOD-6 function is strongly represented by the polynomial $\sum_i X_i$ over $\mathbb{Z}_6$. An even more general definition is *weak representations*, where we only require that the polynomial should be able to distinguish a 0-input from a 1-input.

Definition 3.2 *Polynomial* $P(\mathbf{X}) \in \mathbb{Z}_m[\mathbf{X}]$ *weakly represents function* $f : \{0,1\}^n \to \{0,1\}$ *if for* $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$,

$$f(\mathbf{x}) \neq f(\mathbf{y}) \;\Rightarrow\; P(\mathbf{x}) \neq P(\mathbf{y}) \bmod m.$$

An alternative view of weak representations is that there exists a set $A \subseteq \mathbb{Z}_m$ such that

$$f(\mathbf{x}) = 0 \quad \Rightarrow P(\mathbf{x}) \in A$$

$$f(\mathbf{x}) = 1 \quad \Rightarrow P(\mathbf{x}) \notin A$$

It is clear that strong representations are a special case of weak representations where $A = \{0\}$.

Let us begin by analyzing strong and weak representations in the prime case. Note that strong and weak representations of a function are no longer unique, unlike for exact representations. However if $P(\mathbf{X})$ strongly represents $f(\mathbf{X}) \bmod p$, then by Fermat's theorem $P(\mathbf{X})^{p-1}$ exactly represents $f(\mathbf{X}) \bmod p$. Similarly, one can convert a weak representation to an exact representation with at most a factor p increase in the degree. Since we think of p as constant, this implies that degree lower bounds for exact representations also hold for strong and weak representations, within a factor of p.

The situation is very different over composites. We have seen that the MOD-6 function is strongly represented by the polynomial $\sum_i X_i$ over $\mathbb{Z}_6$. In contrast by Theorem 2.17, the MOD-6 function requires degree $\Omega(n)$ for exact representation over $\mathbb{Z}_6$. In fact it turns out that strong and weak representations are deceptively powerful modulo composites, any many surprising upper bounds are now known for a variety of symmetric Boolean functions.

The task of proving degree lower bounds is now fairly challenging: the best lower bound known to date for weak representations of *any* Boolean function is $\Omega(\log n)$ [74, 41].

One weakness of strong and weak representations is that once cannot compose several such representations together, unlike in exact representations. This means that it is not clear if we can show an upper bound for every function in $\mathrm{AC}^0[m]$, even though we have upper bounds for all the component gates over $\mathbb{Z}_m$. But it is easy to see that strong and weak representations of degree d correspond to depth-2 and depth-3 circuits in $\mathrm{AC}^0[m]$ respcetively of size n^d for computing a function. Thus, proving strong size lower bounds for even depth-2 circuits in $\mathrm{AC}^0[m]$ *implies* degree lower bounds for polynomial representations. Hence it is plausible to suggest that as long as we are unable to show degree lower bounds for simple computational models such as polynomials, the task of proving circuit bounds is beyond our grasp.

3.1.1 Previous Work

The systematic study of polynomial representations modulo composites was initiated by Barrington, Beigel and Rudich [15]. Barrington [14] conjectured that the OR function requires degree $\Omega(n)$ for weak representations over $\mathbb{Z}_m$ just like in the prime case. This conjecture was disproved by Barrington *et al.*[15], who proved an $O(\sqrt{n})$ upper bound for strong representations. They proved their upper bound by constructing a symmetric polynomial, and they also showed a matching lower bound for symmetric polynomials. The question of whether this lower bound holds for general polynomials is still open. The best lower bound is $\Omega(\log n)$ due to Tardos and Barrington, who also conjecture that the right bound is $\Omega(\sqrt{n})$ [74]. An important open problem in this area is whether asymmetric polynomials can give lower degree representations of symmetric Boolean functions than symmetric polynomials. While there are no known examples of symmetric Boolean functions where asymmetry does help, known lower bounds on the degree are often far from optimal. Following the work of Barrington *et al.*, a number of researchers have worked on this problem [74, 6, 76, 40, 43, 44, 20, 21, 35, 47].

Proving lower bounds is considerably easier for strong representations. Lower bounds of

$\Omega(n)$ are known in the strong representation for functions including the $\text{MOD}-m$ function for suitable values of m [15, 76, 40]. Tsai shows a lower bound of $\Omega(k)$ on the degree of T_k for general polynomials using Moebius inversion [76]. As pointed out by Barrington and Tardos [74] the task of proving lower bounds for strong representations is simplified by the fact that $P(\mathbf{x})$ must equal 0 whenever $f(\mathbf{x})$ is 0. The weak representation seems a more natural definition. Here non-trivial upper bounds are known for many functions, including some functions that are hard for strong representations. Far less is known with regard to lower bounds. The best lower bound known in this case for general polynomials is $\Omega(\log n)$ [41, 74]. Grolmusz [41] proves a $\Omega(\log n)$ lower bound for general polynomials weakly representing a certain function called GIP using a connection to the number on the forehead model from communication complexity.

These polynomials have found surprising combinatorial applications. Grolmusz [43, 42] uses this upper bound to construct a super-polynomial size set system where the size of each set is $0 \bmod 6$ but all pairwise intersections are nonzero mod 6. He uses this to construct explicit Ramsey graphs whose parameters almost match the best known construction. In Chapter 5, we will extend this connection and show that in fact several known Ramsey graph constructions are implicitly based on polynomial representations of the OR function.

3.2 Our Results

In this chapter, we will focus on representations by symmetric polynomials. Since symmetric polynomials can only represent symmetric functions, we must also limit ourselves to symmetric Boolean functions. Henceforth, we will let f denote a symmetric Boolean function on $\{0,1\}^n$. We will also think of f as a function defined on the integers $\{0,\ldots,n\}$. Let $\delta(f)$ denote the the smallest degree of a symmetric polynomial that strongly represents f over $\mathbb{Z}_m$. Define $\Delta(f)$ similarly for weak representations of f . We are interested in bounds on $\delta(f)$ and $\Delta(f)$ for a fixed modulus m as an asymptotic function of the number of variables n.

3.2.1 Symmetric Polynomials and Simultaneous Communication Protocols

The new insight in this chapter is an equivalence between computing Boolean functions by symmetric polynomials modulo m and computing the functions by certain one-round simultaneous communication protocols.

A one-round simultaneous communication protocol [78, 56] involves two players Alice and Bob and a referee. Alice receives an input $\mathbf{x}$, Bob receives an input $\mathbf{y}$ and they wish to compute $f(\mathbf{x}, \mathbf{y}) \in \{0, 1\}$. They cannot directly communicate with each other. They simultaneously write messages on a blackboard. A referee reads the messages and decides the value of f. The players and the referee can agree an a strategy beforehand.

We now introduce two kinds of simultaneous protocols called strong and weak protocols for computing a symmetric Boolean function $f : w \in \{0, \ldots, n\} \rightarrow \{0, 1\}$. We will show that they are equivalent to strong and weak representations respectively.

Definition 3.3 *A strong protocol for computing f mod 6 with parameters (k_2, k_3) is a simultaneous protocol involving two players P_2 and P_3.*

- *P_2 is given $j \equiv w \bmod 2^{k_2}$ as input and outputs $P_2(j)$ in $\mathbb{Z}_2$. P_3 is given $i \equiv w \bmod 3^{k_3}$ as input and outputs $P_3(i)$ in $\mathbb{Z}_3$.*
- *If $f(w) = 0$, then both players must output 0. If $f(w) = 1$, at least one player must output a non-zero value.*
- *The cost of the protocol is $\max(2^{k_2}, 3^{k_3})$.*

Definition 3.4 *A weak protocol for computing f mod 6 with parameters (k_2, k_3) is a simultaneous protocol involving two players P_2 and P_3.*

- *P_2 is given $j \equiv w \bmod 2^{k_2}$ as input and outputs $P_2(j)$ in $\mathbb{Z}_2$. P_3 is given $i \equiv w \bmod 3^{k_3}$ as input and outputs $P_3(i)$ in $\mathbb{Z}_3$.*
- *If $f(w) \neq f(w')$ then at least one of the players outputs different values on w and w'.*
- *The cost of the protocol is $\max(2^{k_2}, 3^{k_3})$.*

For m with t distinct prime factors $p_1, \cdots, p_t$, we define protocols with t players where player[1] P_i reads the input in base p_i. We can think of a strong protocol as one where the referee's strategy is fixed: he outputs 0 iff both players say 0. In a weak protocol, the referee can choose any strategy. The reason for defining the cost as above is that it equals the degree of the polynomial mod 6 that the players are computing, this is explained below.

We now make the connection between symmetric polynomials and simultaneous protocols. Recall that in Theorem 2.15, we showed that the symmetric functions $f : \{0,1\}^n \to \mathbb{Z}_p$ that can be computed by a symmetric polynomial $P(\mathbf{X}) \in \mathbb{Z}_p[X]$ of degree $d < p^l$ are exactly those functions that can be computed from the l least significant digits of w in base p or equivalently, from $w \bmod p^l$. By the Chinese Remainder Theorem (CRT), a degree d symmetric polynomial $P(\mathbf{X})$ over $\mathbb{Z}_6$ corresponds to symmetric polynomials $P_2(\mathbf{X}) \in \mathbb{Z}_2[\mathbf{X}]$ and $P_3(\mathbf{X}) \in \mathbb{Z}_3[\mathbf{X}]$ respectively, each of whose degrees are at most d. By Theorem 2.15, the functions computed by P_2 and P_3 can be computed from $w \bmod 2^{k_2}$ and $w \bmod 3^{k_3}$ respectively, where these are the smallest powers of 2 and 3 which exceed d. Thus there is a protocol of cost $\Theta(d)$.

Conversely assume there exists a protocol for f. Using Theorem 2.15, the function computed by players P_2 and P_3 can be computed by symmetric polynomials $P_2(\mathbf{X}) \in \mathbb{Z}_2[\mathbf{X}]$ and $P_3(\mathbf{X}) \in \mathbb{Z}_3[\mathbf{X}]$ having degree no more than 2^{k_2} and 3^{k_3} respectively. We now use the CRT to combine these polynomials and get a polynomial $P(\mathbf{X}) \in \mathbb{Z}_6[\mathbf{X}]$ of degree bounded by $\max(2^{k_2}, 3^{k_3})$. The value of $P(\mathbf{X})$ tells us the messages sent by both players to the referee. So we can now recover the value of f by applying the referee's strategy. Thus for fixed m, the minimum cost of a protocol for f equals the minimum degree of a symmetric polynomial representing f up to a constant factor depending only on m.

3.2.2 Lower Bounds using Communication Complexity

Techniques from communication complexity have been successfully applied to show lower bounds in many areas like circuit complexity, VLSI and data structures [56]. We show how to adapt tools from communication complexity to prove lower bounds on the degree. These

[1]For notational convenience, we use P_i as opposed to P_{p_i}

tools are especially useful for weak representations. In general, proving deterministic lower bounds for simultaneous communication protocols is easy. There is a simple characterization of the deterministic communication complexity in terms of the number of distinct rows and columns of the input matrix. However, in our setting, proving lower bounds is a non-trivial task because we do not have a very explicit description of the input matrix A^f.

For parameters (k_2, k_3) of the protocol, we define an input matrix A^f of size $2^{k_2} \times 3^{k_3}$, where the (i,j)th entry is $f(w)$ where $w \equiv i \bmod 2^{k_2}$ and $w \equiv j \bmod 3^{k_3}$. Thus, the entries of A^f are defined through the CRT. Thus the value of the (i,j)th entry depends on the parameters (k_2, k_3). Further, we are primarily interested in proving linear lower bounds, which correspond to setting $2^{k_2}, 3^{k_3} = \Omega(n)$. The matrix A^f now has roughly n^2 entries, but only n of these correspond to valid inputs $w \leq n$. To prove lower bounds, we need to carefully choose a submatrix of A^f whose entries are known explicitly and show that it has sufficiently many distinct rows or columns.

We show that any symmetric polynomial that weakly represents Mod-k over $\mathbb{Z}_{pq}$ has degree $\Omega(n)$ where $k > p$ and k is relatively prime to pq. We obtain a linear lower bound for the Mod-k function when m has $t > 2$ distinct prime factors for sufficiently large k. This is proved by a reduction to computing the function Exactly-k in the *number on the forehead* model and using a lower bound by Chandra, Furst and Lipton [27]. We give a necessary and sufficient condition for the existence of a strong protocol for a function f. We use this to give simple proofs of known bounds on strong representations for symmetric polynomials. We show a separation between strong and weak representations by constructing a function f which can be weakly represented by polynomials of degree $O(\sqrt{n})$ but both f and $\overline{f}$ need degree $\Omega(n)$ for strong representation.

3.2.3 Threshold Functions and Diophantine Equations

The Threshold-k function T_k is defined to be 1 if the weight of the input is at least k. We study the degree of the Threshold-k function (T_k) for various values of k. T_1 is the OR function and $\delta(T_1) = \Delta(T_1) = \Theta(\sqrt{n})$. T_n is the AND function. It is easy to show that $\delta(T_n) = \Omega(n)$, but $\Delta(T_n) = \Theta(\sqrt{n})$. This raises the question: What is the (strong/weak)

degree of T_k is for $1 < k < n$?

We show that proving bounds on the degree is equivalent to showing that certain Diophantine equations have only finitely many solutions. More precisely, we show that there exists a strong protocol for T_k on n variables with parameters k_2, k_3 iff there are no non-trivial solutions to the equation

$$|a2^{k_2} - b3^{k_3}| = \ell \qquad\qquad a2^{k_2} \leq n,\ b3^{k_3} \leq n,\quad \ell < k$$

When k is a fixed constant, we show that $\delta(T_k) = O(n^{\frac{1}{2}+\varepsilon})$ for any $\varepsilon > 0$. The proof uses a result of Filaseta [30] on factors of numbers of the form $n(n+d)$. We show $\delta(T_k) = O(n^{\frac{1}{t}+\varepsilon})$ when m has t distinct prime factors using a theorem due to Granville [38] which is proved assuming the *abc* conjecture from number theory [39]. We also show that when m has only two prime divisors, the *abc* conjecture implies that $\delta(T_k) = O(nk)^{\frac{1}{2}+\varepsilon}$ for all values of k.

The $O(\sqrt{n})$ upper bound for the OR function can be interpreted as follows: For suitably chosen parameters (k_2, k_3) if $w \bmod 2^{k_2}$ and $w \bmod 3^{k_3}$ are both zero, then in fact the number w must equal 0. Our bounds for T_k give a similar result about the size of w: For suitably chosen parameters (k_2, k_3) if the residues $w \bmod 2^{k_2}$ and $w \bmod 3^{k_3}$ are both less than k, then in fact they are both equal to w itself and $w < k$. Conversely, if $w \geq k$, then one of the residues must be large.

We show a lower bound of $\Omega(n^{\frac{1}{t}} k^{\frac{t-1}{t}})$ for $\delta(T_k)$ over $\mathbb{Z}_m$. This improves the previous bound of $\Omega(\max(k, \sqrt{n}))$ [76]. When $t = 2$, the lower bound nearly matches the upper bound of $(nk)^{\frac{1}{2}+\varepsilon}$ for all values of k. These lower bounds are proved by constructing solutions to the equation above via a pigeonhole argument. Further, when $t = 2$, we also show an $\Omega(\sqrt{nk})$ lower bound for $\Delta(T_k)$.

3.2.4 Organization of This Chapter

We prove the equivalence between polynomials and protocols in Section 3.3. We prove degree bounds for strong representations in Section 3.4 and weak representations in Section 3.5. We study Threshold functions in detail in Sections 3.6 and 3.7 respectively.

3.3 Symmetric Polynomials and Simultaneous Protocols

In this section, we prove our main theorem equating polynomial representations with simultaneous communication protocols.

Theorem 3.1 *There exists a symmetric polynomial over $\mathbb{Z}_m$ of degree d that strongly (weakly) represents f iff there exists a strong (weak) protocol of cost $\Theta(d)$ over $\mathbb{Z}_m$ for computing f.*

PROOF: The constant implicit in that $\Theta(d)$ depends only on m, and can be taken to be $max_i p_i$. We prove the theorem assuming $m = 6$. We prove the equivalence for the strong case, the weak case is similar. Let

$$P(\mathbf{X}) = \sum_{i=0}^{d} a_i S_i(\mathbf{X})$$

be a symmetric polynomial of degree d over $\mathbb{Z}_6$ that strongly represents f. We will construct a strong protocol for computing f with cost at most $3d$. Let $b_i \equiv a_i \bmod 2, c_i \equiv a_i \bmod 3$. Define polynomials $P_2(\mathbf{X})$ over $\mathbb{Z}_2$ and $P_3(\mathbf{X})$ over $\mathbb{Z}_3$ respectively as

$$\begin{aligned} P_2(\mathbf{X}) &\stackrel{\triangle}{=} \sum_{i=0}^{d} b_i S_i(\mathbf{X}) \\ P_3(\mathbf{X}) &\stackrel{\triangle}{=} \sum_{i=0}^{d} c_i S_i(\mathbf{X}) \end{aligned}$$

Both $P_2(\mathbf{X})$ and $P_3(\mathbf{X})$ are symmetric polynomials of degree at most d. Choose k_2, k_3 so that $d < 2^{k_2} \leq 2d, d < 3^{k_3} \leq 3d$. By Theorem 2.15, value of $P_2(\mathbf{x})$ on a 0-1 input $\mathbf{x}$ depends on just the first k_2 bits of the weight w in base 2. This function is computed by player P_2. The function computed by $P_3(\mathbf{x})$ on a 0-1 input $\mathbf{x}$ depends on just the first k_3 digits of w in base 3. This is computed by player P_3. We show that this indeed gives a strong protocol.

Let $\mathbf{x} \in \{0,1\}^n$. Since $P(\mathbf{X})$ strongly represents f,

$$\begin{aligned} f(\mathbf{x}) = 0 \quad &\Rightarrow \quad P(\mathbf{x}) \equiv 0 \bmod 6 \\ &\Rightarrow \quad P_2(\mathbf{x}) \equiv 0 \bmod 2 \quad \text{and} \quad P_3(\mathbf{x}) \equiv 0 \bmod 3 \qquad \text{by the CRT} \\ f(\mathbf{x}) = 1 \quad &\Rightarrow \quad P(\mathbf{x}) \not\equiv 0 \bmod 6 \\ &\Rightarrow \quad P_2(\mathbf{x}) \not\equiv 0 \bmod 2 \quad \text{or} \quad P_3(\mathbf{x}) \not\equiv 0 \bmod 3 \qquad \text{by the CRT} \end{aligned}$$

Hence we have a strong protocol of cost $\max(2^{k_2}, 3^{k_3}) \leq 3d$.

Conversely assume there exists a protocol for f with parameters (k_2, k_3). The function computed by P_2 depends on only the first k_2 bits of the weight w. So it can be computed by a symmetric polynomial $P_2(\mathbf{X})$ in $\mathbb{Z}_2[\mathbf{X}]$ of degree less than 2^{k_2} by Theorem 2.15. Similarly the function computed by P_3 can be computed by a symmetric polynomial $P_3(\mathbf{X})$ in $\mathbb{Z}_3[\mathbf{X}]$ of degree less than 3^{k_3}. Let

$$\begin{aligned} P_2(\mathbf{X}) &= \sum_{i=0}^{d} b_i S_i(\mathbf{X}) \\ P_3(\mathbf{X}) &= \sum_{i=0}^{d} c_i S_i(\mathbf{X}) \end{aligned}$$

By the CRT, we can pick $a_i \in \mathbb{Z}_6$ such that

$$a_i \equiv b_i \bmod 2, \quad a_i \equiv c_i \bmod 3$$

Now set

$$P(\mathbf{X}) = \sum_{i=0}^{d} a_i S_i(\mathbf{X})$$

We will show that $P(\mathbf{X})$ strongly represents f mod 6. If $f(w) = 0$, then both players P_2 and P_3 output 0 on w. Hence if $\mathbf{x} \in \{0,1\}^n$ has weight w, then by the CRT

$$P_2(\mathbf{x}) \equiv 0 \bmod 2 \text{ and } P_3(\mathbf{x}) \equiv 0 \bmod 3 \quad \Rightarrow \quad P(\mathbf{x}) \equiv 0 \bmod 6$$

If $f(w) = 1$, then at least one of P_2 and P_3 outputs a non-zero value on w. Hence if $\mathbf{x} \in \{0,1\}^n$ has weight w, then by the CRT

$$P_2(\mathbf{x}) \not\equiv 0 \bmod 2 \text{ or } P_3(\mathbf{x}) \not\equiv 0 \bmod 3 \quad \Rightarrow \quad P(\mathbf{x}) \not\equiv 0 \bmod 6$$

$P(\mathbf{X})$ is a symmetric polynomial of degree $d < \max(2^{k_2}, 3^{k_3})$. □

Using this theorem, we will prove both upper and lower bounds on the degrees of polynomials for both representations by viewing them as simultaneous communication protocols. We first need some notation. Recall that player P_2 receives $i \equiv w \bmod 2^{k_2}$ and player P_3 receives $j \equiv w \bmod 3^{k_3}$ and they wish to compute $f(w)$. If $2^{k_2} 3^{k_3} \leq n$ there might be multiple values of w between 0 and n satisfying $w \equiv i \bmod 2^{k_2}$ and $w \equiv j \bmod 3^{k_3}$. If $f(w)$

is not the same for all these values, then clearly no protocol with parameters k_2, k_3 exists. Hence assume that the value of f is well defined for every pair (i, j) of possible residues of w. We can define a $2^{k_2} \times 3^{k_3}$ input matrix $A = (a_{ij})$ as follows.

$$\begin{aligned} 0 \leq \quad a_{ij} \quad &\leq 2^{k_2} 3^{k_3} - 1 \\ a_{ij} \quad &\equiv \quad i \bmod 2^{k_2}, \qquad 0 \leq i < 2^{k_2} \\ a_{ij} \quad &\equiv \quad j \bmod 3^{k_3}, \qquad 0 \leq j < 3^{k_3} \end{aligned}$$

We use a_{ij} in place of w since some values a_{ij} could be greater than n. P_2 receives the same input i for all inputs in the same row of A and hence outputs the same value. Similarly inputs in a column are indistinguishable to P_3. Where convenient, we will refer to P_2 and P_3 as the row and column player respectively. For a function f, we then define the $2^{k_2} \times 3^{k_3}$ matrix A^f as below.

$$A^f_{ij} = \begin{cases} f(a_{ij}) & 0 \leq a_{ij} \leq n \\ \text{x} & a_{ij} > n \end{cases}$$

The symbol 'x' indicates that the function is not defined for this value of weight.

In the usual communication complexity setting, there is a fixed function f and a corresponding matrix A^f and we wish to know its communication complexity. In our setting however, the matrix A^f and hence the communication complexity of f depends on k_2 and k_3. There are restrictions on the values that the players can output since $P_2(i) \in \mathbb{Z}_2$ and $P_3(j) \in \mathbb{Z}_3$. As k_2 and k_3 increase, the amount of communication needed can only decrease. For instance, if one player reads all the bits of the input, she could compute $f(w)$ herself and write it on the board. Our goal is now to determine the smallest values of k_2 and k_3 so that the players can compute f with the restrictions on output size.

3.4 Strong Representations

In this section, we present some simple upper and lower bounds for strong representations. A weak representation for f is also a representation for $\overline{f}$ and so $\Delta(f) = \Delta(\overline{f})$, but this need not be true for $\delta(f)$. A strong representation is a special case of a weak representation hence $\Delta(f) \leq \min(\delta(f), \delta(\overline{f}))$.

3.4.1 Lower Bounds

We begin with a proof of the theorem by Barrington *et al.*. We use the following convention throughout this section, the results are stated for general $m = \prod_{i \leq t} p_i$ with t prime divisors. We present the proof only for $m = 6$ when the extension to general m is obvious.

Theorem 3.2 *[15] Over* $\mathbb{Z}_m$ $\delta(OR) = O(n^{\frac{1}{t}})$.

PROOF: We give a strong protocol for OR over $\mathbb{Z}_6$ of cost $\leq 3\sqrt{n}$.

Protocol 3.3 Protocol for OR mod 6

- *Choose* k_2 *and* k_3 *s.t.* $\sqrt{n} < 2^{k_2} \leq 2\sqrt{n}$ *and* $\sqrt{n} < 3^{k_3} \leq 3\sqrt{n}$.
- *If* $i = 0$ *then* $P_2(i) = 0$ *else* $P_2(i) = 1$.
- *If* $j = 0$ *then* $P_3(j) = 0$ *else* $P_3(j) = 1$.

To prove correctness, we need to show that if both players output 0, $w = 0$.

$$w \equiv 0 \bmod 2^{k_2}, \quad w \equiv 0 \bmod 3^{k_3} \Rightarrow w \equiv 0 \bmod 2^{k_2}3^{k_3} \quad \text{(by CRT)}$$

By our choice of (k_2, k_3), $2^{k_2}3^{k_3} > n$ but $w \leq n$. Hence $w = 0$. □

Proposition 3.4 *[15] Over* $\mathbb{Z}_m$ $\Delta(OR) = \Omega(n^{\frac{1}{t}})$.

PROOF: We show that any weak protocol for OR has cost $\Omega(\sqrt{n})$. If $2^{k_2}3^{k_3} \leq n$ then $i = j = 0$ for inputs of weight 0 and $2^{k_2}3^{k_3}$. Hence any protocol will output the same value on these inputs. However, $f(0) \neq f(2^{k_2}3^{k_3})$. So $2^{k_2}3^{k_3} > n$ which implies that $\max(2^{k_2}, 3^{k_3}) > \sqrt{n}$. □

To prove lower bounds better than $\sqrt{n}$ for other functions, we need stronger techniques. The output of a strong protocol on input a_{ij} is zero iff $P_2(i) = P_3(j) = 0$. Hence there exists a protocol for f with parameters (k_2, k_3) iff there exist $I \subset \{0, \cdots, 3^{k_3} - 1\}$ and $J \subset \{0, \cdots, 2^{k_2} - 1\}$ such that

- If $f(a_{ij}) = 0$ then $i \in I, j \in J$.

- If $f(a_{ij}) = 1$ then $i \notin I$ or $j \notin J$.

In other words, all the 0s in A^f must be contained in a single rectangle with no 1s in it. This gives the following necessary and sufficient condition for the existence of a strong protocol. This is essentially a translation of the fooling-set argument from communication complexity [56].

Lemma 3.5 *There is a strong protocol for f with parameters (k_2, k_3) iff $\forall\ i,\ j$ such that $f(a_{ij}) = 1$, either there are no 0s in row i or there are no 0's in column j of A^f.*

PROOF: Assume there exist i, j such that $f(a_{ij}) = 1$ but there are 0s in both row i and column j of A^f. The row player must answer 0 on row i since it contains a 0. Similarly the column player must answer 0 on column j. Hence they both answer 0 on a_{ij} so the protocol is incorrect. Conversely, if row i does not have any 0s, the row player can answer 1 on input i and similarly for the column player. This gives a strong protocol for f. $\square$

Lemma 3.5 gives a condition to test whether a protocol with parameters k_2, k_3 exists. Moreover, it follows from the proof that if the condition is satisfied, Protocol 3.6 given below works correctly. Conversely, if $\delta(f) > \max(2^{k_2}, 3^{k_3})$, then there must be an input w on which Protocol 3.6 with parameters k_2, k_3 is incorrect.

Protocol 3.6 Strong Protocol for general function f

- *If $\exists w \leq n$ such that $w \equiv i \bmod 2^{k_2}$ and $f(w) = 0$ then $P_2(i) = 0$. Else $P_2(i) = 1$.*
- *If $\exists w \leq n$ such that $w \equiv j \bmod 3^{k_3}$ and $f(w) = 0$ then $P_3(j) = 0$. Else $P_3(j) = 1$.*

Let m have t prime factors $p_1, \cdots, p_t$. To extend Lemma 3.5 to t player protocols, we use the notion of a *star*. Our notion of a star is different from the notion used in multi-party protocols [56].

Definition 3.5 *Fix parameters $k_1, \cdots, k_t$. A star in the input matrix A is a set of $t+1$ distinct inputs $w_0, \cdots, w_t \leq n$ such that $w_0 \equiv w_u \bmod p_u^{k_u}$ for $1 \leq u \leq t$. The input w_0 is called the center of the star and $w_1, \cdots w_t$ are called the endpoints.*

Unlike in the multi-party protocol setting, we require the endpoints of the star to agree with the center only on a single co-ordinate. Since we are interested in proving lower bounds of the form $\Omega(n)$, distinct inputs can agree modulo at most one prime power (by the CRT). We prove a condition for the existence of a strong protocol over $\mathbb{Z}_m$ which generalizes Lemma 3.5.

Lemma 3.7 *There exists a strong protocol for computing f over $\mathbb{Z}_m$ with parameters $k_1, \cdots, k_t$ iff there does not exist a star $w_0, \cdots, w_t$ such that $f(w_0) = 1$ and $f(w_u) = 0$ for $1 \leq u \leq t$.*

PROOF: Assume that such a star exists. Then player P_u must answer 0 on input $w_u \bmod p_u^{k_u}$ since $f(w_u) = 0$. This implies that every player outputs 0 on input w_0 since $w_0 \equiv w_u \bmod p_u^{k_u}$. But $f(w_0) = 1$ and so the protocol is incorrect.

Conversely, assume that a star satisfying these conditions does not exist. Then for every w_0 such that $f(w_0) = 1$, there exists an index u such that

$$\forall w_u \text{ s.t. } w_u \equiv w_0 \bmod p_u^{k_u}, \quad f(w_u) = 1 \tag{12}$$

Now consider the following extension of Protocol 3.6 to t players.

On input $j \in [0, \cdots, p_u^{k_u} - 1]$, if $\exists w \leq n$ such that $w \equiv j \bmod p_u^{k_u}$ and $f(w) = 0$ then player P_u outputs 0. Else P_u outputs 1.

If $f(w) = 0$, then every player outputs 0 on input $w \bmod p_u^{k_u}$. If $f(w) = 1$, by Equation 12 there exists u such that $\forall w_u \equiv w \bmod p_u^{k_u}, f(w_u) = 1$. Hence P_u outputs 1 on input $w \bmod p_u^{k_u}$ and the protocol is correct.

□

We can use the above Lemmas to prove degree bounds for various functions. Define the Weight-k function W_k on $\{0,1\}^n$ as $W_k(\mathbf{x}) = 1$ if $w(\mathbf{x}) = k$ and 0 otherwise. Using

an argument similar to the one used for the OR function, one can show that over $\mathbb{Z}_m$, $\delta(\overline{W}_k) = \Theta(n^{\frac{1}{t}}$. We now show bounds on $\delta(W_k)$.

Corollary 3.8 *Over* $\mathbb{Z}_m$, $\delta(W_k) = \Omega(n)$.

PROOF: Let $2^{k_2} \leq \frac{n}{2}, 3^{k_3} \leq \frac{n}{2}$. Assume $k \geq \frac{n}{2}$. Set

$$b = k - 2^{k_2}, \quad c = k - 3^{k_3}$$

Observe that b lies in the same column as k while c lies in the same row. But now

$$f(k) = 1, \quad f(b) = 0, \quad f(c) = 0$$

Hence by Theorem 3.5 such a protocol does not exist. Hence $\max(2^{k_2}, 3^{k_3}) > \frac{n}{2}$. When $k < \frac{n}{2}$, repeat the same argument with $b = k + 2^{k_2}$ and $c = k + 3^{k_3}$. □

Next we prove a simple lower bound for the Threshold-k function. In the next section, we will improve this lower bound to $\Omega(n^{\frac{1}{t}} k^{\frac{t-1}{t}})$ using more sophisticated techniques.

Corollary 3.9 *Over* $\mathbb{Z}_m$, $\delta(T_k) = \Omega(\max(k, n^{\frac{1}{t}})$.

PROOF: We first show a lower bound of $\Omega(\sqrt{n})$ over $\mathbb{Z}_6$. Suppose $2^{k_2} 3^{k_3} \leq n$. We can choose a w so that $w < k \leq w + 2^{k_2} 3^{k_3}$. Both players receive the same inputs for weights w and $w + 2^{k_2} 3^{k_3}$ but $T_k(w) = 0$ while $T_k(w + 2^{k_2} 3^{k_3}) = 1$. Hence the protocol is incorrect. This proves a lower bound of $\sqrt{n}$.

Now suppose $\max(2^{k_2}, 3^{k_3}) < k$. Consider any $w \geq k$. Since $i \equiv w \bmod 2^{k_2}$ and $2^{k_2} < k, i < k$. Similarly $j < k$. The entry i lies in the same row as w while j lies in the same column.

$$T_k(w) = 1, \quad T_k(i) = 0, \quad T_k(j) = 0$$

Now apply Theorem 3.5. Hence $\max(2^{k_2}, 3^{k_3}) > k$. □

Corollary 3.10 *Over* $\mathbb{Z}_m$, $\delta(\overline{T}_k) = \Omega(n)$ *for* $k \leq \frac{n}{2}$.

PROOF: Assume that $2^{k_2}, 3^{k_3} \leq \frac{n}{2}$. There exist multiples $a2^{k_2}, b3^{k_3}$ so that

$$k \leq \frac{n}{2} \leq a2^{k_2}, \quad b3^{k_3} \leq n$$

Now observe that $a2^{k_2}$ and $b3^{k_3}$ are in the same row and column respectively as 0, and

$$\overline{T}_k(0) = 1, \quad \overline{T}_k(a2^{k_2}) = 0, \quad \overline{T}_k(b3^{k_3}) = 0$$

Hence by Theorem 3.5, $\max(2^{k_2}, 3^{k_3}) > \frac{n}{2}$. □

Recall that we defined the Mod-k function M_k on $\{0,1\}^n$ as $M_k(\mathbf{x}) = 1$ if $w(\mathbf{x} \equiv 0 \bmod k$ and 0 otherwise. We can show that if $k \neq 2^a 3^b$ both M_k and its complement have $\delta = \Theta(n)$. If $k = 2^a 3^b$ then $\overline{M}_k$ has degree $O(1)$ while M_k has degree $\Theta(n)$. We skip the proof.

3.5 Weak Representations

In this section we will show lower bounds of $\Omega(n)$ for weak representations of various functions using tools from communication complexity. The lower bounds of $\Omega(n^{\frac{1}{t}})$ do not make use of the *simultaneous* nature of the protocol, the same bounds would hold even if the players were allowed to send their inputs to each other. To prove bounds of $\Omega(n)$, we exploit the fact that the players cannot communicate and there are restrictions on their output size.

3.5.1 Lower Bounds for Two Player Protocols

Using a classical result about deterministic simultaneous communication protocols, we give a necessary and sufficient condition for the existence of a weak protocol in terms of the number of distinct rows and columns in A^f.

Definition 3.6 *Two rows i, i' in the matrix A^f are distinct, if there exists a column index j such that $a_{ij}, a_{i'j} \leq n$ and $f(a_{ij}) \neq f(a_{i'j})$. Rows $i_1, \ldots, i_k$ are said to be distinct if they are pairwise distinct.*

Lemma 3.11 *For a weak protocol for f over $\mathbb{Z}_{pq}$ with parameters (k_p, k_q) to exist, the matrix A^f must have at most p distinct rows and q distinct columns.*

PROOF: We will show that over $\mathbb{Z}_6$, A^f can have at most 2 distinct rows and 3 distinct columns. Assume that there are at least 3 distinct rows. Since P_2 must output a value in

$\mathbb{Z}_2$, she outputs the same value for some two distinct rows i, i'. Since the rows are distinct, there is a column index j such that $a_{i,j}, a_{i',j} \leq n$ and $f(a_{ij}) \neq f(a_{i'j})$. Player P_3 will also output the same value for inputs a_{ij} and $a_{i'j}$ since they lie in the same column. This violates the definition of a weak protocol. □

Recall that we define the function M_k on $\{0,1\}^n$ as $M_k(a) = 1$ if $a \equiv 0 \bmod k$ and 0 otherwise.

Theorem 3.12 *Let $(k,p) = (k,q) = 1$ and $k > min(p,q)$. Over $\mathbb{Z}_{pq}$, $\Delta(M_k) = \Omega(n)$.*

PROOF: We consider the case $k = 5, p = 2, q = 3$. The general case is similar. The values of k_2 and k_3 will be determined later. We exhibit a 3×3 submatrix V of A such that V^f is the identity matrix.

$$V = \begin{pmatrix} 0 & a_1 2^{k_2} & a_2 2^{k_2} \\ b_1 3^{k_3} & a_1 2^{k_2} + b_1 3^{k_3} & a_2 2^{k_2} + b_1 3^{k_3} \\ b_2 3^{k_3} & a_1 2^{k_2} + b_2 3^{k_3} & a_2 2^{k_2} + b_2 3^{k_3} \end{pmatrix}$$

Elements in the same row of V have the same residue modulo 2^{k_2} and elements in a column have the same residue modulo 3^{k_3}. So V is a submatrix of A. Since $2^{k_2}, 3^{k_3} \neq 0 \bmod 5$, we can find $a_1, a_2, b_1, b_2 < 5$ s.t.

$$\begin{aligned} a_1 2^{k_2} &\equiv 1 \bmod 5 & a_2 2^{k_2} &\equiv 2 \bmod 5 \\ b_1 3^{k_3} &\equiv -1 \bmod 5 & b_2 3^{k_3} &\equiv -2 \bmod 5 \end{aligned}$$

Hence

$$V^f = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This implies that A^f has at least 3 different rows and a weak protocol cannot exist by Lemma 3.11. To ensure that all entries are at most n, we pick k_2, k_3 such that

$$4(2^{k_2} + 3^{k_3}) \leq n.$$

To satisfy this, we can take

$$\frac{n}{16} \leq 2^{k_2} \leq \frac{n}{8}, \quad \frac{n}{24} \leq 3^{k_3} \leq \frac{n}{8}, \quad \Rightarrow \quad \min(2^{k_2}, 3^{k_3}) \geq \frac{n}{24}.$$

For T_k over $\mathbb{Z}_{pq}$ the lower bound obtained is $\Omega(\frac{n}{kq})$. □

The same proof works for the Mod-ℓ function where $\ell = p^a q^b k$ provided $(k,p) = (k,q) = 1$ and $k > \min(p,q)$. The condition $k > \min(p,q)$ implies that we cannot for instance show that Mod-2 is hard over $\mathbb{Z}_{15}$. This was left as an open problem in [20]. This was resolved recently by Hansen [47] who showed that Mod-2 is indeed hard over $\mathbb{Z}_{15}$, but not over $\mathbb{Z}_{21}$. In general, he shows that some assumptions about the size of k are required in order to prove a lower bound.

We now show a lower bound for Threshold functions in the two player case.

Theorem 3.13 *Over $\mathbb{Z}_{pq}$, $\Delta(T_k) = \Omega(\max(k, \sqrt{n}))$ for $k \leq \frac{n}{pq}$.*

PROOF: A lower bound of $\sqrt{n}$ is easy to show for all k as in the proof of Corollary 3.9. So we assume that $k > \sqrt{n}$. We consider the case of $\mathbb{Z}_6$. Let $2^{k_2}, 3^{k_3} < k$ and let $3^{k_3+1} \geq k$. We define

$$\begin{aligned} \overline{a} &\equiv 3^{k_3+1} \bmod 2^{k_2} \\ \overline{2a} &\equiv 2 \cdot 3^{k_3+1} \bmod 2^{k_2} \end{aligned}$$

Since $2^{k_2} < k$, $\overline{a} < k$ and $\overline{2a} < k$. Now set

$$\begin{aligned} V &= \begin{pmatrix} 0 & \times & \times \\ 3^{k_3+1} & \overline{a} & \times \\ 2 \cdot 3^{k_3+1} & \overline{a} + 3^{k_3+1} & \overline{2a} \end{pmatrix} \\ \Rightarrow V^f &= \begin{pmatrix} 0 & \times & \times \\ 1 & 0 & \times \\ 1 & 1 & 0 \end{pmatrix} \end{aligned}$$

Clearly V^f has at least three distinct rows for all settings of the x's. To ensure that the entries of V are at most n we need $2 \cdot 3^{k_3+1} < n$. This is possible provided $k \leq \frac{n}{6}$. In the case of $\mathbb{Z}_{pq}$, we can construct a similar matrix of size $(p+1) \times (p+1)$ provided $k \leq \frac{n}{pq}$. □

In Section 5, we will improve this bound to $\Omega(\sqrt{kn})$ for $k \leq \frac{n}{p}$.

3.5.2 Multi-player Protocols for Mod-k

We now consider the case when m has $t > 2$ distinct prime factors and the protocols involve t players. We show a lower bound for the t player case by a reduction to the function Exactly-k in the number on the forehead model. There is a lower bound of $\omega(1)$ on the deterministic complexity of Exactly-k due to Chandra, Furst and Lipton [27]. We first need some results from the number on the forehead model. There are t players $P_1, \ldots, P_t$ and t inputs $x_1, \ldots, x_t$. Player P_j receives inputs x_i for all $i \neq j$. They wish to compute some function $f(x_1, \ldots, x_t)$. $D(f)$ denotes the deterministic complexity of the function f. For further definitions about the model as well as an exposition of the result of Chandra *et al.*, see [56].

Definition 3.7 *For $x_1, \cdots, x_t \in \{0, \cdots, k-1\}$, the Exactly-k function $E_k^t(x_1, \cdots, x_t) = 1$ iff $\sum_{i=1}^t x_i = k$.*

Theorem 3.14 *[27] $D(E_k^t(x_1, \cdots, x_t)) = \omega(1)$.*

Here $\omega(1)$ means that for t fixed, the value of $D(E_k^t(x_1, \cdots, x_t))$ goes to infinity as k tends to infinity. We now define the function M_k^t in the number on the forehead model which should not be confused with the Mod-k function on Boolean inputs.

Definition 3.8 *For $x_1, \cdots, x_t \in \{0, \cdots, k-1\}$, $M_k^t(x_1, \cdots, x_t) = 1$ iff $\sum_{i=1}^t x_i \equiv 0 \bmod k$.*

Lemma 3.15 *$D(M_k^t(x_1, \cdots, x_t)) = \omega(1)$.*

PROOF: We prove the lower bound by reducing computing E_k^t to computing M_k^t. Let $S = \sum_{i=1}^t x_i$. Assume the players have a protocol for M_k^t. If they run this protocol and find that $M_k^t(x_1, \cdots, x_t) = 0$, then $S \neq 0 \bmod k$. Hence $S \neq k$, which implies $E_k^t(x_1, \cdots, x_t) = 0$.

If $M_k^t(x_1, \cdots, x_t) = 1$, then $S \in \{0, k, 2k, \cdots, (t-1)k\}$. However player P_1 (or any other player) can distinguish between these outcomes. In particular if $S = k$, then $1 \leq S - x_1 \leq k$. If $S = Ck$ where $C \neq 1$ then $S - x_1$ cannot take values between 1 and k . But $S - x_1 = \sum_{i=2}^t x_i$ and this can be computed by P_1. P_1 writes one additional bit on the blackboard which tells the referee whether $1 \leq S - x_1 \leq k$. Hence $D(E_k^t) \leq D(M_k^t) + 1$. But now by Theorem 3.14, $D(M_k^t) = \omega(1)$. □

We now prove a lower bound for Mod-k in the t player case.

Theorem 3.16 *Over $\mathbb{Z}_m$, for k sufficiently large as a function of m and $(k,m) = 1$, $\Delta(M_k) = \Omega(n)$.*

PROOF: We consider the case of $\mathbb{Z}_{30}$. The protocols now have three players P_2, P_3 and P_5 who receive $y_2 = w \bmod 2^{k_2}, y_3 = w \bmod 3^{k_3}$ and $y_5 = w \bmod 5^{k_5}$ respectively.

We identify a fooling set comprising of a subset of the inputs. We will show that on this subset, the problem can be reduced to computing M_k^t in the number on the forehead model. The fooling set consists of inputs $a2^{k_2} + b3^{k_3} + c5^{k_5}$ where $a, b, c \in \{0, \cdots, k-1\}$. The values of k_2, k_3 and k_5 will be set later. The inputs received by P_2, P_3 and P_5 respectively are

$$\begin{aligned} u &\equiv b3^{k_3} + c5^{k_5} \bmod 2^{k_2} \\ v &\equiv a2^{k_2} + c5^{k_5} \bmod 3^{k_3} \\ w &\equiv a2^{k_2} + b3^{k_3} \bmod 5^{k_5} \end{aligned}$$

We can give $b3^{k_3}$ and $c5^{k_5}$ as inputs to P_2 since the value of u can be computed from this. Since $(3^{k_3}, k) = 1$ and $b \in \{0, \cdots, k-1\}$, there is a one-to-one correspondence between the numbers $b3^{k_3}$ and $\{0, \cdots, k-1\}$. Hence it is sufficient to give P_2 the inputs

$$\begin{aligned} x_3 &\equiv b3^{k_3} \bmod k \\ x_5 &\equiv c5^{k_5} \bmod k \end{aligned}$$

The values of $b3^{k_3}$ and $c5^{k_5}$ can be recovered from x_3 and x_5 respectively. Similarly set $x_2 \equiv a2^{k_2} \bmod k$. Observe that now $0 \leq x_i \leq k-1$, player P_i has inputs x_j for all $i \neq j$ and they wish to know if the sum of the x_i's is 0 mod k. Thus we have a reduction to the problem of computing $M_k^3(x_2, x_3, x_5)$ in the number in the forehead model.

In any weak protocol over $\mathbb{Z}_{30}$, the number of bits of communication available to the players is bounded by a fixed constant ($\log_2 30$). Lemma 3.15 implies that for k sufficiently large, this is insufficient, hence a weak protocol cannot exist. We now set k_2, k_3 and k_5 so that the entries in our fooling set are no larger than n. The largest entry is bounded by $k(2^{k_2} + 3^{k_3} + 5^{k_5})$. So we set each of $2^{k_2}, 3^{k_3}, 5^{k_5} < \frac{n}{3k}$. This gives a lower bound on the degree of $\Omega(\frac{n}{k})$.

In general over $\mathbb{Z}_m$ where m has t distinct prime factors, we choose k large enough so that $D(M_k^t) \geq \log_2 m$. We then choose a fooling set of inputs of the form $\sum_{i \leq t} a_i p_i^{k_i}$. To ensure that these numbers are less than n, take $p_i^{k_i} \leq \frac{n}{t}$. This gives a degree bound of $\Omega(\frac{n}{tkp_t})$.

□

3.5.3 Separating Strong and Weak Representations

In a strong protocol, w.l.o.g. the players output either 0 or 1. The referee's strategy is fixed. On the other hand, in a weak protocol, a player can output a value from $\mathbb{Z}_p$. The referee is allowed to choose any strategy. A natural question therefore is whether weak protocols are actually more powerful than strong protocols. Recall that $\Delta(f) \leq \min(\delta(f), \delta(\overline{f}))$. We will show a gap between these quantities by constructing a function f such that $\Delta(f) = O(\sqrt{n})$ but $\min(\delta(f), \delta(\overline{f})) = \Omega(n)$.

Choose l_2, l_3 such that $\sqrt{n} < 2^{l_2} \leq 2\sqrt{n}$ and $\sqrt{n} < 3^{l_3} \leq 3\sqrt{n}$. Define $f : \{0, \cdots, n\} \rightarrow \{0, 1\}$ by

$$f(w) = \begin{cases} 1 & \text{exactly one of } 2^{l_2}, 3^{l_3} \text{ divides } w \\ 0 & \text{otherwise} \end{cases}$$

Since $2^{l_2} 3^{l_3} > n$, if both 2^{l_2} and 3^{l_3} divide w, then $w = 0$.

Lemma 3.17 *Over* $\mathbb{Z}_6$, $\min(\delta(f), \delta(\overline{f})) = \Omega(n)$.

PROOF: Let $2^{k_2} + 3^{k_3} \leq n$. Set $m_2 = \max(k_2, l_2)$ and $m_3 = \max(k_3, l_3)$. Observe that

$$\begin{aligned} 2^{m_2} &\equiv 0 \bmod 2^{l_2}, & 2^{m_2} &\not\equiv 0 \bmod 3^{l_3} \\ 3^{m_3} &\not\equiv 0 \bmod 2^{l_2}, & 3^{m_3} &\equiv 0 \bmod 3^{l_3} \\ 2^{m_2} + 3^{m_3} &\not\equiv 0 \bmod 2^{l_2}, & 2^{m_2} + 3^{m_3} &\not\equiv 0 \bmod 3^{l_3} \end{aligned}$$

We now consider the matrix

$$V = \begin{pmatrix} 0 & 3^{m_3} \\ 2^{m_2} & 2^{m_2} + 3^{m_3} \end{pmatrix} \quad \Rightarrow$$

$$V^f = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad V^{\overline{f}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Hence by Lemma 3.5, both f and $\overline{f}$ have strong degree $\Omega(n)$. $\square$

Lemma 3.18 *Over* $\mathbb{Z}_6$, $\Delta(f) = O(\sqrt{n})$.

Protocol 3.19 Weak Protocol for function f

- *Set* $k_2 = l_2$ *and* $k_3 = l_3$.
- *If* $i = 0$ *then* $P_2(i) = 0$ *else* $P_2(i) = 1$.
- *If* $j = 0$ *then* $P_3(j) = 0$ *else* $P_3(j) = 1$.
- *The output of the protocol is* 1 *if* $P_2(i) = P_3(j)$ *and* 0 *if* $P_2(i) \neq P_3(j)$.

It is easy to see that the above protocol computes f with cost $O(\sqrt{n})$. The referee's strategy is to take the XOR of the players outputs which cannot be done in a strong protocol.

Theorem 3.20 *There exists a function* f *for which* $\Delta(f) = O(\sqrt{n})$ *whereas* $\delta(f)$ *and* $\delta(\overline{f})$ *are* $\Theta(n)$.

3.6 Threshold Functions and Diophantine Equations

We now begin a detailed study of the degree of the Threshold-k function for values of k between 1 and n. We prove a theorem that equates showing degree bounds on threshold to the number of solutions to certain families of equations. Note that we already have a lower bound of $\max(k, n^{\frac{1}{t}})$ by Corollary 3.9. Since we wish to minimize the cost of the protocol which is defined as $\max(p_i^{k_i})$, we will assume that $p_i^{k_i}$s are nearly equal and that they are greater than $\max(k, n^{\frac{1}{t}})$.

Theorem 3.21 *There exists a strong protocol for* T_k *over* $\mathbb{Z}_m$ *with parameters* k_i *for* $1 \leq i \leq t$ *iff the following equation has no non-trivial solutions*

$$\begin{aligned} \forall\, i, \quad a_i p_i^{k_i} &\leq n \\ \forall\, i \neq j, \quad |a_i p_i^{k_i} - a_j p_j^{k_j}| &< k \end{aligned} \tag{13}$$

PROOF: Clearly $a_i = 0$ for all i is a solution and we call this a trivial solution. We show that a protocol over $\mathbb{Z}_6$ exists iff the following equation does not have solutions. The extension to general m is easy.

$$|a2^{k_2} - b3^{k_3}| = \ell \qquad\qquad a2^{k_2} \leq n,\ b3^{k_3} \leq n,\quad \ell < k \tag{14}$$

As a first step, we show that it suffices to analyze the following strong protocol. This is essentially the argument for the correctness of Protocol 3.6 specialized to the Threshold-k function.

Protocol 3.22 Strong Protocol for Threshold-k

- *If* $i \geq k$, P_2 *outputs* 1, *else* P_2 *outputs* 0.
- *If* $j \geq k$, P_3 *outputs* 1, *else* P_3 *outputs* 0.

In a strong protocol, if $f(w) = 0$ both players must output 0. Hence when $i < k$, P_2 must output 0 since the input could be i. If $i \geq k$, then clearly $w \geq k$, hence P_2 can w.l.o.g. output 1. Similarly, this is also the best strategy for P_3.

We analyze inputs on which the protocol fails. Let $w \geq k, i < k, j < k$. On such inputs, both players output 0 whereas the value of the function is 1, and so the protocol is incorrect. Note that $i \neq j$ since if $i = j$, by the CRT $w = i$. This contradicts the fact that $w \geq k$. But now

$$w = a2^{k_2} + i = b3^{k_3} + j$$

Assume that $i > j$ and let $i - j = \ell$ where $0 < \ell < k$. Then, we have

$$\begin{aligned} b3^{k_3} - a2^{k_2} &= \ell \\ a2^{k_2}, b3^{k_3} \leq w &\leq n \end{aligned}$$

Hence any such input gives a solution to Equation (14).

Conversely, we will show that any solution to Equation (14) for fixed n gives an input w so that the protocol is incorrect. Assume that we have

$$|a2^{k_2} - b3^{k_3}| = \ell \quad s.t.\ \ a2^{k_2} \leq n,\ b3^{k_3} \leq n,\quad \ell < k$$

Assume $b3^{k_3} > a2^{k_2}$. Set $w = b3^{k_3} = a2^{k_2} + \ell$. From this setting, we obtain

$$i \equiv w \bmod 2^{k_2} = \ell$$

$$j \equiv w \bmod 3^{k_3} = 0$$

Hence we have $w > 2^{k_2} \geq k$ whereas $i, j < k$ and hence the protocol is incorrect. □

As an example, suppose we were trying to show a bound of $n^{\frac{3}{4}}$ on T_2. We set $2^{k_2}, 3^{k_3} > n^{\frac{3}{4}}$. This implies that $a, b < n^{\frac{1}{4}} = (2^{k_2})^{\frac{1}{3}}$. We are looking for solutions to

$$|a2^{k_2} - b3^{k_3}| = 1 \qquad\qquad a < (3^{k_3})^{\frac{1}{3}} \quad b < (2^{k_2})^{\frac{1}{3}}$$

If we relax the constraints on a, b to $a < 3^{k_3}$ and $b < 2^{k_2}$, since $(2^{k_2}, 3^{k_3}) = 1$, by the GCD equation, we will have a solution for every value of k_2, k_3. We are asking how many solutions exist with the constraint that $a, b < (2^{k_2})^{\frac{1}{3}}$. We will show that the answer is only finitely many.

3.6.1 Constant Threshold with Two Players

We now prove an upper bound for constant threshold when m has two prime factors. We set $m = 6$ for convenience. We will use the following result of Filaseta [30].

Proposition 3.23 *Let ℓ be a fixed non-zero integer. Let M be a fixed positive integer. Let $\varepsilon > 0$. Let D be the largest divisor of $N(N - \ell)$ which is relatively prime to M. If N is sufficiently large (depending on ℓ, M and ε), then $D > N^{1-\varepsilon}$.*

Theorem 3.24 *Let $c \geq 1$ be any fixed constant. Over $\mathbb{Z}_{pq}$, $\delta(T_c) = O(n^{\frac{1}{2}+\varepsilon})$ for all $\varepsilon > 0$.*

PROOF: We prove the theorem over $\mathbb{Z}_6$.

Set $2^{k_2} \cdot 3^{k_3} > n^{1+\varepsilon}$. We will show with this setting of parameters, Protocol 3.22 works for sufficiently large n. By Theorem 3.21, the protocol for n fails iff there is a solution to

$$|a2^{k_2} - b3^{k_3}| \;=\; \ell \qquad\qquad a2^{k_2} \leq n,\ b3^{k_3} \leq n,\ \ell < c \tag{15}$$

We first show that for each $\ell < c$, this equation has only finitely many solutions. Set $M = 6$. Take

$$N \;=\; a2^{k_2} \;=\; b3^{k_3} + \ell$$

$$\Rightarrow \quad N(N - \ell) \;=\; ab2^{k_2}3^{k_3}$$

Let D be largest divisor of $N(N-\ell)$ relatively prime to 6. It follows that $D \leq ab$. By our setting of parameters,

$$2^{k_2}3^{k_3} > n^{1+\varepsilon} \geq N^{1+\varepsilon}$$

$$ab2^{k_2}3^{k_3} = N(N-\ell) < N^2$$

$$\Rightarrow \qquad D \leq ab < N^{1-\varepsilon}$$

By Proposition 3.23, this is possible for only finitely many N. Hence, with fixed ℓ, there are only finitely many solutions. There are only finitely many possibilities for ℓ since $1 \leq \ell < c$. Hence Equation 15 has only finitely many solutions in $a2^{k_2}, b3^{k_3}$. This implies an upper bound on n since

$$2^{k_2} \cdot 3^{k_3} \geq n^{1+\varepsilon} \quad \Rightarrow \quad n \leq ab2^{k_2}3^{k_3}$$

Hence there are only finitely many solutions in n. Hence Protocol 3.22 works for all sufficiently large n. We can take 2^{k_2} and 3^{k_3} approximately equal to give the desired degree bound. □

By the CRT, we know that if $2^{k_2}3^{k_3} > n$, and if $w \equiv 0$ modulo 2^{k_2} and 3^{k_3} then in fact $w = 0$. The above theorem states that if $2^{k_2}3^{k_3} > n^{1+\varepsilon}$ for any positive ε, and if the residues of w modulo 2^{k_2} and 3^{k_3} are both less than c then in fact $w < c$ for sufficiently large n. Also we have established an equivalence between proving bounds on the strong degree and showing that certain equations have only finitely many solutions. This equivalence allows us to use number theoretic results to show bounds on degree. On the other hand, it implies than an alternative proof of the degree bound for symmetric polynomials will have interesting number theoretic implications.

3.6.2 Constant Threshold with Multiple Players

In this section we consider the case when m has t distinct prime divisors $p_1, p_2, \cdots p_t$. For T_c with c constant, it is easy to show a lower bound of $\Omega(n^{\frac{1}{t}})$. We will show an upper bound of $O(n^{\frac{1}{t}+\varepsilon})$ for all $\varepsilon > 0$. We will use a result due to Granville which generalizes Filaseta's result. But this result holds only under the assumption of the $abc-$conjecture. This is a

very powerful conjecture which has many important implications, including an asymptotic version of Fermat's Last Theorem [39].

Definition 3.9 *The Radical of M denoted by $R(M)$ is the product of distinct primes dividing M.*

Conjecture 3.25 (The *abc*-conjecture) *Fix $\epsilon > 0$. If a, b, c are coprime positive integers satisfying $a + b = c$, then*

$$c < D \cdot R(abc)^{1+\epsilon}$$

where D is a constant that depends only on ϵ.

Theorem 3.26 *[38] Assume the abc-conjecture is true. Suppose that $g(X) \in \mathbb{Z}[X]$ has no repeated roots. Fix $\varepsilon > 0$. Then for w sufficiently large,*

$$R(g(w)) > |w|^{deg(g)-1-\varepsilon}$$

Using this result, we analyze the following protocol which is the natural generalization of Protocol 3.22.

Protocol 3.27 Threshold-c with multiple players

- *Take $p_1^{k_1} p_2^{k_2} \cdots p_t^{k_t} > n^{1+\varepsilon}$.*
- *Set $w_i \equiv w \bmod p_i^{k_i}$. If $w_i < c$, Player i outputs 0 else player i outputs 1.*

Theorem 3.28 *Let $c \geq 1$ be any fixed constant. Assuming the* abc *conjecture, over $\mathbb{Z}_m$, $\delta(T_c) = O(n^{\frac{1}{t}+\varepsilon})$.*

PROOF: Fix a value of n. If Protocol 3.27 is incorrect, by Theorem 3.21 there must be a non-trivial solution to the following system of equations.

$$\begin{aligned} a_i p_i^{k_i} &\leq n \qquad \forall\, i \in \{1, \cdots t\} \\ |a_i p_i^{k_i} - a_j p_j^{k_j}| &< c \qquad \forall\, i < j \end{aligned} \tag{16}$$

We now set

$$g(X) \quad = \quad X(X-1)\cdots(X-c+1)$$

Clearly $g(X)$ has no repeated roots and we can apply Theorem 3.26. Hence, $\forall \varepsilon > 0$, for all but finitely many n,

$$R(g(w)) > w^{c-1-\varepsilon} \tag{17}$$

We will show that if Protocol 3.27 is incorrect on w, then $g(w)$ is divisible by high prime powers, and so $R(g(w))$ is small, which contradicts Equation (17).

$$g(w) = w(w-1)\cdots(w-c+1)$$

We know that $w - a_i p_i^{k_i} = w_i$ where $0 \leq w_i < c$. Hence for all i,

$$\begin{aligned} & w - w_i \mid g(w) \\ & w - w_i \;=\; a_i p_i^{k_i} \\ \Rightarrow \quad & p_i^{k_i} | g(w) \end{aligned}$$

By the CRT, for a suitable constant C,

$$g(w) = C \prod_i p_i^{k_i}$$

We now bound the size of C.

$$\begin{aligned} \prod_i p_i^{k_i} \;&>\; n^{1+\varepsilon} \;\geq\; w^{1+\varepsilon} \\ g(w) \;&=\; w(w-1)\cdots(w-c+1) \;<\; w^c \\ \Rightarrow\; C \;&=\; \frac{g(w)}{\prod_i p_i^{k_i}} \;<\; w^{c-1-\varepsilon} \end{aligned}$$

This gives an upper bound on $R(g(w))$.

$$\begin{aligned} R(g(w)) \;&<\; C p_1 p_2 \cdots p_t \\ &<\; w^{c-1-\varepsilon} p_1 p_2 \cdots p_t \\ &=\; w^{c-1-\varepsilon'} \end{aligned}$$

The last equality holds since $\prod p_i \leq m$ is a constant. This gives a contradiction to Equation 17. Hence w must be one of only finitely many exceptions. This bounds the value of n since

$$w \geq a_i p_i^{k_i} \geq p_i^{k_i}$$

$$\prod_i p_i^{k_i} > n^{1+\varepsilon}$$

$$\Rightarrow w^t > n^{1+\varepsilon}$$

$$\Rightarrow n < w^{\frac{t}{1+\varepsilon}}$$

Hence there are only finitely many solutions in n and the protocol works correctly for n sufficiently large. The degree bound follows by taking nearly equal powers of p_i. □

3.6.3 Upper Bounds for General Threshold Functions

We now return to the case when m has two prime divisors and show that the $abc-$conjecture implies an upper bound of $O(nk)^{\frac{1+\varepsilon}{2}}$ on T_k for all values of k in the strong representation. We begin with the following technical lemma.

Lemma 3.29 *Assume the* abc *conjecture holds for some $\varepsilon > 0$. For $n > n_0(\varepsilon)$, the equation*

$$|a2^{k_2} - b3^{k_3}| = \ell \qquad\qquad a2^{k_2} \leq n, \quad b3^{k_3} \leq n, \quad 2^{k_2}3^{k_3} \geq (n\ell)^{1+\varepsilon}$$

has no solutions with $a2^{k_2}, b3^{k_3}, \ell$ relatively prime.

PROOF: Assume that we have a solution where $a2^{k_2} > b3^{k_3}$. Applying the *abc* conjecture to the equation $a2^{k_2} = b3^{k_3} + \ell$, we must have

$$D \cdot R(a2^{k_2}b3^{k_3}\ell)^{1+\varepsilon} > a2^{k_2} \geq (a2^{k_2}b3^{k_3})^{\frac{1}{2}} \tag{18}$$

where the last inequality holds since $a2^{k_2} > b3^{k_3}$. We can bound $R(a2^{k_2}, b3^{k_3}, \ell)$ by $6ab\ell$. Plugging this bound into (18), for a suitable constant D' depending only on ε, we get

$$D' \cdot (ab\ell)^{1+\varepsilon} > (ab2^{k_2}3^{k_3})^{\frac{1}{2}} \geq (ab)^{\frac{1}{2}}(n\ell)^{\frac{1+\varepsilon}{2}}$$

The last inequality uses the fact that $2^{k_2}3^{k_3} \geq (n\ell)^{1+\varepsilon}$. Rearranging terms,

$$D' \cdot (ab)^{\frac{1}{2}+\varepsilon}\ell^{1+\varepsilon} > (n\ell)^{\frac{1+\varepsilon}{2}} \tag{19}$$

We now upper bound the size of ab.

$$a2^{k_2}b3^{k_3} \leq n^2, \quad 2^{k_2}3^{k_3} \geq (n\ell)^{1+\varepsilon} \quad \Rightarrow ab \leq \frac{n^{1-\varepsilon}}{\ell^{1+\varepsilon}}$$

A calculation now gives the following bound on the LHS of (19).

$$D' \cdot (ab)^{\frac{1}{2}+\varepsilon}\ell^{1+\varepsilon} \leq D'n^{\frac{1+\varepsilon}{2}-\varepsilon^2}\ell^{\frac{1-\varepsilon}{2}-\varepsilon^2} \tag{20}$$

Plugging this bound into (19), we have

$$D'n^{\frac{1+\varepsilon}{2}-\varepsilon^2}\ell^{\frac{1-\varepsilon}{2}-\varepsilon^2} > (n\ell)^{\frac{1+\varepsilon}{2}}$$

For all $n > n_0(\varepsilon)$, this gives a contradiction. Hence for sufficiently large n, the equation has no solutions. □

There is an easy extension to the case of general p and q. Using this, we can show the following degree bound for T_k over $\mathbb{Z}_{pq}$ assuming the abc-conjecture.

Theorem 3.30 *If the abc−conjecture is true for some $\varepsilon > 0$, over $\mathbb{Z}_{pq}$, $\delta(T_k) = O((nk)^{\frac{1+\varepsilon}{2}})$ for any $k \leq n$.*

PROOF: Note that for a non-trivial bound, we need $\varepsilon < 1$, else $(nk)^{\frac{1+\varepsilon}{2}} = \Omega(n)$ for all k. Take $n > n_0(\varepsilon)$ as in Lemma 3.29. Set $2^{k_2}3^{k_3} \geq (nk)^{1+\varepsilon}$. We claim that there are no solutions to

$$|a2^{k_2} - b3^{k_3}| = \ell \qquad\qquad a2^{k_2} \leq n,\ b3^{k_3} \leq n, \quad \ell < k \tag{21}$$

Assume that a solution exists. Note that $a2^{k_2}, b3^{k_3}, \ell$ need not be coprime. Their GCD can be written as $2^{t_2}3^{t_3}g$ where g is relatively prime to 2 and 3. Dividing throughout we get

$$|a'2^{k_2-t_2} - b'3^{k_3-t_3}| = \ell' \qquad\qquad a'2^{k_2-t_2} \leq n,\ b'3^{k_3} \leq n, \quad \ell' < \frac{k}{2^{t_2}3^{t_3}}$$

Further, we now have that $a'2^{k_2-t_2}, b'3^{k_3-t_3}, \ell'$ are relatively prime. To apply Lemma 3.29, we need to check that $2^{k_2-t_2}3^{k_3-t_3} \geq (n\ell')^{1+\varepsilon}$. It is easy to see that this condition does hold.

$$2^{k_2-t_2}3^{k_3-t_3} \geq \frac{(nk)^{1+\varepsilon}}{2^{t_2}3^{t_3}} \geq \left(\frac{nk}{2^{t_2}3^{t_3}}\right)^{1+\varepsilon} \geq (n\ell')^{1+\varepsilon}$$

However, by Lemma 3.29, our choice of n guarantees that such a solution cannot exist. Hence in fact Equation (21) has no solutions. The degree bound then follows by taking 2^{k_2} and 3^{k_3} nearly equal and applying Theorem 3.21. □

We are unable to extend the above bound to the t-player case for $t \geq 3$.

3.7 Lower Bounds for Threshold Functions

3.7.1 Strong Representations

In this section, we will show a $\Omega(\sqrt{kn})$ lower bound on the strong degree of the T_k function over $\mathbb{Z}_{pq}$. For small ε, this matches the upper bound of the previous section. Over $\mathbb{Z}_m$, when m has t distinct prime factors, we show a lower bound of $\Omega(n^{\frac{1}{t}}k^{1-\frac{1}{t}})$ on the strong degree of T_k.

Theorem 3.31 *Over $\mathbb{Z}_{pq}$, $\delta(T_k) = \Omega(\sqrt{nk})$.*

PROOF: We prove the theorem over $\mathbb{Z}_6$. Set $2^{k_2}, 3^{k_3} \leq \frac{\sqrt{kn}}{2}$. We will construct solutions to the following equation for all n.

$$|a2^{k_2} - b3^{k_3}| = \ell \qquad\qquad a2^{k_2}, b3^{k_3} \leq n, \ell < k \tag{22}$$

By Theorem 3.21 this implies $\delta(T_k) = \Omega(\sqrt{nk})$.

We construct the solutions by a pigeonhole argument. By Lemma 3.9 we may assume $2^{k_2}, 3^{k_3} \geq \max(k, \sqrt{n})$. Consider all pairs (u, v) such that $u2^{k_2} \leq n,\ v3^{k_3} \leq n$. We map the pair (u, v) to the point $P_{uv} = u2^{k_2} - v3^{k_3}$, so that $P_{uv} \in [-n, n]$. Each pair u, v is mapped to a distinct point, since if

$$\begin{aligned}
& P_{uv} = P_{st},\ (u,v) \neq (s,t) \\
\Rightarrow\ & (u-s)2^{k_2} - (v-t)3^{k_3} = 0 \\
\Rightarrow\ & 2^{k_2}3^{k_3} | (u-s)2^{k_2} \\
\Rightarrow\ & |(u-s)2^{k_2}| > n
\end{aligned}$$

However, $|(u-s)2^{k_2}| \leq n$ by our choice of u and s.

We can now count the total number of points $P_{u,v}$. We can take $0 \leq u, v < 2\sqrt{\frac{n}{k}}$. Hence there are $4\frac{n}{k}$ points lying in the interval $[-n, n]$, and hence by the pigeonhole principle, there are two points within a distance of $\frac{(2n+1)k}{4n} < k$. Call them P_{uv} and P_{st}. Hence

$$|(u-s)2^{k_2} - (v-t)3^{k_3}| = \ell \qquad\qquad \ell < k$$

Set $a = u - s$, and $b = v - t$. Assume that $a \geq 0$. This implies that $b \geq 0$, since $2^{k_2} > k, 3^{k_3} > k$ so we cannot add multiples of 2^{k_2} and 3^{k_3} to get $\ell < k$. Also, $a2^{k_2} \leq u2^{k_2} \leq n$ and similarly $b3^{k_3} \leq v3^{k_3} \leq n$. Hence a, b, ℓ give the desired solution to Equation 22. $\square$

Note that the lower bound of $\sqrt{nk}$ almost matches the upper bound of $(nk)^{\frac{1}{2}+\varepsilon}$ implied by the *abc*−conjecture. In [16], Beigel shows unconditionally that the bound of $O(\sqrt{nk})$ holds for *infinitely many* n for $k \leq c\sqrt{n}$ for some constant c.

We now generalize this proof to get a lower bound for the t- player case. This result can also be derived from Dirichlet's theorem on simultaneous Diophantine approximation [48].

Theorem 3.32 *Over* $\mathbb{Z}_m$, $\delta(T_k) = \Omega(n^{\frac{1}{t}}k^{\frac{t-1}{t}})$.

PROOF: Let $p_i^{k_i} < \frac{1}{3}n^{\frac{1}{t}}k^{1-\frac{1}{t}}$ $\forall i$. We will construct solutions to the equation

$$\begin{aligned} \forall\, i, \quad a_i p_i^{k_i} &\leq n \\ \forall\, i \neq j, \quad |a_i p_i^{k_i} - a_j p_j^{k_j}| &< k \end{aligned} \tag{23}$$

By Theorem 3.21, this will imply the desired lower bound.

By Lemma 3.9 we may assume that $p_i^{k_i} > k, n^{\frac{1}{t}}$ $\forall i$. We define t vectors $v_1, \cdots v_t$ in $t-1$ dimensions.

$$\begin{aligned} v_1 &= (p_1^{k_1}, p_1^{k_1}, \cdots, p_1^{k_1}) \\ v_2 &= (p_2^{k_2}, 0, \cdots, 0) \\ v_i &= (0,\ 0,\ p_i^{k_i}, 0) \\ v_t &= (0, 0, \cdots, p_t^{k_t}) \end{aligned}$$

For $i = 1, \cdots t$, consider b_i such that $b_i p_i^{k_i} \leq n$. We map every such t-tuple $b = (b_1, b_2, \cdots, b_t)$ to a point P_b in $t-1$ dimensional space.

$$\begin{aligned} P_b &= b_1 v_1 - b_2 v_2 \cdots - b_t v_t \\ &= (b_1 p_1^{k_1} - b_2 p_2^{k_2}, b_1 p_1^{k_1} - b_3 p_3^{k_3}, \cdots, b_1 p_1^{k_1} - b_t p_t^{k_t}) \end{aligned}$$

We can use the fact that $p_1^{k_1} p_2^{k_2} \cdots p_t^{k_t} > n$ to show that if $b \neq c$, then $P_b \neq P_c$. For each i we can take $0 \leq b_i < 3(\frac{n}{k})^{1-\frac{1}{t}}$. This gives a total of $3^t(\frac{n}{k})^{t-1}$ points. Since each co-ordinate of P_b lies between $[-n, n]$, every point lies in $[-n, n]^{t-1}$ which is a cube of volume $(2n+1)^{t-1}$. We can partition this cube into $\lceil \frac{2n+1}{k-1} \rceil^{t-1} < (\frac{3n}{k})^{t-1}$ smaller cubes with each side of length $k-1$. However there are $3^t(\frac{n}{k})^{t-1}$ distinct points. By the pigeonhole principle, two points lie in the same cube of side $k-1$. Call these points P_b and P_c. This implies for $2 \leq i \leq t$ we have

$$|(b_1 - c_1)p_1^{k_1} - (b_i - c_i)p_i^{k_i}| \leq k - 1$$

Assume that $b_1 - c_1 \geq 0$. Since $p_i^{k_i} > k$ for every i, this implies $b_i - c_i \geq 0$ for every i. We set $a_i = b_i - c_i$. This gives

$$\begin{aligned} \forall i, \quad a_i p_i^{k_i} &\leq b_i p_i^{k_i} &\leq n \\ \forall\ i \neq j, \quad |a_i p_i^{k_i} - a_j p_j^{k_j}| &< k \end{aligned}$$

Hence we get a solution to Equation 23. □

3.7.2 Weak Representations

In Theorem 3.13, we show a lower bound of $\Omega(\max(k, \sqrt{n}))$ for $\Delta(T_k)$ over $\mathbb{Z}_{pq}$. We can improve this to $\Omega(\sqrt{nk})$ using the results obtained above on the strong degree of T_k.

Theorem 3.33 *Over $\mathbb{Z}_{pq}$, for $k \leq \frac{n}{p}$, $\delta(T_k) = \Omega(\sqrt{nk})$.*

PROOF: We prove the bound over $\mathbb{Z}_6$. We apply the construction in the proof of Theorem 3.31 with $\frac{n}{2}$ and $\frac{k}{2}$. Set $2^{k_2}, 3^{k_3} \leq \frac{\sqrt{kn}}{4}$. There exist a, b and ℓ satisfying the following equation.

$$|a2^{k_2} - b3^{k_3}| = \ell \qquad\qquad a2^{k_2}, b3^{k_3} \leq \frac{n}{2}, \ell < \frac{k}{2} \tag{24}$$

We show that there does not exist a weak protocol for T_k of cost $\max(2^{k_2}, 3^{k_3})$. By Lemma 3.11 it suffices to show that A^{T_k} has a submatrix with 3 distinct rows. We use solutions to Equation (24) to construct this submatrix. Assume $a2^{k_2} \geq b3^{k_3}$. By Lemma 3.9 we may assume $2^{k_2}, 3^{k_3} \geq \max(k, \sqrt{n})$ and hence $a2^{k_2} \geq k$. We choose the submatrix V of A

$$
\begin{aligned}
V &= \begin{pmatrix} 0 & a2^{k_2} & 2 \cdot a2^{k_2} \\ \times & a2^{k_2} - b3^{k_3} & 2 \cdot a2^{k_2} - b3^{k_3} \\ \times & \times & 2(a2^{k_2} - b3^{k_3}) \end{pmatrix} \\
&= \begin{pmatrix} 0 & a2^{k_2} & 2 \cdot a2^{k_2} \\ \times & \ell & \ell + a2^{k_2} \\ \times & \times & 2\ell \end{pmatrix} \\
\Rightarrow \quad V^{T_k} &= \begin{pmatrix} 0 & 1 & 1 \\ \times & 0 & 1 \\ \times & \times & 0 \end{pmatrix}
\end{aligned}
$$

We need to ensure that all entries in the fooling set are valid. The largest entry in the fooling set is $2 \cdot a2^{k_2}$. From Equation (24), we have $2 \cdot a2^{k_2} \leq n$. By Lemma 3.11 a weak protocol cannot exist since V^{T_k} has at least 3 distinct columns. Hence $\max(2^{k_2}, 3^{k_3}) > \frac{\sqrt{nk}}{4}$. Note that $2a2^{k_2} \leq n$, on the other hand, $a2^{k_2} \geq k$. Combining the inequalities, we obtain $k \leq \frac{n}{2}$. □

We believe that this bound holds for $k \leq \frac{n}{2}$. It is natural to ask if one can show linear bounds for all $k > \frac{n}{2}$. The next theorem shows that the answer is no (see Figure 2). It explains the remark in the introduction that the weak degree of the AND function is $\Theta(\sqrt{n})$.

Theorem 3.34 $\Delta(T_k) = \Delta(n - k + 1)$.

PROOF: Assume that there is a weak protocol for T_k where the players read k_2 and k_3 digits respectively. On an input w, let $i \equiv w \bmod 2^{k_2}, j \equiv w \bmod 3^{k_3}$. Since both players know the value of n, they can compute

$$i' \equiv (n - i) \bmod 2^{k_2} \equiv (n - w) \bmod 2^{k_2}$$

$$j' \equiv (n - j) \bmod 3^{k_3} \equiv (n - w) \bmod 3^{k_3}$$

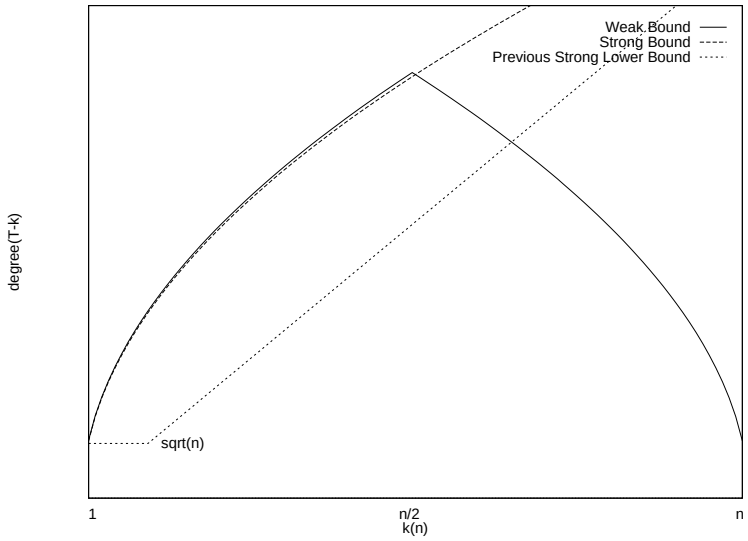

Figure 1: The Degree of Threshold-k functions mod 6

Now if the players use the protocol for T_k with the values i' and j' instead, they can differentiate the values w such that $n-w < k$ and $n-w \geq k$. This is then a weak protocol differentiating values of $w \geq n-k+1$ and $w < n-k+1$ of cost $\max(2^{k_2}, 3^{k_3})$. A symmetric argument shows that a weak protocol for T_{n-k+1} gives a weak protocol for T_k. □

This shows that for $k > \frac{n}{2}$, there is a gap between the strong and weak degree.

We have shown that resolving the degree of Threshold functions for symmetric polynomials is equivalent to questions regarding Diophantine equations. These are rather hard questions and it does not seem that tight upper bounds can be shown unconditionally. Is showing upper bounds for threshold functions using general polynomials any easier? Perhaps we run into hard number theoretic questions because we are restricted to symmetric polynomials and proving upper bounds with general polynomials is easier. Proving lower bounds on the other hand can only be harder for general polynomials. The fact that the best known lower bound for OR is $\Omega(\log n)$ suggests that indeed lower bounds are much harder for general polynomials.

CHAPTER IV

ALGORITHMS FOR INTERPOLATION OVER COMPOSITES

Our goal in this chapter is to understand the structure of polynomials modulo composites; we do so through the lens of polynomial interpolation. Polynomial representations of Boolean functions, which we studied in Chapter 3, as well as combinatorial applications of polynomials modulo composites which we will study in Chapter 5, are concerned with the lowest degree polynomial that satisfies certain constraints on its evaluations. Hence, the problem they address is closely related to polynomial interpolation, and the structural insights gained from this problem are useful for those applications.

4.1 Polynomial Interpolation modulo Composites

The problem of polynomial interpolation is to reconstruct a polynomial from its evaluations. This is a fundamental algorithmic question in algebra with numerous applications. The problem is especially well studied when the polynomial is over a field such as $\mathbb{R}$ or $\mathbb{Z}_p$ dating back to Newton and Lagrange. Relatively less is known about interpolation over rings which contain zero divisors, in particular over $\mathbb{Z}_m$ with m composite. The zero-testing problem is a special case of the interpolation problem where we want to know if the polynomial is 0 everywhere. In this chapter we study the problem of learning a univariate polynomial in $\mathbb{Z}_m[X]$ based on its evaluations at a set $I \subseteq \mathbb{Z}_m$. We ask the question: *Given $I \subseteq \mathbb{Z}_m$, how many evaluations of a polynomial at points in I are required to compute its value at every point in I?*

Throughout, we will consider a polynomial as a function rather than a formal sum and our aim will be to correctly predict its values at every point in I. The polynomial interpolation problem over $\mathbb{Z}_m$ is very different from $\mathbb{Z}_p$ since it is no longer true that a degree d polynomial has at most d zeroes. For instance $X^k \equiv 0 \bmod 2^k$ has 2^{k-1} roots. This

implies that even two polynomials of small degree can agree on a large fraction of points in $\mathbb{Z}_m$. Hence, unlike over $\mathbb{Z}_p$ one cannot interpolate even low degree polynomials from their evaluations at a few arbitrarily chosen points. On the other hand, not every function $f : \mathbb{Z}_m \to \mathbb{Z}_m$ is a polynomial. One restriction on functions defined by polynomials is that they need to satisfy certain congruences. For instance let $m = pq$ and $x, y \in \mathbb{Z}_m$ such that $x \equiv y \bmod p$. Then $P(x) \equiv P(y) \bmod p$ for any polynomial $P(X) \in \mathbb{Z}_m[X]$. Thus the values of a polynomial at a point give some information about its values at other points. This raises the possibility of learning a polynomial by looking at its evaluations at only a few carefully chosen points. This problem has been considered in mathematics. Dueball [63] shows that when $I = \mathbb{Z}_m$, there is a subset S whose size can lie between $\log m$ and m such that the evaluations at S are sufficient for interpolation. However, this result does not answer the more general question stated above.

4.1.1 Problem History

Given a commutative ring R, a function $f : R \to R$ which can be computed by a polynomial in $R[X]$ is called a polynomial function. Polynomial functions over various commutative rings are well studied in algebra [26, 63, 32]. The problem of characterizing polynomial functions over $\mathbb{Z}_m$ was first studied by Carlitz and Spira [73] (see also the book by Narkeiwicz and references therein [63]). Kempner gave a canonical polynomial for every polynomial function over $\mathbb{Z}_m$ [53]. Dueball studied the problem of interpolation over $\mathbb{Z}_m$ [63]. He proved that one can solve the interpolation problem over $\mathbb{Z}_m$ with as few as $O(\log m)$ queries for some composites m. More precisely, he showed the following result: *Let $k(m)$ be the smallest integer such that $k(m)! \equiv 0 \bmod m$. Every polynomial function over $\mathbb{Z}_m$ can be learnt from its values at $\{0, \cdots, k(m) - 1\}$.*

Interpolation and zero-testing for polynomials over $\mathbb{Z}_p$ have been studied extensively in computer science, motivated by applications in coding theory, proof checking and several other areas (see for instance [34]). The problem of zero-testing for polynomials over $\mathbb{Z}_m$ was studied by Agrawal and Biswas [1], motivated by primality testing. They give a non-black-box randomized algorithm for this problem. However, they view polynomials as

formal sums rather than as functions and this is important for their application. Karpinski, van der Poorten and Shparlinski [51] give a black-box algorithm for zero-testing over $\mathbb{Z}_m$. However they require that all non-zero coefficients of the polynomial are relatively prime to m. Bshouty, Tamon and Wilson give a randomized algorithm for interpolation over $\mathbb{Z}_m$ [25]. However if the smallest prime dividing m is p, they require the degree to be at most $\frac{p}{2}$. The results of [25, 51] hold for multivariate polynomials, but in the univariate case Dueball's result is stronger.

4.2 Our Results

4.2.1 The Generalized Interpolation Problem

Our main result is an efficient algorithm to solve the following generalized interpolation problem.

Problem 4.1 GENERALIZED POLYNOMIAL INTERPOLATION: *Given m, a set $I \subseteq \mathbb{Z}_m$ and black-box access to the values of a polynomial $P(X) \in \mathbb{Z}_m[X]$ at points in I. Compute $P(X)$ and minimize the number of black-box queries.*

Minimizing the number of queries is NP-hard, but our algorithm has query complexity close to optimal.

Theorem 4.1 *Let t be the number of distinct prime factors of m. There is an algorithm to solve the general interpolation problem over $\mathbb{Z}_m$, with query complexity within a factor t of the optimum.*

In fact the guarantee is slightly stronger. When the algorithm terminates, it produces a factorization of m into $t' \leq t$ relatively prime factors. The approximation factor is in fact bounded by t'. Thus input sets I which force the algorithm to make several queries must also reveal the factorization of m. The algorithm first computes a set S of queries to ask based on the input set I. Thus the set of queries is chosen non-adaptively. The size of S is within a factor t' of the optimal query complexity. This step takes time proportional to $|S| \cdot |I|$. Once the set S is found, the polynomial can be computed with $|S|$ queries in time $\text{poly}(\log m, |S|)$.

While Dueball's result gives an efficient algorithm for the case when $I = \mathbb{Z}_m$, it does imply anything for the general interpolation problem. The naive approach for this problem would be to write a linear equation for each point in I. We can replace each equation $\sum_j a_{ij}X_{ij} = b_i \bmod m$ with $\sum_j a_{ij}X_{ij} = b_i + y_i m$ where the y_is are integer variables, and find integer solutions to the resulting system of equations. This has query complexity $|I|$, which can be exponentially larger than the complexity of our algorithm.

The generalized zero-testing problem is a special case of the interpolation problem, where we wish to know if some identity holds for every point in I. Theorem 4.1 implies a query-efficient algorithm for this problem. It improves on the algorithms of [25, 51] since there are no restrictions on the degree or coefficients of the polynomial. Our results are incomparable with those of Agrawal and Biswas [1], since they view polynomials as formal sums.

4.2.2 Learning under a Distribution

We give the first efficient algorithms for learning polynomials over $\mathbb{Z}_m$ under a distribution. Here we are given evaluations of the polynomial at points which are drawn from some distribution and we are asked to learn the polynomial. See Section 5 for precise problem definitions.

Theorem 4.2 *Polynomials in $\mathbb{Z}_m[X]$ are exactly learnable under the uniform distribution and PAC-learnable under an arbitrary distribution in polynomial time.*

The algorithm for the uniform distribution learns the polynomial exactly, but its running time is a random variable. These algorithms use the algorithm for the general interpolation problem as a subroutine. For distributional learning it is essential that our algorithm solves the general interpolation problem, where inputs come from some subset $I \subseteq \mathbb{Z}_m$ rather than all of $\mathbb{Z}_m$.

4.2.3 Interpolating Sets

The crux of our algorithm is the notion of an *interpolating set* which we introduce and study here. A set $S \subseteq I$ is an interpolating set for I if knowing the values of any polynomial at

S fixes its value at every point in I. We show that the set of queries of an interpolation algorithm must correspond to an interpolating set for I, thus the problem of designing query-efficient algorithms reduces to finding small interpolating sets.

Let $k(I)$ denote the size of the smallest interpolating set for I. In general $k(I)$ can lie between $\log |I|$ and $|I|$. However the problem of computing a minimum interpolating set for I is NP-hard. We define a related quantity $\bar{k}(I)$, which is the smallest integer such that there is a degree $\bar{k}(I)$ monic polynomial $M(X) \in \mathbb{Z}_m[X]$ which is 0 over I. This quantity can be computed in polynomial time by solving a system of linear equations. We show that for $I \subseteq \mathbb{Z}_m$ where m has t prime divisors, the following relation holds:

$$\bar{k}(I) \;\leq\; k(I) \;\leq\; t \cdot \bar{k}(I)$$

Thus $\bar{k}(I)$ is a factor t approximation to $k(I)$ where t is the number of prime divisors of m. This is where the approximation factor of t in the query complexity of our algorithm comes from.

We sketch the idea behind the algorithm for computing an interpolating set. For the prime-power case, we use a greedy algorithm. There is a natural metric on the points $I \subseteq \mathbb{Z}_{p^a}$, namely p-adic distance. Our algorithm finds a set of points so that the sum of pairwise distances is maximized, this is done by picking a new point in a natural greedy manner. We show that this in fact gives an interpolating set. For the composite case, we essentially try and repeat this greedy approach. However, this approach might fail: firstly, we do not know the factorization of m, and secondly distinct prime divisors p and q give different metrics on the set I. However, we show that when it fails, one can get a factorization $m = m_1 \cdot m_2$ where $(m_1, m_2) = 1$. This allows us to use divide and conquer: we find interpolating sets modulo m_1 and m_2 independently and combine the result using the Chinese Remainder Theorem (CRT).

Interpolating sets over $\mathbb{Z}_{p^a}$ have rich algebraic and combinatorial structure which we study in detail, these properties are also useful in analyzing our algorithm. In proving these properties, we make crucial use of the fact that the underlying space is in fact an ultrametric space (metrics where the following strengthening of the triangle inequality holds:

$d(x,y) \leq \max(d(x,z), d(y,z))$. We show that many algebraic properties of polynomials can be reinterpreted as geometric properties of ultrametric spaces. Further, the proof of these properties for general ultrametric spaces follows directly from the proof for polynomials over $\mathbb{Z}_{p^a}$. We also note that our notion of interpolating sets over $\mathbb{Z}_{p^a}$ is closely related to Very Well Distributed and Well Ordered (V.W.D.W.O) sequences that have been studied in mathematics [26].

4.2.4 Organization of this Chapter

In the next section, we give some basic definitions and results about interpolation over $\mathbb{Z}_m$. We study interpolating sets in Section 4.4. We present our algorithms for interpolation in Section 4.5 and learning algorithms in Section 4.6. We present other algebraic and combinatorial characterizations of interpolating sets in Section 4.7. In Section 4.8, we use these characterizations to prove some combinatorial theorems about ultrametric spaces.

4.3 Preliminaries about Polynomial Interpolation

We will use X to denote a variable, and x for a constant. Let (a,b) denote the GCD of a and b.

Given $x \in \mathbb{Z}_{p^a}$, $x \neq 0$, let its p-adic valuation $\mathrm{val}_p(x)$ be the highest power of p which divides x. Set $\mathrm{val}_p(0) = \infty$. We have the so-called ultrametric inequality which states:

$$\mathrm{val}_p(x+y) \geq \min(\mathrm{val}_p(x), \mathrm{val}_p(y))$$

The p-adic norm of x is defined as

$$|x|_p = p^{-\mathrm{val}_p(x)}.$$

The p-adic norm satisfies the condition:

$$|x+y|_p \leq \max(|x|_p, |y|_p).$$

This induces a metric (the p-adic metric) on $\mathbb{Z}_{p^a}$ given by $d(x,y) = |x-y|_p$. This metric satisfies the following strong form of the triangle inequality:

$$d(x,z) \leq \max(d(x,y), d(y,z)). \tag{25}$$

Metrics which satisfy Equation 25 are known as ultrametrics.

We will also define valuations over $\mathbb{Z}_m$ where m is not a prime power. Assume that p divides m and let p^e be the highest power of p dividing m. For $x \in \mathbb{Z}_m$, we define

$$\mathrm{val}_p(x) \;=\; \mathrm{val}_p(x \bmod p^e).$$

All our results can be stated either in terms of p-adic valuations or norms. For our algorithmic results, it is more natural to work with valuations. For our combinatorial results, we will use p-adic norms, since it is easier to translate these results to general ultrametric spaces.

We start by proving some basic algebraic facts that will be useful to us.

Proposition 4.3 *Let $a, b \in \mathbb{Z}_m$ and $a \not\equiv 0$. The equation*

$$aX \;\equiv\; b \bmod m$$

has a solution in $\mathbb{Z}_m$ iff $(a, m)|b$. If this condition holds, there is a unique solution in the interval $\left[0, \ldots, \frac{m}{(a,m)} - 1\right]$.

PROOF: Let $g = (a, m)$. It is easy to check that the claim holds when $g = 1$.

So assume that $g \neq 1$. The condition $g|b$ is necessary since for any $x \in \mathbb{Z}_m$, $g|ax$, which is the LHS of the equation. Assume that the condition is satisfied. Then it is easy to see that the solutions are given by $x \in \mathbb{Z}_m$ such that

$$\frac{a}{g}x \;\equiv\; \frac{b}{g} \bmod \frac{m}{g}.$$

The solution to this equation is unique modulo $\frac{m}{g}$ since $\frac{a}{g}$ and $\frac{m}{g}$ are relatively prime, it is given by

$$x \;\equiv\; \frac{b}{g}\left(\frac{a}{g}\right)^{-1} \bmod \frac{m}{g}.$$

So there is a unique solution in $\left[0, \ldots, \frac{m}{g} - 1\right]$. □

Proposition 4.4 *Let $M(X)$ be a monic polynomial in $\mathbb{Z}_m[x]$ of degree k. Given $P(X) \in \mathbb{Z}_m[X]$, we can divide it by $M(X)$ and get a remainder of degree at most $k - 1$.*

PROOF: This is just Euclidean division. Let $P(X) = \sum_{i \leq d} c_i X^i$ where $d \geq k$. Since $M(X)$ is monic, $P(X) - c_d X^{d-k} M(X)$ has degree $d-1$. Now repeat the same procedure till we are left with a polynomial of degree $\leq k-1$. □

Proposition 4.5 *Let $N_0(X), \ldots, N_k(X)$ be polynomials in $\mathbb{Z}_m[X]$ where $N_i(X)$ is a monic polynomial of degree i. Every polynomial $P(X)$ of degree at most k can be written as*

$$P(X) = \sum_{i=0}^{k} c_i N_i(X).$$

Further if $P(X)$ is a monic polynomial of degree k, then $c_k = 1$.

PROOF: The proof is by induction on k. When $k = 0$, $N_0(X) = 1$ so there is nothing to prove. Assume the claim holds for $k-1$. Let $P(X) = \sum_{i \leq k} a_i X^i$. Since $N_k(X)$ is monic, $P(X) - a_k N_k(X)$ has degree $k-1$, so we can apply induction to it. Note that the leading coefficient in the monomial basis and the $\{N_i(X)\}$ basis is the same. This shows the second part of the claim. □

We can use this to give a canonical form for polynomial functions over $\mathbb{Z}_m$ due to Kempner [53]. Let $N_0(X) = 1$ and for $j \geq 1$, let

$$N_j(X) = \prod_{i=0}^{j-1} (X - j)$$

Let $k(m)$ be the smallest integer such that $k(m)! \equiv 0 \bmod m$.

Lemma 4.6 *[53] Every polynomial function over $\mathbb{Z}_m$ is computed by a unique polynomial of the form*

$$P(X) = \sum_{j=0}^{k(m)-1} c_j N_j(X) \qquad 0 \leq c_j < \frac{m}{(m, j!)} \tag{26}$$

PROOF: For any $x \in \mathbb{Z}$,

$$N_j(x) \;=\; \prod_{i=0}^{j-1} (x - i) \;=\; \binom{x}{j} j!$$

Hence $N_j(x)$ is divisible by $j!$. Hence the polynomial $\frac{m}{(m,j!)} N_j(X)$ is zero over $\mathbb{Z}_m$. In particular, $N_{k(m)}(X)$ is a degree $k(m)$ monic polynomial which is 0 over $\mathbb{Z}_m$.

Given an arbitrary polynomial $Q(X)$, we first divide by $N_{k(m)}(X)$ to get a polynomial $Q'(X)$ of degree $k(m)-1$. Since the polynomials $N_j(X)$ is monic and of degree j for $j \in \{0, \cdots, k(m)-1\}$, we can write $Q(X)$ as

$$Q'(X) = \sum_{j=0}^{k(m)-1} c_j N_j(X)$$

We can reduce this to the form of Equation 26 by subtracting an appropriate multiple of $\frac{m}{(m,j!)}N_j(X)$ for $j \leq k(m)$. Since, we are only subtracting polynomials that are 0, over $\mathbb{Z}_m$, the resulting polynomial computes the same function as the polynomial $Q(X)$ that we started with.

To show that this representations is unique, take two polynomials

$$P(X) = \sum_{j=0}^{k(m)-1} c_j N_j(X) \qquad 0 \leq c_j < \frac{m}{(m,j!)}$$

$$Q(X) = \sum_{j=0}^{k(m)-1} d_j N_j(X) \qquad 0 \leq d_j < \frac{m}{(m,j!)}$$

Pick the smallest index j so that $c_j \neq d_j$, and assume that $c_j > d_j$. We claim that $P(j) \not\equiv Q(j) \bmod m$. Since $N_i(j) = 0$ for $i > j$ and $c_i = d_i$ for $i < j$, we have

$$P(j) - Q(j) \;=\; (c_j - d_j)N_j(j) \;=\; (c_j - d_j)j! \;\not\equiv\; 0 \bmod m$$

since

$$0 \;<\; c_j - d_j \;<\; \frac{m}{(m,j!)}.$$

□

An easy consequence is the following result (attributed to Dueball in[63]).

Corollary 4.7 *[63] Every polynomial function over $\mathbb{Z}_m$ can be interpolated from its evaluations at $\{0, \cdots, k(m)-1\}$.*

PROOF: Let

$$P(X) = \sum_j c_j N_j(X) \qquad 0 \leq c_j < \frac{m}{(m,j!)}.$$

We let $c_0 = P(0)$. Assuming we know $c_0, \cdots, c_{j-1}$, we solve for c_j from the equation

$$c_j j! \equiv P(j) - \sum_{i<j} c_i N_i(j) \mod m$$

The coefficients of $P(X)$ in the canonical form are a solution to this equation. Further, the solution must be unique, since the canonical form is unique. □

The following estimates for $k(m)$ show that the number of queries can be significantly smaller than m if m is smooth.

Lemma 4.8 *For prime powers,*

$$p(a-1)+1 \;\leq\; k(p^a) \;\leq\; pa.$$

If $m = \prod_j p_j^{a_j}$, *then*

$$k(m) = \max_j k(p_j^{a_j}).$$

PROOF: Since $(pa)! \equiv 0 \bmod p^a$, $k(p^a) \leq pa$. Let $k = \sum_i k_i p^i$ be the base-p expansion of k. Using a formula due to Legendre,

$$\mathrm{val}_p(k!) \;=\; \sum_i \left\lfloor \frac{k}{p^i} \right\rfloor \;=\; \frac{k - \sum k_i}{p-1} \tag{27}$$

Hence if $k! \equiv 0 \bmod p^a$, then

$$\frac{k - \sum k_i}{p-1} \geq a$$

$$\text{Hence} \qquad k \;\geq\; (p-1)a + \sum_i k_i \;\geq\; (p-1)a + 1$$

For $m = \prod_j p_j^{a_j}$, by Chinese Remaindering,

$$k! \equiv 0 \bmod m \iff k! \equiv 0 \bmod p_j^{a_j}$$

Hence $k(m) = \max_j k(p_j^{a_j})$. □

Next we show that the problem of computing $k(m)$ from m is as hard as factoring m.

Lemma 4.9 *The problem of computing* $k(m)$ *given* m *as input is equivalent to factoring* m.

PROOF: One can check in polynomial time if m is a prime power [18], so assume it is not. We will show that $(k(m), m)$ gives a non-trivial factor of m. Note $k(m) = \max_j k(p_j^{a_j})$. Assume this maximum is attained for the prime p_i. Note that $k(p^a) \equiv 0 \bmod p$, else

$$\mathrm{val}_p(k(p^a)!) = \mathrm{val}_p((k(p^a)-1)!)$$

which contradicts the Definition of $k(p^a)$. Then $k(m) = k(p_i^{a_i})$ is divisible by p_i. Further

$$k(p_i^{a_i}) \ \leq \ p_i a_i \ \leq \ p_i^{a_i} \ < \ m$$

since m is not a prime power. Hence

$$p_i \ \leq \ (k(m), m) \ < \ m.$$

Thus we get a non-trivial factor of m. If $(k(m), m)$ is not a prime power, we can repeat this procedure till we get a prime power divisor of m. □

4.4 Interpolating Sets

We say that a polynomial $P(X)$ is 0 over set S if it evaluates to 0 at every point in S.

Definition 4.1 *Given $I \subseteq \mathbb{Z}_m$, $S \subseteq I$ is an interpolating set for I if every polynomial which is 0 over S is 0 over I. Let $k(I)$ denote the size of the smallest interpolating set for I.*

Note that I itself is trivially an interpolating set. However in general there can be interpolating sets which are significantly smaller than I. Note that two polynomials $P(X)$ and $Q(X)$ that agree at S must in fact agree at every point in I, by considering $P(X) - Q(X)$. Thus the values of a polynomial over I are uniquely determined by the values at points in an interpolating set. The next lemma shows that the minimum number of queries to interpolate a polynomial over I is $k(I)$.

Lemma 4.10 *The set of black-box queries of any interpolation algorithm is an interpolating set for I.*

PROOF: Assume that the set S of queries to the black-box is not an interpolating set. Then there exists polynomial $Q(X) \in \mathbb{Z}_m[X]$ such that $Q(x)$ is 0 at all $x \in S$ but non-zero at some point $y \in I$. Hence the algorithm cannot distinguish between polynomials $P(X)$ and $P(X) + Q(X)$ which agree on S but are different at $y \in I$. □

Note that this bound holds even if the algorithm chooses its queries adaptively.

Definition 4.2 *Let $\bar{k}(I)$ be the smallest integer such that there is a degree $\bar{k}(I)$ monic polynomial $M(X) \in \mathbb{Z}_m[X]$ which is 0 over I.*

If S is an interpolating set of size $k(I)$, then the polynomial $\prod_{\alpha \in S}(X - \alpha)$ is a monic polynomial of degree $k(I)$. It is zero over S and hence over I. Hence

$$\bar{k}(I) \leq k(I) \tag{28}$$

This lets us prove lower bounds on $k(I)$ by showing that any polynomial that is 0 over I must have certain degree.

EXAMPLE 1 For $I \subseteq \mathbb{Z}_p$, $\bar{k}(I) = k(I) = |I|$. Since $\mathbb{Z}_p$ is a field, the smallest degree monic polynomial which is 0 over I is $M(X) = \prod_{\alpha \in I}(X - \alpha)$, hence $\bar{k}(I) = |I|$.

EXAMPLE 2 For $I = \mathbb{Z}_m$, $\bar{k}(I) = k(I) = k(m)$. By Corollary 4.7, $S = \{0, \cdots, k(m) - 1\}$ is an interpolating set so $k(I) \leq k(m)$. To show $\bar{k}(I) \geq k(m)$, assume that $M(X)$ is a monic polynomial of degree $d < k(m)$. Writing it in the canonical form, we get $M(X) = \sum_{i \leq d} c_i N_i(X)$ where $c_d = 1$. So $M(X)$ cannot be zero over $\mathbb{Z}_m$ by Lemma 4.6.

One can use the CRT to relate the problem of computing $k(I)$ and $\bar{k}(I)$ for $I \subseteq \mathbb{Z}_m$ for composite m to the prime power case . First we need to introduce some notation. Let $m = \prod_{j=1}^{t} p_i^{a_i}$. Given a set $L \subseteq \mathbb{Z}_m$ we define the *projection* L_j of L mod $p_j^{a_j}$ as

$$L_j = \{y \in \mathbb{Z}_{p_j{}^{a_j}} | \exists\ x \in L, x \equiv y \bmod p_j^{a_j}\}$$

For a polynomial $P(X) \in \mathbb{Z}_m[X]$, we define $P_j(X) \in \mathbb{Z}_{p_j{}^{a_j}}[X]$ to be its *projection* modulo $p_j^{a_j}$ obtained by taking each coefficient of $P(X)$ modulo $p_j^{a_j}$. Conversely, given polynomials $P_j(X) \in \mathbb{Z}_{p_j{}^{a_j}}[X]$ we can combine the coefficients using the CRT to get a unique polynomial $P(X) \in \mathbb{Z}_m[X]$ whose projections are the polynomials $P_j(X)$. We call $P(X)$ the *lift* of the $P_j(X)$s.

Lemma 4.11 *Let $I \subseteq \mathbb{Z}_m$. Then*

$$\bar{k}(I) \quad = \quad \max_j\ \bar{k}(I_j) \tag{29}$$

PROOF: It is easy to show using the CRT that a polynomial $P(X)$ is zero over I iff $P_j(X)$ is zero over I_j for all j. Let $M(X)$ be a monic polynomial which is zero over $I \subseteq \mathbb{Z}_m$. Then by the CRT, $M_j(X)$ is a monic polynomial which is 0 over $I_j \subseteq \mathbb{Z}_{p_j{}^{a_j}}$. Hence $k(I) \geq k(I_j)$ for every j.

Conversely let $\max_j \bar{k}(I_j) = d$. For each j, there is a monic polynomial $M_j(X)$ of degree $d_j \leq d$ which is zero over I_j. The polynomial

$$M'_j(X) = X^{d-d_j} M_j(X)$$

is a degree d monic polynomial which is zero over I_j. Let $M'(X) \in \mathbb{Z}_m[X]$ be the lift of the $M'_j(X)$s. It follows that $M'(X)$ is a monic polynomial of degree d and it is zero over I. □

Lemma 4.12 *Let $I \subseteq \mathbb{Z}_m$. The set S is an interpolating set for I iff S_j is an interpolating set for I_j.*

PROOF: Assume that S_j is an interpolating set for I_j for every j, but S is not an interpolating set for I. Then there is a polynomial $Q(X) \in \mathbb{Z}_m[X]$ such that at every point $x \in S, Q(x) \equiv 0 \bmod m$, but for some $y \in I,\ Q(y) \not\equiv 0 \bmod m$. But then $Q(y) \not\equiv 0 \bmod p_j^{a_j}$ for some j. Consider the polynomial $Q_j(X)$. Since $Q(X)$ is zero over S, $Q_j(X)$ is zero over S_j. However there exists $y' \equiv y \bmod p_j^{a_j}$ in I_j such that $Q_j(y') \not\equiv 0 \bmod p_j^{a_j}$. This contradicts the assumption that S_j is an interpolating set for I_j.

In the other direction, assume that S is an interpolating set for I but S_j is not an interpolating set for I_j. Then there is a polynomial $Q_j(X) \in \mathbb{Z}_{p_j{}^{a_j}}[X]$ such that for every $x \in S_j, Q(x) \equiv 0 \bmod p_j^{a_j}$, but there exists $y \in I_j$ such that $Q(y) \not\equiv 0 \bmod p_j^{a_j}$. Take $Q_i(X) = 0$ for $i \neq j$ and set $Q(X) \in \mathbb{Z}_m[X]$ to be the lift of the $Q_i(X)$s. Then $Q(X)$ is zero over S since it is zero over every S_j. But it is not zero at some point in I since Q_j is not zero over I_j. This contradicts the assumption that S is an interpolating set. □

Corollary 4.13 *Let $I \subseteq \mathbb{Z}_m$. Then*

$$\max_j\ k(I_j) \quad \leq \quad k(I) \quad \leq \quad \sum_{j=1}^{t} k(I_j) \tag{30}$$

PROOF: The bound $k(I) \geq k(I_j)$ follows trivially since $|S| \geq |S_j| \geq k(I_j)$. To prove the other direction, let S_j be a minimum interpolating set for I_j. For $y \in S_j$, there exists a preimage $x \in I$ such that $x \equiv y \bmod p_j^{a_j}$. Define $S'_j \subseteq I$ by choosing one preimage for each y. Set $T = \cup_j S'_j$. T is an interpolating set for I since $S_j \subseteq T_j$ is an interpolating set for I_j. Also $|T| \leq \sum_j k(I_j)$. □

EXAMPLE 3 We give an example where $\bar{k}(I) < k(I)$ and the upper bound in Equation 30 is (near) tight. Let $m = p_1 p_2$, and let

$$I = \{ap_1 | 1 \leq a \leq p_2 - 1\} \cup \{bp_2 | 1 \leq b \leq p_1 - 1\}$$

It is easy to see that $\bar{k}(I_1) = k(I_1) = p_1$, $\bar{k}(I_2) = k(I_2) = p_2$ hence $\bar{k}(I) = \max(p_1, p_2)$. On the other hand, the only interpolating set for I is I itself. Each point of the form ap_1 must be included since it is the only point in its congruence class modulo p_2. Similarly, every point bp_2 must be included. Hence $k(I) = p_1 + p_2 - 2$.

The following extension of the above lemmas can be proved similarly using the CRT.

Corollary 4.14 *Let $m = \prod_{j=1}^{t'} m_j$ and $(m_i, m_j) = 1$. Let I_j denote the projection of I modulo m_j.*

$$\bar{k}(I) = \max_j \bar{k}(I_j)$$

$$\max_j k(I_j) \leq k(I) \leq \sum_j k(I_j)$$

Theorem 4.15 *The problem of computing $k(I)$ given I and m as input is* NP*-hard.*

PROOF: Consider the following decision problem:

Problem 4.2 MIN-INTERPOLATING-SET: *Given m and $I \subseteq \mathbb{Z}_m$, is $k(I) \leq n$?*

We prove this problem is NP-hard by reduction from 3D-matching [33].

Problem 4.3 3-DIMENSIONAL MATCHING: *Given sets U, V, W of size n and a set of edges $E \subseteq U \times V \times W$, is there a subset of E of size n that covers all the vertices?*

Take p_1, p_2, p_3 to be 3 distinct primes greater than n. Let $m = p_1 p_2 p_3$. For each triple $(u_i, v_j, w_k) \in E$ with $i, j, k \leq n$, we add a number $x \in \mathbb{Z}_m$ to I where $x \equiv i \bmod p_1,\ x \equiv j \bmod p_2,\ x \equiv k \bmod p_3$. We claim that there is a matching of size n iff the set I has an interpolating set of size n. We may assume w.l.o.g that every vertex occurs in some edge, hence $|I_1| = |I_2| = |I_3| = n$. Thus S is an interpolating set for I iff $S_j = I_j$ for $1 \leq j \leq 3$. Thus an interpolating set corresponds to a set of edges that cover every vertex. If there is an interpolating set of size n, then there is a cover of size n and vice versa.

In fact, it is possible to show that the MIN-INTERPOLATING-SET problem is NP-complete, we skip the proof. □

In Theorem 4.17, we will show that for $I \subseteq \mathbb{Z}_{p^a}$, $\bar{k}(I) = k(I)$. Combining this with Equations 29 and 30

$$\bar{k}(I) \ \leq \ k(I) \ \leq \ t\bar{k}(I) \tag{31}$$

Thus $\bar{k}(I)$ is a factor t approximation to $k(I)$ where t is the number of prime divisors of m.

4.5 Algorithms for the Generalized Interpolation Problem

4.5.1 The Prime Power Case

We give an algorithm to solve the generalized polynomial interpolation problem over $\mathbb{Z}_{p^a}$ using exactly $k(I)$ queries. We first give a (greedy) algorithm to find a minimum interpolating set.

We start by picking an arbitrary element in I. Suppose we have chosen $\{\alpha_0, \cdots, \alpha_{i-1}\}$ so far. If the polynomial $N_i^S(X) = \prod_{j<i}(X - \alpha_j)$ is 0 over I we stop. Else we choose the next element $\alpha_i \in I$ so that $\mathrm{val}_p(N_i(\alpha_i))$ is minimized.

Algorithm 4.1 IntSet(I, p^a)

Input: Set $I \subseteq \mathbb{Z}_{p^a}$.

Output: Interpolating set S for I.

Pick $\alpha_0 \in I$ arbitrarily. Set $S = \{\alpha_0\}, i = 1$.

Repeat

  Let $N_i^S(X) = \prod_{j<i}(X - \alpha_j)$.

  If $N_i^S(x)$ is zero for all $x \in I$,

    Output $S = \{\alpha_0, \cdots, \alpha_{i-1}\}$. Stop.

  Else

    Find $x \in I$ that minimizes $\mathrm{val}_p(N_i^S(x))$.

    Set $\alpha_i = x$, $i = i + 1$.

Assume that the algorithm outputs a set $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$ of size k and let $e_i = \mathrm{val}_p(N_i^S(\alpha_i))$.

Lemma 4.16 *Every polynomial function over I is computed by a unique polynomial of the form*

$$P(X) = \sum_{j=0}^{k-1} c_j N_j^S(X) \qquad 0 \le c_j < p^{a-e_j} \tag{32}$$

PROOF: The Proof is similar to that of Lemma 4.6. Given any polynomial $Q(X)$, we give an algorithmic procedure to construct $P(X)$ with the above form that agrees with $Q(X)$ on I. By the termination condition, the polynomial $N_k^S(X) = \prod_{j<k}(X - \alpha_j)$ is identically zero over I. Dividing $Q(X)$ by $N_k^S(X)$ and taking the remainder, we get $Q'(X)$ of degree $k-1$ that computes the same function on I. Let us set $N_0^S(X) = 1$. Since the polynomials $N_j^S(X)$ is monic and of degree j for $j \in \{0, \cdots, k-1\}$, we can write any polynomial of degree at most $k-1$ as a linear combination of these polynomials. Hence we have

$$Q'(X) = \sum_{j<k} c_j N_j^S(X)$$

Note that by our choice of α_j,

$$e_j \;=\; \mathrm{val}_p(N_j^S(\alpha_j)) \;\le\; \mathrm{val}_p\, N_j^S(x) \;\text{ for } x \in I.$$

So the polynomials $p^{a-e_j}N_j^S(X)$ are 0 over I. Hence, by subtracting appropriate multiples of these polynomials from $Q'(X)$ we can get a polynomial $P(X)$ where $0 \leq c_j < p^{a-e_j}$ that computes the same function as $Q(X)$.

To show uniqueness of this representation, consider two polynomials

$$P(X) = \sum_{j=0}^{k-1} c_j N_j^S(X) \quad 0 \leq c_j < p^{a-e_j}$$

$$Q(X) = \sum_{j=0}^{k-1} d_j N_j^S(X) \quad 0 \leq d_j < p^{a-e_j}$$

with different canonical forms. Pick the smallest j such that $c_j \neq d_j$. We claim that $P(\alpha_j) \neq Q(\alpha_j)$. Note that

$$P(\alpha_j) - Q(\alpha_j) = \sum_i (c_i - d_i) N_i^S(\alpha_j).$$

Since $N_i^S(\alpha_j) = 0$ for $i > j$ and $c_i = d_i$ for $i < j$, we have

$$P(\alpha_j) - Q(\alpha_j) = (c_j - d_j) N_j^S(\alpha_j).$$

Since $\mathrm{val}_p(N_j^S(\alpha_j)) = e_j$ and $\mathrm{val}(c_j - d_j) < a - e_j$ hence

$$P(\alpha_j) - Q(\alpha_j) \not\equiv 0 \bmod p^a.$$

□

Theorem 4.17 *The set S is a minimum interpolating set. In fact $\bar{k}(I) = k(I) = |S|$.*

PROOF: Since S is an interpolating set of size k, $k(I) \leq k$. It suffices to show that $\bar{k}(I) \geq k$. Let $M(X)$ be a monic polynomial of degree $d \leq k-1$. We can put $M(X)$ in the canonical form using the procedure above to get

$$M(X) = \sum_{j \leq d} c_j N_j^S(X) \qquad 0 \leq c_j < p^{a-e_j}.$$

Since $M(X)$ is monic, it follows that $c_d = 1$. Hence $M(X)$ does not compute the 0 function by Lemma 4.16. So $\bar{k}(I) \geq k$. □

Note that Algorithm 4.1 for picking a minimum interpolating set is essentially a greedy algorithm: at each stage it picks a new element x that minimizes $\sum_{j<i} \mathrm{val}_p(x - \alpha_j)$. One

can ask what objective function is being optimized by this greedy algorithm. In Theorem 4.30 (proved in Section 4.7), we prove that this algorithm minimizes the power of p that divides the Vandermonde determinant of $\prod_{i<j}(\alpha_i - \alpha_j)$. In other words, the minimum interpolating sets of I are all subsets $S = \{\beta_i\}$ of size $k(I)$ that minimize

$$\sum_{i<j\leq k(I)} \mathrm{val}_p(\beta_i - \beta_j).$$

This gives a simple algorithm to check if a set $T = \{\beta_i\}$ is a minimum interpolating set for I. We first compute an interpolating set $S = \{\alpha_i\}$ using Algorithm 4.1 and then check that $|T| = |S|$ and that $\sum \mathrm{val}_p(\beta_i - \beta_j) = \sum \mathrm{val}_p(\alpha_i - \alpha_j)$.

We now give an algorithm for Polynomial Interpolation over $\mathbb{Z}_{p^a}$ whose query complexity is optimal. One can show that this Algorithm computes the canonical form of $P(X)$ using an argument similar to Corollary 4.7.

Algorithm 4.2 Interpolate($I, \mathbb{Z}_{p^a}$)

Input: Set $I \subseteq \mathbb{Z}_{p^a}$, an black-box for $P(X)$ evaluated at I.

Output: The polynomial $P(X)$.

Compute $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$ using IntSet(I, p^a).

For $i = 0, \cdots, k-1$,

  Query $P(\alpha_i)$.

  Compute c_i so that $0 \leq c_i < p^{a-e_i}$ and

$$P(\alpha_i) \equiv \sum_{j\leq i} c_j N_j^S(\alpha_i) \bmod p^a$$

Output $P(X) = \sum_{i<k} c_i N_i^S(X)$.

4.5.2 The General Interpolation Problem

We first give an algorithm to find interpolating sets over $\mathbb{Z}_m$. The algorithm is given I as input, it does not have the factorization of m. It computes an interpolating set for I. We sketch the idea of the algorithm for $m = pq$. The algorithm tries to add elements to

S greedily like in the prime power case. Assume we have picked $\{\alpha_0, \cdots, \alpha_{i-1}\}$ and let $N_i^S(X) = \prod_{j<i}(X - \alpha_i)$. We compute $g(x) = (N_i^S(x), m)$ for every $x \in I$. This quantity plays the role of $\mathrm{val}_p(N_i^S(X)$ in Algorithm 4.1.

1. If there is an x such that $\mathrm{val}_p(N_i^S(X))$ and $\mathrm{val}_q(N_i^S(X))$ are both minimized at x, then $g(x)|g(y)$ for all $y \in I$. We add x to S and proceed.

2. If $\mathrm{val}_p(N_i^S(X))$ and $\mathrm{val}_q(N_i^S(X))$ are minimized at distinct points x and y, then $g(x) \not| g(y)$ and vice versa. Here the greedy approach fails. But in this case we can efficiently factor $m = pq$ using $g(x)$ and $g(y)$. We then use divide and conquer.

For general m, in case 2 we compute a factorization $m = m_1 m_2$ where $(m_1, m_2) = 1$ using the subroutine Factor and then use divide and conquer.

Algorithm 4.3 IntSet(I, m)

Input: Set $I \subseteq \mathbb{Z}_m$.

Output: An factorization $m = \prod_{j=1}^{t'} m'_j$ where $(m'_i, m'_j) = 1$ and a minimum interpolating set S_j for $I_j = I \bmod m'_j$.

Pick $\alpha_0 \in I$ arbitrarily. Set $S = \{\alpha_0\}$, $i = 1$.

Repeat

  Let $N_i^S(X) = \prod_{j<i}(X - \alpha_j)$.

  If $N_i^S(x)$ is zero for all $x \in I$,

    Output $S = \{\alpha_0, \cdots, \alpha_{i-1}\}, m$. Stop.

  Else

    For each $x \in I$, set $g(x) = (N_i^S(x), m)$

    If some $g(x)$ divides $g(y)$ for all $y \in I$,

      Set $\alpha_i = x$, $i = i + 1$.

    Else

      Find $g(x), g(y)$ that do not divide each other.

      Factor$(m, g(x), g(y))$ = $m_1 \cdot m_2$

      Return IntSet(I_1, m_1), IntSet(I_2, m_2). Stop.

We first analyze the Algorithm when Factor is not called. We then present the factoring subroutine. In particular, Lemmas 4.18, 4.19, 4.20 all assume that Factor was not called. In this case, the behavior of the algorithm is similar to the prime power case.

Let $m = \prod_{j=1}^{t} p_j^{a_j}$. Let $S = \{\alpha_i\}$ be the set output. Let I_j and S_j be the projections of I and S modulo $p_j^{a_j}$. We show that if Factor is not called, then the Algorithm finds a minimum interpolating set by showing $\bar{k}(I) = k(I) = |S|$. This is done by simulating Algorithm 4.1 on I_j and showing that it would produce the same outcome.

Fix a prime p_j. Let $\alpha'_i \equiv \alpha_i \bmod p_j^{a_j}$. We take T to be the projection of the first $k(I_j)$ elements of I. In other words, let $T = \{\alpha'_1, \ldots, \alpha'_{k(I_j)}\}$. Note that $T \subseteq S_j$.

Lemma 4.18 *The set T is a minimum interpolating set for I_j.*

PROOF: We will show that $\mathrm{val}_{p_j}(N_i^T(x))$ is minimized over I_j at α_i'. Hence the set T is a possible output when we run Algorithm 4.1 on the set I_j.

Assume that there is a $y' \in I_j$ such that

$$\mathrm{val}_{p_j}(N_i^T(y')) < \mathrm{val}_{p_j}(N_i^T(\alpha_i')).$$

Choose $y \in I$ so that $y \equiv y' \bmod p_j^{a_j}$. Note that

$$\mathrm{val}_{p_j}(N^T(y')) = \mathrm{val}_{p_j}(g(y)), \quad \mathrm{val}_{p_j}(N^T(\alpha_i')) = \mathrm{val}_{p_j}(g(\alpha_i))$$

$$\text{Hence} \qquad \mathrm{val}_{p_j}(g(y)) < \mathrm{val}_{p_j}(g(\alpha_i)).$$

So $g(\alpha_i)$ cannot divide $g(y)$. But since Factor is not used, α_i satisfies $g(\alpha_i)|g(y)$ for all $y \in I$, which is a contradiction. □

Lemma 4.19 *The set S is a minimum interpolating set for I. In fact, $\bar{k}(I) = k(I) = |S|$.*

PROOF: By Lemma 4.18, the set S_j is an interpolating set for I_j so S is an interpolating set for I. We will show that $\bar{k}(I) = k(I) = |S|$.

For each j, the polynomial $\prod_{i<k(I_j)}(X - \alpha_i)$ is 0 mod $p_j^{a_j}$ over I_j because the first $k(I_j)$ elements are an interpolation set for I_j. Take $k = \max_j k(I_j)$. The polynomial $\prod_{i<k}(X-\alpha_i)$ is 0 mod m over I. Since this is the termination condition for Algorithm 4.3, it will stop after k steps and output S of size $k = \max_j k(I_j)$. By Equation 30, we have

$$\max_j k(I_j) \;\leq\; k(I).$$

Hence the set S is a minimum interpolating set. Further by Equation 29,

$$\bar{k}(I) \;= \max_j \bar{k}(I_j)$$

But for prime powers, $\bar{k}(I_j) = k(I_j)$. Hence we conclude that

$$\bar{k}(I) \;=\; \max_j \bar{k}(I_j) \;=\; \max_j k(I_j) \;=\; k(I).$$

□

$P(X)$ can be computed by a procedure similar to Algorithm 4.2.

Lemma 4.20 *The polynomial $P(X)$ can be computed from the values at points in S.*

PROOF: The proof of correctness is similar to Corollary 4.7. Note that

$$\frac{m}{(m, N_i^S(\alpha_i))} N_i^S(\alpha_i) \equiv 0 \bmod m$$

and for every $x \in I$,

$$N_i^S(x) = y \cdot N_i^S(\alpha_i) \bmod m \qquad \text{for some } y \in \mathbb{Z}_m.$$

Hence for $i \leq k$, the polynomials

$$\frac{m}{(m, N_i^S(\alpha_i))} N_i^S(X)$$

are 0 over I.

Hence every polynomial function over I can be canonically represented as

$$P(X) = \sum_{i=0}^{k-1} c_i N_i^S(X) \qquad 0 \leq c_i < \frac{m}{(m, N_i^S(\alpha_i))}$$

To compute the canonical form of $P(X)$, we query the value of $P(X)$ at every point in S. For $i \leq k-1$ we solve the equation

$$c_i N_i^S(\alpha_i) \equiv P(\alpha_i) - \sum_{j<i} c_j N_j^S(\alpha_i) \mod m \qquad 0 \leq c_i < \frac{m}{(m, N_i^S(\alpha_i))}$$

The unique solution to this system is the canonical from of $P(X)$. □

We now turn to the subroutine for factoring. The idea is to use $g(x)$ and $g(y)$ to get m_1, m_2 which divide m and are relatively prime. Their product $m_1 m_2$ might be less than m. At each step, we take a non-trivial divisor of $\frac{m}{m_1 m_2}$ and multiply either m_1 or m_2 by it. We do this in such a way that they stay relatively prime.

Algorithm 4.4 Factor(m,g(x),g(y))

Input: A number m and $g(x), g(y)$ that divide m but do not divide each other.

Output: $m_1 \cdot m_2 = m$ and $(m_1, m_2) = 1$.

Let $g = (g(x), g(y))$. Let $m_1 = \frac{g(x)}{g}, m_2 = \frac{g(y)}{g}$.

Repeat

  Set $c = \frac{m}{m_1 \cdot m_2}$.

  If $(c, m_1) = 1$, Set $m_2 = m_2 \cdot c$.

  Else, Set $m_1 = m_1 \cdot (c, m_1)$.

  If $m_1 \cdot m_2 = m$, Output m_1, m_2. Stop.

Lemma 4.21 *Factor$(m, g(x), g(y))$ returns m_1, m_2 such that $m_1 \cdot m_2 = m$ and $(m_1, m_2) = 1$.*

PROOF: At the start of the algorithm,

$$m_1 = \frac{g(x)}{(g(x), g(y))}, \quad m_2 = \frac{g(y)}{(g(x), g(y))}$$

so $(m_1, m_2) = 1$. Also m_1, m_2 are non-trivial divisors of m since $g(x)$ and $g(y)$ do not divide each other.

Let $c = \frac{m}{m_1 m_2}$. If $(m_1, c) = 1$, since $(m_1, m_2) = 1$, we have $(m_1, cm_2) = 1$. So we set $m_2 = cm_2$ and we are done. If $(c, m_1) = d > 1$, then since d divides m_1, we have $(d, m_2) = 1$. So we set $m_1 = dm_1$. In either case, the product $m_1 m_2$ increases by a factor of 2, hence the algorithm terminates in $O(\log m)$ iterations. □

The subroutine above is somewhat inefficient. The running time can be considerably improved by running the factor refinement algorithm of Bernstein [19] on $g(x), g(y)$ and m. This algorithm gives a factorization into co-primes in near linear time.

If we find factors m_1, m_2 which are relatively prime, then we run IntSet(I_1, m_1) and IntSet(I_2, m_2). In doing so we could find further factors of m_1 and m_2, but these will be

relatively prime, since m_1 and m_2 are relatively prime. Hence finally, the algorithm returns a factorization $m = \prod_{i \leq t'} m'_i$ where the m'_is are relatively prime. If m has t distinct prime factors then clearly $t' \leq t$. We now solve the interpolation problem modulo m'_j using Lemma 4.20 and combine the results using the CRT.

Algorithm 4.5 Interpolate($I, \mathbb{Z}_m$)

Input: Set $I \subseteq \mathbb{Z}_m$, an black-box for $P(X)$ evaluated at I.

Output: The polynomial $P(X)$.

Using IntSet(I, m), compute $m = \prod_{j=1}^{t'} m'_j$ and interpolating sets S_j for I_j.

For each $j \in 1, \cdots, t'$

  For each $y \in S_j$,

    Query $P(X)$ at $x \in I$ s.t. $x \equiv y \bmod m'_j$.

  Use these to compute $P_j(X) \bmod m'_j$.

Lift the polynomials $P_j(X)$ to a polynomial $P(X) \in \mathbb{Z}_m[X]$ using the CRT.

Lemma 4.22 *Algorithm 4.5 solves the interpolation problem over $\mathbb{Z}_m[X]$. The number of queries is within a factor t' of the optimal.*

PROOF: The proof that the polynomial $P(X)$ is correct follows by the CRT. The number of queries is at most $\sum_{j \leq t'} |S_j|$. By Lemma 4.19 since each m_j is not factored further, the set S_j is a minimum interpolating set for I_j. Hence $|S_j| = k(I_j)$. Hence by Corollary 4.14,

$$\max_j |S_j| \ \leq \ k(I) \ \leq \ \sum_{j \leq t'} |S_j| \ \leq \ t' k(I).$$

□

Also by Corollary 4.14,

$$\bar{k}(I) \ = \ \max_j \bar{k}(I_j) \ = \ \max_j |S_j|.$$

Hence Algorithm 4.3 can be used to compute $\bar{k}(I)$ exactly. (Note that this can also be done by solving a system of linear equations).

4.6 Learning Algorithms

One can use the algorithms in the previous section to design efficient algorithms for interpolation over $\mathbb{Z}_m$ in various learning theoretic settings. We consider the problem of learning under the uniform distribution and PAC-learning under an arbitrary distribution. In the uniform distribution problem, we are given evaluations of a polynomial $P(X)$ at points x chosen at random from $\mathbb{Z}_m$. In the PAC-learning problem, the samples are drawn from an unknown distribution $\mathcal{D}$ over $\mathbb{Z}_m$. We are required to output a polynomial that computes $P(X)$ correctly with good probability on points chosen from the same distribution. In this setting, it is necessary to allow some error probability. Consider a distribution D which is concentrated on a set I which does not contain an interpolating set for $\mathbb{Z}_m$. A polynomial time algorithm cannot distinguish between the 0 function and a function which is 0 on I but non-zero elsewhere.

For learning algorithms, the notion of polynomial running time needs to be defined carefully. Let $F(m)$ denote the number of polynomial functions over $\mathbb{Z}_m$. The algorithm is required to output some polynomial function which requires at least $\log F(m)$ bits to represent. Hence we say the algorithm runs in polynomial time if the running time is $\text{poly}(\log F(m))$.

From Theorem 4.6, we get

$$F(m) = \prod_{0 \leq j < k(m)} \frac{m}{(m, j!)} \tag{33}$$

Note that $\log F(m)$ can vary between $\log m$ and m depending on the prime factorization of m. We compute a rough lower bound on $\log F(m)$ in terms of its factorization. Note that if $m = p^a$, it follows from Theorem 4.6 that

$$F(p^a) \;\geq\; (p^a)^p \;=\; p^{ap}.$$

Hence if $m = \prod_{j \leq t} p_j^{a_j}$, then

$$F(m) \geq \prod_j p_j^{a_j p_j} \;\Rightarrow\; \log F(m) \geq \sum_{j \leq t} a_j p_j \log p_j.$$

4.6.1 Learning under the Uniform Distribution

Problem 4.4 LEARNING UNDER THE UNIFORM DISTRIBUTION: *Given samples $(x, f(x))$, where x is drawn uniformly from $\mathbb{Z}_m$ and f is a polynomial function, find a polynomial $P(X)$ that computes f.*

Algorithm 4.6 Interpolation under the Uniform Distribution

Input: `Black-box for evaluations of` $P(X)$ `under the uniform distribution.`

Output: `The polynomial` $P(X)$`.`

`Compute the factorization` $m = \prod_{j=1}^{t} p_j^{a_j}$ `.`

`Draw samples till we have an interpolating set` S_j `for` $\mathbb{Z}_{p_j{}^{a_j}}$ `.`

`Compute` $P_j(X)$ `for each` j`.`

`Let` $P(X)$ `be the lift of the` $P_j(X)$`s.`

We compute the factorization using brute force which takes time $O(\sum_j p_j a_j) = O(F(m))$. We now bound the number of samples needed till we have an interpolating set modulo p^a. Let p^b be the smallest power of p such that $p^b > k(p^a)$. By Lemma 4.8, $p^b < p^2 a$. Let $S = \{\alpha_0, \cdots, \alpha_{k(p^a)-1}\}$ where $\alpha_i \equiv i \bmod p^b$.

Lemma 4.23 *The set S is an interpolating set for $\mathbb{Z}_{p^a}$.*

PROOF: By Corollary 4.7, $T = \{0, \cdots, k(p^a) - 1\}$ is an interpolating set for $\mathbb{Z}_{p^a}$. By the choice of α_i, α_j

$$\alpha_i - \alpha_j \equiv i - j + cp^b \bmod p^a$$

Since $0 \leq i \neq j < p^b$, $\mathrm{val}_p(\alpha_i - \alpha_j) = \mathrm{val}_p(i - j)$. Hence

$$\sum_{i<j\leq k(p^a)} (\alpha_i - \alpha_j) = \sum_{i<j\leq k(p^a)} (i - j)$$

So by Theorem 4.30, S is an interpolating set. □

Lemma 4.24 *Algorithm 4.6 requires $O(\log^2 F(m))$ samples with high probability.*

PROOF: The uniform distribution over $\mathbb{Z}_m$ induces the uniform distribution over congruence classes modulo $p_j^{b_j}$ since $b_j < a_j$. By the coupon collector's problem, in time $O(p_j^{b_j} \log(p_j^{b_j}))$ we will see a sample from each congruence class with high probability. By Lemma 4.23, this gives an interpolating set modulo $p_j^{a_j}$. Overall the number of samples needed can be bounded by $O(\log^2 F(m))$ w.h.p. □

Theorem 4.25 *Algorithm 4.6 learns the polynomial $P(X)$ exactly under the uniform distribution. It runs in time $O(\log^2 F(m))$ with high probability.*

We needed to factor m to check whether the set of points seen so far is an interpolating set for $\mathbb{Z}_m$. Is there an algorithm to check if S is an interpolating set for $\mathbb{Z}_m$ that does not need to factor m?

4.6.2 PAC Learning

Problem 4.5 PAC LEARNING: *Given samples $(x, f(x))$, where x is drawn from an unknown distribution $\mathcal{D}$ and f is a polynomial function, find a polynomial $P(X)$ that computes f over $\mathcal{D}$ with probability $1 - \varepsilon$.*

Polynomial functions are PAC learnable under an arbitrary distribution in polynomial time. Once we have drawn the set of samples, the problem reduces to one of general interpolation. The number of samples to be drawn can be determined from $F(m)$ using Occam's Razor [52]. We first compute $F(m)$ using Equation 33. This can be done in time $O(\log F(m))$. We then draw $\frac{1}{\varepsilon} \log \frac{F(m)}{\delta}$ samples from $\mathcal{D}$ and solve the interpolation problem on these inputs using Algorithm 4.5. The proof that this suffices for PAC-learning is standard [52].

Theorem 4.26 *Polynomial over $\mathbb{Z}_m$ are PAC learnable in polynomial time using $\frac{1}{\varepsilon} \log \frac{F(m)}{\delta}$ queries.*

4.7 Algebraic Structure of Interpolating Sets modulo Prime Powers

In this section we study the algebraic properties of interpolating sets modulo prime powers. We give alternate algebraic characterizations of such sets (Theorems 4.29, 4.30 and 4.34).

In this section and the next, we use p-adic distance as opposed to valuations.

Recall that by the definition of Interpolating sets, every polynomial which is non-zero over I is in fact non-zero over some point in S. The next Lemma generalizes this to show that in fact, the norm of every polynomial is maximized over I at some point in S.

Lemma 4.27 *A set S is an interpolating set iff for any polynomial $P(X)$, there exists $\alpha \in S$ such that*

$$|P(\alpha)|_p \geq |P(x)|_p \quad \forall x \in I \tag{34}$$

PROOF:($\Rightarrow$) Assume there exists $P(X) \in \mathbb{Z}_{p^a}$ such that $|P(x)|_p > |P(\alpha)|_p \ \forall \alpha \in S$. But then for an appropriately chosen e, $p^e P(X)$ is non-zero at x, but 0 everywhere in S. Hence S cannot be an interpolating set.

($\Leftarrow$) Assume that S satisfies equation 34. There cannot exist a polynomial $P(X)$ which is 0 on S, but $P(x) \neq 0$ for some $x \in \mathbb{Z}_{p^a}$, since this implies that $|P(x)|_p > |P(\alpha)|_p \ \forall \alpha \in S$. Hence S is an interpolating set. □

This property of interpolating sets allows us to order its elements in a natural manner. Given an ordered set $S = \{\alpha_0, \alpha_1, \cdots\}$, let $N_j^S(X) = \prod_{i<j}(X - \alpha_i)$.

Algorithm 4.7 Ordering an Interpolating Set

Input: An interpolating set T of I.

Output: An ordered set $S \subseteq T$ which is a (minimal) interpolating set.

Pick $\alpha_0 \in T$ arbitrarily, put it in S.

Given $S = \{\alpha_0, \cdots, \alpha_{j-1}\}$.

  If $N_j^S(X)$ is 0 over T, stop and output $S = \{\alpha_0, \cdots, \alpha_{j-1}\}$.

  Else pick $\alpha_j \in T$ which maximizes $|N_j^S(\alpha_j)|_p$ and add it to S.

Assume that the above procedure outputs an ordered set $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$ of size k. Let $e_j = \mathrm{val}_p(N_j^S(\alpha_j))$ for $i \leq j \leq k-1$. Observe that $0 \leq e_j < a$. Using the argument of Theorem 4.16, we can show that every polynomial function over I is computed by a unique

polynomial of the form

$$P(X) = \sum_{j=0}^{t} c_j N_j^S(X) \quad 0 \leq c_j < p^{a-e_j}$$

Using the canonical form above, one can show that all minimal interpolating sets over $\mathbb{Z}_{p^a}$ have the same size. The proof is similar to that of Theorem 4.17.

Corollary 4.28 *The set $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$ is an interpolating set iff $|N_j^S(\alpha_j)|_p \geq |N_j^S(x)|_p$ for all $x \in I$.*

PROOF: Clearly an interpolating set with the canonical ordering has this property. To prove the other direction, simply take $T = I$ in Algorithm 4.7. Since $|N_j^S(x)|_p$ is maximized at α_j, we can add α_j to S at step j, giving the interpolating set $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$. $\square$

Henceforth we will assume that interpolating sets are canonically ordered and that polynomials are in the canonical form. Lemma 4.27 states that for any polynomial function $P(X)$, $|P(x)|_p$ is maximized at some point $\alpha \in S$. We strengthen this to show that if the degree of $P(X)$ is d, such an α can be found among the first $d+1$ elements in S.

Theorem 4.29 *The set $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$ is an interpolating set iff for every polynomial $P(X)$ of degree d, there exists $\alpha \in \{\alpha_0, \cdots, \alpha_d\}$ such that $|P(\alpha)|_p \geq |P(x)|_p \quad \forall x \in I$.*

PROOF: Clearly a set with this property is an interpolating set by Lemma 4.27. We prove the other direction. The proof is by induction on d. The base case when $d = 0$ is trivial. Assume the claim holds for $d-1$. Let $P(X) = Q(X) + c_d N_k^S(X)$ where $\deg(Q) \leq d-1$.

$$|P(x)|_p \leq \max(|Q(x)|_p, |c_d N_d^S(x)|_p) \qquad \text{(Ultrametric inequality)} \tag{35}$$

We bound $|Q(x)|_p$ using the inductive hypothesis. Since $Q(X)$ has degree $d-1$,

$$|Q(x)|_p \leq \max(|Q(\alpha_0)|_p, \cdots, |Q(\alpha_{d-1})|_p) \tag{36}$$

By our choice of α_d,

$$\begin{aligned}
|c_d N_d^S(x)|_p &\leq |c_d N_d^S(\alpha_d)| \\
&= |P(\alpha_d) - Q(\alpha_d)|_p \\
&\leq \max(|Q(\alpha_d)|_p, |P(\alpha_d)|_p) \\
&\leq \max(|Q(\alpha_0)|_p, \cdots, |Q(\alpha_{d-1})|_p, |P(\alpha_d)|_p) \qquad \text{(Induction on } Q(X)) \quad (37)
\end{aligned}$$

Hence from Equations 35, 36 and 37 we get

$$|P(x)|_p \leq \max(|Q(\alpha_0)|_p, \cdots, |Q(\alpha_{d-1})|_p, P(\alpha_d)|_p)$$

Since $N_d^S(\alpha_j) = 0$ for $j < d$, we have $Q(\alpha_j) = P(\alpha_j)$ for $j < d$. Hence

$$|P(x)|_p \leq \max(|P(\alpha_0)|_p, \cdots, |P(\alpha_{d-1})|_p, |P(\alpha_d)|_p) \quad (38)$$

□

We use this to show that interpolating sets are greedy solutions for the problem of maximizing the p-adic norm of the Vandermonde determinant. Since the determinant could vanish mod p^a, we define the norm of the Vandermonde determinant as follows

$$\text{Let} \quad |V(\alpha_0, \cdots, \alpha_{k-1})|_p \;=\; \prod_{0 \leq i < j \leq k-1} |(\alpha_i - \alpha_j)|_p \;=\; \prod_{j=1}^{k-1} |N_j^S(\alpha_j)|_p$$

This is equivalent to regarding the determinant as an integer and taking its norm.

Theorem 4.30 *The set $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$ is an interpolating set for I iff for all subsets $\{x_0, \cdots, x_{k-1}\}$ of I,*

$$|V(\alpha_0, \cdots, \alpha_{k-1})|_p \geq |V(x_0, \cdots, x_{k-1})|_p \quad (39)$$

PROOF:($\Rightarrow$). We will show a stronger statement: for $1 \leq j \leq k-1$, for any subset $\{x_0, \cdots, x_j\}$ of I,

$$|V(\alpha_0, \cdots, \alpha_j)|_p \geq |V(x_0, \cdots, x_j)|_p \quad (40)$$

Consider the polynomial $Q(X) = \prod_{i<j}(X - x_i)$. By Theorem 4.29, there exists $\alpha_i \in \{\alpha_0, \cdots, \alpha_j\}$ such that $|Q(\alpha_i)|_p \geq |Q(x_j)|_p$. Hence one can replace x_j by α_i without decreasing the norm of the Vandermonde determinant. Now repeat the same argument for

the set $\{x_0, \cdots, x_{j-1}, \alpha_i\}$ and the element x_{j-1} and so on. We get

$$|V(\alpha_0, \cdots, \alpha_j)|_p \;\geq\; \cdots \;\geq\; |V(x_0, \cdots, x_{j-1}, \alpha_i)|_p \;\geq\; |V(x_0, \cdots, x_j)|_p$$

($\Leftarrow$). Assume we have a set S satisfying Equation 39. Assume that the α_is are ordered canonically. We will show that $|N_j^S(\alpha_j)|_p \geq |N_j^S(x)|_p$ for all $x \in I$. This implies S is an interpolating set by Corollary 4.28.

Assume that there exists β such that $|N_j^S(\alpha_j)|_p < |N_j^S(\beta)|_p$. Pick $\alpha_i \in \{\alpha_j, \cdots, \alpha_{k-1}\}$ such that $|\beta - \alpha_i|_p$ is minimized. We will show that replacing α_i by β will increase the norm of the Vandermonde determinant. Observe that for any $\ell \neq i$ and $\ell \geq j$,

$$\begin{aligned}
& |\alpha_\ell - \alpha_i|_p \;\leq\; \max(|\beta - \alpha_\ell|_p, |\beta - \alpha_i|_p) \\
\text{But} \quad & |\alpha_\ell - \beta|_p \;\geq\; |\beta - \alpha_i|_p \qquad \text{by choice of } \alpha_i \\
\text{Hence} \quad & |\alpha_i - \alpha_\ell|_p \;\leq\; |\beta - \alpha_\ell|_p \\
\Rightarrow & \prod_{\ell > j, \ell \neq i} |\alpha_i - \alpha_\ell|_p \;\leq\; \prod_{\ell \geq j, \ell \neq i} |\beta - \alpha_\ell|_p \qquad (41)
\end{aligned}$$

We also have

$$|N_j^S(\alpha_i)|_p \;\leq\; |N_j^S(\alpha_j)|_p \;<\; |N_j^S(\beta)|_p$$

The first inequality is because S is ordered canonically, the second is by the definition of β. Hence from the definition of $N_j^S(X)$,

$$|\prod_{\ell < j} (\alpha_i - \alpha_\ell)|_p \;<\; |\prod_{\ell < j} (\beta - \alpha_\ell)|_p \qquad (42)$$

Combining Equations 41 and 42, we get

$$|\prod_{\ell \neq i} (\alpha_i - \alpha_\ell)|_p \;<\; |\prod_{\ell \neq i} (\beta - \alpha_\ell)|_p \qquad (43)$$

Hence replacing α_i with β increases the norm of the Vandermonde determinant, which contradicts the assumption that the norm is maximized at S. $\square$

Corollary 4.31 *The parameters $e_1, \cdots, e_{k-1}$ are independent of choice of interpolating set.*

PROOF: Note that

$$|N_j^S(\alpha_j)| \;=\; p^{-e_j}$$

and

$$|V(\alpha_0,\cdots,\alpha_j)|_p \;=\; \prod_{i\le j}|N_i^S(\alpha_i)|_p \;=\; p^{-\sum_{i\le j}e_i}$$

This quantity is maximized over subsets of I at every interpolating set. So $\sum_{i\le j}e_i$ and hence e_i is the same for every interpolating set of I. □

Finally we show that an interpolating set S for I is a union of interpolating sets for I restricted to each congruence class modulo p. This is interesting because a similar statement is not true for congruence classes mod p^2. Also, this generalizes the fact that the values of a polynomial at different points in $\mathbb{Z}_p$ are independent. The proof involves the construction of indicator polynomials for congruence classes mod p, using the modulus amplifying polynomials of Beigel and Tarui [17].

Lemma 4.32 *[17] Let*

$$M(X) \;=\; \sum_{i=a}^{2a-1}\binom{2a-1}{i}X^i(1-X)^{2a-1-i}$$

Then for any $x\in\mathbb{Z}$,

$$\begin{aligned} x\equiv 0 \bmod p \;&\Rightarrow\; M(x)\equiv 0 \bmod p^a \\ x\equiv 1 \bmod p \;&\Rightarrow\; M(x)\equiv 1 \bmod p^a \end{aligned} \tag{44}$$

PROOF: We have

$$\begin{aligned} M(X) &= X^a\sum_{i=a}^{2a-1}\binom{2a-1}{i}X^{i-a}(1-X)^{2a-1-i} \\ &\equiv 0 \bmod X^a \end{aligned}$$

Observe that one can also write

$$\begin{aligned} M(X) &= 1-\sum_{i=0}^{a-1}\binom{2a-1}{i}X^i(1-X)^{2a-1-i} \\ &= 1-(1-X)^a\sum_{i=0}^{a-1}\binom{2a-1}{i}X^i(1-X)^{a-1-i} \\ &\equiv 1 \bmod (1-X)^a \end{aligned}$$

Equation 44 follows from these observations. □

Lemma 4.33 *There exist polynomials $\Delta_\ell(X)$ over $Z_{p^a}[X]$ for $0 \leq \ell < p$ such that*

$$x \equiv \ell \bmod p \quad \Rightarrow \quad \Delta_\ell(x) \equiv 1 \bmod p^a$$

$$x \not\equiv \ell \bmod p \quad \Rightarrow \quad \Delta_\ell(x) \equiv 0 \bmod p^a$$

PROOF: When $a = 1$, we can take

$$\delta_0(X) = 1 - X^{p-1}$$

$$\delta_\ell(X) = \delta_0(X - \ell)$$

For $a > 1$, we construct $\Delta_\ell(X)$ by applying $M(X)$ to the above polynomials *i.e.*

$$\Delta_\ell(X) = M(\delta_\ell(X))$$

The correctness of this construction follows from Lemma 4.32. □

Given $0 \leq \ell \leq p-1$ let us define $I(\ell) = \{x \in I, x \equiv \ell \pmod p\}$. Define $S(\ell)$ similarly.

Theorem 4.34 *S is an interpolating set for I iff $S(\ell)$ is an interpolating set for $I(\ell)$ for $0 \leq \ell \leq p-1$.*

PROOF: Assume that each $S(\ell)$ is an interpolating set for $I(\ell)$. If $P(X)$ is zero over $\cup_\ell S(\ell) = S$, then it is zero over each $I(\ell)$, and hence over $\cup_\ell I(\ell) = I$. Hence S is an interpolating set for I.

In the other direction, let us assume that for some fixed ℓ, $S(\ell)$ is not an interpolating set for $I(\ell)$. Then there exists a polynomial $Q(X)$ which is zero at $S(\ell)$ but non-zero for some point $y \in I(\ell)$. We claim that the polynomial $Q(X)\Delta_\ell(X)$ is zero over S but not zero at y and hence S is not an interpolating set for I.

Since $Q(y) \not\equiv 0$ and $\Delta_\ell(y) \equiv 1$, it follows that $Q(y)\Delta_\ell(y) \not\equiv 0$. On the other hand, given $x \in S$,

$$x \not\equiv \ell \bmod p \;\Rightarrow\; \Delta_\ell(x) \equiv 0$$

$$x \equiv \ell \bmod p \;\Rightarrow\; Q(x) \equiv 0 \qquad \text{Since } x \in S(\ell)$$

Hence $Q(X)\Delta_\ell(X)$ is zero at every point in S. □

4.8 Some Combinatorial Properties of Ultrametric Spaces

We show that many of our results for interpolating sets can be translated into properties of general ultrametric spaces. Further, the proof of these properties for general ultrametric spaces follows directly from the proof for polynomials over $\mathbb{Z}_{p^a}$.

Definition 4.3 *Let T be a tree rooted at a vertex r, such that the distances of all leaves from the root r are equal. The metric space whose points are the leaves of the tree and distance is the shortest path in the tree is called an* equidistant tree *and denoted by (T, d).*

It is easy to show that (T, d) is an ultrametric . In fact, the converse is also true.

Fact 4.35 *Every finite ultrametric space embeds isometrically into an equidistant tree.*

Every equidistant tree can in turn be associated with $I \subseteq \mathbb{Z}_{p^a}$ for appropriate choices of p, a and I.

Lemma 4.36 *There is a mapping from any equidistant tree T to $I \subseteq \mathbb{Z}_{p^a}$ for some p, a such that*

$$|x - y|_p = p^{\frac{d(x,y)}{2} - a} \qquad \text{for } x \neq y$$

PROOF: There is a natural way to associate $\mathbb{Z}_{p^a}$ with an equidistant tree of degree p and depth a [11]. The root is at depth 0. The edges from each vertex to its descendants are labeled $\{0, \cdots, p-1\}$. Given a point $x = \sum_i x_i p^i \in \mathbb{Z}_{p^a}$, we associate it with a leaf of the tree as follows: Start from the root. At depth i, follow the edge labeled x_i. Thus the leaf nodes correspond to points in $\mathbb{Z}_{p^a}$, while nodes at depth d correspond to congruence classes modulo p^d. If $d(x, y)$ is the tree distance between the points, then $|x - y|_p = p^{\frac{d(x_i, x_j)}{2} - a}$.

Given an equidistant tree T, we take p to be a prime larger than the maximum degree of T, and a to be the depth of the tree. For any node, we arbitrarily label the edges to its descendants with $\{0, \cdots, p-1\}$. This can be done since there are at most p of them. This will map the leaves of T to $I \subseteq \mathbb{Z}_{p^a}$ and it is easy to verify that the distance satisfies the desired condition. □

Based on this correspondence, we can translate properties of interpolating sets to properties of ultrametric spaces. We consider the following NP-hard optimization problem.

Problem 4.6 MAX-DIST-K: *Given a metric space (X, d), pick a subset S of k points such that the sum of pairwise distances is maximized.*

For ultrametrics, the problem can be solved greedily.

Algorithm 4.8 Greedy Algorithm for Max-Dist-k

Input: An n point ultrametric space (T, d).

Output: A subset S of size k maximizing the sum of pairwise distances.

Pick $\alpha_0 \in T$ arbitrarily.

For $j \leq k-1$,

  Pick α_j so that $\sum_{i<j} d(\alpha_j, \alpha_i)$ is minimized.

Output $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$.

Lemma 4.37 *The greedy algorithm solves Max-Dist-k over Ultrametric Spaces.*

PROOF: Associate T with $I \subseteq \mathbb{Z}_{p^a}$. For any subset $(x_0, \cdots, x_{k-1})$ of size k,

$$\prod_{i<j} |x_i - x_j|_p = p^{\sum_{i,j} \frac{d(x_i,x_j)}{2} - \binom{k}{2} a}$$

Hence MAX-DIST-K on an ultrametric reduces to choosing k points in I such that $|V(x_0, \cdots, x_{k-1})|_p$ is maximized, by Theorem 4.30 this can be done by choosing the points greedily. □

Next we consider the problem of finding a point in a metric space that is farthest from a given set of points.

Problem 4.7 FARTHEST-POINT: *Given a metric space (X, d), and a set $\{y_1, \ldots, y_{k-1}\}$ of size $k-1$, find the point $x \in X$ that maximizes $\sum_{i<k} d(x, y_i)$.*

This problem is easy to solve for arbitrary metric spaces, we can just try every point in X and pick the best. However, ultrametric spaces admit a more efficient solution with some pre-processing. In the pre-processing step, we find a solution S to MAX-DIST-K using the greedy algorithm. This step is oblivious of the y_is. We then return the point $x \in S$ that

maximizes $\sum_{i<k} d(x, y_i)$. The running time of this step depends only on k, it is independent of the number of points n.

Lemma 4.38 *Let $y_1, \cdots, y_{k-1}$ be any subset of size $k-1$ in T. Let $S = \{\alpha_0, \cdots, \alpha_{k-1}\}$ be a greedy solution to Max-Dist-k. The quantity $\sum_{i<k} d(x, y_i)$ is maximized over T at a point $\alpha \in S$.*

PROOF: Associate T with $I \subseteq \mathbb{Z}_{p^a}$. Given points y_i, consider the polynomial

$$P(X) = \prod_{i<k} (X - y_i).$$

Note that

$$|P(x)|_p = p^{\sum \frac{d(x,y_i)}{2} - a(k-1)}$$

Hence, maximizing the distance is equivalent to maximizing $|P(x)|_p$. Since $P(X)$ is of degree $k-1$, by Theorem 4.29 its norm over I is maximized at some point in $\{\alpha_0, \cdots, \alpha_{k-1}\}$. □

The case $k = 2$ of this Lemma is a direct consequence of the ultrametric inequality. We are unaware of a direct combinatorial proof of Lemma 4.38 for $k \geq 3$ and higher.

Korte and Lovasz introduced to notion of a greedoid to explain the optimality of some greedy algorithms like Prim's algorithm which are not explained by matroids but where there is a natural ordering of elements in a set [23]. Lemma 4.38 allows us to show that ultrametric spaces have a greedoid structure.

Definition 4.4 *[23] A greedoid is a pair (T, G), where $G \subseteq 2^T$ is a set system satisfying*

1. *For every $S \in G$, there is an $\alpha \in S$ such that $S \setminus \{\alpha\}$ is in G.*
2. *For $R, S \in G$ with $|R| < |S|$ there is an $\alpha \in S$ such that $R \cup \{\alpha\} \in G$.*

Given an ultrametric space (T, d), we take G to be all possible sets output by the greedy algorithm for MAX-DIST-K for various values of k. We refer to sets in G as greedy sequences (since there is a natural ordering on them).

Lemma 4.39 *The pair (T, G) is a greedoid.*

PROOF: The first condition is easy, since given a greedy sequence $\{\alpha_0, \cdots, \alpha_k\}$ removing α_k leaves us with a greedy sequence.

For the second condition, let $S = \{\alpha_0, \cdots, \alpha_k\}$ while $R = \{\beta_0, \cdots, \beta_{k-1}\}$. To extend R, we need to add $x \in T$ that maximizes $\sum_{i<k} d(x, \beta_i)$. But applying Lemma 4.38 with $y_i = \beta_i$, we can always find a point $\alpha \in S$ that satisfies this condition. $\square$

CHAPTER V

RAMSEY GRAPHS FROM POLYNOMIAL REPRESENTATIONS

This chapter studies a problem at the intersection of combinatorics and computational complexity.

The combinatorial problem is that of explicitly constructing Ramsey graphs. Ramsey's theorem shows that every graph on 2^n vertices has either a clique or an independent set of size $n/2$. In his seminal 1947 paper introducing the probabilistic method, Erdős showed that there exist graphs with 2^n vertices where $\alpha(G), \omega(G) \leq (2 + o(1))n$ [29]. He posed the question of constructing such *Ramsey graphs* explicitly and offered a prize of \$100 for it. This is a central open problem in explicit combinatorial constructions; the best known constructions to date are far from the probabilistic bound. The first breakthrough on this problem was due to Frankl and Wilson in 1981 [31]; their construction gives $\alpha(G), \omega(G) \leq c^{\sqrt{n \log n}}$. For over two decades, there was no improvement on this bound despite much effort. However there were other constructions known due to Grolmusz and Alon [43, 5] that achieved exactly the same bound, and also extended to the problem of constructing multi-color Ramsey graphs, which is to t-color the edges of the complete graph so that there is not large monochromatic clique. At first sight, the construction of Grolmusz is quite different from that of Alon and Frankl-Wilson, yet it gives exactly the same bound. All three constructions use algebraic techniques, though in different ways. Very recently in 2006, the Frankl-Wilson bound was beaten by a new construction due to Barak, Rao, Shaltiel and Wigderson [13] which relies on machinery from extractors and pseudorandomness.

The complexity-theoretic problem which we have encountered in Chapter 3 is to show degree bounds for polynomial representations of Boolean functions modulo m. Theorem 3.2 due to Barrington, Beigel and Rudich [15] shows that over $\mathbb{Z}_6$, the OR function can be strongly represented by polynomials of degree n. We will refer to this construction as the

BBR polynomial. Proposition 3.4 shows that this bound is tight for symmetric polynomials. The question of whether better representations exist using asymmetric polynomials is still open. Tardos and Barrington proved a lower bound of $\Omega(\log n)$ [74]. A more general open question is whether asymmetric polynomials can give lower degree representations of symmetric functions than symmetric polynomials.

A surprising connection between these two problems was discovered by Grolmusz, who used the BBR polynomials to construct Ramsey graphs [43, 44]. As an intermediate step, he constructed a set system of size $n^{\omega(1)}$ on n elements where all set sizes are 0 mod 6 but all intersections are non-zero mod 6, settling an open problem in extremal set theory. He constructed Ramsey graphs from this set system and showed that lower degree OR representations mod 6 would give better Ramsey graphs.

5.1 Our Results

Our work generalizes and extends the connection between OR polynomials and Ramsey graphs. We propose a new definition of an OR representation: *a pair of polynomials represent the* OR *function on n variables if the union of their zero sets contains all points in $\{0,1\}^n$ except the origin.* We give a simple construction of a Ramsey graph from such representations. This viewpoint based on OR polynomials unifies the constructions of Frankl-Wilson, Alon and Grolmusz: they can all be derived from various OR representations of degree $O(\sqrt{n})$ based on symmetric polynomials. Thus the barrier to better Ramsey constructions through algebraic techniques appears to be the construction of lower degree representations. On one hand, since the best lower bound for any of these representations is only $\Omega(\log n)$ there is the possibility of better constructions. On the other hand, we show that further improvements cannot come from representations using symmetric polynomials; we prove an $\Omega(\sqrt{n})$ lower bound for such representations.

5.1.1 Ramsey Graphs from OR Representations:

We recall the definition of weak representations due to Barrington *et al.*[15].

Definition 5.1 *Polynomial $P(\mathbf{X}) \in \mathbb{Z}_m[\mathbf{X}]$ weakly represents the function f mod m if for*

$\mathbf{x}, \mathbf{y} \in \{0,1\}^n$, *if* $f(\mathbf{x}) \neq f(\mathbf{y})$ *then* $P(\mathbf{x}) \not\equiv P(\mathbf{y}) \bmod m$.

We note that for the OR function strong and weak representations are equivalent, since if $P(\mathbf{X})$ is a weak representation of the OR function, then $P(\mathbf{X}) - P(0,\ldots,0)$ strongly represents the OR function mod m.

We propose the following definition of an OR representation.

Definition 5.2 *Polynomials* $P(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ *and* $Q(\mathbf{X}) \in \mathbb{Z}_q[\mathbf{X}]$ *represent the* OR *function on* n *variables if*

$$P(0,\ldots,0) \equiv 1 \bmod p \quad \textit{and} \quad Q(0,\ldots,0) \equiv 1 \bmod q$$

and for $\mathbf{x} \in \{0,1\}^n \setminus (0,\ldots,0)$

$$P(\mathbf{x}) \equiv 0 \bmod p \quad \textit{or} \quad Q(\mathbf{x}) \equiv 0 \bmod q$$

where p, q *are primes. The degree of the representation is* $d = \max(\deg(P), \deg(Q))$.

One can combine the two polynomials using the Chinese Remainder Theorem (CRT) to get a single polynomial that weakly represents OR mod pq. However the specific choice of values output by the weak representation is important for our application. The construction of Barrington *et al.*gives a degree $O(\sqrt{n})$ OR representation using polynomials over $\mathbb{Z}_2$ and $\mathbb{Z}_3$. A simple representation of degree $O(\sqrt{n})$ with $n = pq - 1$ using polynomials over $\mathbb{Z}_p$ and $\mathbb{Z}_q$ can be derived from Alon's construction. This highlights another difference about our definition and weak representations: for Ramsey constructions, we are not restricted to any fixed moduli p and q, we are free to choose them in any way, (possibly as functions of n) so that the degree is minimized as a function of n.

We give a simple Ramsey construction based on OR representations: the vertex set is $\{0,1\}^n$ and we add edge $(\mathbf{x}, \mathbf{y})$ to G if $\mathbf{x} \oplus \mathbf{y}$ is in the zero set of $P(\mathbf{X})$, where $\mathbf{x} \oplus \mathbf{y}$ denotes the symmetric difference of $\mathbf{x}$ and $\mathbf{y}$. In order to bound $\alpha(G)$ and $\omega(G)$, we use the notion of representations of graphs over spaces of polynomials introduced by Alon [5]. The idea is to assign polynomials to the vertices of G so that the polynomials assigned to vertices in a clique are linearly independent.

Definition 5.3 *Let $G(V, E)$ be a graph and $\mathcal{F}$ be a set polynomials in n variables over field $\mathbb{F}$. A polynomial representation of G over $\mathbb{F}$ is an assignment of a polynomial $P_v(\mathbf{X}) \in \mathcal{F}$ and a point $\mathbf{x}_v \in \mathbb{F}^n$ to $v \in V$ where:*

1) *For each $v \in V$, $P_v(\mathbf{x}_v) \neq 0$.*

2) *If $(u, v) \in E$ then $P_v(\mathbf{x}_u) = 0$.*

It is easy to see that $\omega(G) \leq dim(\mathcal{F})$ which is the dimension of the $\mathbb{F}$ vector space spanned by polynomials in $\mathcal{F}$. We use the polynomial $P(\mathbf{X})$ to construct a representation of G over $\mathbb{Z}_p$ and $Q(\mathbf{X})$ to construct a representation of $\overline{G}$ over $\mathbb{Z}_q$. The Frankl-Wilson construction can also be viewed in this framework, where we represent G over $\mathbb{Z}_p$ and $\overline{G}$ over $\mathbb{Q}$. However, quoting Alon *'It seems that this construction does not extend to the case of more than 2 colors'* [5]. We propose a definition of OR representation which leads to such an extension.

Definition 5.4 *Polynomials $P(\mathbf{X}) \in \mathbb{Z}_{p^a}[\mathbf{X}]$ and $Q(\mathbf{X}) \in \mathbb{Z}_{p^b}[\mathbf{X}]$ represent the* OR *function on n variables if*

$$P(0, \ldots, 0) \not\equiv 0 \bmod p^a \quad \textit{and} \quad Q(0, \ldots, 0) \not\equiv 0 \bmod p^b$$

and for $\mathbf{x} \in \{0,1\}^n \setminus (0, \ldots, 0)$

$$P(\mathbf{x}) \equiv 0 \bmod p^a \quad \textit{or} \quad Q(\mathbf{x}) \equiv 0 \bmod p^b$$

where p is prime and $a, b \geq 1$. The degree of the representation is $d = \max(\deg(P), \deg(Q))$.

To differentiate the representations of Definitions 5.2 and 5.4, we refer to them as prime representations and prime-power representations respectively. The Frankl-Wilson construction can be used to show that for $n = p^2 - 1$, there exist OR representations of degree $O(\sqrt{n})$. The interesting feature of this representation is that it does not use the Chinese Remainder Theorem (CRT). The construction of Ramsey graphs from prime-power representations stays the same; the difference is in the analysis. For this, we introduce polynomial representations of G over $\mathbb{Z}_{p^a}$.

Definition 5.5 *Let $G(V,E)$ be a graph and $\mathcal{F}$ a set of polynomials in n variables over $\mathbb{Z}$. A polynomial representation of G over $\mathbb{Z}_{p^a}$ is an assignment of a polynomial $P_v(\mathbf{X}) \in \mathcal{F}$ and a point $\mathbf{x}_v \in \mathbb{Z}^n$ to $v \in V$ s.t.:*

1) *For each $v \in V$, $P_v(\mathbf{x}_v) \not\equiv 0 \bmod p^a$.*

2) *If $(u,v) \in E$ then $P_v(\mathbf{x}_u) \equiv 0 \bmod p^a$.*

We show that the polynomials assigned to a clique are linearly independent over $\mathbb{Q}$ so $\omega(G)$ is bounded by the dimension of the $\mathbb{Q}$-vector space spanned by $\mathcal{F}$. Like polynomial representations of graphs over $\mathbb{Q}$, representations over $\mathbb{Z}_{p^a}$ assign linearly independent polynomials over $\mathbb{Q}$ to vertices in a clique. A crucial difference is that *representations over $\mathbb{Q}$ tensor, those over $\mathbb{Z}_{p^a}$ do not.* This means that if we have sets of polynomials $\mathcal{F}_1$ and $\mathcal{F}_2$ that represent G_1 and G_2 over $\mathbb{Q}$, then $\mathcal{F}_1 \otimes \mathcal{F}_2$ represents $G_1 \cdot G_2$ over $\mathbb{Q}$ for an appropriate definition of graph product (see [5] for definitions and proofs). This is important for the original application of these representations, which was to bound the Shannon capacity of the graph. However, this property implies that one cannot get low dimensional representations of both G and $\overline{G}$ over $\mathbb{Q}$, since $G \cdot \overline{G}$ always has a large clique. But since $\mathbb{Z}_{p^a}$ has zero divisors, we lose this tensor product property, so we can simultaneously get low dimensional representations of G and $\overline{G}$ over $\mathbb{Z}_{p^a}$.

We could restate this argument from the viewpoint of OR representations. We cannot get low degree prime representations by taking $p = q$ since then $P(\mathbf{X})Q(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ is 0 at every point in $\{0,1\}^n$ except the origin, and such a polynomial requires degree n. But this argument does not extend to prime-power representations because of zero-divisors.

All the OR representations above achieve a bound of $O(\sqrt{n})$ using symmetric polynomials. Plugging them into the simple construction above gives $\alpha(G), \omega(G) \leq c_1^{\sqrt{n}\log n}$ as opposed to the best bound of $c^{\sqrt{n\log n}}$. However, by massaging the polynomials and working with set intersections as opposed to distances, we can get exactly the constructions of Frankl-Wilson, Grolmusz and Alon. Here are some advantages of our unified view of these constructions:

- It places the constructions of Alon and Frankl-Wilson in the context of OR polynomials, and raises the possibility of getting better constructions from low degree representations. The notions of prime-power representations of graphs and Boolean functions arising from the Frankl-Wilson construction are of independent interest.

- It relates the construction of Grolmusz to those of Frankl-Wilson and Alon, which look very different at first. Our Ramsey graph construction from the OR polynomial of Barrington *et al.*is simple and direct. In fact it takes some work to show that we get the same graph as Grolmusz. Viewing this construction in terms of set intersections, we derive improved bounds for set systems with restricted intersections modulo prime powers.

- In this view, all the constructions naturally extend to multicolor Ramsey graphs. To construct t-color Ramsey graphs, we define OR representations involving t polynomials over $\mathbb{Z}_{q_1}, \ldots, \mathbb{Z}_{q_t}$ where $q_1 \ldots, q_t$ are prime powers. Taking powers of the same prime p extends the Frankl-Wilson construction.

5.1.2 Limitations to Symmetric Constructions

A natural question is to show tight degree bounds for OR representations. A better upper bound would lead to better Ramsey graphs. Lower bounds are interesting from the complexity-theory viewpoint of understanding polynomial representations over composites. For the OR function, we believe Definition 5.2 is the right one to use, since it seems to eliminate dependence of the degree on the modulus pq. Also it places the problem in the context of understanding the zero-sets of low degree polynomials over $\mathbb{Z}_p$. This question has been studied in various other contexts including low degree testing, zero-testing and derandomization. Prime-power representations are interesting since they do not rely on the CRT. Interestingly, the $\Omega(\log n)$ lower bound [74] also does not use the CRT, so it applies to prime-power representations too. It is possible that proving bounds for prime-power representations is easier than the prime case.

Degree lower bounds extend a line of work in combinatorics aimed at understanding why explicit Ramsey graphs are hard to construct, by showing limitations to various natural

techniques. A conjecture of Babai states that one cannot construct good Ramsey graphs based on the sign of a set of real polynomials. There has been considerable progress towards proving this conjecture by Alon *et al.*[3, 7]. Degree lower bounds are weaker since they say the known technique for bounding $\alpha(G)$ and $\omega(G)$ does not yield good bounds, as opposed to showing either $\alpha(G)$ or $\omega(G)$ is large. But on the other hand, there are no good Ramsey constructions using signs of real polynomials, while OR representations are the best technique known for this problem. Further, the Ramsey graph constructions based on symmetric polynomials result in graphs where the vertex set is $\{0,1\}^n$ and where edges are added between vertices based on the Hamming distance between them. Such graphs possess a high degree of symmetry which is unlikely in a random graph. Our degree lower bound suggests that perhaps such Ramsey graph constructions cannot give better parameters (see Chapter 6).

We show a degree $\Omega(\sqrt{n})$ lower bound for OR representations by symmetric polynomials. Thus better representations if they exist must use asymmetric polynomials. Proposition [15] gives a lower bound of $\Omega(\sqrt{n})$ for symmetric polynomials that weakly represent OR mod 6 [15]. One might guess that similar arguments should work even for our new definition of OR representations, but this is incorrect. In fact those arguments will not suffice even for prime representations. The precise bound they prove, and which holds for all weak representations is $\deg(P) \cdot \deg(Q) \geq n/(pq)$. This is good enough when p, q are small, but if $n < pq$ as in Alon's construction, this gives a bound of 1. One cannot hope for a stronger result since the polynomial $\sum_i X_i$ of degree 1 weakly represents OR on $n < pq$ variables over $\mathbb{Z}_{pq}$. Our definition restricts the values output by the weak representation, making it possible to show bounds independent of the modulus m. But exploiting this difference calls for new techniques, beyond the periodicity based arguments used for weak representations that were used in Chapter 3.

5.1.3 Our Lower-Bound Techniques

While lower bounds for the prime and prime-power cases are very different, they have similar high-level structure: an *algebraic part* where we show that if the zero-set of the polynomial

has certain structure, then the polynomial must have high degree, and a *combinatorial part* where we argue that there is no good partition of hypercube, that any partition results in one of the polynomials having high degree.

For the prime-power case, we translate the problem to one about univariate polynomials modulo $\mathbb{Z}_{p^a}$. However over $\mathbb{Z}_{p^a}$ it is no longer true that a degree d polynomial can have only d roots (take X^a for instance); so we need new tools for degree lower bounds. Building on Algorithm 4.2 for interpolation over $\mathbb{Z}_{p^a}$ from Chapter 4, we define a *greedy sequence*, which roughly is a sequence that is distributed uniformly among various congruence classes modulo powers of p. We show that the longest greedy sequence in the zero-set gives a lower bound on the degree of a polynomial. Then a combinatorial argument shows that in any partition of integers $\{1, \ldots, n\}$ into A and B, one of them contains a long greedy sequence.

For the prime case, we view a symmetric polynomial P acting on a 0-1 vector $\mathbf{x}$ as a polynomial $\bar{P}$ acting on in the digits of the base p expansion of the weight $wt(\mathbf{x})$ following Bhatnagar *et al.*[20]. There it was shown that $\bar{P}$ can be used to bound $\deg(P)$ within a factor of p; we introduce a notion of weighted degree of $\bar{P}$ that exactly captures the degree of P. The combinatorial part of the proof uses a number theoretic lemma which seems of independent interest. It says that if p, q are primes, $n < pq$ and $A \subseteq \mathbb{Z}_p^*$ and $B \subseteq \mathbb{Z}_q^*$ are subsets so that every number in $[1, \ldots, n]$ lies in A mod p or in B mod q, then one of A or B has to be *large*.

5.1.4 Organization of this Chapter

We analyze our simple Ramsey graph construction based on OR polynomials in Section 5.2. In Section 5.3, we give a construction based on set intersections, and show how to derive the constructions of Frankl-Wilson, Grolmusz and Alon from it. We prove lwoer bounds for prime-power representations in Section 5.4 and for prime representations in Section 5.5.

5.1.5 Preliminaries

Let $\mathbf{0} = (0, \ldots, 0)$. Given $\mathbf{x} \in \{0,1\}^n$ let $wt(\mathbf{x})$ denote its Hamming weight. Given $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$, $\mathbf{x} \oplus \mathbf{y}$ denotes symmetric difference, $\mathbf{x} \cap \mathbf{y}$ denotes the bitwise AND and $d(\mathbf{x}, \mathbf{y})$

denotes Hamming distance. For $i \leq d$, let

$$\binom{n}{\leq d} \stackrel{\triangle}{=} \sum \binom{n}{i}.$$

For $x \in \mathbb{Z}$, let the valuation of x denoted $\mathrm{val}_p[x]$ be the highest power of p that divides x. Let $\mathrm{val}_p[0] = \infty$. Th valuations have the following properties:

1. $\mathrm{val}_p[x+y] \geq \min(\mathrm{val}_p[x], \mathrm{val}_p[y])$. (Ultrametric inequality)

2. $\mathrm{val}_p[xy] = \mathrm{val}_p[x] + \mathrm{val}_p[y]$.

We use the notation $\mathrm{val}_p[\ ]$ to distinguish valuations for integers from valuations for $\mathbb{Z}_{p^a}$ which we encountered in Chapter 4 which we denoted by $\mathrm{val}_p(\ )$. Note that for those valuations, Property 2 was replaced by the inequality $\mathrm{val}_p(xy) \geq \mathrm{val}_p(x) + \mathrm{val}_p(y)$.

We will also consider the problem of constructing t-color Ramsey graphs which is defined as follows:

Problem 5.1 MULTI-COLOR RAMSEY GRAPHS: *Give a t-coloring of the edges of the complete graph so that the size of the largest monochromatic clique is minimized.*

We say a Ramsey graph $G(V,E)$ is explicit if there is a deterministic $\mathrm{poly}(|V|)$ time algorithm to compute the adjacency matrix and very explicit if there is a deterministic $\mathrm{poly}(\log|V|)$ algorithm that computes the adjacency relation. We briefly describe the known explicit constructions of Ramsey graphs in chronological order.

– **Frankl-Wilson** [31]: Take p prime and $m = p^3$. The vertex set consists of all subsets of $[m]$ of size $p^2 - 1$. Two vertices S and T are adjacent if $|S \cap T| \not\equiv -1 \bmod p$. One can bound the size of $\alpha(G)$ and $\omega(G)$ using well-known results from extremal set theory [31, 10].

– **Grolmusz** [43, 44] : The main step is to construct a set system $\mathcal{F}$ on $[n]$ of size $n^{\omega(1)}$ so that $|S| \equiv 0 \bmod 6$ but $|S \cap T| \not\equiv 0 \bmod 6$. The vertices of the graph G are sets of $\mathcal{F}$ and S, T are adjacent if $|S \cap T|$ is odd. One can bound $\alpha(G)$ and $\omega(G)$ using results from extremal set theory.

– **Alon** [5]: Take $p < q$ to be nearly equal primes and $m = p^3$. The vertex set consists of all subsets of $[m]$ of size $pq - 1$. Two vertices S and T are adjacent if $|S \cap T| \not\equiv -1 \bmod p$. To bound $\alpha(G)$ and $\omega(G)$, we construct representations of G over $\mathbb{Z}_p$ and $\overline{G}$ over $\mathbb{Z}_q$.

– **Barak** [12]: Barak gives a product based construction (discovered independently by Pudlak and Rodl) where we first explicitly search for a good Ramsey graph in a small sample space and then use the Abbot product to get a larger graph. This gives $|V| = 2^n$ and $\alpha(G), \omega(G) \leq 2^{\epsilon\sqrt{n}\log n}$ for any $\epsilon > 0$. A similar product based construction, but with worse parameters is given by Naor [62].

– **Barak-Rao-Shaltiel-Wigderson** [13]: In a recent breakthrough, Barak *et al.*give a construction that achieves $\alpha(G), \omega(G) \leq 2^{n^{o(1)}}$. In fact they solve a more general problem, which is to construct bipartite Ramsey graphs. Their construction is rather intricate and makes significant use of machinery developed for extracting randomness from weak random sources.

Except for Barak's product-based construction, all the constructions mentioned above are highly explicit.

5.2 *Constructing Ramsey graphs using* OR *Polynomials*

In this section, we prove the correctness of the construction described in the introduction. While the graphs obtained are not quite optimal, the construction is simple and best explains the close connections between OR representations and Ramsey graphs.

If graph G has a representation over a field $\mathbb{F}$ as in Definition 5.3, it is easy to show that $\omega(G) \leq dim(\mathcal{F})$ where $dim(\mathcal{F})$ is the dimension of the $\mathbb{F}$-vector space spanned by $\mathcal{F}$ [5]. For representations over $\mathbb{Z}_{p^a}$, we show that $\omega(G) \leq dim(\mathcal{F})$ where $dim(\mathcal{F})$ is the dimension of the $\mathbb{Q}$-vector space spanned by $\mathcal{F}$. The proof is by a valuation based argument similar to one used by Babai *et al.*[11].

Lemma 5.1 *If $G(V, E)$ has a polynomial representation over $\mathbb{Z}_{p^a}$, then $\omega(G) \leq dim(\mathcal{F})$.*

PROOF: Let $K \subseteq V$ be a clique. We claim that the polynomials $P_v(\mathbf{X})$ for $v \in K$ are linearly independent over $\mathbb{Q}$. Assume for contradiction that

$$\sum_{v \in K} \lambda_v P_v(\mathbf{X}) \;=\; 0$$

By clearing denominators, w.m.a that $\lambda_v \in \mathbb{Z}$, and by removing common factors w.m.a that

p does not divide λ_u for some $u \in K$. Rearranging terms, we have

$$\lambda_u P_u(\mathbf{X}) = -\sum_{v \in K, v \neq u} \lambda_v P_v(\mathbf{X})$$

Substituting $\mathbf{X} = \mathbf{x}_u$,

$$\lambda_u P_u(\mathbf{x}_u) = -\sum_{v \in K, v \neq u} \lambda_v P_v(\mathbf{x}_u)$$

Since $P_u(\mathbf{x_u}) \not\equiv 0 \bmod p^a$ and $\mathrm{val}_p[\lambda_u] = 0$, we have $\mathrm{val}_p[\lambda_u P_u(\mathbf{x}_u)] \leq a - 1$. But $v \in K$ and $v \neq u$, then (v, u) is an edge, hence $P_v(\mathbf{x}_u) \equiv 0 \bmod p^a$. So the RHS is divisible by p^a, which is a contradiction. □

Construction 5.1 `Graph` $G(V, E)$ `from` OR `polynomials.`

- `Let` $V(G) = \{0,1\}^n$.
- `If` $P(\mathbf{x} \oplus \mathbf{y}) \equiv 0$, `add an edge` $(\mathbf{x}, \mathbf{y})$.

Theorem 5.2 *Given a degree d* OR *representation, graph G has 2^n vertices and $\alpha(G), \omega(G) \leq \binom{n}{\leq d}$.*

PROOF: Assume that we have a prime representation. We give a polynomial representation of G over $\mathbb{Z}_p$. For each vertex $\mathbf{v} \in \{0,1\}^n$, let

$$Y_i = \begin{cases} 1 - X_i & \text{if } \mathbf{v}_i = 1 \\ X_i & \text{if } \mathbf{v}_i = 0 \end{cases}$$

Define $P_{\mathbf{v}}(X_1, \ldots, X_n)$ to be the polynomial obtained by multi-linearizing $P(Y_1, \ldots, Y_n)$ (i.e setting $X_i^d = X_i$). Note that for $\mathbf{u} \in \{0,1\}^n$, $P_{\mathbf{v}}(\mathbf{u}) = P(\mathbf{v} \oplus \mathbf{u})$. Hence

$$P_{\mathbf{v}}(\mathbf{v}) = P(\mathbf{0}) \not\equiv 0 \bmod p.$$

On the other hand, from our construction, if $(\mathbf{u}, \mathbf{v}) \in E$ then $P(\mathbf{v} \oplus \mathbf{u}) \not\equiv 0 \bmod p$. Hence

$$P_{\mathbf{v}}(\mathbf{u}) = P(\mathbf{v} \oplus \mathbf{u}) \not\equiv 0 \bmod p.$$

Thus we get a polynomial representation of G over $\mathbb{Z}_p$. Since the $P_{\mathbf{v}}(\mathbf{X})$s are all multilinear polynomials of degree at most d in n variables, they lie in a vector space of dimension $\binom{n}{\leq d}$. This shows that $\omega(G) \leq \binom{n}{\leq d}$. Similarly, if $(\mathbf{u}, \mathbf{v})$ is not an edge then

$P(\mathbf{v} \oplus \mathbf{u}) \not\equiv 0 \bmod p$, hence $Q(\mathbf{v} \oplus \mathbf{u}) \equiv 0 \bmod q$. Using this we construct a representation of $\overline{G}$ over $\mathbb{Z}_q$ and bound $\alpha(G)$.

For prime-power representations of OR we can represent G and $\overline{G}$ over $\mathbb{Z}_{p^a}$ by the same argument. □

One can construct explicit Ramsey graphs by plugging in various OR representations described below; all of which give $d = O(\sqrt{n})$ using symmetric polynomials. This gives a bound of $c_1^{\sqrt{n}\log n}$ for some constant c_1 on the clique size. In fact the constructions below are very explicit, since given vertices $\mathbf{x}, \mathbf{y} \in \{0,1\}^n$, the color of the edge $(\mathbf{x}, \mathbf{y})$ can be computed in time $O(n)$.

1) **Alon** [5]: Let $p < q$ be primes and let $n = pq - 1$. Define $P(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ and $Q(\mathbf{X}) \in \mathbb{Z}_q[\mathbf{X}]$ as

$$\begin{aligned} P(\mathbf{X}) &= 1 - \left(\sum X_i\right)^{p-1} \\ Q(\mathbf{X}) &= 1 - \left(\sum X_i\right)^{q-1} \end{aligned} \tag{45}$$

For $\mathbf{x} \neq \mathbf{0}$, since $1 \leq \sum_i w(\mathbf{x}) \leq pq - 1$, by the CRT $w(\mathbf{x}) \not\equiv 0 \bmod p$ or $w(\mathbf{x}) \not\equiv 0 \bmod q$. By Fermat's Theorem, in the former case $P(\mathbf{x}) \equiv 0 \bmod p$, in the latter $Q(\mathbf{x}) \equiv 0 \bmod q$. Taking p, q nearly equal gives degree $d = (1 + o(1))\sqrt{n}$.

2) **BBR** [15]: Let $n = 2^k 3^\ell - 1$. Define $P(\mathbf{X}) \in \mathbb{Z}_2[\mathbf{X}]$ and $Q(\mathbf{X}) \in \mathbb{Z}_3[\mathbf{X}]$ as

$$\begin{aligned} P(\mathbf{X}) &= \binom{\sum_i X_i + 2^k - 1}{2^k - 1}, \\ Q(\mathbf{X}) &= \binom{\sum_i X_i + 3^\ell - 1}{3^\ell - 1} \end{aligned} \tag{46}$$

Since $\binom{\sum_i x_i}{k} = S_k(\mathbf{x})$ for $\mathbf{x} \in \{0,1\}^n$, $P(\mathbf{X})$ and $Q(\mathbf{X})$ in fact have coefficients from $\mathbb{Z}_2$ and $\mathbb{Z}_3$. For $x \neq \mathbf{0}$, $1 \leq \sum_i w(\mathbf{x}) \leq 2^k 3^\ell - 1$. Lucas' theorem implies that if $w(\mathbf{x}) \not\equiv 0 \bmod 2^k$ then $P(\mathbf{x}) \equiv 0 \bmod 2$, and if $w(\mathbf{x}) \not\equiv 0 \bmod 3^\ell$ then $Q(\mathbf{x}) \equiv 0 \bmod 3$. We can choose k, ℓ s.t. $d = (1 + \varepsilon)\sqrt{n}$ for any $\varepsilon > 0$ [57].

Both representations above are prime representations, we now construct prime power representations. For ease of exposition, we restate Definition 5.4 of prime-power representations in terms of rational polynomials; we omit the simple proof of equivalence.

Definition 5.6 *Polynomials* $P(\mathbf{X}), Q(\mathbf{X}) \in \mathbb{Q}[\mathbf{X}]$ *represent the* OR *function on* $\{0,1\}^n$ *if*

$$P(0,\ldots,0) \equiv 1 \bmod p \quad \textit{and} \quad Q(0,\ldots,0) \equiv 1 \bmod p$$

and for $\mathbf{x} \in \{0,1\}^n \setminus (0,\ldots,0)$

$$P(\mathbf{x}) \equiv 0 \bmod p \quad \textit{or} \quad Q(\mathbf{x}) \equiv 0 \bmod p$$

for a prime p*. The degree of the representation is* $d = \max(\deg(P), \deg(Q))$*.*

Note that in general $P(\mathbf{x})$ could be rational. When we say $P(\mathbf{x}) \equiv 0/1 \bmod p$, we mean $P(\mathbf{x})$ is an integer satisfying the condition. However, if $\mathbf{x} \neq \mathbf{0}$ and $Q(\mathbf{x}) \equiv 0 \bmod p$, then $P(\mathbf{x})$ need not be an integer and vice versa.

3) **Frankl-Wilson** [31]: Take p prime and $n = p^2 - 1$. Define $P(\mathbf{X}), Q(\mathbf{X}) \in \mathbb{Q}[\mathbf{X}]$ as

$$\begin{aligned} P(\mathbf{X}) &= \prod_{j=1}^{p-1} \left(\sum_i X_i - j \right) \\ Q(\mathbf{X}) &= \prod_{j=1}^{p-1} \left(\frac{\sum_i X_i}{p} - j \right) \end{aligned} \tag{47}$$

For a non-zero vector $\mathbf{x} \in \{0,1\}^n$ we have $1 \leq w(\mathbf{x}) \leq p^2 - 1$. If $w(\mathbf{x}) \not\equiv \mathbf{0} \bmod \mathbf{p}$ then $P(\mathbf{x}) \equiv 0 \bmod p$. If $w(\mathbf{x}) \equiv 0 \bmod p$, then $1 \leq \frac{w(\mathbf{x})}{p} \leq p - 1$ hence $Q(\mathbf{x}) \equiv 0 \bmod p$. The degree is $d = p - 1 < \sqrt{n}$.

4) We construct representations with the prime fixed and n varying, analogous to [15]. Let $n = 2^{2k} - 1$. Define $P(\mathbf{X}), Q(\mathbf{X}) \in \mathbb{Q}[\mathbf{X}]$ as

$$\begin{aligned} P(\mathbf{X}) &= \binom{\sum_i X_i + 2^k - 1}{2^k - 1} \\ Q(\mathbf{X}) &= \binom{\frac{\sum_i X_i}{2^k} + 2^k - 1}{2^k - 1} \end{aligned} \tag{48}$$

The proof of correctness is through Lucas' theorem. If $w(\mathbf{x}) \not\equiv 0 \bmod 2^k$ then $P(\mathbf{x}) \equiv 0$. If $\mathbf{x} \neq 0$ but $w(\mathbf{x}) \equiv 0 \bmod 2^k$ then $Q(\mathbf{x}) \equiv 0$.

Plugging any of these polynomials into construction 5.1 gives the following type of graph : Add $(\mathbf{x}, \mathbf{y})$ to E if $d(\mathbf{x}, \mathbf{y}) \not\equiv 0 \bmod \ell$ where ℓ is either a prime or a prime power close to $\sqrt{n}$.

For the problem of constructing t-color Ramsey graphs, we define OR representations with t polynomials $P_1(\mathbf{X}), \ldots, P_t(\mathbf{X})$ such that the union of their zero sets if $\{0,1\}^n \setminus \{\mathbf{0}\}$. We can extend constructions in Equations 45, 46 by taking t distinct primes. To extend the construction of Equation 47, let $n = p^t - 1$. For $1 \leq \ell \leq t$ define

$$P_\ell(\mathbf{X}) \;=\; \prod_{j=1}^{p-1} \left(\frac{\sum_i X_i}{p^\ell} - i \right) \tag{49}$$

We can similarly extend Equation 48, we omit the details.

5.3 Ramsey Graphs based on Set Intersections

The constructions of Frankl-Wilson, Alon and Grolmusz use a coloring scheme based on the size of set intersections. In this section we show that these can be constructed from certain polynomials that are closely related to OR polynomials. These polynomials are also used by Grolmusz [44] and Kutin [57] to give simple constructions of set systems with restricted intersections mod 6.

Definition 5.7 *The weight-n function W_n is a partial function defined on $\{0,1\}^m$ for $m \geq n$ as follows*

$$W_n(\mathbf{x}) = 0 \;\; \textit{if} \;\; w(\mathbf{x}) = n \qquad W_n(\mathbf{x}) = 1 \;\; \textit{if} \;\; w(\mathbf{x}) < n$$

The function is undefined for $w(\mathbf{x}) \in [n+1, \ldots, m]$.

Note that W_n on n variables is simply the NAND function. We now define polynomial representations of W_m. We give an extension of Definition 5.2, a similar extension holds for Definition 5.4

Definition 5.8 *Polynomials $P(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ and $Q(\mathbf{X}) \in \mathbb{Z}_q[\mathbf{X}]$ represent the W_n function on m variables if*

$$\begin{array}{llll} P(\mathbf{x}) \not\equiv 0 \bmod p & \textit{and} & Q(\mathbf{x}) \not\equiv 0 \bmod q & \textit{if } w(\mathbf{x}) = n \\ P(\mathbf{x}) \equiv 0 \bmod p & \textit{or} & Q(\mathbf{x}) \equiv 0 \bmod q & \textit{if } w(\mathbf{x}) < n \end{array}$$

The degree of the representation is $d = \max(\deg(P), \deg(Q))$.

Assume that $P(X_1, \ldots, X_m)$ and $Q(X_1, \ldots, X_m)$ represent W_n with degree d for some $m \geq n$. Define $\hat{P}(X_1, \ldots, X_n)$ and $\hat{Q}(X_1, \ldots, X_n)$ to be polynomial by substituting $1 - X_i$ for X_i when $i \leq n$ and setting $X_i = 0$ for $i \geq n$. It is easy to verify that $\hat{P}$ and $\hat{Q}$ represent OR on n variables with degree at most d. Further if P and Q were symmetric polynomials, then so are $\hat{P}$ and $\hat{Q}$. Thus lower bounds for OR representations imply lower bounds for W_n representations. In particular our lower bounds for OR representations rule out representations of W_n with symmetric polynomials of degree $o(\sqrt{n})$.

Conversely one can construct degree d representations of W_n from degree d symmetric polynomials representing OR on n variables. We do not know if a similar statement is true for asymmetric polynomials.

Lemma 5.3 *A degree d representation of* OR *on n variables using symmetric polynomials gives a degree d representation of W_n on n variables for all $m \geq n$,*

PROOF: Let $P(\mathbf{X}), Q(\mathbf{X})$ be symmetric polynomials of degree at most d on m variables that represent OR. Replace each X_i by $1 - X_i$ and multi-linearize. It is easy to show that the resulting polynomials $P'(\mathbf{X}), Q'(\mathbf{X})$ represent W_n on n variables. Further, they are symmetric multilinear polynomials of degree d, hence we can write them as

$$\begin{aligned} P'(X_1, \ldots, X_n) &= \sum_{i \leq d} a_i S_i(X_1, \ldots, X_n) \\ Q'(X_1, \ldots, X_n) &= \sum_{i \leq d} b_i S_i(X_1, \ldots, X_n) \end{aligned}$$

We obtain new polynomials P'' and Q'' by replacing $S_i(X_1, \ldots, X_n)$ by $S_i(X_1, \ldots, X_m)$.

$$\begin{aligned} P''(X_1, \ldots, X_m) &= \sum_{i \leq d} a_i S_i(X_1, \ldots, X_m) \\ Q''(X_1, \ldots, X_m) &= \sum_{i \leq d} b_i S_i(X_1, \ldots, X_m) \end{aligned}$$

Since the value of a symmetric function on a $0,1$-vector depends only on the weight of the vector, one can show that P'' and Q'' represent W_n on m variables. □

We can use the $O(\sqrt{n})$ OR representations to construct representations of W_n. We give a construction of explicit Ramsey graphs from representations of W_n.

Construction 5.2 Ramsey Graph $G(V,E)$ from representations of W_n.

- $V(G)$ consists of vectors $\mathbf{x} \in \{0,1\}^m$ of weight n.
- If $P(\mathbf{x} \cap \mathbf{y}) \equiv 0$, add $(\mathbf{x},\mathbf{y})$ to $E(G)$.

Theorem 5.4 *Given a degree d representation of the weight-n function on $\{0,1\}^m$, the graph G has $\binom{m}{n}$ vertices and $\alpha(G), \omega(G) \leq \binom{m}{\leq d}$.*

PROOF: Assume we have prime representation of W_n. We associate a polynomial $P_{\mathbf{v}}(\mathbf{X})$ with each vertex $\mathbf{v}$ so that $P_{\mathbf{v}}(\mathbf{u}) = P(\mathbf{v} \cap \mathbf{u})$. Given $\mathbf{v} = (v_1, \ldots, v_m)$, let $Y_i = 0$ if $v_i = 0$ and $Y_i = X_i$ if $v_i = 1$. Set $P_{\mathbf{v}}(\mathbf{X}) = P(Y_1, \ldots, Y_m)$ and multi-linearize. Using an argument like in Theorem 5.2, we can show that this gives a polynomial representation of G over $\mathbb{Z}_p$. Since the $P_{\mathbf{v}}(\mathbf{X})$s are multilinear polynomials of degree d, we get $\omega(G) \leq \binom{m}{\leq d}$. Similarly we bound $\alpha(G)$ by representing $\overline{(}G)$ over $\mathbb{Z}_q$.

For prime-power representations of OR we can represent G and $\overline{G}$ over $\mathbb{Z}_{p^a}$ and get a similar bound. □

The OR representation of Equation 45 gives the following construction due to Alon [5]. Let $n = pq - 1, m = n^2$. The vertices are all subsets of $[m]$ of size n. Add $(\mathbf{x},\mathbf{y})$ to $E(G)$ if $|\mathbf{x} \cap \mathbf{y}| \not\equiv -1 \bmod p$.

The OR representation of Equation 47 gives the Frankl-Wilson construction: Let $n = p^2 - 1, m = n^2$. The vertices are all subsets of $[m]$ of size n. Add edge $(\mathbf{x},\mathbf{y})$ to $E(G)$ if $|\mathbf{x} \cap \mathbf{y}| \not\equiv -1 \bmod p$. This construction can be extended to $t \geq 2$ colors using the polynomials constructed in Equation 49.

Construction 5.3 Extending the Frankl-Wilson construction to t colors.

- Take $n = p^t - 1, m = p^{t+1}$. Vertices are all n subsets of m.
- Edges are colored $\{0, \ldots, t - 1\}$. Edge $(\mathbf{x},\mathbf{y})$ is given color $\mathrm{val}_p(1 + |\mathbf{x} \cap \mathbf{y}|)$.

The OR representation of Equation 46 gives the following graph $G(V,E)$. Let $n = 2^k 3^\ell - 1, m = n^2$. The vertices are all subsets of $[m]$ of size n. Add $(\mathbf{x},\mathbf{y})$ to $E(G)$ if

$|\mathbf{x} \cap \mathbf{y}| \not\equiv -1 \bmod 2^k$. In fact the graph obtained is the same as Grolmusz. To show this, we first present his construction, following the simplified exposition of Grolmusz himself [44] and Kutin [57].

1) Let $n = 2^k 3^\ell - 1$. The BBR polynomials give the following representation of W_n.

$$P(\mathbf{X}) = \binom{\sum X_i}{2^k - 1}, \quad Q(\mathbf{X}) = \binom{\sum X_i}{3^\ell - 1}$$

Define $R(\mathbf{X}) \in \mathbb{Z}_6[\mathbf{X}]$ to be the polynomial obtained by combining these polynomials using the CRT. It follows that $R(\mathbf{x}) \equiv 1 \bmod 6$ when $w(\mathbf{x}) = n$ and $R(\mathbf{x}$ is divisible by 2 or 3 when $w(\mathbf{x}) < n$.

2) We can view $R(\mathbf{X})$ as an integer polynomial with coefficients in $\{0, \ldots, 5\}$. By repeating each monomial sufficiently many times, we can write

$$R(\mathbf{X}) = \sum_{\alpha} \prod_{i \in \alpha} X_i$$

The elements of the universe are the monomials. If α is repeated c times in $R(\mathbf{X})$, then there are c elements in the universe, one for each occurance of α. For each $\mathbf{x} \in \{0,1\}^m$ of weight n, the set $S(\mathbf{x})$ consists of monomials that evaluate to 1 on $\mathbf{x}$. One can verify that this system has restricted intersections mod 6 since $|S(\mathbf{x}) \cap S(\mathbf{y})| = R(\mathbf{x} \cap \mathbf{y})$.

3) The vertices of the graph H are the sets $S(\mathbf{x})$. We add edge $(S(\mathbf{x}), S(\mathbf{y}))$ if $|S(\mathbf{x}) \cap S(\mathbf{y})| \equiv 0 \bmod 2$.

We wish to show that this graph H is the same as the graph G constructed above. We identify $\mathbf{x} \in V(G)$ with $S(\mathbf{x}) \in V(H)$. In H, we add an edge between $S(\mathbf{x})$ and $S(\mathbf{y})$ if $R(\mathbf{x} \cap \mathbf{y}) \equiv 0 \bmod 2$. By the CRT, this implies that $P(\mathbf{x} \cap \mathbf{y}) \equiv 0 \bmod 2$. By Lucas' theorem, this happens if $w(\mathbf{x} \cap \mathbf{y}) \not\equiv -1 \bmod 2^k$. But this is precisely when $(\mathbf{x}, \mathbf{y})$ is an edge of G.

This equivalence can also be seen from Kutin's construction [57]. While Grolmusz's set system construction is an important result, our approach seems to be simpler for the purpose of Ramsey graph construction. One can interpret the bounds on clique and independent set size as coming from an extension of the modular Ray-Chaudhuri-Wilson theorem to prime powers, which we prove in Appendix A (Theorem A.5).

5.4 Lower Bounds for Prime-Power Representations

In this section we prove a lower bound for prime-power representations by symmetric polynomials.

Theorem 5.5 *Let* $P(\mathbf{X}) \in \mathbb{Z}_{p^a}[\mathbf{X}]$ *and* $Q(\mathbf{X}) \in \mathbb{Z}_{p^b}[\mathbf{X}]$ *be symmetric polynomials that represent the* OR *function on* n *variables. Then* $(\deg(P)+1)\cdot(\deg(Q)+1) \geq \frac{n}{2}$.

Since a symmetric polynomial on a 0-1 vector is essentially a polynomial in the weight of the vector, we can restate Theorem 5.5 in terms of integer polynomials. The formal proof is easy and is omitted. While we could also work with polynomials in $\mathbb{Z}_{p^a}[X]$, the presence of zero divisors would make it messier to use valuations.

Proposition 5.6 *Let* $P(X), Q(X) \in \mathbb{Z}[X]$ *be univariate polynomials such that for* $x \in \{1,\ldots,n\}$,

$$\mathrm{val}_p[P(0)] < \mathrm{val}_p[P(x)] \quad \textit{or} \quad \mathrm{val}_p[Q(0)] < \mathrm{val}_p[Q(x)] \tag{50}$$

Then $(\deg(P)+1)\cdot(\deg(Q)+1) \geq \frac{n}{2}$.

The next two Lemmas (5.7 and 5.8) develop tools to show degree bounds for such polynomials.

Definition 5.9 *A sequence* $S = (\alpha_1,\ldots,\alpha_d)$ *of integers is called a greedy sequence if for all* j,

$$\sum_{i<j} \mathrm{val}_p[\alpha_j - \alpha_i] \;\leq\; \sum_{i<j} \mathrm{val}_p[\alpha_k - \alpha_i] \quad \textit{for} \quad k \neq j$$

Let us define $N_1(X) = 1$ and $N_j(X) = \prod_{i<j}(X-\alpha_i)$ for $j > 1$. The definition of a greedy sequence can be restated as $\mathrm{val}_p[N_j(\alpha_j)] \leq \mathrm{val}_p[N_j(\alpha_k)]$ for $k \neq j$. Given any set S, we can order it elements *greedily* as follows to get a greedy sequence: we choose α_1 arbitrarily; having chosen $(\alpha_1,\ldots,\alpha_{j-1})$ we choose $\alpha_j \in S$ to be the element that minimizes $\mathrm{val}_p[N_j(\alpha_j)]$.

The definition of greedy sequences is reminiscent of interpolating sets modulo prime-powers; indeed they are analogues of interpolating sets over the integers. An example of

a greedy sequence is when $(\alpha_1, \ldots, \alpha_d)$ are consecutive integers. The intuition for next Lemma is from Algorithm 4.2 from Chapter 4 for polynomial interpolation over $\mathbb{Z}_{p^a}$. Given a set $S \subseteq \mathbb{Z}_{p^a}$, and evaluations of some polynomial in $\mathbb{Z}_{p^a}[X]$, Algorithm 4.2 will output the smallest degree polynomial $P(X)$ that fits the data, provided it sees the elements of S in the above greedy order. If the polynomial is 0 on every element but the last, then the algorithm is forced to output a polynomial of degree $d-1$. Greedy sequences are an analogue of this over the integers.

Lemma 5.7 *Let $S = (\alpha_1, \ldots, \alpha_d)$ be a greedy sequence. Let $P(X) \in \mathbb{Z}[X]$ be such that*

$$\mathrm{val}_p[P(\alpha_d)] \;<\; \mathrm{val}_p[P(\alpha_i)] \quad \textit{for } i \leq d-1$$

Then $\deg(P) \geq d-1$.

PROOF: The proof is by induction on d. We will show the converse, namely that if $\deg(P) \leq d-2$.

$$\mathrm{val}_p[P(\alpha_d)] \;\geq\; \min_{i \leq d-1} \mathrm{val}_p[P(\alpha_i)]$$

The base case $d = 2$ is trivial, in this case P is constant so it is clear that $\mathrm{val}_p[P(\alpha_2)] = \mathrm{val}_p[P(\alpha_1)]$. Assume the property holds for greedy sequences of length $d-1$. Given a polynomial $P(X)$ of degree $d-2$, since $N_{d-1}(X)$ is a monic polynomial of degree $d-2$, we write $P(X) = Q(X) + c_{d-1}N_{d-1}(X)$, where $Q(X)$ is a polynomial of degree $d-3$. Substituting $X = \alpha_d$,

$$P(\alpha_d) \;=\; Q(\alpha_d) + c_{d-1}N_{d-1}(\alpha_d)$$

hence by the ultrametric inequality

$$\mathrm{val}_p[P(\alpha_d)] \;\geq\; \min\{\mathrm{val}_p[Q(\alpha_d)], \mathrm{val}_p[c_{d-1}N_{d-1}(\alpha_d)]\} \tag{51}$$

To lower bound $\mathrm{val}_p[Q(\alpha_d)]$, note that the sequence $(\alpha_1, \ldots, \alpha_{d-2}, \alpha_d)$ of length $d-1$ obtained by deleting α_{d-1} is also greedy. Hence applying the inductive hypothesis to $Q(\alpha_d)$, we get

$$\mathrm{val}_p[Q(\alpha_d)] \;\geq\; \min_{i \leq d-2} \mathrm{val}_p[Q(\alpha_i)] \;=\; \min_{i \leq d-2} \mathrm{val}_p[P(\alpha_i)] \tag{52}$$

The last equality follows since $N_{d-1}(\alpha_i) = 0$ for $i \leq d-2$, hence $Q(\alpha_i) = P(\alpha_i)$. We now lower bound $\mathrm{val}_p[c_{d-1}N_{d-1}(\alpha_d)]$. Using the greedy property of the sequence $(\alpha_1, \ldots, \alpha_d)$,

$$\mathrm{val}_p[c_{d-1}N_{d-1}(\alpha_d)] \;\geq\; \mathrm{val}_p[c_{d-1}N_{d-1}(\alpha_{d-1})]$$

$$c_{d-1}N_{d-1}(\alpha_{d-1}) \;=\; P(\alpha_{d-1}) - Q(\alpha_{d-1})$$

Hence we have

$$\mathrm{val}_p[c_{d-1}N_{d-1}(\alpha_{d-1})] \;\geq\; \min\{\mathrm{val}_p[P(\alpha_{d-1})], \mathrm{val}_p[Q(\alpha_{d-1})]\} \tag{53}$$

Since $(\alpha_1, \ldots, \alpha_{d-1})$ is a greedy sequence and Q has degree $d-3$, we get by induction that

$$\mathrm{val}_p[Q(\alpha_{d-1})] \;\geq\; \min_{i \leq d-2} \mathrm{val}_p[Q(\alpha_i)] \;=\; \min_{i \leq d-2} \mathrm{val}_p[P(\alpha_i)] \tag{54}$$

Combining Equations 51, 52, 53, 54 gives the desired result. □

Next we define the notion of a greedy array which we use to construct long greedy sequences. Given a t-dimensional matrix A of dimension $d_0 \times \ldots \times d_{t-1}$, we use $A[\mathbf{i}]$ to denote $A[i_0, \ldots, i_{t-1}]$.

Definition 5.10 *A t-dimensional matrix of distinct integers A is called a greedy array if*

$$\mathrm{val}_p[A[\mathbf{i}] - A[\mathbf{j}]\,] = \min\{a | i_a \neq j_a\} \tag{55}$$

We define an ordering of the array indices, which is essentially the reverse lexicographic (revlex) ordering.

Definition 5.11 *Given a t-dimensional integer vectors $\mathbf{i}$ and $\mathbf{j}$, let $\ell = \max\{a | i_a \neq j_a\}$. Then $\mathbf{i} < \mathbf{j}$ if $i_\ell < j_\ell$.*

Note that for a greedy array, the valuation should equal the smallest index where $\mathbf{i}$ and $\mathbf{j}$ differ. However to order elements, we look at the largest index where they differ. For example, consider the $p \times \cdots \times p$ array where $A[i_0, \ldots, i_{t-1}] = i_0 + i_1 p \ldots i_{t-1} p^{t-1}$ and $0 \leq i_j \leq p-1$. Thus the array contains $i \in \{0, \ldots, p^t - 1\}$ with numbers indexed by their base-p expansion. Since $\mathrm{val}_p[i-j]$ depends on the smallest digit where i and j, this is a greedy array. The ordering defined above is the usual ordering of integers, it depends on the largest digit where the expansions differ.

Lemma 5.8 *Ordering elements of a greedy array gives a greedy sequence.*

PROOF: We want to show that for $\mathbf{k} \neq \mathbf{j}$

$$\sum_{\mathbf{i}<\mathbf{j}} \mathrm{val}_p[A[\mathbf{j}] - A[\mathbf{i}]\ ] \quad \leq \quad \sum_{\mathbf{i}<\mathbf{j}} \mathrm{val}_p[A[\mathbf{k}] - A[\mathbf{i}]\ ] \tag{56}$$

For $0 \leq a \leq t-1$, we define the set

$$S_a = \{\mathbf{i} | i_a < j_a,\ i_{a+1} = j_{a+1}, \ldots,\ i_{t-1} = j_{t-1}\}$$

The indices $i_0, \ldots, i_{a-1}$ are unrestricted. Note that the S_as are disjoint and they partition the set $\{\mathbf{i}|\mathbf{i} < \mathbf{j}\}$. We show that for every a, and for $\mathbf{k} \neq \mathbf{j}$

$$\sum_{\mathbf{i}\in S_a} \mathrm{val}_p[A[\mathbf{j}] - A[\mathbf{i}]\ ] \quad \leq \quad \sum_{\mathbf{i}\in S_a} \mathrm{val}_p[A[\mathbf{k}] - A[\mathbf{i}]\ ] \tag{57}$$

Equation 56 will follow by summing over all a. Hence consider a fixed a. Note that if $\mathbf{i} \in S_a$ then $0 \leq \mathrm{val}_p[A[\mathbf{j}] - A[\mathbf{i}]\ ] \leq a$. Accordingly we partition S_a into $J(0), \ldots, J(a)$ as follows: for $0 \leq \ell \leq a-1$,

$$\begin{aligned} J(\ell) &= \{\mathbf{i} \in S_a | i_0 = j_0, \ldots, i_{\ell-1} = j_{\ell-1}, i_\ell \neq j_\ell\} \\ &= \{\mathbf{i} \in S_a |\ \mathrm{val}_p[A[\mathbf{j}] - A[\mathbf{i}]\ ] = \ell\} \\ J(a) &= \{\mathbf{i} \in S_a | i_0 = j_0, \ldots, i_{a-1} = j_{a-1}\} \end{aligned}$$

For $\mathbf{i} \in J(a)$ we have $\mathrm{val}_p[A[\mathbf{j}] - A[\mathbf{i}]\ ] = a$ since for all $\mathbf{i} \in S_a$, $i_a < j_a$ so $i_a \neq j_a$. Now given $\mathbf{k} \neq \mathbf{i}$ let us define the sets $K(0), \ldots, K(a)$ as follows. For $0 \leq \ell \leq a-1$,

$$\begin{aligned} K(\ell) &= \{\mathbf{i} \in S_a | i_0 = k_0, \ldots, i_{\ell-1} = k_{\ell-1}, i_\ell \neq k_\ell\} \\ &= \{\mathbf{i} \in S_a |\ \mathrm{val}_p[A[\mathbf{k}] - A[\mathbf{i}]\ ] = \ell\} \\ K(a) &= \{\mathbf{i} \in S_a | i_0 = k_0, \ldots, i_{a-1} = k_{a-1}\} \end{aligned}$$

Unlike for $J(a)$, for $i \in K(a)$ it could be that $i_a = k_a$, so we have $\mathrm{val}_p[A[\mathbf{k}] - A[\mathbf{i}]\ ] \geq a$. Since the indices $i_0, \ldots, i_{a-1}$ are unrestricted in S_a, we have $|J(\ell)| = |K(\ell)|$ for $0 \leq \ell \leq a$.

We now prove Equation 57.

$$\begin{aligned}
\sum_{\mathbf{i}\in S_a} \mathrm{val}_p[A[\mathbf{j}] - A[\mathbf{i}]\ ] &= \sum_{0\le\ell\le a} \sum_{\mathbf{i}\in J(\ell)} \mathrm{val}_p[A[\mathbf{j}] - A[\mathbf{i}]\ ] \\
&= \sum_{0\le\ell\le a} \ell \cdot |J(\ell)| \\
\sum_{\mathbf{i}\in S_a} \mathrm{val}_p[A[\mathbf{k}] - A[\mathbf{i}]\ ] &= \sum_{0\le\ell\le a} \sum_{\mathbf{i}\in K(\ell)} \mathrm{val}_p[A[\mathbf{k}] - A[\mathbf{i}]\ ] \\
&\ge \sum_{0\le\ell\le a} \ell \cdot |K(\ell)| \\
&\ge \sum_{0\le\ell\le a} \ell \cdot |J(\ell)|
\end{aligned}$$

Hence the claim follows. □

A two-dimensional greedy array is a matrix G of integers such that elements in the same row are congruent mod p, while elements in distinct rows are not congruent mod p. Lemma 5.8 says that ordering the elements of G column-wise gives a greedy sequence.

This concludes the algebraic step of the proof. Let us sketch the rest of the proof when $n = p^2 - 1$, which corresponds to the Frankl-Wilson construction (see Figure 2). Define the sets

$$A = \{0\} \cup \{x \in \{1, \ldots, p^2-1\} \mid \mathrm{val}_p[P(0)] < \mathrm{val}_p[P(x)]\}$$

$$B = \{0\} \cup \{x \in \{1, \ldots, p^2-1\} \mid \mathrm{val}_p[P(0)] \ge \mathrm{val}_p[P(x)]\}$$

Note that $\mathrm{val}_p[Q(0)] < \mathrm{val}_p[Q(x)]$ for every $x \neq 0$ in B. Further A and B partition the set $\{1, \ldots, p^2-1\}$ and they intersect only at 0. We will show that A and B contain *large* greedy arrays.

1. Arrange $\{0, \ldots p^2-1\}$ in $p \times p$ grid, each row corresponding to a congruence class mod p.

2. Within each row, place elements lying in A before those in B. Since 0 lies in $A \cap B$, place all other elements in A which are 0 mod p before 0.

3. Sort the rows according to how many elements from A they contain.

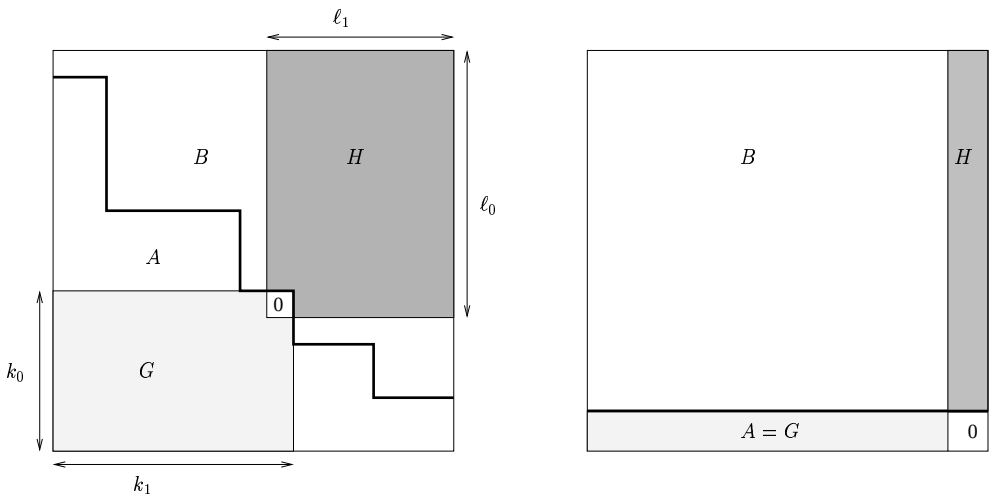

Figure 2: Lower bound for $p^2 - 1$ and the Frankl-Wilson construction

This reordering is illustrated in Figure 2, the dark line separates A and B. It is clear that A and B contain greedy arrays G of size $k_0 \times k_1$ and H of size $\ell_0 \times \ell_1$ respectively (indicated by shaded regions) so that $k_0 + \ell_0 = k_1 + \ell_1 = p + 1$. From this it follows that $|G||H| \geq p^2$. Also, we can ensure that 0 is the last element of these arrays in the column-wise ordering. So $\mathrm{val}_p[P(x)]$ is minimized at the last element in G, hence by Lemma 5.8 $\deg(P) \geq |G| - 1$. Similarly $\deg(Q) \geq |H| - 1$, which proves the desired bound. Also, $|G||H|$ is minimized when $A = \{x | x \equiv 0 \bmod p\}$, and $B = 0 \cup \{x | x \not\equiv 0 \bmod p\}$ (or vice versa), the corresponding polynomials give exactly the Frankl-Wilson construction.

The proof for general n is a high dimensional extension of this argument. The next lemma (Lemma 5.9) says that any disjoint partition of $\{0, \ldots, n-1\}$ into A, B will result in one of the partitions having a greedy array of size $\sqrt{n}$. In fact we prove something stronger, we can choose the dimensions of the array to be any solution to Equation 58. We also assume that n is of the form $n' p^{t-1}$ which makes it easier to use induction.

Lemma 5.9 *Let $1 \leq n' \leq p$. Let A, B be disjoint sets of integers such that $A \cup B = \{0, \ldots, n' p^{t-1} - 1\}$. Given any positive integers $k_0, \ldots, k_{t-1}, \ell_0, \ldots, \ell_{t-1}$ satisfying*

$$\begin{aligned} \text{For } i \leq t-2 \qquad k_i + \ell_i &= p + 1 \\ k_{t-1} + \ell_{t-1} &= n' + 1 \end{aligned} \tag{58}$$

either A contains a greedy array of size $k_0 \times \cdots \times k_{t-1}$ or B contains a greedy array of size

$\ell_0 \times \cdots \times \ell_{t-1}$.

PROOF: The proof is by induction on t (the dimension of the greedy arrays).

When $t = 1$, we have disjoint sets A, B so that $A \cup B = \{0, \ldots, n'-1\}$ hence $|A|+|B| = n'$. Since $n'-1 < p$ any ordering of A and B gives greedy arrays of size $|A|$ and $|B|$ respectively. Given k_0, ℓ_0 such that $k_0 + \ell_0 = n' + 1$, if $|A| \leq k_0 - 1$, then $|B| \geq n' + 1 - k_0 = \ell_0$.

Assume that the claim is true up to $t-1$. For $0 \leq i \leq p-1$, we define the following sets

$$A(i) = \{x \in A | x \equiv i \bmod p\}$$

$$\hat{A}(i) = \{(x - i)/p \mid x \in A(i)\}$$

We define sets $B(i)$ and $\hat{B}(i)$ similarly. Note that for each i, $\hat{A}(i)$ and $\hat{B}(i)$ are disjoint, further $\hat{A}(i) \cup \hat{B}(i) = \{0, \ldots, n'p^{t-2} - 1\}$. So the induction hypothesis applied to $\hat{A}(i)$ and $\hat{B}(i)$ with $k_1, \ldots, k_{t-1}, \ell_1, \ldots, \ell_{t-1}$ implies that either A contains a greedy array of size $k_1 \times \cdots \times k_{t-1}$ or B contains a greedy array of size $\ell_1 \times \cdots \times \ell_{t-1}$. We define the following sets

$$S = \{i | \hat{A}(i) \text{ has a greedy array } \hat{G}_i \text{ of size } k_1 \times \cdots \times k_{t-1}\}$$

$$T = \{i | \hat{B}(i) \text{ has a greedy array } \hat{H}_i \text{ of size } \ell_1 \times \cdots \times \ell_{t-1}\}$$

Since S, T are disjoint and $|S| + |T| = p$ we have either $|S| \geq k_0$ or $|T| \geq \ell_0$. Assume $|S| \geq k_0$. We define a greedy array G of size $k_0 \times \cdots \times k_{t-1}$ as follows. Choose $S' \subset S$ of size k_0. For each $i \in S'$, the i^{th} row of G contains the pre-image G_i of $\hat{G}_i$ in $A(i)$ of dimension $k_1 \times \cdots \times k_{t-1}$.

We need to verify that G satisfies $\text{val}_p(G[\mathbf{i}] - G[\mathbf{j}]) = \min\{a | i_a \neq j_a\}$. Given $\mathbf{i}$ and $\mathbf{j}$, if $i_0 \neq j_0$, then $G[\mathbf{i}] \not\equiv G[\mathbf{j}] \bmod p$ so the condition holds. Now assume that $i_0 = j_0$, so that $\mathbf{i} = (i_0, \mathbf{i}'), \mathbf{j} = (i_0, \mathbf{j}')$. Since $G[\mathbf{i}]$ and $G[\mathbf{j}]$ are in the same row, $G[\mathbf{i}] \equiv G[\mathbf{j}] \equiv c \bmod p$ for

$0 \le c \le p-1$. So

$$\begin{aligned} G[\mathbf{i}] - G[\mathbf{j}] &= G_c[\mathbf{i}'] - G_c[\mathbf{j}'] \\ &= (p\hat{G}_c[\mathbf{i}'] + c) - (p\hat{G}_c[\mathbf{j}'] + c) \\ &= p(\hat{G}_c[\mathbf{i}'] - \hat{G}_c[\mathbf{j}']) \\ \Rightarrow \ \mathrm{val}_p[G[\mathbf{i}] - G[\mathbf{j}]\ ] &= 1 + \mathrm{val}_p[G_c[\mathbf{i}'] - G_c[\mathbf{j}']\ ] \\ &= 1 + \min\{a | i'_a \neq j'_a\} \end{aligned}$$

Note that $\min\{a|i_a \neq i_a\} = 1 + \min\{a|i'_a \neq i'_a\}$. Hence G is a greedy array of the right dimension. □

The next Lemma is the key step in the combinatorial argument. Now we consider sets A and B which intersect only at 0, and we want to produce greedy arrays that end at 0 by our ordering. We show that such arrays exist whose dimensions satisfy Equation 58.

Lemma 5.10 Partition Lemma: *Let $1 \le n' \le p$. Let A, B be sets of integers such that*

$$A \cup B = \{0, \ldots, n'p^{t-1} - 1\}, \quad A \cap B = \{0\}$$

Then there exist positive integers $k_0, \cdots, k_{t-1}, \ell_0, \cdots, \ell_{t-1}$ satisfying Equation 58, so that A contains a greedy array G of size $k_0 \times \cdots \times k_{t-1}$ and B contains a greedy array H of size $\ell_0 \times \cdots \times \ell_{t-1}$, and both G and H contain 0 as the last element.

PROOF: The proof is by induction on t.

When $t = 1$, we have sets A, B so that $A \cup B = \{0, \ldots, n' - 1\}$ and $A \cap B = \{0\}$ so $|A| + |B| = n' + 1$. We take $k_0 = |A|, \ell_0 = |B|$. Define G to be an ordering of A where 0 comes last, similarly for H.

Assume that the claim holds up to $t-1$. For $0 \le i \le p-1$, we define the sets $A(i), \hat{A}(i), B(i), \hat{B}(i)$ as before. Note that

$$\hat{A}(0) \cup \hat{B}(0) = \{0, \ldots, n'p^{t-2} - 1\}$$
$$\hat{A}(0) \cap \hat{B}(0) = \{0\}$$

By induction, there exist $k_1, \ldots, k_{t-1}$ and $\ell_1, \ldots, \ell_{t-1}$ as above so that $\hat{A}(0)$ contains a greedy array of size $k_1 \times \cdots \times k_{t-1}$ and $\hat{B}(0)$ contains a greedy array of size $\ell_1 \times \cdots \times \ell_{t-1}$. For $1 \leq i \leq p-1$ we have

$$\begin{aligned} \hat{A}(i) \cup \hat{B}(i) &= \{0, \ldots, n'p^{t-2} - 1\} \\ \hat{A}(i) \cap \hat{B}(i) &= \phi \end{aligned}$$

Hence applying Lemma 5.9, either $\hat{A}(i)$ contains an array of size $k_1 \times \cdots \times k_{t-1}$ or $\hat{B}(0)$ contains a greedy array of size $\ell_1 \times \cdots \times \ell_{t-1}$. Again we define the sets

$$S = \{i | \hat{A}(i) \text{ has a greedy array } \hat{G}_i \text{ of size } k_1 \times \cdots \times k_{t-1}\}$$
$$T = \{i | \hat{B}(i) \text{ has a greedy array } \hat{H}_i \text{ of size } \ell_1 \times \cdots \times \ell_{t-1}\}$$

Let $k_0 = |S|, \ell_0 = |T|$. Since $S \cap T = \{0\}$ and $S \cup T = \{0, \ldots, p-1\}$ we have $k_0 + \ell_0 = p+1$. Order S and T so that 0 is the last element. We define a greedy array G of size $k_0 \times \cdots \times k_{t-1}$ as follows. For each $i \in S$, the i^{th} row of G contains the pre-image G_i of $\hat{G}_i$ in $A(i)$ of dimension $k_1 \times \cdots \times k_{t-1}$. Similarly we define H where the i^{th} row contains the pre-image H_i of $\hat{H}_i$ in $B(i)$. The proof that these are greedy arrays follows Lemma 5.9. They both contain 0 as the last element by induction. □

We now complete the proof of Theorem 5.5.

PROOF OF THEOREM 5.5:

Assume that $p^{t-1} \leq n < p^t$. We can choose n' so that $1 \leq n' \leq p$ and $n/2 \leq n'p^{t-1} - 1 \leq n$. Define the sets

$$A = \{0\} \cup \{x \mid 1 \leq x \leq n'p^{t-1} - 1,\ \text{val}_p[P(0)] < \text{val}_p[P(x)]\}$$
$$B = \{0\} \cup \{x \mid 1 \leq x \leq n'p^{t-1} - 1,\ \text{val}_p[P(0)] \geq \text{val}_p[P(x)]\}$$

Applying Lemma 5.10 implies that A and B contain greedy arrays G and H of size $k_0 \times \cdots \times k_{t-1}$ and $\ell_0 \times \cdots \times \ell_{t-1}$ respectively where k_i and ℓ_i satisfy Equation 58. Applying Lemma 5.8, by ordering G we get a greedy sequence $\{\alpha_1, \ldots, \alpha_{d-1}, 0\}$ in A of length $d = \prod_j k_j$. By the definition of set A, $\text{val}_p[P(0)] < \text{val}_p[P(\alpha_i)]$ for $i \leq d$. So by Lemma 5.7 $\deg(P) \geq \prod_j k_j - 1$.

Similarly we get a greedy sequence of length $\prod_j \ell_j$ in B ending in 0. Note that by Equation 50, $x \in B$ and $x \neq 0$ implies $\text{val}_p[Q(0)] < \text{val}_p[Q(x)]$. So by Lemma 5.7, $\deg(Q) \geq \prod_j \ell_j - 1$. By Equation 58, $k_j \ell_j \geq p$ for $j \leq t-2$ and $k_{t-1}\ell_{t-1} \geq n'$. Hence

$$\begin{aligned}(\deg(P)+1)(\deg(Q)+1) &\geq \prod_j k_j \ell_j \\ &\geq n' p^{t-1} > n/2.\end{aligned}$$

For the Frankl-Wilson construction where $n = p^2 - 1$, we get $(\deg(P)+1)(\deg(Q)+1) \geq p^2$ which is tight. □

5.5 Lower Bounds for Prime Representations

In this section we prove a lower bound for prime representations using symmetric polynomials.

Theorem 5.11 *Let $P(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ and $Q(\mathbf{X}) \in \mathbb{Z}_q[\mathbf{X}]$ be symmetric polynomials that represent the* OR *function on n variables. Then* $\deg(P).\deg(Q) \geq n/10$.

Note that this requires $\deg(P), \deg(Q) \geq 1$ but if $\deg(P) = 0$, then it is easy to show that $\deg(Q) = n$, so this case is not interesting. The hard case of this theorem is when p and q are fast-growing functions of n, as in Alon's construction. To handle this case, we prove a partition lemma (Lemma 5.12) which says that taking p and q large does not help.

Definition 5.12 *Let $p < q$ be distinct primes, let $n < pq$. Let $A \subseteq \mathbb{Z}_p^*$ and $B \subseteq \mathbb{Z}_q^*$. We say that x is covered by A if $x \bmod p \in A$. We say A and B cover $[n]$ if every $x \in \{1, \ldots, n\}$ is covered by A or B.*

If $n < pq$, we can cover $[n]$ by taking $A = \mathbb{Z}_p^*$ and $B = \mathbb{Z}_q^*$. Given $A \subseteq \mathbb{Z}_p^*$ and $B \subseteq \mathbb{Z}_q^*$, the number of elements in $\{1, \ldots, pq\}$ that are covered by A or B is $|A|q + |B|p - |A||B|$ which can be much larger than $|A||B|$. The partition lemma states that to cover the first n integers however, $|A||B|$ needs to be $\Omega(n)$.

Lemma 5.12 Partition Lemma: *If $A \subseteq \mathbb{Z}_p^*$ and $B \subseteq \mathbb{Z}_q^*$ cover $[n]$, then $(|A|+1) \cdot (|B|+1) > \frac{n}{2}$.*

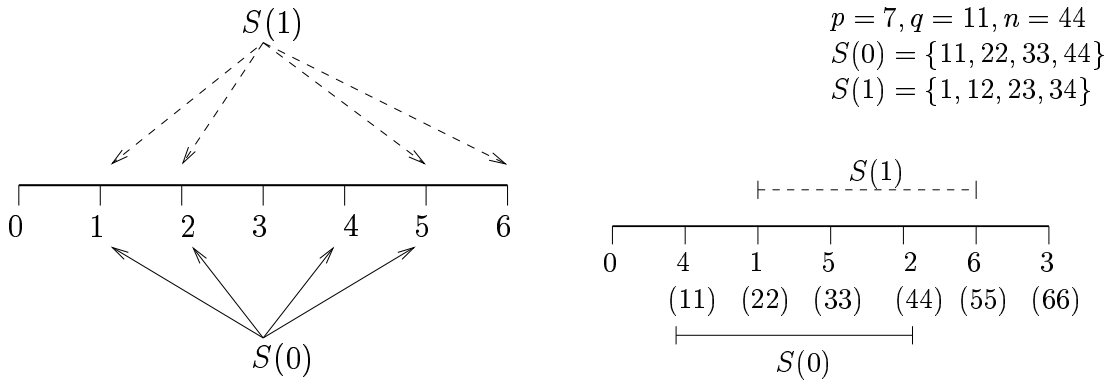

Figure 3: Proof of the Partition Lemma

Using $|A|+1$ rather than $|A|$ in the product lets us ignore the case when $|A|=0$. Let us sketch the idea behind the proof of the Partition Lemma. Let $n = n_q q$. Assume that to begin with, we have $B = \mathbb{Z}_q^*$ and $A = \{q, 2q, \ldots, n_q q\}$. It is clear that A and B cover n, however $(|A|+1)(|B|+1) > n$. One could try and reduce $|B|$ by removing elements from it. We want to show that this results in an increase in $|A|$. Removing $i \in \mathbb{Z}_q^*$ from B results in the numbers $\{i, i+q, \ldots, i+(n_q-1)q\}$ being uncovered. Call this set $S(i)$. The various elements of $S(i)$ are less than pq and they are congruent mod q, hence the CRT implies they cannot also be congruent mod p. But the problem is for $i \neq j$,there could be considerable overlap between the residues of $S(i)$ and $S(j)$ mod p. Hence it is not clear that removing many elements from B does actually cause $|A|$ to increase. However, by suitably reordering the elements of $\mathbb{Z}_p$, we show that every element removed from B causes the size of A to increase by at least 1. In fact Figure 3 shows that $|A|$ could increase by just 1. This is sufficient to prove the Partition Lemma.

Set $n_p = \left\lfloor \frac{n}{p} \right\rfloor$ and $n_q = \left\lfloor \frac{n}{q} \right\rfloor$. Given set S of integers, define $S \bmod p \subseteq \mathbb{Z}_p$ to be the set $\{x \bmod p | x \in S\}$.

Proposition 5.13 *Let $n \geq q$. If A and B cover $[n]$ and $|B| = q - \ell$ then $|A| \geq \left\lfloor \frac{n}{q} \right\rfloor + \ell - 1$.*

PROOF: Note that since $n \geq q$, $n_q \geq 1$. Let $\overline{B}$ denote the complement of B in $\mathbb{Z}_q$, so $0 \in \overline{B}$. For each $i \in \overline{B}$, take $S(i)$ to be the first n_q numbers in $\{1, \ldots, n\}$ congruent to i mod p. In

other words, $S(0) = \{q, 2q \ldots, n_q q\}$ and for $i \neq 0, S(i) = \{i, i+q, \ldots, i+(n_q-1)q\}$. Let

$$S = \bigcup_{i \in \overline{B}} S(i)$$

If $x \in S$, then x is not covered by B so it must be covered by A. We want to lower bound the size of S mod p.

Let us reorder the set $\mathbb{Z}_p$ as $\{0, q, 2q, \ldots, (p-1)q\}$ (this is a reordering since $(q,p) = 1$). It sends $j \bmod p$ to $c(j)q$ such that $c(j)q \equiv j \bmod p$. This map sends $S(0) \bmod p$ to $\{q, \ldots, n_q q\}$ and the set $S(i) \bmod p$ to the interval $\{c(i)q, (c(i)+1)q, \ldots, (c(i)+n_q-1)q\}$ of length n_q for $i \neq 0$. None of these intervals contain 0, since that would give $x \in \{1, \ldots, n\}$ such that $x \equiv i \bmod q$ and $x \equiv 0 \bmod p$. Such an x is not covered by A or B. Each interval $S(i) \bmod p$ begins at a distinct point $c(i)$. Sorting the intervals by their starting points, it follows that the union of ℓ such intervals of length n_q contains at least $n_q + \ell - 1$ elements of $\mathbb{Z}_p^*$. □

Figure 3 illustrates this argument for $p = 7, q = 11, n = 44$. Here $B = \mathbb{Z}_{11} \setminus \{0, 1\}$.

PROOF OF LEMMA 5.12:

We consider the cases $n < p$, $p < n \leq q$ and $q \leq n$ separately. The non-trivial case is when $q \leq n$.

1. Let $n \leq p < q$. Numbers $\{1, \ldots, n\}$ lie in distinct congruence classes mod p and q. Hence

$$|A| + |B| \geq n \quad \Rightarrow \quad (|A|+1) \cdot (|B|+1) > n$$

2. Let $p < n \leq q$. The numbers $\{1, \ldots, p\}$ lie in distinct congruence classes modulo p and q. Hence $|A| + |B| \geq p$ and $(|A|+1) \cdot (|B|+1) > p$. This proves the claim if $n \leq 2p$ so let $n > 2p$.

   Let $|A| = p - k$ for $1 \leq k < p$. There are n_p numbers $\leq n$ in each congruence class mod p. Thus $n_p k$ numbers are not covered by A and have to be covered by B. Since $n \leq q$, they lie in distinct congruence classes mod q. Hence $|B| \geq n_p k$. Using the fact that $n \geq 2p$ hence $pn_p \geq n/2$ we get

$$(|A|+1) \cdot (|B|+1) \; > \; (p-k+1)kn_p \; \geq \; pn_p \; > \; n/2$$

3. Let $n > q$. By Prop. 5.13, if $|B| = q - \ell$, then $|A| \geq n_q + \ell - 1$. Since $|A| \leq p - 1$, we get $1 \leq \ell \leq p - n_q$. Hence for $1 \leq \ell \leq p - n_q$ we have

$$(|A| + 1)(|B| + 1) \;\geq\; (q - \ell + 1)\,(n_q + \ell)$$

We will show that this is lower bounded by $n/2$. By differentiating, this bound is minimized at one of the extreme values of ℓ, so it suffices to check the bound is at least $\frac{n}{2}$ for those values. When $\ell = 1$,

$$(q - \ell + 1)\left(\left\lfloor \frac{n}{q} \right\rfloor + \ell\right) \;=\; q\left(\left\lfloor \frac{n}{q} \right\rfloor + 1\right) \;>\; n$$

When $\ell = p - \left\lfloor \frac{n}{q} \right\rfloor$

$$\begin{aligned}(q - \ell + 1)\left(\left\lfloor \frac{n}{q} \right\rfloor + \ell\right) &= \left(q - p + \left\lfloor \frac{n}{q} \right\rfloor + 1\right) p \\ &\geq \left(q - p + \frac{n}{q}\right) p \\ &= (q - p)p + \frac{np}{q}\end{aligned}$$

One of $(q - p)p$ and $(np)/q$ is at least $n/2$: If $q < 2p$, $(np)/q > n/2$. If $q > 2p$, then $(q - p)p \;>\; n/2$.

□

We now proceed to the algebraic step of the proof. Every symmetric polynomial $P(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ computes a symmetric function $w(\mathbf{x}) \to \mathbb{Z}_p$ where $w(\mathbf{x}) \in \{0, \ldots, n\}$. Let $w(\mathbf{x}) = \sum_{i \leq \ell} w_i p^i$. Every polynomial $\bar{P}(w_0, \ldots, w_\ell)$ also computes a function $w(\mathbf{X}) \to \mathbb{Z}_p$. Theorem 2.15 tells us that functions which can be computed by symmetric polynomials $P(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ of degree less than p^{j+1} are the functions which depend only on $w_0, \ldots, w_j$. In other words, they are computed by some polynomial $\bar{P}(w_0, \ldots, w_j)$. For each variable w_j, let $\deg(w_j)$ denote the degree of w_j in $\bar{P}$. If j is the largest index so that $\deg(w_j) > 0$ then $p^j \leq \deg(P) < p^j$. Thus Theorem 2.15 gives a bound with an error factor of p. By defining an appropriate weighted degree of $\bar{P}$, we will make the correspondence exact (Theorem 5.14).

Definition 5.13 *Given $\bar{P}(w_0, \ldots, w_\ell) \in \mathbb{Z}_p[w_0, \ldots, w_\ell]$, the degree of a monomial $\prod_i w_i^{d_i}$ with $d_i \leq p-1$ is defined as $\deg(\prod_i w_i^{d_i}) = \sum_i d_i p^i$. The degree of $\bar{P}$ denoted $\deg(\bar{P})$ is the maximum degree over all monomials.*

Note that if j is the largest index such that $\deg(w_j) > 0$ then $\deg(\bar{P})/2 \leq \deg(w_j)p^j \leq \deg(\bar{P})$.

Theorem 5.14 *For every symmetric polynomial $P(\mathbf{X}) \in \mathbb{Z}_p(\mathbf{X})$ there is a unique polynomial $\bar{P}(w_0, \ldots, w_\ell)$ that computes the same function $w(\mathbf{X}) \to \mathbb{Z}_p$ and vice versa. This correspondence preserves the degree.*

PROOF: Given a symmetric multilinear polynomial $P(\mathbf{X}) \in \mathbb{Z}_p[\mathbf{X}]$ of degree d, write it as $P(\mathbf{X}) = \sum_{k \leq d} c_k S_k(\mathbf{X})$. On at 0-1 vector $\mathbf{x}, S_k(\mathbf{x}) = \binom{w(\mathbf{x})}{k}$. By Lucas' Theorem

$$\binom{w(\mathbf{x})}{k} \equiv \prod_{i \leq \ell} \binom{w_i}{k_i} \bmod p$$

Further the polynomial $\prod_{i \leq \ell} \binom{w_i}{k_i}$ has degree $\sum_i k_i p^i = k$. Thus $P(\mathbf{X})$ computes the same function as

$$\bar{P}(w_0, \ldots, w_\ell) = \sum_{k=0}^{d} c_k \prod_{i \leq \ell} \binom{w_i}{k_i} \bmod p$$

and they have the same degree.

To prove the other direction, observe that the monomials $\prod_{i \leq \ell} \binom{w_i}{k_i}$ with $k_i \leq p-1$ form a basis for polynomials in $\mathbb{Z}_p[w_0, \ldots, w_\ell]$ with degree at most $p-1$ in each w_i. Further writing a polynomial in this basis does not change the degree as defined above. Let $k = \sum_i k_i p^i$ be the degree of the monomial $\prod_i \binom{w_i}{k_i}$. Hence given $\bar{P}(w_0, \ldots, w_\ell)$ with degree d, one can write

$$\bar{P}(w_0, \ldots, w_{\ell-1}) = \sum_{k=0}^{d} c_k \prod_{i < \ell} \binom{w_i}{k_i}$$

By Lucas' theorem, this computes the same function as the polynomial $P(\mathbf{X}) = \sum_{k \leq d} c_k S_k(\mathbf{X})$.
□

For $w \in \{0, \ldots, n\}$, let $w = \sum_{i=0}^{\ell} u_i p^i = \sum_{j=0}^{k} v_j q^j$ denote the base p and base q expansions of w. For $\bar{P}(u_0, \ldots, u_\ell) \in \mathbb{Z}_p[u_0, \ldots, u_\ell]$ let $\bar{P}(w)$ denote the polynomial $\bar{P}$ applied to the base p expansion of w. As consequence of Theorem 5.14, to prove Theorem 5.11 it suffices to prove the following Proposition.

Proposition 5.15 *Let* $\bar{P}(w) \in \mathbb{Z}_p[u_0, \ldots, u_\ell]$ *and* $\bar{Q}(w) \in \mathbb{Z}_q[v_0, \ldots, v_k]$ *be polynomials such that*

$$\bar{P}(0) \equiv 1 \bmod p \quad \textit{and} \quad \bar{Q}(0) \equiv 1 \bmod q$$

For $1 \leq w \leq n$,

$$\bar{P}(w) \equiv 0 \bmod p \quad \textit{or} \quad \bar{Q}(w) \equiv 0 \bmod q$$

Further $\deg(\bar{P}) \cdot \deg(\bar{Q}) \geq \frac{n}{10}$.

PROOF: Let a denote the largest index such that $\deg(u_a) \geq 1$ in $\bar{P}$. This implies $\deg(\bar{P}) \geq p^a$ and $\bar{P}(w) = \bar{P}(u_0, \ldots, u_a)$. Similarly let b be the largest index so that $\deg(v_b) \geq 1$. Then $\deg(\bar{Q}) \geq q^b$ and $\bar{Q}(w) = \bar{Q}(v_0, \ldots, v_b)$. Hence $\deg(\bar{P}) \cdot \deg(\bar{Q}) \geq p^a q^b$. This proves the desired bound for $n < 10p^a q^b$. So we may assume that $n \geq 10p^a q^b$.

Also $n < p^{a+1} q^{b+1}$, since if $w = p^{a+1} q^{b+1} \leq n$, then $u_0, \ldots, u_a = 0$ and $v_0, \ldots, v_b = 0$ so

$$\bar{P}(p^{a+1} q^{b+1}) = \bar{P}(0, \ldots, 0) \equiv 1 \bmod p$$

$$\bar{Q}(p^{a+1} q^{b+1}) = \bar{Q}(0, \ldots, 0) \equiv 1 \bmod p$$

which contradicts the hypothesis. Let $\hat{n} = \left\lfloor \frac{n}{p^a q^b} \right\rfloor < pq$.

Let us consider weights of the form $w = yp^a q^b$ where $0 \leq y \leq \hat{n}$. Observe that this implies $u_0, \ldots, u_{a-1} = 0$ and $u_a \equiv yq^b \bmod p$. Similarly $v_0, \ldots, v_{b-1} = 0$ and $v_b \equiv yp^a \bmod q$. Define polynomials $R(Y) \in \mathbb{Z}_p[Y]$ as $R(Y) = \bar{P}(0, \ldots, 0, Yq^b)$, and $S(Y) \in \mathbb{Z}_q[Y]$ as $S(Y) = \bar{Q}(0, \ldots, 0, Yp^a)$. This implies $\deg(R) = \deg(u_a) \leq p - 1$ and $\deg(S) = \deg(v_b) \leq q - 1$. Note that

$$\begin{aligned} R(0) \equiv 1 \bmod p \quad &\text{and} \quad S(0) \equiv 1 \bmod q \\ R(y) \equiv 0 \bmod p \quad &\text{or} \quad S(y) \equiv 0 \bmod q \quad 1 \leq y \leq \hat{n} \end{aligned} \tag{59}$$

We define $A \subseteq \mathbb{Z}_p^*$ and $B \subseteq \mathbb{Z}_q^*$ to be the 0 sets of $R(Y)$ and $S(Y)$ respectively. By equation 59 A and B cover $[\hat{n}]$. So by Lemma 5.12,

$$(\deg(u_a) + 1)(\deg(v_b) + 1) \;\geq\; \hat{n}/2$$

Since $\deg(u_a), \deg(v_b) \geq 1$, this implies that $\deg(u_a) \cdot \deg(v_b) \geq \hat{n}/8$. Since $\deg(\bar{P}) \geq p^a \deg(u_a)$ and $\deg(\bar{Q}) \geq q^b \deg(v_b)$,

$$\deg(\bar{P}) \cdot \deg(\bar{Q}) \geq \frac{\hat{n}}{8} p^a q^b > \frac{1}{8}\frac{9n}{10} > \frac{n}{10}$$

The second inequality uses the fact that $n \geq 10 p^a q^b$ hence $p^a q^b \left\lfloor \frac{n}{p^a q^b} \right\rfloor > \frac{9n}{10}$. □

Following the breakthrough of Barak *et al.* [13], the algebraic construction described here are no longer the best constructions known. However, the appeal of these constructions is their simplicity and elegance. So we believe that it is important to resolve the question of whether this approach can beat the Frankl-Wilson bound. As we have seen, this problem is intimately linked to well-studied questions in complexity theory. We will discuss some approaches to this problem in the Chapter 6.

CHAPTER VI

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we have studied a number of prime versus composite problems and explored some connections between them. We have also discovered some new structural properties of polynomials that might be of independent interest. The outstanding open problems in this area are proving lower bounds for $AC^0[m]$, and constructing explicit Ramsey graphs meeting the probabilistic bound. Both these are well-studied questions that have been open for the last twenty years, and are presumably hard problems. Below we present a list of problems that we think are more tractable, but at the same time might lead to some new insights regarding the harder problems above.

6.1 Resolving the Symmetry versus Asymmetry Question

The outstanding open problem regarding Boolean function representations modulo m is whether asymmetric polynomials can give better (i.e lower degree) representations of symmetric Boolean functions than symmetric polynomials. There are no known examples of symmetric Boolean functions where asymmetry does help, but there are no degree lower bounds better than $\Omega(\log n)$ known for any Boolean function. The question of whether there are lower degree weak representations of the OR function mod 6 has been open for a while. Our work on Ramsey constructions raises the question of whether low degree OR representations exist for our definition.

Better upper bounds would give better Ramsey graphs. Lower bounds for Prime representations will imply lower bounds for weak representations mod 6. Prime-power representations are exciting from the lower bound viewpoint since they have not been studied previously and might turn out to be easier to work with. It is interesting that prime-power representations do not invoke the Chinese Remainder Theorem. Interestingly, the $\Omega(\log n)$ lower bound of Barrington and Tardos [74] also does not invoke the CRT, hence it applies

to prime-power representations as well.

A conjecture attributed to Barrington, Beigel and Rudich in [74] is that $\Omega(\sqrt{n})$ is the right bound for weak representations modulo 6. We do not have a strong bias for or against this conjecture. It seems quite possible that much like explicit Ramsey constructions, the problem of constructing low degree asymmetric OR representations is a hard explicit construction problem. It is interesting to note that for the Ramsey graph problem, it is easy to construct a graph on 2^n vertices with $\alpha(G), \omega(G) \leq 2^{\frac{n}{2}}$. There was a conjecture due to Turan that this was in fact optimal, which was disproved by Erdős in 1947. However it was not until 1972 that a better explicit construction was found [10].

One possible reason for believing $\Omega(\sqrt{n})$ is the right lower bound is that it comes from Chinese remaindering. However, as we have seen prime-power representations do not use the CRT, similarly, there might be other low-degree asymmetric constructions that do not use Chinese remaindering. It is also tempting to think that symmetric polynomials ought to give the best representations of symmetric Boolean functions. However, our new definition of OR representations frames the problem as a covering problem for points on the hypercube. In this setting, choosing symmetric polynomials does not seem as natural.

The question of symmetry versus asymmetry is especially interesting for Threshold-k functions where $2 \leq k \leq n-1$, since here we have seen that resolving the degree for symmetric polynomials is equivalent to questions regarding Diophantine equations. These are rather hard questions in number theory regarding Diophantine equations and unconditional results seem out of the reach of current techniques. There is the tantalizing possibility that we run into hard number theoretic questions because we are restricted to symmetric polynomials and one can prove unconditional upper bounds with asymmetric polynomials.

6.2 Towards Better Degree Lower bounds

It is rather embarrassing that the best lower bounds for weak representations of any function is $\Omega(\log n)$. We believe that our lower bounds for symmetric polynomials representing the OR function suggest some algebraic problems that need to be solved in order to make progress on this front. Both our lower bounds for symmetric polynomials follow a similar

scheme: we characterize the zero-sets of low-degree symmetric polynomials and then show that there is no good partition of the hypercube. A natural question is whether such a scheme could extend to the general case. The first step would be to give a of characterization of zero-sets of low degree polynomials. Motivated by this we pose the following problems:

1. Given $S \subseteq \{0,1\}^n \setminus \mathbf{0}$, let $\deg_p(S)$ denote the smallest degree of a polynomial in $\mathbb{Z}_p[\mathbf{X}]$ which is 0 at every point in S but not at the origin. Give a lower bound on $\deg_p(S)$.

2. Given $S \subset \{0,1\}^n$, let $\deg'_p(S)$ denote the smallest degree of a polynomial in $\mathbb{Z}_p[\mathbf{X}]$ which is 0 over S but not at every point in $\{0,1\}^n$. Give a lower bound on $\deg'_p(S)$.

Note that both these quantities are easy to compute, since they involve checking whether a system of equations is feasible. We are looking for a combinatorial lower bound, perhaps analogous to Lemma 5.7. The latter quantity $\deg'_P(S)$ is closely related to the notion of *the degree of a subset* studied by Smolensky with a view towards proving circuit lower bounds [71]. The main difference is that he requires the zero-set to be exactly the set S.

6.3 Limitations to Distance-based Ramsey Constructions

We have shown that using symmetric polynomials in out construction, current techniques cannot give better bounds on $\alpha(G), \omega(G)$. Note that for the constructions of Alon, Frankl-Wilson and Grolmusz, this technique gives tight bounds.This raises the question: *do constructions based on symmetric polynomials contain either a large clique or independent set?*

Using a symmetric polynomial in our construction gives a graph where edges are added between vertices based on the Hamming distance between them. More formally, let $D \subset \{1, \cdots, n\}$. The graph $G(D)$ is defined as follows: The vertex set is $\{0,1\}^n$. We add $(\mathbf{x}, \mathbf{y})$ to E if $d(\mathbf{x}, \mathbf{y}) \in D$. Is it true that for every choice of D, $G(D)$ contains a large clique or independent set?

Similarly in Construction 5.2, symmetric polynomials give graphs where the vertices are sets and edges are added based on intersection sizes. Do such graphs always contain large cliques or independent sets?

6.4 *Tight Bounds for* MOD *functions*

Our $\Omega(n)$ lower bounds for weak protocols for Mod-k in both the two player and multi-player cases (Theorems 3.12 and 3.16) required r to be sufficiently large. For instance we are unable to prove a lower bound for Mod-2 over $\mathbb{Z}_{15}$. In [20], we had suggested that since the only cases for which upper bounds are known were $k = p_1^{a_1} \cdots p_t^{a_t}$ one should expect a lower bound of $\Omega(n)$ for all other k. Surprisingly, recent work due to Hansen shows that this is not the case [47]. He shows that Mod-2 requires degree $\Omega(n)$ over $\mathbb{Z}_{15}$, but he proves an upper bound of $O(\sqrt{n})$ for $\mathbb{Z}_{21}$. He obtains a fairly right characterization of the degree of the Mod-k function over $\mathbb{Z}_{pq}$, which roughly shows that non-trivial upper bounds are possible if p and q are sufficiently large compared to k. Obtaining tight bounds for the case when m has more than two prime factors is open.

The reason why this problem is interesting is that it sheds light on why obtaining lower bounds on $\mathrm{AC}^0[m]$ is hard. We know that for any odd number m, computing the Mod-m function for $\mathrm{AC}^0[2]$ requires circuits of size $2^{\Omega(\frac{1}{d})}$. One might expect the Mod-2 function to be as hard for $\mathrm{AC}^0[m]$ where m is odd. However Hansen's results imply that for any $\epsilon > 0$, by taking sufficiently large m, one can compute the Mod-2 function by $\mathrm{AC}^0[m]$ circuits of size 2^{n^ϵ} and depth 3. This suggests that simple functions like Mod-2 might be insufficient for the task of proving strong lower bounds on $\mathrm{AC}^0[m]$.

6.5 *Set-Systems with Restricted Intersections*

There are huge gaps in the known upper and lower bounds for set systems with restricted intersections modulo composites. For set systems with restricted intersections mod 6, Grolumusz constructs a systems of size $n^{\frac{\log n}{\log\log n}}$ on n elements. The only non-trivial upper bound known, due to Sgall is $2^{\frac{n}{2}}$ [57]. It would be interesting to close this gap. Grolmusz has shown that lower-degree weak representations modulo 6 would give larger set-systems. Is there a converse to this?

Similarly, in the prime-power case, there are gaps between upper and lower bounds on the size of set-systems with restricted intersections, as a function of the size s of the set of intersections modulo p^a. The upper bound due to Babai *et al.* [11] is $O(n^{2^{s-1}})$ while

the best known construction due to Kutin is bounded by n^{2s} [57]. The upper bound of Babai *et al.* proceeds by constructing certain separating polynomials modulo prime powers; it is possible that one might be able to use facts about interpolating sets to improve their construction.

APPENDIX A

EXTENSIONS TO PRIME POWERS

A.1 Symmetric Polynomials over Prime Powers

We first show that low degree polynomials depend on only a few bits of the base p representation of the weight. The proof uses Kummer's Theorem, a proof of which can be found in [37].

Theorem A.1 (Kummer's Theorem) *The largest power of p that divides $\binom{n}{k}$ equals the number of carries when k and $n-k$ are added in base p.*

Corollary A.2 *If $k < p^l$, $\binom{w}{k} \bmod p^a$ depends only on the first $l+a-1$ digits of w in base p.*

PROOF: This is equivalent to proving

$$\binom{w}{k} \equiv \binom{w+p^{\ell+a-1}}{k} \bmod p^a$$

Let $1 \leq j \leq k$. Then $j < p^\ell$ which implies $j_i = 0$ for $i \geq \ell$. When we add j and $(p^{\ell+a-1}-j)$ we get at least a carries. By Kummer's theorem,

$$\begin{aligned}
\binom{p^{\ell+a-1}}{j} &\equiv 0 \bmod p^a \quad 1 \leq j < p^\ell & (60)\\
\binom{w+p^{l+a-1}}{k} &= \sum_{j=0}^{k} \binom{w}{k-j}\binom{p^{l+a-1}}{j} \\
&\equiv \binom{w}{k} \bmod p^a \qquad \text{By Equation 60}
\end{aligned}$$

□

Corollary A.3 *Let $k < p^\ell$. Let $f : \{0,1\}^n \to \mathbb{Z}_{p^a}$ be computed by a symmetric polynomial $P(X)$ of degree k. Then f is a function of only the $\ell+a-1$ least significant digits of w in base p.*

We next show that a function depending on few lower order digits of the weight can be computed by a low degree polynomial. Recall that in Lemma 4.33, we constructed univariate polynomials $\Delta_c(X)$ for $0 \leq c \leq p-1$ with $\deg(\Delta_c) = (2a-1)(p-1)$ that satisfy the following condition

$$\begin{aligned} x \equiv c \bmod p &\Rightarrow \Delta_c(x) \equiv 1 \bmod p^a \\ x \not\equiv c \bmod p &\Rightarrow \Delta_c(x) \equiv 0 \bmod p^a \end{aligned}$$

Theorem A.4 *Let $f : \{0,1\}^n \to \mathbb{Z}_{p^a}$ be a symmetric function which depends only on the first ℓ digits of w in base p. Then f is computed by $P(X) \in \mathbb{Z}_{p^a}[X]$ where $deg(P) < 2p^\ell a$.*

PROOF: Let $\mathbf{x} \in \{0,1\}^n$ have weight w. By Lucas' Theorem, we have

$$S_{p^j}(\mathbf{x}) \equiv w_j \bmod p$$

Hence

$$\Delta_c(S_{p^j}(\mathbf{x})) \equiv \Delta_c(w_j) \equiv \begin{cases} 1 \bmod p^a & \text{if } w_j = c \bmod p \\ 0 \bmod p^a & \text{otherwise} \end{cases}$$

Similarly the polynomial

$$\prod_{j=0}^{\ell-1} \Delta_{c_j}(S_{p^j}(\mathbf{X}))$$

is 1 mod p^a only if $w_j = c_j$ for $j \leq \ell$.

Let $f(c_0, \cdots c_{\ell-1})$ denote the value of f when the first ℓ digits are set to $c_0, \cdots, c_{\ell-1}$ in base p. The desired polynomial is

$$P(\mathbf{X}) = \sum_{c_0, \cdots c_{\ell-1}} \left(f(c_o, \cdots c_{\ell-1}) \cdot \prod_{j=0}^{\ell-1} \Delta_{c_j}(S_{p^j}(\mathbf{X})) \right)$$

The degree of this polynomial is bounded by

$$\sum_{j=0}^{\ell-1} p^j \cdot (p-1)(2a-1) = \leq (2a-1)(p^\ell - 1)$$

□

A.2 Set Systems with restricted Intersections modulo Prime Powers

Definition A.1 *A set system $\mathcal{F} = \{S_i\}$ on $[n]$ is said to have restricted intersections mod q if there exists $L \subset \mathbb{Z}_q$ so that $|S_i| \bmod q \notin L$ but $|S_i \cap S_j| \bmod q \in L$.*

For a fixed modulus q, we study the problem of how large $|\mathcal{F}|$ can be as a function of n. When $q = p$ is a prime, the non-uniform modular Ray-Chaudhuri Wilson theorem proved by Deza, Frankl and Singhi [10] gives a bound of

$$|\mathcal{F}| \leq \binom{n}{\leq |L|} \leq \binom{n}{\leq p-1}$$

When q is not a prime power, Grolmusz shows a lower bound of $n^{\omega(1)}$ [43]. We give a near-tight bound of $\binom{n}{\leq p^a-1}$ for the prime power case. This improves the bound of $\binom{n}{\leq 2^{|L|-1}}$ due to Babai *et al.*[11]. Previously stronger bounds than ours were known for the special case when $|L| = p^a - 1$ i.e. when all set sizes are congruent to $k \mod p^a$ for some k (see theorems 5.30 and 7.18 in the book by Babai and Frankl [10]). To prove our result, we use the fact that every function from $\mathbb{Z}_{p^a}$ to $\mathbb{Z}_p$ can be written as a polynomial.

Theorem A.5 *Let $\mathcal{F}$ be a set system with restricted intersections modulo p^a. Then $|\mathcal{F}| \leq \binom{n}{\leq p^a-1}$.*

Proof: We construct a univariate integer-valued polynomial $P(X) \in \mathbb{Q}[X]$ of degree $p^a - 1$ such that

$$P(x) \equiv \begin{cases} 1 \bmod p & x \bmod p^a \in L \\ 0 \bmod p & x \bmod p^a \notin L \end{cases}$$

By Lucas' theorem,

$$\binom{x}{p^a-1} \equiv \begin{cases} 1 \bmod p & x \equiv p^a - 1 \bmod p^a \\ 0 \bmod p & x \not\equiv p^a - 1 \bmod p^a \end{cases}$$

$$\Rightarrow \binom{x-\ell+p^a-1}{p^a-1} \equiv \begin{cases} 1 \bmod p & x \equiv \ell \bmod p^a \\ 0 \bmod p & x \not\equiv \ell \bmod p^a \end{cases}$$

$$\text{Set} \quad P(X) = \sum_{\ell \in L} \binom{X - \ell + p^a - 1}{p^a - 1}$$

By Lemma 3.1, of [11] this implies the desired bound. We sketch the argument below.

We will use $\mathbf{S_i}$ to denote the incidence vector of set S_i. Let $P_i(X_1, \cdots, X_n) = P(\sum_{j \in S_i} X_j)$ and multi-linearize. It is easy to show that

$$P_i(\mathbf{S_j}) = P(|S_i \cap S_j|) \equiv \begin{cases} 1 \bmod p & i = j \\ 0 \bmod p & i \neq j \end{cases}$$

Using this one can show that the polynomials $P_i(\mathbf{X})$ are linearly independent over $\mathbb{Q}$. Since they are multilinear polynomials in n variables of degree $p^a - 1$, the bound follows. $\square$

REFERENCES

[1] Agrawal, M. and Biswas, S., "Primality and identity testing via Chinese Remaindering," *Journal of the ACM*, vol. 50(4), pp. 429–443, 2003.

[2] Agrawal, M., Kayal, N., and Saxena, N., "PRIMES is in P," *Annals of Mathematics*, vol. 160(2), pp. 781–793, 2004.

[3] Alon, N., "Ramsey graphs cannot be defined by real polynomials," *Journal of Graph Theory 14*, pp. 651–661, 1990.

[4] Alon, N., "Tools from higher algebra," in *Handbook of Combinatorics*, pp. 1749–1783, North-Holland, 1995.

[5] Alon, N., "The Shannon capacity of a union," *Combinatorica*, vol. 18(3), pp. 301–310, 1998.

[6] Alon, N. and Beigel, R., "Lower bounds for approximations by low degree polynomials over $\mathbb{Z}_m$," *Proceedings of the* 16^{th} *IEEE Conference on Computational Complexity (CCC'01)*, 2001.

[7] Alon, N., Pach, J., Pinchasi, R., Rados, R., and Sharir, M., "Crossing patterns of semi-algebraic sets," *Journal of Combinatorial Theory Ser A 111*, pp. 310–326, 2005.

[8] Arora, S., Lund, C., Motwani, R., Sudan, M., and Szegedy, M., "Proof verification and the hardness of approximation problems," *J. of the ACM*, vol. 45, no. 3, pp. 501–555, 1998.

[9] Arora, S. and Safra, S., "Probabilistic checking of proofs : A new characterization of NP," *J. of the ACM*, vol. 45, no. 1, pp. 70–122, 1998.

[10] Babai, L. and Frankl, P., "Linear algebra methods in combinatorics, preliminary version 2." University of Chicago, 1992.

[11] Babai, L., Frankl, P., Kutin, S., and Stefankovic, D., "Set systems with restricted intersections modulo prime powers," *Journal of Combinatorial Theory, Ser. A*, vol. 95(1), 2001.

[12] Barak, B., "A simple explicit construction of an $n^{\tilde{o}(\log n)}$ Ramsey graph," *arXiv.org, math.CO/0601651*, 2006.

[13] Barak, B., Rao, A., Shaltiel, R., and Wigderson, A., "2-source dispersers for $n^{o(1)}$ entropy and Ramsey graphs beating the Frankl-Wilson construction," in *Proceedings of the* 38^{th} *Symposium on Theory of Computing (STOC)*, 2006.

[14] Barrington, "Some problems involving Razborov-Smolensky polynomials," in *Boolean Function Complexity, M. Paterson (Ed.),*, London Mathematical Society Lecture Note Series 169, Cambridge University Press, 1992.

[15] Barrington, D. A., Beigel, R., and Rudich, S., "Representing Boolean functions as polynomials modulo composite numbers," *Computational Complexity*, vol. 4, pp. 367–382, 1994.

[16] Beigel, R., "Personal communication," 2003.

[17] Beigel, R. and Tarui, J., "On ACC," *Computational Complexity*, vol. 4, pp. 350–366, 1994.

[18] Bernstein, D. J., "Detecting perfect powers in essentially linear time," *Mathematics of Computation*, vol. 67, pp. 1253–1283, 1998.

[19] Bernstein, D. J., "Factoring into coprimes in essentially linear time," *Journal of Algorithms*, vol. 54, pp. 1–30, 2005.

[20] Bhatnagar, N., Gopalan, P., and Lipton, R. J., "Symmetric polynomials over $\mathbb{Z}_m$ and simultaneous communication protocols," in *Proceedings of the* 44^{th} *Annual Symposium on the Foundations of Computer Science (FOCS'03)*, pp. 451–459, 2003.

[21] Bhatnagar, N., Gopalan, P., and Lipton, R. J., "The degree of threshold mod 6 and Diophantine equations," Tech. Rep. ECCC TR04-022, Electronic Colloqium on Computational Complexity, 2004.

[22] Bhatnagar, N., Gopalan, P., and Lipton, R. J., "Symmetric polynomials over $\mathbb{Z}_m$ and simultaneous communication protocols," *Journal of Computer and System Sciences*, vol. 72, pp. 252–285, 2006.

[23] Bjorner, A. and Ziegler, G. M., "Introduction to greeoids," in *Matroid Applications*, pp. 284–348, 1992.

[24] Blum, A. and Langley, P., "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.

[25] Bshouty, N. H., Tamon, C., and Wilson, D. K., "Learning matrix functions over rings," in *European Conference on Computational Learning Theory*, pp. 27–37, 1997.

[26] Cahen, P.-J. and Chabert, J.-L., *Integer-Valued Polynomials.* AMS Mathematical Surveys and Monographs, Vol. 48, 1996.

[27] Chandra, A., Furst, M., and Lipton, R. J., "Multi-party protocols," in *Proceedings of the* 15^{th} *Annual ACM Symposium on Theory of Computing (STOC'83)*, pp. 94–99, 1983.

[28] Cohen, H., *A Course in Computational Algebraic Number Theory.* Graduate Texts in Mathematics 138, Springer-Verlag, 1993.

[29] Erdős, P., "Some remarks on the theory of graphs," *Bulletin of the A. M. S.*, vol. 53, pp. 292–294, 1947.

[30] Filaseta, M., "A generalization of an irreducibility theorem of I. Schur," *Acta Arithmetica*, vol. 58, no. 3, pp. 251–272, 1991.

[31] Frankl, P. and Wilson, R., "Intersection theorems with geometric consequences," *Combinatorica*, vol. 1, pp. 357–368, 1981.

[32] Frisch, S., "Polynomial functions on finite commutative rings," in *Advances in commutative ring theory (Fez, 1997)*, vol. 205 of *Lecture Notes in Pure and Appl. Math.*, pp. 323–336, New York: Dekker, 1999.

[33] Garey, M. and Johnson, D., *Computers and Intractability.* New York: Freeman, 1979.

[34] Goldreich, O., Rubinfeld, R., and Sudan, M., "Learning Polynomials with Queries: The Highly Noisy Case," *SIAM Journal on Discrete Mathematics, 13(4):535-570*, 2000.

[35] Gopalan, P., "Constructing Ramsey graphs from Boolean function representations," in *Proceedings of the* 21^{st} *IEEE Conference on Computational Complexity (CCC'06)*, 2006.

[36] Gopalan, P., "Query-efficient algorithms for polynomial interpolation over composites," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA'06)*, pp. 908–917, 2006.

[37] Granville, A., "Arithmetic properties of binomial coefficients," *Canadian Mathematical Society Conference Proceedings*, vol. 20, pp. 253–275, 1997.

[38] Granville, A., "*abc* means we can count squarefrees," *International Mathematical Research Notices*, vol. 19, pp. 1224–1231, November 1998.

[39] Granville, A. and Tucker, T. J., "It's as easy as *abc*," *Notices of the AMS*, vol. 49, no. 10, pp. 991–1009, 2002.

[40] Green, F., "Complex Fourier technique for lower bounds on the mod-m degree," *Computational Complexity*, vol. 9, pp. 16–38, 2000.

[41] Grolmusz, V., "On the weak mod m representation of Boolean functions," *Chicago Journal of Theoretical Computer Science*, vol. 2, 1995.

[42] Grolmusz, V., "Low rank co-diagonal matrices and Ramsey graphs," *Electronic Journal of Combinatorics*, vol. 7(1), 2000.

[43] Grolmusz, V., "Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs," *Combinatorica*, vol. 20, no. 1, pp. 71–86, 2000.

[44] Grolmusz, V., "Constructing set systems with prescribed intersection sizes," *Journal of Algorithms*, vol. 44, no. 2, pp. 321–337, 2002.

[45] Guruswami, V., *List Decoding of Error-Correcting Codes.* Springer, 2004.

[46] Guruswami, V. and Sudan, M., "Improved decoding of Reed-Solomon codes and algebraic-geometry codes," *IEEE Transactions on Information Theory*, vol. 45(6), pp. 1757–1767, 1999.

[47] Hansen, K. A., "On modular counting with polynomials," in *Proceedings of the* 21^{st} *IEEE Conference on Computational Complexity (CCC'06)*, 2006.

[48] Hardy, G. and Wright, E., *An Introduction to the Theory of Numbers.* Clarendon Press, Oxford, 1985.

[49] HÅSTAD, J., "Some optimal inapproximability results," in *Proceedings of the* 29^{th} *Annual ACM Symposium on Theory of Computing (STOC'97)*, pp. 1–10, 1997.

[50] KAHN, J., KALAI, G., and LINIAL, N., "The influence of variables on Boolean functions," in *Proceedings of the* 29^{th} *Annual Symposium on Foundations of Computer Science, (FOCS'88)*, pp. 68–80, 1988.

[51] KARPINSKI, M., VAN DER POORTEN, A., and SHPARLINSKI, I., "Zero testing of p-adic and modular polynomials," *Theoretical Computer Science*, vol. 233, no. 1-2, pp. 309–317, 2000.

[52] KEARNS, M. J. and VAZIRANI, U. V., *An Introduction to Computational Learning Theory.* MIT Press, 1994.

[53] KEMPNER, A., "Polynomials and their residue systems," *Transactions of the AMS*, vol. 22, no. 2, pp. 240–288, 1921.

[54] KLIVANS, A., O'DONNELL, R., and SERVEDIO, R., "Learning intersections and thresholds of halfspaces," in *Proceedings of the* 43^{rd} *Annual Symposium on Foundations of Computer Science (FOCS'02)*, pp. 177–186, 2002.

[55] KLIVANS, A. and SERVEDIO, R., "Learning DNF in time $2^{\tilde{O}(n^{1/3})}$," in *Proceedings of the* 33^{rd} *Annual Symposium on Theory of Computing (STOC'01)*, pp. 258–265, 2001.

[56] KUSHILEVITZ, E. and NISAN, N., *Communication Complexity.* Cambridge University Press, 1997.

[57] KUTIN, S., "Constructing large set systems with given intersection sizes modulo composite numbers," *Combinatorics, Probability and Computing*, vol. 11(5), 2002.

[58] LANG, S., *Algebra, 3rd ed.* Addison-Wesley, 1992.

[59] LINIAL, N., MANSOUR, Y., and NISAN, N., "Constant depth circuits, Fourier transform and learnability," *Journal of the ACM*, vol. 40, no. 3, pp. 607–620, 1993.

[60] MINSKY, M. and PAPERT, S., *Perceptrons: an Introduction to Computational Geometry.* MIT Press, 1968.

[61] MOSSEL, E., O'DONNELL, R., and SERVEDIO, R., "Learning juntas," in *Proceedings of the* 35^{th} *Annual ACM Symposium on the Theory of Computing (STOC'03).*, 2003.

[62] NAOR, M., "Constructing Ramsey graphs from small probability spaces," *Manuscript, available online*, 1993.

[63] NARKIEWICZ, W., *Polynomial Mappings.* Springer Lecture Notes in Mathematics, 1600, 1995.

[64] NISAN, N. and SZEGEDY, M., "On the degree of Boolean functions as real polynomials," in *Proceedings of the* 24^{th} *Annual ACM Symposium on the Theory of Computing (STOC'92)*, pp. 462–467, 1992.

[65] O'DONNELL, R., *Computational applications of noise sensitivity.* PhD thesis, MIT, 2003.

[66] RABIN, M. O., "Digitalized signatures and public-key functions as intractable as factorization," tech. rep., MIT Laboratory for Computer Science, 1979.

[67] RABIN, M. O., "Probabilistic algorithm for testing primality," *Journal of Number Theory*, vol. 12(1), pp. 128–138, 1980.

[68] RAZBOROV, A., "Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$," *Methematical Notes of the Academy of Science of the USSR*, vol. 41, pp. 333–338, 1987.

[69] SHAMIR, A., "On the generation of polynomials which are hard to factor," in *Proceedings of the* 25^{th} *Annual ACM Symposium on the Theory of Computing (STOC'93)*, pp. 796–804, 1993.

[70] SMOLENSKY, R., "Algebraic methods in the theory of lower bounds for Boolean circuit complexity," in *Proceedings of the* 19^{th} *Annual ACM Symposium on Theoretical Computer Science (STOC'87)*, pp. 77–82, 1987.

[71] SMOLENSKY, R., "On representations by low-degree polynomials," in *Proceedings of the* 34^{th} *Annual Symposium on Foundations of Computer Science (FOCS'93)*, 1993.

[72] SOLOVAY, R. M. and STRASSEN, V., "A fast Monte-Carlo test for primality," *SIAM Journal on Computing*, vol. 6(1), pp. 84–85, 1977.

[73] SPIRA, R., "Polynomial interpolation over commutative rings," *American Mathemtical Monthly*, vol. 75, no. 6, pp. 638–640, 1968.

[74] TARDOS, G. and BARRINGTON, D., "A lower bound on the mod 6 degree of the OR function," *Computational Complexity*, vol. 7, pp. 99–108, 1998.

[75] TARUI, J., "Probabilistic polynomials. ac^0 functions and the polynomial-time hierarchy," *Theoretical Computer Science*, vol. 113, pp. 167–183, 1993.

[76] TSAI, S.-C., "Lower bounds on representing Boolean functions as polynomials in $\mathbb{Z}_m$," *SIAM Journal of Discrete Mathematics*, vol. 9, pp. 55–62, 1996.

[77] VON ZUR GATHEN, J. and ROCHE, J., "Polynomials with two values," *Combinatorica*, vol. 17(3), pp. 345–362, 1997.

[78] YAO, A. C., "Some complexity questions related to distributive computing," *Proceedings of the 11th Annual ACM Symposium on Theory of Computation*, pp. 209–213, 1979.