On Hitting Sets for Special Depth-4 Circuits

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

Master of Technology

by

Pranav Bisht
Roll No. : 15111028

under the guidance of

Prof. Nitin Saxena

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur

June, 2017

Statement of Thesis Preparation

1. Thesis title: ...On Hitting Sets for Special Depth-4 Circuits

2. Degree for which the thesis is submitted: ..MTech.............................

3 Thesis Guide was referred to for preparing the thesis. ✓

4. Specifications regarding thesis format have been closely followed. ✓

5. The contents of the thesis have been organized based on the guidelines. ✓

6. The thesis has been prepared without resorting to plagiarism. ✓

7. All sources used have been cited appropriately. ✓

8 The thesis has not been submitted elsewhere for a degree. ✓

Pranav

(Signature of the student)

Name: PRANAV BISHT

Roll No.: 15111028

Department/~~IDP~~: CSE

CERTIFICATE

It is certified that the work contained in the thesis titled "On Hitting Sets for Special Depth-4 Circuits", by "Pranav Bisht", has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

NSaxena 21 Jun'17

Dr. Nitin Saxena
Department of Computer Science & Engineering
Indian Institute of Technology Kanpur

June, 2017

Abstract

We study the Polynomial Identity Testing (PIT) problem in this thesis. When the input polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d, is given in the form of an arithmetic circuit of size s, it asks for an efficient algorithm to test whether the polynomial is identically zero or not. By efficient we mean, an algorithm that makes use of only poly(s, n, d) many $\mathbb{F}$ operations. There are two versions of this problem. In a Blackbox PIT algorithm, we are allowed only to evaluate the circuit on polynomially many points from $\mathbb{F}^n$, and cannot 'look' inside the circuit. Whereas in a whitebox PIT algorithm, we have access to the internal gates of the circuit. Blackbox algorithms are more lucrative as they have interesting connections with circuit lower bounds.

We give an efficient blackbox PIT algorithm for the special class of diagonal depth 4 circuits, with top fan-in 3 and power gates having equal fan-in. We will motivate why studying and solving this restricted model is worthwhile in the quest for the general PIT problem. We use sparse PIT map to efficiently test whether $f_1^a + f_2^a = f_3^a$, where f_1, f_2 and f_3 are sparse polynomials. The algorithm we give is tailor made for this model, as it uses the polynomial analog of Fermat's Last Theorem which has a similar equation setup for integers. While attacking other more general instances of depth four circuits, we encountered and proved following structural results also, which might be useful.

We explain a new form of rank concentration measure called cone closure. We observe that a general polynomial under a random shift has a cone closed basis, and give a simple proof for the same using the derivative operator. This derivative operator restricts this proof for polynomials belonging to base fields with zero or large characteristic. For small characteristic fields, we give a new meaningful definition for cone closure, as the old definition fails in this regime. We make use of a clever transformation to prove cone closed basis here.

Lastly, for a given set of linearly independent polynomials, we question the linear independence of their positive powers. We get the answer by proving a new theorem which tells that the powers of these linearly independent polynomials will almost always be linearly independent. We give a quadratic upper bound on the number of exceptions. This property has a surprisingly elementary proof, making use of the Wronskians.

Acknowledgments

The first credit always goes to the teacher. I would take this opportunity to personally thank my supervisor Prof. Nitin Saxena for grooming me so professionally. This work was possible only with his insight and active participation. He sets the best example of how a disciplined, punctual lifestyle is conducive to research. I learnt so much from his courses. And a special mention to his fine sense of sarcasm, which will never cease to entertain me.

I would like to express my gratitude for my parents, who made me able to pursue a difficult line as research, with their quality support, education and culture.

I cannot thank my seniors enough. Sumanta was and will always be my goto man for my doubts. His grasp of the basics, and his clarity of exposition is amazing. I also thank Amit Da, for his guidance and the great width of knowledge he shares. I thank the Department of Computer Science and Engineering - the staff and the faculty members for granting students great facilities and academic flexibility. Thank you, Prof. Raghunath Tewari for your theory courses. I would also like to thank Neeraj Kayal, Michael A. Forbes, Rohit Gurjar and Arpita Korwar for their surveys and PhD Thesis, which were my main source of learning in the field of Polynomial Identity Testing. The precision and clarity of their writing was very helpful.

All the love for my dearest peers. Thank you Chandan, for being such a great friend. I would not have survived here without our lazy discussions. Thanks for Pink Floyd and pool. Thank you Ravi for being my partner in crime and defeating me in chess. Special thanks to my two theory brothers - Ashish Dwivedi and Kartik Kale. I like Ashish for his true love for mathematics, and his genuine, unadulterated nature. I hope we rock together in our PhD. Working together in the same area with Kartik was insanely helpful. I am simply in awe of his ability to understand so quickly, and doing complex proofs in his mind. I will miss your presence during my PhD. I wish you all the happiness in the coming future.

Pranav Bisht

Dedicated to

My father, whose simple and humble conduct of life inspires me.

The art of doing mathematics consists in finding that special case which contains all the germs of generality.

– David Hilbert

Contents

Chapter 1

Notations

Before we start, let us clear out certain terms and short hand notations, which we will be using repeatedly throughout this work.

Throughout this thesis $\mathbb{N}$ will denote the set of non-negative integers. And we will use the shorthand $[n]$ to mean the set $\{1, 2, \ldots, n\}$. We will not always write $x_1, x_2, \ldots x_n$, but use $\overline{x}$ to mean the same. For example, $\mathbb{F}[\overline{x}]$ means $\mathbb{F}[x_1, x_2, \ldots, x_n]$. A point or vector $\overline{\alpha} \in \mathbb{F}^n$ means $(\alpha_1, \alpha_2, \ldots, \alpha_n)$. We will say a vector $\bar{a} = (a_1, a_2, \cdots, a_n) \leq$ another vector $\bar{b}$ if $a_1 \leq b_1, a_2 \leq b_2, \cdots, a_n \leq b_n$. In addition to that, $\bar{e}!$ will be the short for $e_1!e_2! \cdots e_n!$, $\binom{\bar{a}}{\bar{b}}$ for $\binom{a_1}{b_1}\binom{a_2}{b_2}\cdots\binom{a_n}{b_n}$, and $F(\bar{x} + \bar{t})$ for $F(x_1 + t_1, x_2 + t_2, \ldots, x_n + t_n)$. Also the partial derivative $\partial_{\bar{t}^{\bar{f}}}$ will mean $\partial_{t_1^{f_1} t_2^{f_2} \cdots t_n^{f_n}}$.

Whenever we say polynomial, we mean a multivariate polynomial $\in \mathbb{F}[x_1, \ldots, x_n]$ unless specified otherwise. By default n is the number of variables, s is the size of arithmetic circuit computing it, and d is the total degree of the polynomial. A monomial $m_{\bar{e}} = \bar{x}^{\bar{e}}$ will mean $x_1^{e_1} x_2^{e_2} \cdots x_n^{e_n}$. Similarly, a polynomial $f = c_1 x_1^{e_{11}} x_2^{e_{12}} \ldots x_n^{e_{1n}} + \ldots + c_m x_1^{e_{m1}} x_2^{e_{m2}} \ldots x_n^{e_{mn}}$ can be written compactly as $f = \sum_{\bar{e}} c_{\bar{e}} \bar{x}^{\bar{e}}$. coeff$_f(\bar{x}^{\bar{e}})$ will be used to denote the coefficient of monomial $\bar{x}^{\bar{e}}$ in the polynomial f. We define degree of a monomial as the sum of exponents of each variable occurring in that monomial. Total degree of a polynomial is the maximum of all the monomial degrees. Individual degree of a variable (deg$_{x_i}$) in a polynomial is the maximum exponent of that variable (x_i) over all the monomials. Individual degree of a polynomial (ideg) is the maximum individual degree over all variables. More formally,

$$\begin{aligned}
\text{Total degree}(f) &= \text{max } \left\{ \sum_{i=1}^{n} e_i \mid \bar{e} \in \mathbb{N}^n \text{ and } \text{coeff}_f(\bar{x}^{\bar{e}}) \neq 0 \right\} \\
\text{deg}_{x_i}(f) &= \text{max } \{ e_i \mid \bar{e} \in \mathbb{N}^n \text{ and } \text{coeff}_f(\bar{x}^{\bar{e}}) \neq 0 \} \\
\text{ideg}(f) &= \text{max } \left\{ \text{deg}_{x_i}(f) \mid i \in [n] \right\}
\end{aligned}$$

Also note that by default, PIT stands for Polynomial Identity Testing, $\mathcal{H}$ for Hitting Set, gcd for greatest common divisor, W for Wronskian, and $\log x$ for $\log_2 x$, unless stated otherwise.

Chapter 2

Introduction

2.1 Complexity Theory

Today, everyone is aware of the power and usefulness of computers, which have touched almost every field out there. It has become an indispensable part of our daily life, as we delegate a number of our tasks to these machines. Computer science teaches, in a broad sense, the art of problem solving while exploiting the computational power of these machines. This is realized through algorithms, which simply speaking is a sequence of rules and logical operations, meant to solve a problem. But our machines have a limited memory, and we also want to be able to solve the problems efficiently. Thus, resources of time and space are critical in design of a good algorithm. In the field of Complexity Theory, we abstract out the nitty-gritty details, and question the very limits of computation. We try to separate the easy or time efficient problems, from the difficult or time consuming ones. But a thoughtful and informed person may object that running an algorithm on a supercomputer will be much more efficient than running the same algorithm on his personal computer. This is why we give a universal mathematical model known as Turing machine, and we analyze the number of steps Turing Machine takes in solving a problem, and classify the problems accordingly. The complexity class P comprises all the problems that have a known deterministic polynomial time algorithm, while the class NP is the collection of problems, for which given a proof, its correctness can be verified in time polynomial to the size of input. And the famous open problem whether $\mathsf{P} = \mathsf{NP}$? is still haunting the researchers in this field. We can say that this whole field is the output of different endeavours to tackle this problem.

There are other models of computation as well - like Probabilistic Turing Machines, Boolean circuits etc, which have their own resources based on which they classify problems. We also have problems of algebraic nature, and it makes sense to have an algebraic model defined for them. A natural algebraic object is polynomial, which we can use to model a number of algebraic problems. For example, the problem of checking whether a graph has a perfect matching, can also be put as whether the determinant polynomial of its Tutte matrix is zero or not. Here, intuitively we wish to measure the complexity of a polynomial. For example computing the polynomial $(x_1+1)(x_2+1)\cdots(x_n+1)$ as a function is easier than substituting the values in its fully expanded form which has 2^n monomials. Thus, representation of polynomial matters in determining its complexity.

2.2 Arithmetic Circuits

A very compact, natural and useful representation of polynomials is that of arithmetic circuits. For a multivariate polynomial $f \in \mathbb{F}[\overline{x}]$, it is formally defined as a directed acyclic graph, where we have input nodes as variables and constants of the field, output node(s) which computes the final polynomial f, and intermediate nodes are addition, and multiplication nodes. The edges of graph are labeled with field constants (by default 1).

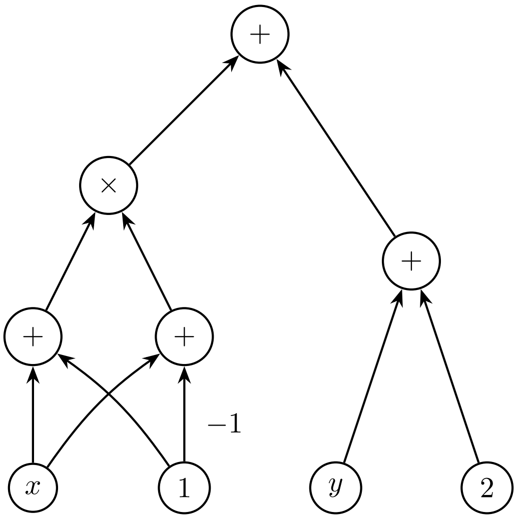

FIGURE 2.1: A circuit computing the polynomial x^2+y+1

We can define *size* of an arithmetic circuit as the number of edges in the graph. Some researchers like to define it as number of nodes in the graph, but we will follow the former convention as it is more accurate in some situations. We define *depth* of circuit as the number of edges in the longest path from a leaf node to output node. The nodes are called gates of the circuit. In-degree of a gate is termed as fan-in, and out degree as fan-out. An arithmetic circuit where each gate has fan-out $= 1$ is called an *arithmetic formula*. The degree of a gate is the total degree of the polynomial computed by that gate and *degree* of a circuit is the maximal degree of a gate, in the circuit. It is important to note that degree of circuit may be more than the degree of the polynomial computed by the circuit, as a sum gate may lead to cancellations of the highest total degree monomials in the inputs of that gate.

2.3 Arithmetic Complexity

In the model of arithmetic circuits, the two main resources are size and depth. Based on size, we define class VP as the family of circuits $\{\mathcal{C}_n\}$ computing polynomials such that number of variables, degree and size of the circuit is polynomially bounded in n. Observe that monomial x^{2^n}, can be computed by $O(n)$ sized circuit, but it is not in VP as the degree is exponential in n. This degree restriction is motivated for making this class relatable to boolean classes. This class is arithmetic analog of P. Similarly, we have the class VNP as the arithmetic analog of NP, which we will not define here. Based on depth, we have classes of constant depth circuits like depth-2, depth-3 and depth-4 circuits. Interested readers may read in detail about the definition, motivation and interrelation of these classes in [SY10], [For14].

It should also be noted that the arithmetic circuit model is a non-uniform model, unlike the Turing Machine model which is uniform as the transition function of a Turing machine is independent of the size of input. There is a single Turing Machine for all input sizes. But in the arithmetic model, we have different circuits for different number of variables. Thus, one can cheat with non-uniformity to even compute undecidable language using a constant sized circuit. Therefore, classes in this setting may not be directly relatable to boolean world. Nonetheless, we have interesting lower bound problems here.

2.4 Polynomial Identity Testing

This is the primary topic of this thesis work. Polynomial Identity Testing, PIT in short, is simply testing whether a given multivariate polynomial is zero or not. This statement is quite ambiguous, as it can be interpreted in various ways. One interpretation is that the problem statement asks whether a polynomial $f \in \mathbb{F}[\overline{x}]$ is zero when evaluated at some point $\overline{\alpha} \in \mathbb{F}^n$ or decide if there is no such point? The search version for this interpretation would be to find a point where the input polynomial is zero, or output no if there is no such point. This is the commonly known, root finding problem. This interpretation is not the concern of this thesis, rather we are interested in knowing whether polynomial is always (identically) zero or not.

The definition of PIT, which we follow in this thesis is that, given an input polynomial, determine whether coefficients of the all the monomials are zero or is there some monomial in the input polynomial which has a non-zero coefficient? This question may rather seem absurd, if the input polynomial is given in the form of a list of coefficients along with the corresponding list of monomials (which is the case in traditional polynomial factorization questions), as one needs to only check whether the list of coefficients has a non-zero entry or not. But the question becomes interesting when the input polynomial is given in the form of an arithmetic circuit. If we take the same example of polynomial $f = (x_1+1)(x_2+1)\cdots(x_n+1)$, we can see that it has a simple circuit and efficient PIT algorithm. But if we follow the brute force approach of simply expanding out the polynomial, and checking if there is some non-zero coefficient, then it will take exponential time, as it has exponential number of monomials. So, the question of PIT is that given a polynomial in the form of circuit, test the zeroness of polynomial in time polynomial in the size of circuit. We will define it more formally in Chapter 3.

It is also important here to remark that, an identically non-zero polynomial over a finite field, may evaluate to zero at all the points in the base field, yet as per this definition we will consider the polynomial to be non-zero. For example, $x^2 + x$ is a zero function over $\mathbb{Z}_2$, but still a syntactically non-zero polynomial as the coefficients are non-zero. For infinite sized fields though, a polynomial evaluates to zero $\forall \overline{\alpha} \in \mathbb{F}$ if and only if it is identically zero. This follows via simple inductive argument, and the fact that a non-zero univariate polynomial of degree d over a field has at most d roots. This problem to determine whether a given polynomial evaluates to zero on all the points of the base field, is also known as Evaluates to Zero Everywhere (EZE) problem and it is infact coNP-hard, as one can easily show a reduction from $\overline{\text{SAT}}$ to EZE over $\mathbb{F}_2$. Simply convert a boolean formula by replacing $\bar{x}$ with $1-x$, $x \wedge y$ with $x \cdot y$, and $x \vee y$ with $x + y + xy$. Now, the formula is unsatisfiable if and only if the corresponding polynomial evaluates to zero everywhere. On the contrary, PIT is in coRP due to Schwartz-Zippel Lemma [Sch80]. coRP is a subclass of BPP which is conjectured to be same as P. And the whole endeavour of this area is to derandomize PIT so as to put it in P.

There are two versions of PIT problem - whitebox and blackbox.

- **Whitebox PIT:** In this setting, we are given the polynomial as an arithmetic circuit, and we have full access to all the gates inside the circuit. For example the circuit class $\prod\sum$, has an easy whitebox PIT algorithm. This class computes polynomials of the form $\prod_{i=1}^{k} l_i$, where $l_i's$ are linear polynomials. For example $f = (x_1+1)(x_2+1)\cdots(x_n+1)$, belongs to this class. Since fields are by definition integral domains, testing zeroness of $\prod_{i=1}^{k} l_i$ reduces to testing zeroness of each l_i, which can be done in time $O(k\cdot$ max fan-in of sum gate), which is polynomial in size of the input circuit.

- **Blackbox PIT:** Here, we are given the polynomial as a blackbox arithmetic circuit. That means, we are not allowed to look inside the circuit, but we only have oracle access to the circuit, to evaluate it at any field point. The goal is to give a PIT algorithm in time polynomial in the circuit size, while using the circuit only for evaluations. Now blackbox PIT for $\prod\sum$ circuit class will not be as trivial as the whitebox PIT algorithm we described above. Whitebox PIT looks easier than the blackbox setting, but till now for almost all the models where we have a whitebox PIT algorithm, we also have a blackbox algorithm, with little worse parameters in few cases. Blackbox PIT is equivalent to the concept of hitting sets. Hitting sets is a collection of field points which suffice to test zeroness of a circuit class. They are designed in a way that any non-zero polynomial must evaluate to a non-zero value on at least one point in the hitting set, and if a circuit evaluates to zero on all the points in the hitting set, then we mark it a zero polynomial. Note that different non-zero polynomials of the same class may hit a non-zero value on different points in the hitting set.

2.5 Applications of PIT

- Testing equivalence of two polynomials f, g can also be posed as a PIT question of checking zeroness of the polynomial $f - g$. Many researchers find such change of representation more useful, as they have now to deal only with a single polynomial.

- As mentioned earlier also, the problem of deciding existence of perfect matching in a graph efficiently can be seen as a question of finding efficient PIT algorithm for the determinant polynomial of the graph's *Tutte matrix*. The $n \times n$ Tutte matrix A of a graph $G = (V, E)$ with n vertices is defined as:

  $$A_{i,j} = \begin{cases} x_{i,j} & \text{if } (i,j) \in E \text{ and } i < j \\ -x_{j,i} & \text{if } (i,j) \in E \text{ and } i > j \\ 0 & \text{otherwise} \end{cases}$$

  Tutte proved in 1947 that the multivariate determinant polynomial of matrix A is identically non-zero if and only if there exists a perfect matching in G. This immediately gives a fast RNC randomized parallel algorithm, since PIT has a randomized algorithm, and computing determinant has known fast parallel algorithms. Mulmuley et al. [MVV87] showed using their famous isolation lemma that even the search version of finding a perfect matching is in RNC. It is still open to completely derandomize isolation lemma. Recently, Rohit Gurjar (almost) derandomized Isolation Lemma for the case of planar bipartite graphs, which puts the problem in quasiNC for such graphs.

- We know primality testing was one of the first problems that demonstrated the power of randomized algorithms. The first deterministic primality testing algorithm was the celebrated AKS Primality Testing [AKS04], which finally put the problem in P. It was solved by formulating the problem as a PIT question. It was observed that the univariate polynomial $f = (x+1)^n - (x^n + 1)$ is identically zero if and only if n is prime (converse is evident from Frobenius endomorphism). In AKS, the authors were able to give a poly($\log n$) time polynomial identity testing algorithm by testing zeroness of f modulo few $O(\log n)$ degree polynomials.

- PIT also has interesting connections with lower bounds in arithmetic circuit complexity. [KI03] proved that PIT $\in$ P $\Rightarrow$ NEXP $\not\subseteq$ P$_{/poly}$ or VNP $\neq$ VP. [AV08] showed that solving blackbox PIT (or hitting set) for a circuit class gives a hard polynomial for that class. There is almost a converse also, that is finding a hard function for the circuit class VP immediately gives a quasi polynomial hitting set (or blackbox PIT) for that class. So, both the approaches are almost equivalent.

2.6 Our results

Chapter 4 gives polynomial time hitting set for the circuit class $\sum^3 \bigwedge \sum \prod$, which is the diagonal depth four model with top fan-in three, and power gates having equal fan-ins. This uses the polynomial analog of Fermat's last theorem, which is a direct corollary from Mason Stother's Theorem, which itself is the polynomial analog of abc conjecture. We prove that sparse PIT algorithm which works for $\sum \prod$ circuit class also works in this model. We also show proof of Mason's Theorem [Sto81, Mas84] for the sake of completeness.

In Chapter 5, we first explain and motivate the concept of cone closure after applying shift to a polynomial. Then, for the case of characteristic zero fields, we give a very simple proof of the presence of cone closure in the least basis of coefficient space obtained after applying a random shift to the variables of any polynomial. Then, we extend the result for characteristic p fields.

In Chapter 6, we will observe an intriguing property about the linear independence of powers of polynomials. We will prove that if a set of k polynomials is just pairwise linearly independent, then their constant powers will be almost always mutually linearly independent, except for at most $\binom{k-1}{2}$ exceptions. This is an interesting structural property we came across, and which has a very elementary proof using the tool of Wronskians.

Chapter 3

Background

In this chapter, we briefly discuss few concepts, lemmas, and theorems which shall be used in the results of subsequent chapters. Let us start with formal definitions of PIT and hitting sets.

3.1 Formal Definitions

Definition 3.1 (Polynomial Identity Testing [For14]). *Let $\mathcal{C}$ be a class of circuits having size $\leq s$, which compute polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ of degree $< d$. The PIT problem for this class $\mathcal{C}$ asks for a deterministic algorithm to test whether a polynomial f_C, computed by a circuit $C \in \mathcal{C}$, is identically zero or not. The algorithm is considered efficient if it uses only poly(s,n,d) $\mathbb{F}$ operations.*

If the algorithm uses the input circuit C, to only evaluate it at points in $\mathbb{F}^n$, then it is called a blackbox algorithm, and whitebox when it also considers the internal gates of C.

Definition 3.2 (Hitting Set [For14]). *Let $\mathcal{C}$ be a class of circuits having size $\leq s$, which compute polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ of degree $< d$. A hitting set $\mathcal{H} \subseteq \mathbb{F}^n$ for the circuit class $\mathcal{C}$ is a set of points such that if a circuit $C \in \mathcal{C}$ computes a non-zero polynomial f_C, then $\exists \overline{\alpha} \in \mathcal{H}$ such that $f(\overline{\alpha}) \neq 0$. The converse is trivial, that is, if C computes a zero polynomial f_C, then $f_C(\overline{\alpha}) = 0, \forall \overline{\alpha} \in \mathcal{H}$.*

Thus, giving a poly(s, n, d) sized hitting set $\mathcal{H}$ for a circuit class $\mathcal{C}$, gives an efficient blackbox PIT for $\mathcal{C}$. For an input circuit $C \in \mathcal{C}$ computing a polynomial f_C, we just evaluate f_C on all the points in $\mathcal{H}$. If there exists, a point $\overline{\alpha} \in \mathcal{H}$ such that $f_C(\overline{\alpha}) \neq 0$, then we output NON-ZERO, otherwise ZERO. Note that each evaluation takes at most $O(s)$ $\mathbb{F}$ operations. Thus, in total, we have a poly(s, n, d) time blackbox algorithm. (In PIT setting, we assume that each $\mathbb{F}$ addition or multiplication takes unit time).

For univariate polynomials, a simple blackbox PIT algorithm is to pick any distinct $d+1$ points from $\mathbb{F}$, where d is the degree of polynomial. If the polynomial evaluates to 0, on all the points, then it is zero, otherwise non-zero. And a trivial whitebox algorithm for univariate polynomials would be to simply expand out the polynomial, and check each coefficient. This can be done efficiently since maximum number of monomials is $d+1$. However, these trivial algorithms fail for multivariate polynomials. For example, a simple multivariate polynomial like $x - y$ has infinite set of roots S $= \{\dots, (-1, -1), (0, 0), (1, 1), \dots\}$. This means that we do not have a trivial *deterministic* blackbox algorithm. As for the whitebox case, recall the fact that a n-variate degree d polynomial has at most $\binom{n+d}{d}$ many monomials which is quite large for big values of n, d and thus simply expanding out the polynomial won't also work.

Now, we will show that PIT has an efficient randomized algorithm, due to famous Schwartz Zippel Lemma.

Lemma 3.3 (Schwartz-Zippel Lemma [Sch80]). *Let $f \in \mathbb{F}[x_1, \dots, x_n]$ be a non-zero polynomial of total degree $d \geq 0$. Let S be any finite subset of $\mathbb{F}$, and let $\alpha_1, \alpha_2, \dots, \alpha_n$ be elements selected independently, uniformly and randomly from S. Then*

$$Pr_{\alpha_1,\dots,\alpha_n \in S}[f(\alpha_1, \dots, \alpha_n) = 0] \leq \frac{d}{|S|}$$

This lemma has a simple inductive proof, where the base case $n = 1$ is true from the fact that a univariate polynomial of degree d has at most d roots. And the lemma generalizes this fact for multivariate polynomials. This immediately gives the following randomized blackbox algorithm which puts PIT in coRP: Given a circuit C computing a polynomial f_C. Pick any arbitrary set S of size $> d+1$. Pick a random $(\alpha_1, \dots, \alpha_n) \in S^n$. If $f_C(\alpha_1, \dots, \alpha_n) = 0$, then simply output ZERO else output NON-ZERO. In other words, PIT is asking for an efficient derandomization of Schwartz Zippel Lemma. A trivial derandomization is to check $(d+1)^n$ many points, but that is inefficient for large values of n, d. This derandomization is formally stated in the following lemma by N. Alon known as the combinatorial nullstellensatz.

Lemma 3.4 (Combinatorial Nullstellensatz [AT99]). *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero polynomial of individual degree d. Let S be a set of distinct field values of size $> d$. Then, there exists a point $\overline{\alpha} \in S^n$ such that $f(\overline{\alpha}) \neq 0$.*

3.2 Sparse PIT

We start with the simplest model first - $\sum\prod$. A circuit C of $\sum\prod$ class computes a polynomial f_C, which is simply sum of monomials, where the number of monomials is bounded by size of circuit C, say s. We call such a polynomial f_C, a sparse polynomial. The whitebox PIT for this class is trivial, since we simply collect coefficients of at most s monomials. The Blackbox PIT is not so straightforward. But, as we will see, the PIT for this model is very important and fundamental in this field.

Kronecker Map: Let f be a polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ with maximum individual degree d, of sparsity m computed by a circuit C of size s . Then consider the polynomial $g = f(y, y^d, y^{d^2}, \ldots, y^{d^{n-1}}) \in \mathbb{F}[y]$, where we substitute variable $x_i \rightarrow y^{d^{i-1}}$. Observe that a monomial $m_{\bar{e}} = c \cdot x_1^{e_1} x_2^{e_2} \ldots x_n^{e_n}$ will map to $m'_{\bar{e}} = c \cdot y^{e_1 + e_2 d + \ldots + e_n d^{n-1}}$. This is same as viewing the exponent vector $\bar{e} = (e_n, e_{n-1}, \ldots, e_1)$ in d-ary representation $(e_n, e_{n-1}, \ldots, e_1)_d$. Therefore, two different monomials (two different exponent vectors) will map to two different exponents of y, since each $e_i < d$ and d-ary representation is unique. Also, the map leaves the constant term untouched, and is thus oblivious of constants. Hence, PIT for a multivariate sparse polynomial f reduces to PIT for a univariate polynomial g. But, note that the power to reduce number of variables comes at the cost of increasing the degree. After the Kronecker substitution of f, the maximum possible degree of g can be of the order d^n. Therefore for PIT, evaluating g at deg$(g) + 1$ many points is inefficient. But fortunately, we do not have to evaluate at so many points as elicited by the following theorem.

Theorem 3.5 (Sparse PIT [Agr05, Sax09]). *Let f be a non-zero polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ with maximum individual degree d, and sparsity m. Then there exists $1 \leq r \leq (mn \log d)^2$ such that, $f(y, y^d, y^{d^2}, \ldots, y^{d^{n-1}}) \neq 0($mod $y^r - 1)$.*

This immediately gives a blackbox algorithm which runs in time poly(s, m). For each $r \in [(mn \log d)^2]$, we compute $d, d^2, \ldots, d^{n-1}$ mod r using repeated squaring and evaluate the circuit at $C(y, y^d, \ldots, y^{d^{n-1}})$mod $(y^r - 1)$. We declare f to be ZERO if and only if all these evaluations are zero.

Another noteworthy point is that, sparse PIT map ϕ can be seen as a variable map for whichever value of $r \in [(mn \log d)^2]$ works out from Theorem 3.5. This makes ϕ a ring homomorphism from $\mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[y]$, and thus it preserves polynomial addition and multiplication. Sparse PIT also has a plethora of other algorithms and proofs, for which the interested reader is referred to [SY10, KS01, BHLV09]. The $\prod \sum$ class has an even simpler PIT algorithm, and we leave it for the readers to verify it themselves.

3.3 Depth Reduction

Next, we consider the depth-3 circuit class $\prod \sum \prod$, which computes polynomials of the type $f = f_1 f_2 \ldots f_k$, where each f_i is a sparse polynomial. Note that such a polynomial f can be zero if and only if each f_i is zero. Thus, the PIT for this class reduces to PIT for $\sum \prod$ class, which is the sparse PIT itself.

Now, consider the depth-3 circuit class $\sum \prod \sum$, which computes polynomials of the type $f = \sum_{i=1}^{k} \prod_{j=1}^{d_i} l_{ij}$, where each l_{ij} is a linear polynomial in $\mathbb{F}[x_1, \ldots, x_n]$. PIT for depth-3 model is still open, but it has been solved in the restricted cases: like depth-3 with constant top fan-in has a polynomial time PIT algorithm ([KS09, SS12, SS13]). Set multilinear depth-3 and depth-4 models have quasi-polynomial time PIT algorithms ([ASS13, FSS14, AGKS13]). Multilinear depth-3 PIT is still open.

This gives us insight that depth-3 and depth-4 models are very general and capture the complexity of PIT for general arithmetic circuits. This intuition is not misplaced, as was proved by the depth-3 and depth-4 chasm results in [VSBR83, AV08, GKKS13]. Most recently, it has been proved in [AFGS17] that even PIT for tiny diagonal depth four circuit class $(\sum^k \bigwedge^a \sum \prod^b)$ in time poly$(s, 2^{O(n+b)}, \mu(a))$ implies quasi-polynomial PIT for VP circuit class, where $n = O(\log s)$, $b = O(\log s)$, and a is arbitrary small non-constant. This motivates Chapter 4 of our thesis, in which we give polynomial time PIT algorithm for $\sum^3 \bigwedge^a \sum \prod$ circuit class, which is diagonal depth 4 circuit with top fan-in 3 and where powering gates have equal fan-in a. $\bigwedge$ gates are simply $\prod$ gates where all inputs to the gate is a single input polynomial, variable or constant.

3.4 Shifts and concentration

In the past two decades, a lot of algebraic tools and techniques have been devised to better understand the complexity of polynomials computed by some class of circuits. Xi Chen et al. in their survey [CKW11] explain how **partial derivatives** can be employed to give lower bounds and PIT for few circuit classes.

Another technique is that of **faithful morphisms** where we devise a homomorphism $\phi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[y_1, \ldots, y_k]$ that preserves a certain algebraic property of the polynomials that belong to a certain class. The property to be preserved in the image space should be such that it suffices to give an efficient hitting set for the polynomials of that class. For example, sparse PIT map $\phi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[y]$ in Theorem 3.5 preserves the non-equality of two monomials of a sparse polynomial. Similarly, it can be shown that an efficient homomorphism ϕ that preserves gcd of two sparse polynomials, will yield hitting set for the circuit class $\sum^2 \prod \sum \prod$, which is the depth four model with top fan-in 2. Blackbox PIT for this model is still open while the whitebox PIT for the tiny version of this model can be seen in [Kal17]. Use of faithful morphisms in giving PIT algorithms can be studied in [SS12, BMS13, ASSS12].

The last and more important technique in context of this thesis is that of **rank concentration**. In [FSS14, ASS13, AGKS13] it has been shown that after applying a shift to a polynomial computed by a ROABP (read once arithmetic branching program which also subsumes the class of constant depth set multilinear circuits), the shifted polynomial obtains the following property: The rank of its coefficients viewed as $\mathbb{F}$-vectors is concentrated in low support monomials. We will briefly discuss low support concentration but the interested reader is refered to [For14, Gur16, Kor16] for detailed explanation on ROABPs and low support concentration. We will also discuss about a new form of rank concentration called cone size concentration, and its implications.

Low support concentration: Consider a polynomial $f \in \mathbb{F}^k[\overline{x}]$, where the coefficients are from a k dimensional vector space. By low support concentration in polynomial f, we mean that the coefficients of the low support monomials span the whole coefficient space of the polynomial. Coefficient space is simply the span of the coefficients of the monomials of f. Therefore a $< l$ support concentrated polynomial f is non-zero if and only if there is at least one monomial of support $< l$ which has a non-zero coefficient. Therefore for such cases, PIT reduces to PIT of low support monomials. We formally define l-concentration now:

Definition 3.6 (l-concentration [Kor16]). *The polynomial $f \in \mathbb{F}^k[\overline{x}]$ is l-concentrated if rank$_{\mathbb{F}}$ $\{$coeff$_f$ $\bar{x}^{\bar{e}} \mid \bar{e} \in \mathbb{N}^n$, supp$(\bar{e}) < l\,\}$ = rank$_{\mathbb{F}}$ $\{$coeff$_f$ $\bar{x}^{\bar{e}} \mid \bar{e} \in \mathbb{N}^n\,\}$.*

If a polynomial is l-concentrated, then its PIT reduces to PIT for a l variate polynomial. Since maximum number of monomials is $\binom{l+d}{l}$ which is $O(d^l)$. Therefore PIT of f can be done in polynomial time if l is constant and quasi-polynomial time if $l = O(\log s)$. In [AGKS15], the authors have shown $O(\log s)$ support concentration after a shift, thus giving a quasi polynomial time hitting set for ROABPs and sum of constantly many set multilinear circuits.

Mostly, the polynomial by itself is not low support concentrated, but becomes one after a suitable efficiently designed shift. As a simple example, consider the polynomial $f = x_1x_2 \ldots x_n$, which is not $< n$-concentrated, but becomes $< 1-$concentrated after a simple shift $x_i \to x_i + 1$. The new polynomial $f' = (x_1 + 1) \ldots (x_n + 1)$ has a non-zero constant term 1. Also, note that a general polynomial f is zero if and only if the shifted polynomial f' is zero, since shift is an invertible operation. Therefore PIT for f is equivalent to PIT for f'. Shifts are also used in factorization algorithms. Next, we talk about a new form of rank concentration called, cone size concentration.

Cone size concentration: In the tiny models regime, the arity of polynomials is restricted to $O(\log s)$, where s is the size of circuits. Note that a polynomial f with this much arity will have at most quasi polynomial number of monomials, because $\binom{n+d}{d} = s^{O(\log s)}$, for $n = O(\log s)$ and $d = O(s)$. Therefore, we need a strictly polynomial sized hitting set for such models, because quasi-polynomial is already trivial. This calls for a better measure than $< \log s$ support concentration. Cone size $\leq k$-concentration is one such measure. We start with basic definitions.

A monomial $m_{\bar{a}} = c_a\, x_1^{a1} x_2^{a2} \ldots x_n^{a_n}$ belongs in the cone of another monomial $m_{\bar{b}} = c_b\, x_1^{b_1} x_2^{b_2} \ldots x_n^{b_n}$ if $\bar{x}^{\bar{a}}$ divides $\bar{x}^{\bar{b}}$, or equivalently $\bar{a} \leq \bar{b}$, where $\leq$ is component wise. We define **cone** for a monomial or equivalently for its exponent vector $\bar{e}$ as:

$$cone(\bar{e}) = \{\, \bar{f} \in \mathbb{Z}^n \mid \bar{0} \leq \bar{f} \leq \bar{e}\,\}$$

For example $cone(x^d) = \{\,1, x, x^2, \cdots, x^d\,\}$, and $cone(x^2y) = \{\,1, x, x^2, y, xy, x^2y\,\}$. And now we define **cone size** of a monomial as simply the number of monomials which divide it, that is the number of monomials in its cone.

$$cs(\bar{e}) = |cone(\bar{e})| = \prod_{i\in[n]}(e_i + 1)$$

Lemma 3.7 (cs concentration ⇒ Hitting set [AFGS17])**.** *The number of monomials of maximum arity n, and cone size $\leq k$ is $O(2^n \cdot k^2)$.*

Proof. First, let us suppose the monomials have a fixed arity n, and we wish to upper bound the cardinality of set $\mathcal{I} = \{m \mid m \text{ has arity } n \text{ and cone size}(m) \leq k\}$. Let $T(n,k)$ denote the cardinality of $\mathcal{I}$. Observe that a monomial $x_1^{e_1}x_2^{e_2}\dots x_n^{e_n}$ of arity n has cone size $\leq k$ if $(e_1+1)(e_2+1)\dots(e_n+1) \leq k$. Therefore,

$$T(n,k) = \left|\left\{(e_1,\dots,e_n) \mid \prod_{i=1}^{n}(e_i+1) \leq k \text{ and } \forall i, e_i > 0\right\}\right|$$
$$T(n,k) \leq T\left(n-1,\frac{k}{2}\right) + T\left(n-1,\frac{k}{3}\right) + \dots + T\left(n-1,1\right)$$

This recurrence relation has a simple inductive proof. Letting $T(n,k) \leq k^2$ works because,

$$T(n,k) \leq \frac{k^2}{2^2} + \frac{k^2}{3^2} + \dots + 1$$
$$T(n,k) \leq k^2\left(-1 + \sum_{i=1}^{\infty}\frac{1}{i^2}\right)$$
$$T(n,k) \leq k^2\left(\frac{\pi^2}{6} - 1\right)$$
$$T(n,k) \leq k^2$$

This fixes $|\mathcal{I}| \leq k^2$, where the monomials had arity exactly n. Now let $\mathcal{J}$ be the set of all monomials of arbitrary arity $\leq n$ with cone size $\leq k$. $\mathcal{J} = \{m \mid \text{cone size}(m) \leq k\}$. This means $|\mathcal{J}| \leq 2^n \cdot k^2$. □

For, arity $O(\log s)$ polynomials, this translates to poly(s) sparsity. Thus, if we are able to show rank concentration in the form of cone size $\leq$ poly(s)-concentration in such polynomials, the question of PIT reduces to PIT of cone size $\leq$ poly(s) monomials which further reduces to sparse PIT from the arguments in above lemma.

All this is good in theory, but one may ask, is cone size concentration even achievable? Our result in Chapter 5 shows that on applying a random shift to a general polynomial $f \in \mathbb{F}^k[\overline{x}]$, we achieve cone size $\leq k$ concentration, which is strictly better than the low support concentration. For the purposes of PIT, we need to construct a deterministic, efficiently computable shift by exploiting the inherent structure of a model. The usefulness of this measure has come to light in the recent result of [AFGS17], where the authors achieve cone size concentration in diagonal depth 3 model by applying a basis isolating weight assignment shift, thus giving blackbox PIT algorithm of complexity $sd2^{O(n)}$, which is polynomial for $\log s$ number of variables.

Chapter 4

Special Diagonal Depth-4 PIT

4.1 Introduction

A polynomial f computed by a $\sum^3 \bigwedge^a \sum \prod$ circuit of size s is of the form $f = f_1^a + f_2^a + f_3^a$, where f_1, f_2, f_3 have sparsity less than s, and powering gates have equal fan-in a. So, the problem of identity testing of polynomial f asks to check whether $f_1^a + f_2^a = f_3^a$ in blackbox and in time polynomial in input circuit size s. Most of the PIT algorithms have a common trend. Design a map ϕ, which reduces the number of variables (to constant, in most cases). Thus, the simple blackbox PIT algorithm is to simply apply variable map ϕ, which significantly reduces the complexity of polynomial and find a hitting set for it. For example, in our case we will apply sparse PIT to make it a univariate polynomial of polynomial degree and test on degree + 1 many points. All the effort in PIT goes mainly in proving why the map ϕ works. That is, to prove $C = 0 \iff \phi(C) = 0$.

Originally, we were attacking two variations of depth four model. The first one is the tiny depth four model $\sum^k \prod^a \sum \prod^b$ which computes polynomials of the form $f = \prod_{i=1}^{a_1} f_{1i} + \cdots + \prod_{i=1}^{a_k} f_{ki}$ with added restrictions that total number of variables is $O(\log s)$, and fan-in of bottom $\prod$ gate is $b = O(\log s)$, (in other words total degree of each f_{ji} is bounded by $O(\log s)$). The second variation we studied was the tiny diagonal depth four circuit model $\sum^k \bigwedge^a \sum \prod^b$ which is same as tiny depth four model with its restrictions, the only difference being the powering gate instead of top product gate. We note that tiny diagonal depth four circuit reduces to tiny depth four circuit. More precisely $\sum^{2k} \bigwedge^a \sum \prod^b$ reduces to $\sum^k \prod^a \sum \prod^b$, since $f_1^a + f_2^a + f_3^a + f_4^a +$

$\cdots + f_{2k-1}^a + f_{2k}^a = \prod_{i=1}^a (f_1 + \omega_1 f_2) + \prod_{i=1}^a (f_3 + \omega_2 f_4) + \cdots + \prod_{i=1}^a (f_{2k-1} + \omega_k f_{2k})$, where $\omega_1, \cdots, \omega_k$ are some a^{th} roots of -1 (not necessarily distinct).

Our result does not assume tiny restrictions in the diagonal depth four model, but it is has two other restrictions — top fan-in 3, and equal fan-ins of powering gates. Also, it requires that characteristic of the base field $\mathbb{F}$ does not divide a. In the subsequent sections, we show an easy hitting set of $\sum^2 \bigwedge^a \sum \prod$, followed by our main result of hitting set for $\sum^3 \bigwedge^a \sum \prod$. Kartik Kale in his MTech thesis [Kal17] gives the whitebox PIT algorithm for tiny $\sum^2 \prod^a \sum \prod^b$, which also covers only the whitebox PIT for tiny $\sum^4 \bigwedge^a \sum \prod^b$, by the reduction argument discussed above.

For the $\sum^3 \bigwedge^a \sum \prod$ case, we make use of Mason Stother's theorem to show that the variable reduction map of sparse PIT from Theorem 3.5 also works for this case. A direct corollary of Mason's theorem is the polynomial analog of Fermat's Last Theorem. We observed that the PIT problem for this case looked similar to FLT, and successfully used that FLT theorem to show that a linear independence preserving variable map suffices. Since, the sparse PIT map preserves linear independence of a constant number of sparse polynomials, it suffices for this model. The detailed proof can be verified in Section 4.3. Our main result is stated formally in the following theorem:

Theorem 4.1. *Let $f \in \mathbb{F}[\overline{x}]$ be a polynomial computed by a $\sum^3 \bigwedge^a \sum \prod$ circuit of size s, such that $f = f_1^a + f_2^a + f_3^a$, where f_i's are of sparsity $O(s)$. Then, there is poly(s) sized hitting set for f.*

4.2 Top fan-in 2 diagonal depth 4

First, we warm up with the simple problem of finding hitting set for the class of $\sum^2 \bigwedge^a \sum \prod$ circuits. One can skip this portion, and jump to Section 4.3 which proves our main theorem, which is quite independent of the discussion here.

Lemma 4.2. *Let $f \in \mathbb{F}[\overline{x}]$ be a polynomial computed by a $\sum^2 \bigwedge^a \sum \prod$ of size s, such that $f = f_1^a + f_2^a$, where f_1, f_2 are of sparsity $O(s)$. Then, there is poly(s) sized hitting set for f.*

Proof. Testing $f = 0$ can be rephrased as testing equivalence of polynomials f_1^a, f_2^a. Note that though f_1, f_2 are sparse, f_1^a, f_2^a may not be sparse. Hence, sparse PIT map cannot be trivially used. But we show that nevertheless, sparse PIT map will work, which follows from the simple observation that $f_1^a = f_2^a$ if and only if $f_1 = \omega f_2$, where ω is some a^{th} root of unity. Note that if the base field does not contain primitive roots of unity, then we are in a simpler case, where $f_1^a = f_2^a$ if and only if $f_1 = f_2$. In other words f_1 must be proportional or linearly dependent on f_2, for their a^{th} powers to be equal. Since, the sparse PIT map preserves proportionality, it suffices.

More formally, let $\omega_1, \cdots, \omega_a$ be the a^{th} roots of unity. Then, if I can find a point $\overline{\alpha}$ such that $\forall i \in [a], f_1 - \omega_i f_2(\overline{\alpha}) \neq 0$, then $f_1^a + f_2^a(\overline{\alpha}) \neq 0$. Let f_1, f_2 be of sparsity s_1, s_2 respectively, and let $\mathcal{H}$ be the blackbox hitting set for $f_1 - \omega_i f_2$, $\forall i \in [a]$, obtained in poly(s) time using sparse PIT map for the class of polynomials of sparsity $s_1 + s_2 = O(s)$. Then the same hitting set $\mathcal{H}$ will work for $f_1^a + f_2^a$ because if $f_1^a + f_2^a \neq 0$, then $\forall \omega_i$, $f_1 - \omega_i f_2 \neq 0 \Rightarrow$ we can find hitting set $\mathcal{H}$ in poly(s) time such that $\exists \overline{\alpha}$ and $f_1 - \omega_i f_2 \neq 0$, $\forall i \in [a] \Rightarrow f_1^a + f_2^a(\overline{\alpha}) \neq 0$. For the converse side, if $f_1^a + f_2^a = 0$, then $\forall \overline{\alpha} \in \mathcal{H}$, $f_1^a + f_2^a = 0$ trivially. Note that the hitting set obtained by sparse PIT map for the class of circuits of sparsity $s_1 + s_2$ is special for us, in the sense that $\exists$ a single $\overline{\alpha} \in \mathcal{H}$ such that $f_1 - \omega_i f_2 \neq 0$ for all ω_i. This is because sparse PIT map is only a variable map which is oblivious of constants (and does not look inside the circuit), hence changing the constant ω_i will not affect the PIT map. □

4.3 Top fan-in 3 diagonal depth-4

In this section, we give proof of our main theorem, Theorem 4.1. First, let us state the corollary of Mason Stother's theorem which we will prove later in Section 4.4, but will use it directly here in our proof.

Corollary 4.3 (FLT polynomial analog [Lan02]). *Let $a(t), b(t), c(t) \in \mathbb{F}[t]$ be three co-prime polynomials. If $a(t)^n + b(t)^n = c(t)^n$ with $n > 2$, and if characteristic of base field $\mathbb{F}$ does not divide n, then $a(t), b(t), c(t)$ are all constant polynomials.*

First, let us directly state the simple PIT algorithm for this model, which is like every other model. All the effort is in proving its correctness, which we will discuss in detail afterwards.

Algorithm: Given input a $\sum^3 \bigwedge^a \sum \prod$ circuit computing polynomial f of the form, $f = f_1^a + f_2^a + f_3^a$. Apply sparse PIT map Φ on f. Then $f = 0 \iff \Phi(f) = 0$. Check whether $\Phi(f) = 0$ in polynomial time. Output ZERO, if $\Phi(f) = 0$, and NON-ZERO otherwise.

Proof of Theorem 4.1. If $f = 0$, then $\Phi(f) = 0$ trivially, since Φ is only a variable map. Now, we need to prove that if $\Phi(f) = 0$, then $f = 0$. As a preprocessing step, assume $a \geq 3$, because for $a = 1$, or 2, f itself will be sparse, and hence sparsePIT map Φ will work. Also, for the sake of clarity, let $f = f_1^a + f_2^a - f_3^a$, so that $f = 0 \iff f_1^a + f_2^a = f_3^a$. This can be done safely as negative sign can be absorbed inside the polynomial f_3. Now, we prove the converse. Assume that $\Phi(f) = 0$. Then, since Φ is a ring homomorphism, it is additive and multiplicative. Thus,

$$\begin{aligned}\Phi(f) &= 0\\ \Phi(f_1^a + f_2^a) &= \Phi(f_3^a)\\ \Phi(f_1)^a + \Phi(f_2)^a &= \Phi(f_3)^a \qquad (4.1)\end{aligned}$$

Now, we can use Corollary 4.3 since $\Phi(f_1), \Phi(f_2), \Phi(f_3)$ are univariate polynomials, and $a > 2$. But it may be the case that $\Phi(f_1), \Phi(f_2), \Phi(f_3)$ are not co-prime which is a condition required to apply Corollary 4.3. Suppose that they are not co-prime, say $\Phi(f_1)$ and $\Phi(f_2)$ share a common factor, then the equality dictates that $\Phi(f_3)$ must also share the same factor. Let $g = gcd(\Phi(f_1), \Phi(f_2), \Phi(f_3))$. Then divide equation Equation (4.1) by g^a on both sides, to get

$$g_1^a + g_2^a = g_3^a, \text{ where } g_i = \frac{\Phi(f_i)}{g} \qquad (4.2)$$

Now since, g_1, g_2, g_3 are co-prime we can use Corollary 4.3 to deduce that all of g_1, g_2, g_3 are constants. This means,

$$\begin{aligned}\Phi(f_1) = g \cdot g_1 &\Rightarrow f_1 = h_1 \cdot g_1, \text{ where } \Phi(h_1) = g\\ \Phi(f_2) = g \cdot g_2 &\Rightarrow f_2 = h_2 \cdot g_2, \text{ where } \Phi(h_2) = g\\ \Phi(f_3) = g \cdot g_3 &\Rightarrow f_3 = h_3 \cdot g_3, \text{ where } \Phi(h_3) = g\end{aligned}$$

The above equations hold since Φ is $\mathbb{F}$-algebra homomorphism, in particular Φ(constant $\cdot$ polynomial) = constant $\cdot$ Φ(polynomial). Note that $\Phi(h_1) = \Phi(h_2) = \Phi(h_3) \Rightarrow \Phi(h_1 - h_2) = \Phi(h_2 - h_3) = 0$. Since $h_1 - h_2$ and $h_2 - h_3$ are also sparse, sparse PIT map Φ gives that $\Phi(h_1 - h_2) = 0 \Rightarrow h_1 - h_2 = 0 \Rightarrow h_1 = h_2$. Similarly,

$h_2 = h_3 = h_1 = h$ (say). This means

$$f = f_1^a + f_2^a - f_3^a$$
$$f = h^a \cdot (g_1^a + g_2^a - g_3^a)$$
$$f = 0 \text{ [From Equation (4.2)]}$$

Thus $\Phi(f) = 0 \Rightarrow f = 0$. □

Note that we can construct Φ in poly(s) time which separates every pair of monomials from the set which contains all the different monomials of sparse polynomials h_1, h_2, h_3 together. Thus, a single map Φ for $h_1 + h_2 + h_3$ will preserve zeroness or non-zeroness of $h_1 - h_2$ and $h_2 - h_3$. Thus instead of preserving non-zeroness of f_1^a, f_2^a, f_3^a which would have been costly, we are just preserving non-zeroness of h_1, h_2, h_3 or equivalently f_1, f_2, f_3 which suffices for this model.

4.4 Mason Stothers Theorem

For the sake of completeness, in this section, we will give the proof of Mason Stothers Theorem and the consequent Corollary 4.3. Blackbox PIT for $\sum^3 \bigwedge^a \sum \prod$ has already been covered in Section 4.3. The theorem was reportedly discovered first by W. Wilson Stothers in 1981, and rediscovered by R.C. Mason few years later. Mason Stother Theorem is the proved polynomial version of famous ABC conjecture in number theory, which is open till date.

Theorem 4.4 (Mason's Theorem [Sto81, Mas84]). *Let $a(t)$, $b(t)$, $c(t) \in \mathbb{F}[t]$ be relatively prime polynomials over field $\mathbb{F}$ such that $a + b = c$ and not all of them have vanishing derivative. Then,*

$$\mathit{max}\ \{\mathit{deg}(a), \mathit{deg}(b), \mathit{deg}(c)\} \leq \mathit{deg}(\mathit{rad}(abc)) - 1$$

where rad(f) is the product of all distinct irreducibles of polynomial f (analogous to radical of an integer).

We will cover Noah Snyder's [Sny00] proof version of this theorem. First, consider this lemma, which we shall use in the proof.

Lemma 4.5. *[Sny00] Let f be a non-zero polynomial in $\mathbb{K}[x]$. Then, deg(gcd(f, f')) $\geq$ deg(f) − deg(rad(f)).*

Proof. Let $f = g_1^{e_1} g_2^{e_2} \cdots g_k^{e_k}$, where g_i's are irreducibles. Then deg$(f) = e_1d_1 + e_2d_2 + \cdots + e_kd_k$, where d_i is degree of g_i. Note that rad$(f) = g_1 \cdots g_k$, and deg$(rad(f)) = d_1 + \cdots + d_k$. Now, taking formal derivative of f, we get

$$f' = \sum_{i=1}^{k} e_i g_i^{e_i - 1} g_i' \cdot g_1^{e_1} \cdots g_{i-1}^{e_{i-1}} \cdot g_{i+1}^{e_{i+1}} \cdots g_k^{e_k}$$

This implies that

$$\begin{aligned} g_1^{e_1-1} g_2^{e_2-1} \cdots g_k^{e_k-1} &\mid \text{gcd}(f, f') \\ \text{deg}(\text{gcd}(f, f')) &\geq (e_1 - 1)d_1 + \cdots + (e_k - 1)d_k \\ &= e_1d_1 + \cdots + e_kd_k - (d_1 + \cdots + d_k) \\ &= \text{deg}(f) - \text{deg}(rad(f)) \end{aligned}$$

□

Now, we are ready to prove Theorem 4.4.

Proof. Hypothesis gives,

$$a + b = c \tag{4.3}$$

$$a' + b' = c' \tag{4.4}$$

Multiply equation Equation (4.3) by a' and subtracting it from Equation (4.4) multiplied by a, we get $ab' - a'b = ac' - a'c$. Similarly, we get $ab' - a'b = cb' - c'b = ac' - a'c$. Let $W = W(a, b) = W(a, c) = W(c, b)$ denote the value of these equal two dimensional Wronskians. We claim that $W \neq 0$ since , if it were zero then $ab' = a'b \Rightarrow a \mid a'$, since a, b are co-prime $\Rightarrow a' = 0$. Similarly, then $b' = 0, c' = 0$ which contradicts our hypothesis that at least one of a, b, c has a non-vanishing derivative. Hence, the Wronskian W is non-zero. Also, observe that gcd(a, a'), gcd(b, b'), gcd(c, c') all divide W, and since these gcd's are all co-prime, this implies that

$$\begin{aligned} \text{gcd}(a, a') \cdot \text{gcd}(b, b') \cdot \text{gcd}(c, c') &\mid W \\ \text{deg}(\text{gcd}(a, a')) + \text{deg}(\text{gcd}(b, b')) + \text{deg}(\text{gcd}(c, c')) &\leq \text{deg}(W) \qquad (4.5) \end{aligned}$$

Lemma 4.5 then gives the following:

$$\text{deg}(a) - \text{deg}(\text{rad}(a)) \leq \text{deg}(\text{gcd}(a, a')) \qquad (4.6)$$

$$\text{deg}(b) - \text{deg}(\text{rad}(b)) \leq \text{deg}(\text{gcd}(b, b')) \qquad (4.7)$$

$$\text{deg}(c) - \text{deg}(\text{rad}(c)) \leq \text{deg}(\text{gcd}(c, c')) \qquad (4.8)$$

We know that deg(rad(abc)) $=$ deg(rad(a)) $+$ deg(rad(b)) $+$ deg(rad(c)), since a, b, c are co-prime. Then, equations Equation (4.6), Equation (4.7), Equation (4.8) together with Equation (4.5) implies

$$\text{deg}(a) + \text{deg}(b) + \text{deg}(c) - \text{deg}(\text{rad}(abc)) \leq \text{deg}(W) \qquad (4.9)$$

Since, $W = ab' - a'b$, we have that deg$(W) \leq$ deg(a) $+$ deg$(b) - 1$. Therefore, Equation (4.9) then implies

$$\begin{aligned} \text{deg}(a) + \text{deg}(b) + \text{deg}(c) - \text{deg}(\text{rad}(abc)) &\leq \text{deg}(a) + \text{deg}(b) - 1 \\ \text{deg}(c) &\leq \text{deg}(\text{rad}(abc)) - 1 \end{aligned}$$

Similarly,

$$\begin{aligned} \text{deg}(b) &\leq \text{deg}(\text{rad}(abc)) - 1 \\ \text{deg}(a) &\leq \text{deg}(\text{rad}(abc)) - 1 \end{aligned}$$

This proves that $max\{deg(a), deg(b), deg(c)\} \leq deg(rad(abc)) - 1$ □

Finally, now we prove Corollary 4.3 which was the main tool we used in our PIT proof. We restate it here for clarity.

Corollary 4.3 (FLT polynomial analog [Lan02]). *Let $a(t), b(t), c(t) \in \mathbb{F}[t]$ be three co-prime polynomials. If $a(t)^n + b(t)^n = c(t)^n$ with $n > 2$, and if characteristic of base field $\mathbb{F}$ does not divide n, then $a(t), b(t), c(t)$ are all constant polynomials.*

Proof. We simply apply Mason's Theorem [Sto81, Mas84] to the co-prime polynomials $a(t)^n$, $b(t)^n$, $c(t)^n$. Since the characteristic of $\mathbb{F}$ does not divide n, the polynomials $a(t)^n, b(t)^n, c(t)^n$ have vanishing derivative only when $a(t), b(t), c(t)$ have vanishing derivative, in other words they are all constant. If they are not all constant, then the equality is impossible to achieve as shown ahead. By Mason's Theorem [Sto81, Mas84], we get

$$\begin{aligned} \max\{\deg(a^n), \deg(b)^n, \deg(c^n)\} &\leq \deg(\mathrm{rad}(a^n b^n c^n)) - 1 \\ n \cdot \max\{\deg(a), \deg(b), \deg(c)\} &\leq \deg(\mathrm{rad}(abc)) - 1 \end{aligned}$$

Assume without loss of generality that $deg(a)$ is maximum, then

$$\begin{aligned} n \cdot \deg(a) &\leq \deg(a) + \deg(b) + \deg(c) - 1 \\ 3 \cdot \deg(a) &\leq \deg(a) + \deg(b) + \deg(c) - 1 \quad [\text{ since } n \geq 3 \,] \\ \deg(a) + \deg(b) + \deg(c) &\leq \deg(a) + \deg(b) + \deg(c) - 1 \\ 0 &\leq -1 \end{aligned}$$

which is a contradiction. This completes the proof of our corollary. Note that the equality can hold in case of field constants. This does not contradict Fermat's Last Theorem for integers, which requires a, b, c to be integers. For example in $\mathbb{C}[x]$, $1^3 + 2^3 = \omega^3$, where ω is cube-root of 9. We have infinite such examples in $\mathbb{C}[x]$. □

4.5 Conclusion and Future work

This section described blackbox PIT algorithms for diagonal depth four models with top fan-in 2 and 3 with the additional restriction that powering gates have equal fan-in. The first step would be to remove the equal fan-in restriction. But more importantly, we wish to make the top fan-in flexible. First, solving for constant k, itself seems very non-trivial. After that, solving for a general k, would imply solving PIT for the class of VP circuits (a quasi-polynomial time algorithm), because of the depth four chasm result due to [AV08]. But more recently, it has been shown in [AFGS17] that even solving PIT for the so called tiny diagonal depth four model, namely $\sum^k \bigwedge^a \sum \prod^b$ circuits in time poly$(s, 2^{O(n+b)}, \mu(a))$ suffices to give quasiP PIT algorithm for VP circuits, where μ is some function. We also have depth-3 chasm result, so one may question the need of working on depth four models. It is simply because, depth four model gives us more special instances to work on, and we hope that the tools and insights used in solving these very special cases, give us a better insight for finding the solution to the more general and difficult PIT models. The ultimate destination of every researcher working in the field of PIT is, of course, to solve the problem for the class of VP circuits.

Chapter 5

Cone Closure

5.1 Introduction

In this chapter, we explain the concept of cone closure and its form of occurrence in a shifted polynomial. Here, we give only a structural result and not any PIT algorithm, as we use a random shift. Though, we do explain its connection with cone size concentration and hence with PIT. Derandomizing the shift in an efficient way will yield a true PIT algorithm, which was recently done for diagonal depth-3 circuits in [AFGS17]. That result requires 0 or large characteristic fields. But here we complete the notion of cone closure and prove its existence even for the small characteristic fields. But the traditional definition fails for such fields, and thus, we require a different definition of cone for such fields, which makes the resulting PIT less efficient.

This chapter is notation heavy, and the reader is advised to look up Chapter 1 before proceeding. First we talk about 0 or large characteristic fields. Recall the definitions of cone, cone size and the connection of cone size with hitting sets from Section 3.4. Here, we give an even stronger notion of cone closed basis. We say that a set S of monomials is **cone-closed** if for every monomial $m \in S$, its cone is also present in S. More formally,

$$\forall \bar{e} \in S,\ cone(\bar{e}) \subseteq S$$

For example the set $S_1 = \{1, x, y, y^2, xy, xy^2, z, xz\}$ is cone closed, while set $S_2 = \{1, x, y, xy, z, xz, xyz\}$ is not, since $yz \notin S_2$ but $yz \in cone(xyz)$.

5.2 Coefficients in a randomly shifted polynomial

Before we discuss the result for general multivariate polynomials, let us first consider univariate polynomials to develop intuition for cone closure. Let $F(x) \in \mathbb{F}^k[x]$ be a univariate polynomial of degree d. Let us shift the variable x by a formal variable t. The notion of a shift by a random point and shift by a formal variable is equivalent. Let $F(x+t)$ be the shifted polynomial over $\mathbb{F}(t)[x]$, where we include formal variable t in the base field, and treat it as a constant.

$$F(x) = \sum_{i=0}^{d} z_i x^i$$
$$F(x+t) = \sum_{i=0}^{d} z_i'(t) x^i$$

where $z_i'(t) \in \mathbb{F}(t)$ (field of fractions of $\mathbb{F}[t]$). Derive the following relation between the coefficients of $F(x)$ and $F(x+t)$ by applying binomial theorem.

$$\begin{bmatrix} 0!z_0' & 1!z_1' & \cdots & d!z_d' \end{bmatrix} = \begin{bmatrix} 0!z_0 & 1!z_1 & \cdots & d!z_d \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{t}{1!} & 1 & 0 & \cdots & 0 \\ \frac{t^2}{2!} & \frac{t}{1!} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{t^d}{d!} & \frac{t^{d-1}}{(d-1)!} & \frac{t^{d-2}}{(d-2)!} & \cdots & 1 \end{bmatrix}$$
$$\partial_t(i!z_i') = (i+1)!z_{i+1}', \forall 0 \le i \le d-1$$
$$\partial_t(z_i') = (i+1)z_{i+1}'$$

This has a strong implication. If $z_0' = 0$, then $z_i' = 0, \forall i \in [d]$. This means that if for a polynomial $F(x) \in \mathbb{F}^k[x]$, the constant term (x-free term) in the shifted polynomial $F(x+t) = 0$ then the shifted polynomial is identically zero (since all the coefficients are 0), which means $F(x)$ is identically zero. Again, the catch here is that the shift by formal variable t means the constant term is itself a univariate polynomial in t, infact of degree d again. But, the good thing is that the rank of the coefficient space has become concentrated in the first term z_0'. In other words, we achieve cone size $= 1$ concentration. Therefore, PIT through this route is still randomized, but may be derandomized more easily with this approach. Nonetheless, it is still an intriguing property, which can also be seen from the Taylor series of F at point t.

Now, we proceed to the main multivariate case. Let us shift a general polynomial randomly, and try to find relation among its coefficients. Let $F(\bar{x}) \in (\mathbb{F}^k)[\bar{x}]$ be a non-zero polynomial with individual degree d. Let $F'(\bar{x}+\bar{t})$ be the shifted polynomial where $x_i \to x_i + t_i$. Assume $ch(\mathbb{F}) = 0$.

$$\begin{aligned} \text{Let } F(\bar{x}) &= \sum_{\bar{0}\le\bar{e}\le\bar{d}} z_{\bar{e}} \cdot \bar{x}^{\bar{e}}, \text{ where } z_{\bar{e}} \text{ are coefficients } \in \mathbb{F}^k \\ F'(\bar{x}+\bar{t}) &= \sum_{0\le\bar{e}\le\bar{d}} z'_{\bar{e}} \cdot \bar{x}^{\bar{e}}, \text{ where } z'_{\bar{e}} \in \mathbb{F}^k(\bar{t}) \end{aligned}$$

Again by applying binomial expansion and collecting coefficients, we get:

$$\bar{e}! \cdot z'_{\bar{e}} = \sum_{\bar{e}\le\bar{f}\le\bar{d}} (\bar{f}! \cdot z_{\bar{f}}) \cdot \frac{\bar{t}^{\bar{f}-\bar{e}}}{(\bar{f}-\bar{e})!}$$

This implies that:

$$\begin{aligned} z'_{\bar{e}} &= \left(\frac{\partial_{\bar{x}^{\bar{e}}}}{\bar{e}!} \cdot F'\right)\Bigg|_{\bar{x}=\bar{0}} \\ &= \left(\frac{\partial_{\bar{x}^{\bar{e}}}}{\bar{e}!} \cdot F\right)\Bigg|_{\bar{x}=\bar{t}} \\ \frac{\partial_{\bar{t}^{\bar{f}}}}{\bar{f}!} \cdot (z'_{\bar{e}}) &= \left(\frac{\partial_{\bar{x}^{\bar{e}+\bar{f}}}}{\bar{e}!\bar{f}!} \cdot F\right)\Bigg|_{\bar{x}=\bar{t}} \\ &= \frac{(\bar{e}+\bar{f})!}{\bar{e}!\bar{f}!} \cdot z'_{\bar{e}+\bar{f}} \end{aligned}$$

$$\frac{\partial_{\bar{t}^{\bar{f}}}}{\bar{f}!} \cdot (z'_{\bar{e}}) = \binom{\bar{e}+\bar{f}}{\bar{e}} \cdot z'_{\bar{e}+\bar{f}} \tag{5.1}$$

This means that the higher degree coefficient can be obtained by applying ∂ operator on lower degree coefficient (which is in its cone). In other words two coefficients in the shifted polynomial are related if there is a sub-monomial relationship between them. Now, we use this result to prove an interesting theorem in the next section.

5.3 Cone closed basis

First, define a valid monomial ordering, say the simple lexical ordering $(1 < x_1 < x_2 < \cdots < x_n)$. We can consider a least basis of coefficients B in the shifted polynomial based on this ordering (over $\mathbb{F}[\bar{t}]$. Just apply the greedy approach. First consider the constant term coefficient, put it in the basis, then consider coefficient of next term as per the lexical ordering, check if it is linearly independent on the coefficients already in basis constructed till now, if not include it in the basis, otherwise not. Then consider the immediate next term and so on.

Theorem 5.1. *Let $F(\bar{x}) \in (\mathbb{F}^k)[\bar{x}]$ be a non-zero polynomial. Let $F'(\bar{x} + \bar{t})$ have the least basis B over $\mathbb{F}(\bar{t})$, with respect to a monomial ordering, say lexical ordering $(x_1 < x_2 < \cdots < x_n)$. Then B is cone-closed.*

Proof. Assume for the purpose of contradiction that B is not cone-closed. Then by definition of cone-closure $\exists z'_{\bar{e}} \in B$ such that $z'_{\bar{e}'} \notin B$ for some $\bar{e}' < \bar{e}$ (strictly $<$ in at least one coordinate). This implies

$$z'_{\bar{e}'} \in \left\langle z'_{\bar{f}} \mid \bar{f} <_{lex} \bar{e}' \right\rangle_{\mathbb{F}(\bar{t})}$$

Note that $<_{lex}$ is based on the monomial ordering. It does not imply that $\bar{f} \in cone(\bar{e}')$.

$$\begin{aligned}
\partial_{\bar{t}^{\bar{e}-\bar{e}'}}\left(z'_{\bar{e}'}\right) &\in \left\langle z'_{\bar{f}} \mid \bar{f} <_{lex} \bar{e}' \right\rangle_{\mathbb{F}(\bar{t})} + \left\langle \partial_{\bar{t}^{\bar{e}-\bar{e}'}}\left(z'_{\bar{f}}\right) \mid \bar{f} <_{lex} \bar{e}' \right\rangle_{\mathbb{F}(\bar{t})} \\
\partial_{\bar{t}^{\bar{e}-\bar{e}'}}\left(z'_{\bar{e}'}\right) &\in \left\langle z'_{\bar{f}} \mid \bar{f} <_{lex} \bar{e}' \right\rangle_{\mathbb{F}(\bar{t})} + \left\langle z'_{\bar{f}+\bar{e}-\bar{e}'} \mid \bar{f} <_{lex} \bar{e}' \right\rangle_{\mathbb{F}(\bar{t})}, \text{ using Equation (5.1)} \\
\partial_{\bar{t}^{\bar{e}-\bar{e}'}}\left(z'_{\bar{e}'}\right) &\in \langle z'_{\bar{u}} \mid \bar{u} <_{lex} \bar{e} \rangle_{\mathbb{F}(\bar{t})} \left(\text{Since } \bar{f} <_{lex} \bar{e}, \bar{f} + \bar{e} - \bar{e}' <_{lex} \bar{e}\right) \\
z'_{\bar{e}} &\in \langle z'_{\bar{u}} \mid \bar{u} <_{lex} \bar{e} \rangle_{\mathbb{F}(\bar{t})}, \text{ using Equation (5.1)}
\end{aligned}$$

This means $z'_{\bar{e}}$ is linearly dependent on lexically strictly smaller coefficients. By the way basis is constructed this implies $z'_{\bar{e}} \notin B$ which is a contradiction to our initial assumption. Hence, B is cone-closed. □

Corollary 5.2. *$F'(\bar{x} + \bar{t})$ is $(cs \leq k)$-concentrated.*

Proof. $F(\bar{x}) \neq 0 \Rightarrow F'(\bar{x}+\bar{t}) \neq 0 \Rightarrow$ at least one of the coefficients in the basis is non-zero (in the shifted polynomial if all basis coefficients were zero then all the coefficients which linearly depend on basis elements will be zero). Since the coefficients belong to $\mathbb{F}^k(\bar{t}), |B| \leq k$. This together with the fact that B is the least basis constructed in greedy manner and is cone-closed implies $(cs \leq k)$-concentration in the shifted polynomial. □

Corollary 5.3. *$F'(\bar{x}+\bar{t})$ is $\log_2 k$ -support concentrated.*

Proof. Since it is $cs \leq k$-concentrated. We wish to maximize s in the given equation $\prod_{i\in[s]}(e_i+1) \leq k$, where $e_i \geq 1$. This will happen when $\forall i$, $e_i = 1$. This gives $2^s \leq k \Rightarrow s \leq \log_2 k$. □

B is cone-closed $\Rightarrow$ $(cs \leq k)$-concentration $\Rightarrow$ $(\log_2 k)$-support concentration. And each implication here is a strict one, that is the converses are not true. We leave it as a nice exercise for the reader to verify it. **Thus we have proved a structural property strictly stronger than the $\log_2 k$-support concentration proved in [ASS13].**

5.4 Small characteristic case

Note that the above proof will fail for the case where $0 < \mathrm{ch}(\mathbb{F}) = p < d$, where d is the individual degree of some variable in F, since ∂ operator will not work. In fact, a general polynomial over small characteristic fields is not cone closed. We can observe this from the following simple counterexample:

$$\text{Let } F(x) = \begin{bmatrix}1\\0\end{bmatrix} x^p + \begin{bmatrix}0\\1\end{bmatrix} 1$$

$$F'(x) = F(x+t) = \begin{bmatrix}1\\0\end{bmatrix} (x+t)^p + \begin{bmatrix}0\\1\end{bmatrix} 1$$

$$F'(x) = \begin{bmatrix}1\\0\end{bmatrix} x^p + \begin{bmatrix}1\\0\end{bmatrix} t^p + \begin{bmatrix}0\\1\end{bmatrix} 1$$

$$F'(x) = \begin{bmatrix}1\\0\end{bmatrix} x^p + \begin{bmatrix}t^p\\1\end{bmatrix} 1$$

Note that $\mathrm{coeff}_{F'}(x^p) = \begin{bmatrix}1\\0\end{bmatrix}$ and $\mathrm{coeff}_{F'}(1) = \begin{bmatrix}t^p\\1\end{bmatrix}$ both belong to basis B as they are linearly independent over $\mathbb{F}(t)$ and span the coefficient space of F'. But B is not cone closed since $x^p \in B$ but the elements of its cone $\{x^{p-1}, \ldots, x\} \notin B$. This calls for an adjustment in the definition of cone and cone closure when the base field has small prime characteristic. When a variable has degree $< p$ in a monomial, after shifting it will produce its old cone (cone as per the old definition). But the problematic case is when degree $> p$.

After observing a number of examples, we give the following correction for characteristic p fields: **Consider prime powers of a variable to be unit or indivisible**. We write the power of any variable in p-ary representation. Let us first observe few examples before formally defining the new cone. Observe that for $p = 2$, shifting x^{40}, we get $(x+t)^{40} = (x+t)^{32} \cdot (x+t)^8 = (x^{32}+t^{32}) \cdot (x^8+t^8) = (x^{40}+t^8x^{32}+t^{32}x^8+t^{40})$. The binary representation of 40 yields $40 = 1.2^5 + 1.2^3$. And the terms of x which we get after shift are basically $\{x^{1.2^5}, x^{0.2^5}\} \times \{x^{1.2^3}, x^{0.2^3}\} = \{x^{32}, 1\} \times \{x^8, 1\}$. Now, for $p = 3$, consider x^{20}, where $20 = 2.3^2 + 2.3^0$. Learning from the last example, we should get the following powers of x after shift: $\{x^{2.3^2}, x^{1.3^2}, x^{0.3^2}\} \times \{x^{2.3^0}, x^{1.3^0}, x^{0.3^0}\} = \{x^{18}, x^9, 1\} \times \{x^2, x^1, 1\} = \{x^{20}, x^{19}, x^{18}, x^{11}, x^{10}, x^9, x^2, x, 1\}$. Note that these will be the exact terms if we expand manually: $(x+t)^{20} = (x+t)^{18} \cdot (x+t)^2 = (x^9+t^9)^2 \cdot (x+t)^2 = (x^{18}+2t^9x^9+t^{18}) \cdot (x^2+2tx+t^2)$.

Formally, let the power of a variable x_i appearing in a monomial be $e = \sum_{i=0}^{m} a_i p^i$, where each a_i lies in the range $0 \leq a_i < p$ (p-ary representation). Then for a characteristic p field, we **define (new) cone′ size** of x^e is $\text{cs}'(x^e) = \prod_{i=0}^{m}(a_i+1)$. And the new **definition of cone′** is as follows:

$$\text{cone}'(x^e) = \{x^{(a_m).p^m}, x^{(a_m-1).p^m}, \ldots, x^{0.p^m}\} \times \ldots \times \{x^{(a_0).p^0}, x^{(a_0-1).p^0}, \ldots, x^{0.p^0}\}$$

Now, for general n variables, we define $\text{cone}'(\bar{x}^{\bar{e}})$ as $\text{cone}'(x_1^{e_1}) \times \text{cone}'(x_2^{e_2}) \times \ldots \times \text{cone}'(x_n^{e_n})$, and $\text{cs}'(\bar{x}^{\bar{e}}) = \prod_{j=1}^{n} \text{cs}'(x_j^{e_j})$.

Now with this new definition of cone, we prove the same cone closure result as Theorem 5.1 for characteristic p fields. Before, we do that, we need to find a substitute for the ∂ operator which is not useful in this setting. We state the following lemma without proof. Interested reader can look up [AFGS17] for a detailed analysis and proof. Also from now on, we will say cone when we are talking about the old definition of cone which works in 0 or large characteristic fields, and cone′, for small characteristic fields.

Lemma 5.4. *[AFGS17] Let $\mathbb{F}$ be a field of 0 or large characteristic and $F(\bar{x}) \in (\mathbb{F}^k)[\bar{x}]$ be a non-zero polynomial. Let $\phi : \mathbb{F}[x_1, \dots, x_n] \to \mathbb{F}[t]$ be a variable map that keeps all monomials of F in the image space of ϕ distinct. Then $F' = F(x_1 + \phi(x_1), \dots, x_n + \phi(x_n))$ has a cone closed basis over $\mathbb{F}(\bar{t})$, with respect to the monomial ordering induced by ϕ.*

Recall that Kronecker map (Section 3.2) has the property that it keeps all monomials of F distinct from each other. Moreover, it totally orders them. Shifting the variables by any such map achieves cone closed basis in the shifted polynomial. Therefore $F'(\bar{x}, t) = F(x_1 + t, x_2 + t^d, x_3 + t^{d^2}, \dots, x_n + t^{d^{n-1}}) = F(x_1 + \phi(x_1), \dots, x_n + \phi(x_n))$, where d is the individual degree of F has a cone closed basis over 0 or large characteristic fields. In fact, [AFGS17] show that even shifting by a basis isolating weight assignment works. Now, we are ready to prove the main theorem of this section. Note that in the above lemma also the field must have 0 or large characteristic, but in the proof of next theorem, we will show that after applying a transformation we can reduce small characteristic case to large characteristic case.

Theorem 5.5. *Let $\mathbb{F}$ be a field of small characteristic p and $F(\bar{x}) \in (\mathbb{F}^k)[\bar{x}]$ be a non-zero polynomial. If we shift F to get $F'(\bar{x}, \bar{t}) = F(x_1 + t_1, x_2 + t_2, \dots, x_n + t_n)$, then F' has a cone′ closed basis over $\mathbb{F}(\bar{t})$, with respect to any given monomial ordering.*

Proof. Let m be the largest power of $p \leq$ individual degree d of F. Apply shift on F to get $F' = F(x_1 + t_1, \dots, x_n + t_n)$. Then, we apply the transformation ψ on F', which acts on the monomial $x_1^{e_1} x_2^{e_2} \dots x_n^{e_n} = x_1^{e_{1m}p^m + \dots + e_{10}p^0} \dots x_n^{e_{nm}p^m + \dots + e_{n0}p^0}$ as follows:

$$\begin{gathered} x_1^{e_{1m}p^m + \dots + e_{10}p^0} \dots x_n^{e_{nm}p^m + \dots + e_{n0}p^0} \\ \downarrow^{\psi} \\ (y_{1m}^{e_{1m}} \dots y_{10}^{e_{10}}) \cdots (y_{nm}^{e_{nm}} \dots y_{n0}^{e_{n0}}) \end{gathered}$$

Observe that ψ^{-1} is $\psi^{-1} : y_{ij} \mapsto x_i^{p^j}$. After applying transformation ψ on F', it is same as shifting variables y_{ij} by $t_i^{p^j}$. Observe that, if we first apply shift on F, and then apply transformation ψ, we get the same polynomial as the one obtained after first applying ψ, and then applying shift on variables $y_{ij} \mapsto y_{ij} + t_i^{p^j}$ for $i \in [n]$ and $j \in \{0, 1, \dots, m\}$. Formally, $\psi(F(\bar{x} + \bar{t})) = F(\psi(\bar{x}) + t_i^{p^j})$.

The nice property of the polynomial $\psi(F(\bar{x}+\bar{t}))$ is that all the variables y_{ij} have individual degree $< p$. Thus, we have reduced the small characteristic case to large characteristic case, by applying the transformation ψ. In fact, ψ^{-1} is the Kronecker map (where instead of individual degree, we have p in the exponents), which has the property of reducing the number of variables and increasing the degree (Refer Section 3.2). Contrary to that, ψ is the inverse Kronecker map, which increases the number of variables but reduces the individual degrees.

Therefore, we can apply old cone closure result on the polynomial obtained. By applying Lemma 5.4 on the polynomial $\psi(F(\bar{x}+\bar{t}))$, we see that $\psi(F(\bar{x}+\bar{t}))$ has a cone closed basis. The shift $y_{ij} \mapsto y_{ij} + t_i^{p^j}$ is same as shifting by Kronecker map since individual degree of each variable $y_{ij} < p$. Now, we claim that (old) cone closure in $\psi(F(\bar{x}+\bar{t})) = F(\psi(\bar{x}) + t_i^{p^j})$ implies (new) cone$'$ closure in $F' = F(\bar{x}+\bar{t})$.

Claim 5.6. *If $\psi(F(\bar{x}+\bar{t}))$ has (old) cone closed basis, then $F(\bar{x}+\bar{t})$ has a (new) cone$'$ closed basis.*

Proof. We will prove the contrapositive form of the implication. Suppose $F(\bar{x}+\bar{t})$ does not have a cone$'$ closed basis. Recall how the least basis is constructed which was described in Section 5.3. First we pick a monomial of $F(\bar{x}+\bar{t})$ in a greedy manner as per the monomial ordering. If its coefficient is linearly independent to the the coefficients present in the basis constructed till now, we add it to the basis. If it is linearly dependent, we check the next monomial and so on.

The hypothesis of $F(\bar{x}+\bar{t})$ not having a cone$'$ closed basis means that there exists a monomial $m_{\bar{e}}$ in the constructed basis B, such that at least one of its cone$'$ elements does not appear in B. Now, consider the following monomial order for variables y_{ij} of the polynomial $\psi(F(\bar{x}+\bar{t}))$, where $i \in [n], 0 \leq j \leq m$. Across index i, we keep the same monomial order as that followed by x_i's in $F(\bar{x}+\bar{t})$. For example, if $x_1 < x_2 < \ldots < x_n$, then $y_{1j} < y_{2j} \ldots < y_{nj}$, for all j in $0 \leq j \leq m$. And across index j, we will have the natural monomial order inspired from least significant bit to most significant bit. That is, $y_{i0} < y_{i1} < \ldots < y_{im}$ for all $i \in [n]$. Thus, $y_{ab} < y_{cd}$ if and only if $a < c$ (where $<$ is as per the monomial order followed in $F(\bar{x}+\bar{t})$), or $a = c$ and $b < d$ (where $<$ is the usual strictly less than). This monomial order for $\psi(F(\bar{x}+\bar{t}))$ exactly preserves the monomial order followed while constructing basis B

for $F(\bar{x}+\bar{t})$. Therefore, the basis B' constructed for $\psi(F(\bar{x}+\bar{t}))$ will be exactly

$$B' = \left\{ \text{coeff}_{\psi(F(\bar{x}+\bar{t}))}(\bar{y_{ij}}^{\bar{f_{ij}}}) \mid \bar{y_{ij}}^{\bar{f_{ij}}} = \psi(\bar{x}^{\bar{e}}) \text{ and } \bar{x}^{\bar{e}} \in B \right\}$$

Therefore, the ψ image of the element which was not present in the $cone'$ of the monomial $m_{\bar{e}}$ will also not be present in the basis B' which will contain $\psi(m_{\bar{e}})$. And since individual degree of each variable y_{ij} in $\psi(F(\bar{x}+\bar{t}))$ is $< p$, B' is not (old) cone closed. □

Thus, existence of (old) cone closed basis together with Claim 5.6 implies that $F(\bar{x}+\bar{t})$ has a (new) cone′ closed basis. [End of proof of Theorem 5.5] □

A brief look into the factorization algorithms over finite fields will tell that they also face the problem of vanishing derivative due to small characteristics. The idea of transformation ψ was inspired from the solution adopted in the factorization algorithms, of seeing the polynomial $f(x)^p$ as $f(x^p)$ over $\mathbb{F}_p[x]$.

5.5 Conclusion and Future work

There are few important things we wish to remark here. By derandomizing the shift of Theorem 5.1, we achieve cone size $\leq k$-concentration which implies a polynomial sized hitting set in the tiny regime, where support n is bounded by $O(\log s)$. k is usually the top fan-in of the circuit, when we view diagonal depth 3 circuit as a polynomial over $\mathbb{F}^k[\overline{x}]$. Note that we do not obtain a polynomial sized hitting set from (new) cone′ closed basis from the arguments stated in Lemma 3.7, because cone′ definition allows high degree monomials. Nevertheless, it does not undermine the work done for small characteristic case. It helped in completing the notion of cone closure as a structural result over all fields.

The next step should be to obtain deterministic efficient shifts for various tiny models in order to get blackbox PIT algorithms for the same. These will be conditional to the characteristic of base field being 0 or large enough. Also, we need to get a new link from cone closure to hitting set for small characteristic fields.

Chapter 6

Diagonal Circuits and Wronskian

6.1 Introduction

Let us start with a question. Suppose $f_1, f_2, \ldots, f_k$ are k linearly independent polynomials. What can we say about the linear independence of their powers? That is, will ${f_1}^d, {f_2}^d, \ldots, {f_k}^d$ be linearly independent, $\forall d \in \mathbb{Z}^+$? The answer is NO, one of the counter examples being : x^2+y^2, x^2-y^2, xy which are linearly independent, but for $d=2$, they are dependent.

$$(x^2+y^2)^2 - (x^2-y^2)^2 - 4(xy)^2 = 0$$

But in the next section we will see that there are very few such exceptional values of d, and that most of the integral powers of linearly independent polynomials are also independent (actually, our theorem requires even weaker condition that the polynomials are not constant multiples of each other, that is, they are just pairwise linearly independent).

6.2 Wronskian

One of the tools to study about linear independence of functions is the Wronskian which is quite frequently used in linear algebra. It is a special determinant which was first introduced by Józef Hoene-Wroński and named by Thomas Muir [MM03]. For n

real or complex valued functions $f_1, f_2, \ldots, f_n$, which are $n-1$ times differentiable, the Wronskian $W(f_1, f_2, \ldots, f_n)$ as a function is defined by

$$W(f_1, f_2, \ldots, f_n)(x) = \begin{vmatrix} f_1(x) & f_2(x) & \ldots & f_n(x) \\ f_1'(x) & f_2'(x) & \ldots & f_n'(x) \\ \vdots & \vdots & \ddots & \vdots \\ f_1^{(n-1)}(x) & f_2^{(n-1)}(x) & \ldots & f_n^{(n-1)}(x) \end{vmatrix}$$

If the functions are linearly dependent, then so are their derivatives (as differentiation is a linear operation). This means the columns of the Wronskian matrix will be linearly dependent and hence, the Wronskian vanishes. Thus, the Wronskian can be used to show that a set of functions are linearly independent by showing that it is not an identically zero polynomial. The converse is not true in general as pointed out by Peano in 1889, but is true for analytic functions, specifically polynomials. That is the vanishing of Wronskian implies that they are linearly dependent. But we only require the first side in our proof.

For multivariate functions, we have the **generalized Wronskian**. For functions with m variables $x_1, \ldots, x_m$ let

$$\Delta_s = \left(\frac{\partial}{\partial x_1}\right)^{j_1} \cdots \left(\frac{\partial}{\partial x_m}\right)^{j_m} \quad \text{with} \quad j_1 + \cdots + j_m \leq s$$

$$W(f_1, \cdots, f_n)(\overline{x}) = \begin{vmatrix} f_1 & \ldots & f_n \\ \Delta_1(f_1) & \ldots & \Delta_1(f_n) \\ \vdots & \ddots & \vdots \\ \Delta_{n-1}(f_1) & \ldots & \Delta_{n-1}(f_n) \end{vmatrix}$$

Note that there are finitely many Wronskians produced based on which partial derivatives you take. Also i^{th} row has derivative of functions upto i^{th} order (may not be equal to i) for $0 \leq i \leq n-1$.

Lemma 6.1. *[Wol89, BD10] If any of the Wronskians $W(f_1, f_2, \cdots, f_n)$ does not vanish identically then $f_1, f_2, \cdots, f_n$ are linearly independent polynomials.*

Proof. If $f_1, \cdots, f_n$ are linearly dependent, then so are $\Delta_i(f_1), \cdots, \Delta_i(f_n)$ for $1 \leq i \leq n-1$, since differentiation is a linear operation. Also polynomials are infinitely differentiable. Therefore all the Wronskians vanish for linearly dependent polynomials.

□

6.3 Main Theorem

Now, let us formally state the main theorem we wish to present.

Theorem 6.2. *Let $f_1(\overline{x}), f_2(\overline{x}), \cdots, f_k(\overline{x})$ be polynomials from $\mathbb{F}[\overline{x}]$ which are pairwise linearly independent over $\mathbb{F}$, then polynomials $f_1(\overline{x})^d, f_2(\overline{x})^d, \ldots, f_k(\overline{x})^d$ are linearly dependent for at most $\binom{k-1}{2}$ values of $d \in \mathbb{Z}^+$.*

Proof Idea: We will consider one of the Wronskian of powers of these polynomials, that is, $W({f_1}^d, \cdots, {f_k}^d)$ where we will take the $k-1$ partial derivatives with respect to only a single variable, say x_1. We will show that this Wronskian does not vanish identically, and that it is a polynomial in $\mathbb{F}(\overline{x})[d]$, with its degree in $d = \binom{k-1}{2}$. And thus it will have at most these many roots for d in $\mathbb{F}$.

Now, we will break the complete proof into various lemmas. For the purpose of simplicity and visualization, let us first see Wronskian for only 3 polynomials, that is $k = 3$.

$$\begin{aligned}\Delta_1({f_i}^d) &= d.{f_i}^{d-1}.{f_i}'\\ \Delta_2({f_i}^d) &= d.{f_i}^{d-1}.{f_i}'' + d.(d-1).{f_i}^{d-2}.({f_i}')^2\end{aligned}$$

By splitting the determinant at the last row, we get

$$W({f_1}^d, {f_2}^d, {f_3}^d) = \begin{vmatrix} {f_1}^d & {f_2}^d & {f_3}^d \\ \Delta_1({f_1}^d) & \Delta_1({f_2}^d) & \Delta_1({f_3}^d) \\ \Delta_2({f_1}^d) & \Delta_2({f_2}^d) & \Delta_2({f_3}^d) \end{vmatrix}$$

$$=$$

$$\begin{vmatrix} {f_1}^d & {f_2}^d & {f_3}^d \\ d.{f_1}^{d-1}.{f_1}' & d.{f_2}^{d-1}.{f_2}' & d.{f_3}^{d-1}.{f_3}' \\ d.(d-1).{f_1}^{d-2}.({f_1}')^2 & d.(d-1).{f_2}^{d-2}.({f_2}')^2 & d.(d-1).{f_3}^{d-2}.({f_3}')^2 \end{vmatrix}$$

$$+$$

$$\begin{vmatrix} {f_1}^d & {f_2}^d & {f_3}^d \\ d.{f_1}^{d-1}.{f_1}' & d.{f_2}^{d-1}.{f_2}' & d.{f_3}^{d-1}.{f_3}' \\ d.{f_1}^{d-1}.{f_1}'' & d.{f_2}^{d-1}.{f_2}'' & d.{f_3}^{d-1}.{f_3}'' \end{vmatrix}$$

$$=$$

$$d^2.(d-1).{f_1}^d.{f_2}^d.{f_3}^d \begin{vmatrix} 1 & 1 & 1 \\ \frac{{f_1}'}{f_1} & \frac{{f_2}'}{f_2} & \frac{{f_3}'}{f_3} \\ \left(\frac{{f_1}'}{f_1}\right)^2 & \left(\frac{{f_2}'}{f_2}\right)^2 & \left(\frac{{f_3}'}{f_3}\right)^2 \end{vmatrix}$$

$$+$$

$$d^2.{f_1}^d.{f_2}^d.{f_3}^d \begin{vmatrix} 1 & 1 & 1 \\ \frac{{f_1}'}{f_1} & \frac{{f_2}'}{f_2} & \frac{{f_3}'}{f_3} \\ \frac{{f_1}''}{f_1} & \frac{{f_2}''}{f_2} & \frac{{f_3}''}{f_3} \end{vmatrix}$$

Lemma 6.3. *Let $f_1(\overline{x}), f_2(\overline{x}), \cdots, f_k(\overline{x})$ be polynomials from $\mathbb{F}[x_1, \ldots, x_n]$ which are pairwise linearly independent over $\mathbb{F}$, then the degree of variable d, in the reduced Wronskian polynomial $W'({f_1}^d, \cdots, {f_k}^d) \in \mathbb{F}(\overline{x})[d]$, is exactly $\binom{k-1}{2}$.*

Proof. By reduced Wronskian polynomial, we mean the polynomial W' left after taking out d^{k-1} common from the last $k-1$ rows and ${f_1}^d.{f_2}^d.\cdots{f_k}^d$ from the Wronskian $W({f_1}^d, \cdots, {f_k}^d)$.

$$\begin{aligned} W({f_1}^d, \cdots, {f_k}^d) &= d^{k-1}.{f_1}^d \cdots {f_k}^d. \quad W'({f_1}^d, \cdots, {f_k}^d) \\ W = 0 \quad &\Rightarrow \quad W' = 0 \quad \text{(since } d, f_1, \cdots, f_k \text{ are all non-zero)} \end{aligned}$$

Therefore we only need to consider degree of d in W'

$$W({f_1}^d, \cdots, {f_k}^d) = \begin{vmatrix} {f_1}^d & {f_2}^d & \cdots & {f_3}^d \\ \Delta_1({f_1}^d) & \Delta_1({f_2}^d) & \cdots & \Delta_1({f_k}^d) \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{k-1}({f_1}^d) & \Delta_{k-1}({f_2}^d) & \cdots & \Delta_{k-1}({f_k}^d) \end{vmatrix}$$

Let the rows of W be numbered from 0 to $k-1$. Then the contribution of row i in degree of d in $W = i$.

$$\begin{aligned}
\text{degree of } d \text{ in } W &= \sum_{i=0}^{k-1} i \\
\text{degree of } d \text{ in } W' &= \sum_{i=0}^{k-1} i \quad - (k-1) \\
&= \sum_{i=0}^{k-2} i \quad = \frac{(k-2)(k-1)}{2} \\
&= \binom{k-1}{2}
\end{aligned}$$

In Lemma 6.4, we will prove that coefficient of degree $\binom{k-1}{2}$ term is indeed non-zero, thus making it the actual degree of d in $\mathbb{F}(\overline{x})[d]$ □

Lemma 6.4. *Let $f_1(\overline{x}), f_2(\overline{x}), \cdots, f_k(\overline{x})$ be polynomials from $\mathbb{F}[x_1, \ldots, x_n]$ which are pairwise linearly independent over $\mathbb{F}$, then the wronskian polynomial $W({f_1}^d, \ldots, {f_k}^d)$ does not vanish identically.*

Proof. Consider the polynomial $\Delta_p({f_1}^d)$ (element at row p, column 0), where $0 \leq p \leq k-1$. Let us describe the unique term in this polynomial with the highest degree of d. We claim that it will be $\prod_{j=0}^{p-1}(d-j) \quad {f_1}^{d-p}\big(f_1'\big)^p$. The proof will proceed by induction on p. For the base case $p = 1$, it is straightforward. Suppose the claim is true for p, and we wish to prove it for $p+1$. Note that the highest degree term in $\Delta_{p+1}({f_1}^d)$ would come by differentiating the highest degree term in $\Delta_p({f_1}^d)$ only. And using induction hypothesis,

$$\text{Highest degree term in } \Delta_{p+1}({f_1}^d) = \left(\prod_{j=0}^{p-1}(d-j) \quad {f_1}^{d-p}\big(f_1'\big)^p\right)'$$

$$= \prod_{j=0}^{p-1}(d-j)\left\{{f_1}^{d-p}.p.(f_1')^{p-1}.f_1'' + (f_1')^p.(d-p).f_1^{d-p-1}.f_1'\right\}$$

Extracting the highest degree term from above expression we get,

$$\prod_{j=0}^{(p+1)-1}(d-j) \quad {f_1}^{d-(p+1)}\big(f_1'\big)^{p+1}$$

as expected for $p+1$. The calculations will be same for $f_2, \cdots, f_k$. Now, for getting the highest degree term in $W({f_1}^d, \cdots, {f_k}^d)$, we will split the determinant recursively at highest degree terms in each row. The Wronskian which will have the highest degree of d will be as follows:

$$= \prod_{j=0}^{k-2}(d-j) \begin{vmatrix} {f_1}^d & {f_2}^d & \dots & {f_k}^d \\ d.{f_1}^{d-1}.{f_1}' & d.{f_2}^{d-1}.{f_2}' & \dots & d.{f_k}^{d-1}.{f_k}' \\ \vdots & \vdots & \ddots & \vdots \\ {f_1}^{d-k+1}\left(f_1'\right)^{k-1} & {f_2}^{d-k+1}\left(f_2'\right)^{k-1} & \dots & {f_k}^{d-k+1}\left(f_k'\right)^{k-1} \end{vmatrix}$$

$$= \quad d^{k-1}.(d-1)^{k-2}\cdots(d-k+2)^1$$
$$\begin{vmatrix} {f_1}^d & {f_2}^d & \dots & {f_k}^d \\ {f_1}^{d-1}.{f_1}' & {f_2}^{d-1}.{f_2}' & \dots & {f_k}^{d-1}.{f_k}' \\ \vdots & \vdots & \ddots & \vdots \\ {f_1}^{d-k+1}\left(f_1'\right)^{k-1} & {f_2}^{d-k+1}\left(f_2'\right)^{k-1} & \dots & {f_k}^{d-k+1}\left(f_k'\right)^{k-1} \end{vmatrix}$$

$$= \quad d^{k-1}.(d-1)^{k-2}\cdots(d-k+2).{f_1}^d{f_2}^d\dots{f_k}^d$$
$$\begin{vmatrix} 1 & 1 & \dots & 1 \\ \frac{{f_1}'}{f_1} & \frac{{f_2}'}{f_2} & \dots & \frac{{f_k}'}{f_k} \\ \vdots & \vdots & \ddots & \vdots \\ \left(\frac{{f_1}'}{f_1}\right)^{k-1} & \left(\frac{{f_2}'}{f_2}\right)^{k-1} & \dots & \left(\frac{{f_k}'}{f_k}\right)^{k-1} \end{vmatrix} \tag{6.1}$$

Now we will prove that the determinant in Equation (6.1) is non-zero, thus proving that the coefficient of highest degree term is non-zero.

Claim 6.5. *f_i, f_j are pairwise linearly dependent if and only if $\frac{f_i'}{f_i} = \frac{f_j'}{f_j}$.*

Proof. Observe that by pairwise dependence, we mean that f_i is a constant multiple of f_j.

$$f_i = c.f_j \quad c \in \mathbb{F}$$
$$f_i' = c.f_j' \quad \text{(Differentiating on both sides)}$$
$$\frac{f_i'}{f_i} = \frac{f_j'}{f_j}$$

For the other side suppose,

$$\frac{f_i'}{f_i} = \frac{f_j'}{f_j}$$

$$f_i'.f_j - f_j'.f_i = 0$$

$$\frac{f_i'.f_j - f_j'.f_i}{f_j^2} = 0$$

$$\left(\frac{f_i}{f_j}\right)' = 0$$

$$f_i = c.f_j$$

□

Since all our polynomials are pairwise independent, $\frac{f_i'}{f_i} \neq \frac{f_j'}{f_j}$. Therefore the determinant appearing in Equation (6.1) is that of a **Vandermonde matrix**, and equal to $\prod_{1\leq i\leq j\leq k}\left(\frac{f_i'}{f_i} - \frac{f_j'}{f_j}\right) \neq 0$. Hence the coefficient of highest degree term in $W({f_1}^d, \cdots, {f_k}^d)$ is non-zero and Wronskian is identically non-zero polynomial. [End of Lemma 6.4] □

Proof of Main Theorem. Lemma 6.3 and Lemma 6.4 essentially state that $W({f_1}^d, \cdots, {f_k}^d)$ is identically non-zero polynomial in $\mathbb{F}(\overline{x})[d]$, with degree bound of $\binom{k-1}{2}$ in d. This implies, it has at most $\binom{k-1}{2}$ roots in $\mathbb{F}(\overline{x})$, and thus at most these many roots in $\mathbb{F}$ also (since $\mathbb{F} \subset \mathbb{F}(\overline{x})$). Thus by Lemma 6.1 they will be linearly independent for all other values of d. The same argument will hold for other identically non-zero wronskians obtained when differentiation is carried out with some variable other than x_1. □

6.4 Conclusion

Let us now try to see its connection with the PIT problem, because of which, we stumbled upon this result. Consider a polynomial which has a diagonal circuit representation with top fan-in k, that is, it can be expressed as sum of powers of k polynomials. Let $F = f_1^d + f_2^d + \cdots + f_k^d$. If $f_i's$ are not constant multiples of each other (pairwise linearly independent), then by our Main Theorem 6.2, F will be identically zero for very few constant values of d (at most $\mathcal{O}(k^2)$). This is because for all other values of d, the Wronskian will not vanish identically and by standard Lemma 6.1, $f_1^d, \cdots, f_k^d$

will be linearly independent, and hence their sum is non-zero. If it were not for these exceptions, then the sparse PIT map could be extended to give a polynomial sized hitting set for F. In that case, for the map to preserve non-zeroness of F, it would simply suffice to preserve pairwise linear independence of all (f_i, f_j) pairs, which sparse PIT map can do. Although this result does not immediately solve PIT for any model, it is still a surprising structural result in itself, with a very simple proof, that uses only the tool of Wronskian. Also, note that for the counterexample we took in the beginning, x^2+y^2, x^2-y^2, xy, our Main Theorem implies that $d = 2$ is the only identity possible. (since $\binom{k-1}{2} = 1$ in this case). That means $c_1.(x^2+y^2)^d + c_2.(x^2-y^2)^d + c_3.(xy)^d \neq 0$ for $d > 2$, and for any constant values of c_1, c_2, c_3 not all zero.

Chapter 7

Conclusion and Future Work

This thesis is the result of endeavour to solve some special, yet very non-trivial instances of Polynomial Identity Testing problem. We discussed briefly how solving PIT for depth-4 or depth-3 models almost solves the general problem. Depth-4 model has a lot of special cases to solve compared to depth-3 model. In Chapter 4, we give the first polynomial time blackbox algorithm for one such special instance, which we call diagonal depth-4 model with top fan-in 3. It essentially tells, that we can test zeroness of a polynomial computed by a $\sum^3 \bigwedge^a \sum \prod$ in polynomial time. The restrictions are mainly the top fan-in $= 3$, and the power gates having equal fan-in a. The next step in this line of work would be to remove the equal fan-in restrictions, but more importantly, designing an efficient blackbox algorithm for any constant top fan-in. The ultimate goal however, is to solve it for a general fan-in (diagonal depth-4), as it will put PIT in quasiP. Also, there are plenty of other open models - multilinear depth-3, arithmetic branching programs (ABPs), general depth-3 et cetera.

In Chapter 5, we explain a new measure of rank concentration, namely cone closure, which we show is better than the older measure of low support concentration. We prove the existence of a cone closed basis in a general polynomial on applying a random shift. We complete the definition and proof of cone closed basis, also for the small characteristic fields, which do not follow the traditional sense of cone closure. This is mainly a structural result, which does not immediately give any PIT algorithm. But this work was mainly motivated to provide cone closure as a tool or technique that can be used to design PIT algorithms for various models. Recently, it was used in

[AFGS17] to solve PIT for diagonal depth-3 circuits. It will be interesting to use it similarly for other models also.

Lastly, we give another structural result in Chapter 6. It proves a very interesting property that positive powers of linearly independent polynomials are also linearly independent most of the time. We prove that the number of powers d which can make the polynomials $f_1^d, f_2^d, \ldots, f_k^d$ linearly dependent is upper bounded by $O(k^2)$. This tells us that the density of identically zero polynomials among the polynomials computed by diagonal circuits is very low. Although there are very few values of d where $f_1^d + f_2^d + \ldots + f_k^d = 0$, since we cannot pinpoint the exact values of d, we fail to get a hitting set. Nevertheless, it is an interesting property which happens to have a very simple proof, which we show using the Wronskians. It may have applications in problems which are even outside the domain of PIT.

References

[AFGS17] Manindra Agrawal, Michael Forbes, Sumanta Ghosh, and Nitin Saxena. Small hitting-sets for tiny arithmetic circuits or: How to turn bad designs into good. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:35, 2017.

[AGKS13] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for low-distance multilinear depth-3. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:174, 2013.

[AGKS15] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for roabp and sum of set-multilinear circuits. *SIAM Journal on Computing*, 44(3):669–697, 2015.

[Agr05] Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 92–105. Springer, 2005.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in P. *Annals of mathematics*, pages 781–793, 2004.

[ASS13] Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial hitting-set for set-depth-δ formulas. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 321–330. ACM, 2013.

[ASSS12] Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: hitting-sets, lower bounds for depth-d occur-k formulas & depth-3 transcendence degree-k circuits. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 599–614. ACM, 2012.

[AT99] Noga Alon and M Tarsi. Combinatorial nullstellensatz. *Combinatorics Probability and Computing*, 8(1):7–30, 1999.

[AV08] Manindra Agrawal and V Vinay. Arithmetic circuits: A chasm at depth four. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 67–75. IEEE, 2008.

[BD10] Alin Bostan and Philippe Dumas. Wronskians and linear independence. *The American Mathematical Monthly*, 117(8):722–727, 2010.

[BHLV09] Markus Bläser, Moritz Hardt, Richard J Lipton, and Nisheeth K Vishnoi. Deterministically testing sparse polynomial identities of unbounded degree. *Information Processing Letters*, 109(3):187–192, 2009.

[BMS13] Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Information and Computation*, 222:2–19, 2013.

[CKW11] Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity and beyond. *Foundations and Trends® in Theoretical Computer Science*, 6(1–2):1–138, 2011.

[For14] Michael Andrew Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs*. PhD thesis, Massachusetts Institute of Technology, 2014.

[FSS14] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 867–875, New York, NY, USA, 2014. ACM.

[GKKS13] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 578–587. IEEE, 2013.

[Gur16] Rohit Gurjar. *Derandomizing PIT for ROABP and Isolation Lemma for Special Graphs*. PhD thesis, Indian Institute of Technology Kanpur, 2016.

[Kal17] Kartik Kale. FPT algorithms for computing division, gcd and identity testing of polynomials. Master's thesis, Indian Institute of Technology, Kanpur, 2017.

[KI03] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 355–364. ACM, 2003.

[Kor16] Arpita Korwar. *Polynomial Identity Testing and Lower Bounds for Sum of Special Arithmetic Branching Programs*. PhD thesis, Indian Institute of Technology Kanpur, 2016.

[KS01] Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 216–223. ACM, 2001.

[KS09] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 198–207. IEEE, 2009.

[Lan02] Serge Lang. Algebra revised third edition. *Graduate Texts in Mathematics*, 1(211):ALL–ALL, 2002.

[Mas84] Richard C Mason. *Diophantine equations over function fields*, volume 96. Cambridge University Press, 1984.

[MM03] Thomas Muir and William Henry Metzler. *A Treatise on the Theory of Determinants*. Courier Corporation, 2003.

[MVV87] Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 345–354. ACM, 1987.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009.

[Sch80] Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.

[Sny00] Noah Snyder. An alternate proof of mason's theorem. *Elemente der Mathematik*, 55(3):93–94, 2000.

[SS12] Nitin Saxena and Comandur Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM Journal on Computing*, 41(5):1285–1298, 2012.

[SS13] Nitin Saxena and Comandur Seshadhri. From sylvester-gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *Journal of the ACM (JACM)*, 60(5):33, 2013.

[Sto81] W Wilson Stothers. Polynomial identities and hauptmoduln. *The Quarterly Journal of Mathematics*, 32(3):349–370, 1981.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010.

[VSBR83] Leslie G. Valiant, Sven Skyum, Stuart Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM Journal on Computing*, 12(4):641–644, 1983.

[Wol89] Kenneth Wolsson. Linear dependence of a function set of m variables with vanishing generalized wronskians. *Linear Algebra and its applications*, 117:73–80, 1989.