Expanders in arithmetic circuit lower bound: Towards a separation between ROABPs and multilinear depth 3 circuits

A THESIS

SUBMITTED FOR THE DEGREE OF

Master of Science (Engineering)

IN THE

Faculty of Engineering

BY

Vineet Nair

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

June 2015

Signature of the Author: .....

Vineet Nair
Dept. of Computer Science and Automation
Indian Institute of Science, Bangalore

Signature of the Thesis Supervisor: .....

Chandan Saha
Assistant Professor
Dept. of Computer Science and Automation
Indian Institute of Science, Bangalore

Acknowledgements

First and foremost, I am extremely thankful to my advisor Chandan Saha. The past two years working with Chandan has been a deep learning curve for me. I have tried my best to imbibe in me the suggestions given by him. Words would not suffice to express my gratitude towards him. In the coming years I hope I can inculcate in me at least a part of the enthusiasm and dedication he has towards his work and research. In this thesis we have tried to resolve the questions asked by him in many of the insightful discussions we had over the past two years.

Studying at IISc has been amazing. I am extremely indebted to all the faculties of the Department of Computer Science and Automation. Specially I am thankful to Arnab Bhattacharya for his valuable guidance on various occasions which helped me immensely. Arnab also pointed out some corrections in the language used in section A.1 and brought to our notice the work of [KMS98] which is used in the proof of theorem 3.2.2.

I am greatly thankful to Neeraj Kayal who pointed out the possibility of constructing hard polynomial used in subsection 4.2.1 in one of the discussions we had. Neeraj also taught us a course on Arithmetic Circuits in the winter/summer of 2015. This course helped me understand the fundamental problems in Algebraic Complexity Theory better and grasp the motivation behind the research done in this area. I am also thankful to Nitin Saxena, Rohit Gurjar and Arpita Korwar for the stimulating discussions we had during their visit to IISc in 2014. The observation in section 4.1 was made in one such discussion. I am extremely thankful to Rohit in particular who was subjected to numerous questions by me. He was patient enough to answer each one of them, however stupid they seemed to be.

I made quite a few friends in IISc. Life in IISc was far more enjoyable because of them. In particular I am thankful to Rafia Sabih and Srinivas Karthik for the engaging discussions we had on Complexity Theory and sometimes on philosophy. Heartfelt thanks to Sabuj, Kiran, Vishesh, Vibhuti, Monika and Srivarun for their constant company.

Acknowledgements

Last but not the least, I am extremely thankful to Umang Mammaniya, Vrishali Shah, Saurabh Shertukde and Neha Nair for providing me constant support and love through all the ups and downs in my life in the past two years. Finally, I owe a lot to my family. I thank my sister for her love, protection, care and valuable career advices during my early student life. I thank my father who inspired me to ask basic questions in science as a child and thus sparked my interest in research. Lastly, I am thankful to my mother for every little thing she has done for me. I would not have been here if she had not inspired me to pursue my interest in maths. I dedicate this work to her.

DEDICATED TO

My mother, Geeta Nair

who understands me even before I do.

Abstract

Consider the problem of Polynomial Identity Testing(PIT): we are given an arithmetic circuit computing a multivariate polynomial over some field and we have to determine whether that polynomial is identically zero or not. PIT is a fundamental problem and has applications in both algorithms and complexity theory. In this work, our aim is to study PIT for the model of multilinear depth three circuits for which no deterministic polynomial time identity test is known. An $n^{O(\log n)}$ time blackbox PIT for set-multilinear depth three circuits (a special kind of multilinear depth three circuits) is known due to [ASS12], [FS13]. To get a better understanding of the problem at hand, we move towards multilinear depth three circuits by considering intermediate circuit classes which encompasse more polynomials than set-multilinear depth three circuits and are 'natural' subclasses of multilinear depth three circuits. One such model is 'superposition of set-multilinear depth 3 circuits'. Our initial observations are:

- There is an $n^{O(\log n)}$ whitebox PIT for superposition of two set-multilinear depth 3 circuits.
- There is a sub-exponential time whitebox PIT for superposition of constantly many set-multilinear depth 3 circuits.

The second observation is subsumed by the recent independent and almost simultaneous work by [OSV15] that gives sub-exponential time hitting set for multilinear depth three circuits.

A recent line of research considers hitting set for Read Once Oblivious Algebraic Branching Programs (ROABP's) which subsumes set-multilinear depth three circuits. An $n^{O(\log n)}$ black box PIT is given for ROABP's of width polynomial in the number of variables in [AGKS14]. It is natural to ask whether this result on ROABP PIT can be used to give efficient (meaning polynomial or quasi-polynomial time) PIT for multilinear depth three circuits. For instance, the result by [OSV15] elegantly uses ROABP PIT as a 'base case' (in a certain sense) to give a sub-exponential time PIT algorithm for multilinear depth three circuits. At this point, we

wondered if multilinear depth three circuits of size s could also be computed by an ROABP of size polynomial in s. If true then this would immediately imply a quasi-polynomial time hitting set for multilinear depth three circuits, which is a long standing open problem in algebraic complexity theory. For instance, it can be shown that any multlinear depth three circuit with top fan-in two and just two variables per linear polynomial can be computed by an ROABP with constant width. But we show in our main result that this is not true for general multilinear depth three circuits that are superpositions of only two set-multilinear depth three circuits.

- There is a polynomial computed by a superposition of two set-multilinear depth 3 circuits with top fan-in just three and size polynomial in the number of variables n, such that any ROABP computing the polynomial has width $2^{\Omega(n)}$.
- There is a polynomial computed by a superposition of three set-multilinear depth 3 circuits with top fan-in just two and size polynomial in the number of variables n, such that any ROABP computing the polynomial has width $2^{\Omega(n)}$.

This means the approach of directly converting a multilinear depth 3 circuit (even a superposition of set-multilinear depth 3 circuits) to an ROABP and then applying the existing PIT for ROABP will not work. However the underlying techniques in [ASS12], [AGKS14] and [OSV15] might still be useful. The proofs of the above lower bounds are based on explicit construction of expander graphs that can be used to design multilinear depth three circuits (in particular superposition of set-multilinear depth 3 circuits) with high '*evaluation dimension*' - a complexity measure that is well suited to capture a 'weakness' of ROABPs.

Contents

List of Figures

Chapter 1

Introduction

Leslie Valiant defined the complexity classes VP and VNP in [Val79a] and [Val79b]. VP and VNP are algebraic analogs of P and NP respectively. It was later proved in [SV85] that P≠NP implies VP≠VNP (more precisely, if 'Permanent' polynomial can be computed by a circuit C of size polynomial in the number of variables n and the field constants appearing in C have bit lengths bounded by $n^{O(1)}$ then P = NP). VP and VNP are in some sense the algebraic analogs of P and NP respectively. Valiant's work inspired many works related to proving lower bounds for specific arithmetic circuit classes, but our knowledge with respect to lower bounds or separating various algebraic complexity classes is far from complete. Specially the answer to the question posed by Valiant, whether VP equals VNP still eludes us. Sometimes we find in complexity theory that proving lower bounds for a specific complexity class has connections to finding an efficient algorithm for a problem. For example connections between finding a sub-exponential time algorithm for polynomial identity testing and proving arithmetic circuit lower bounds are known due to [KI04],[HS80],[Aga05]. We elaborate on this in section 2.2. As part of this thesis we worked on lower bounds and polynomial identity testing for restricted circuit classes.

The problem of Polynomial Identity Testing(PIT) is significant and has profound applications. Here we have an arithmetic circuit computing a multivariate polynomial as input and we need to check whether this polynomial is formally zero (i.e all its coefficients are zero) and not just zero as a function over the field. To understand the difference between these two notions of zero consider the polynomial $x^2 - x$, which is a zero function over $\mathbb{F}_2$ but not the zero polynomial. Checking polynomial identities is a central question in a number of problems. For example, some efficient parallel algorithms for perfect matching are based on testing whether a symbolic determinant is formally zero or not [Lov79, KUW86, MVV87, CRS95]. In complexity

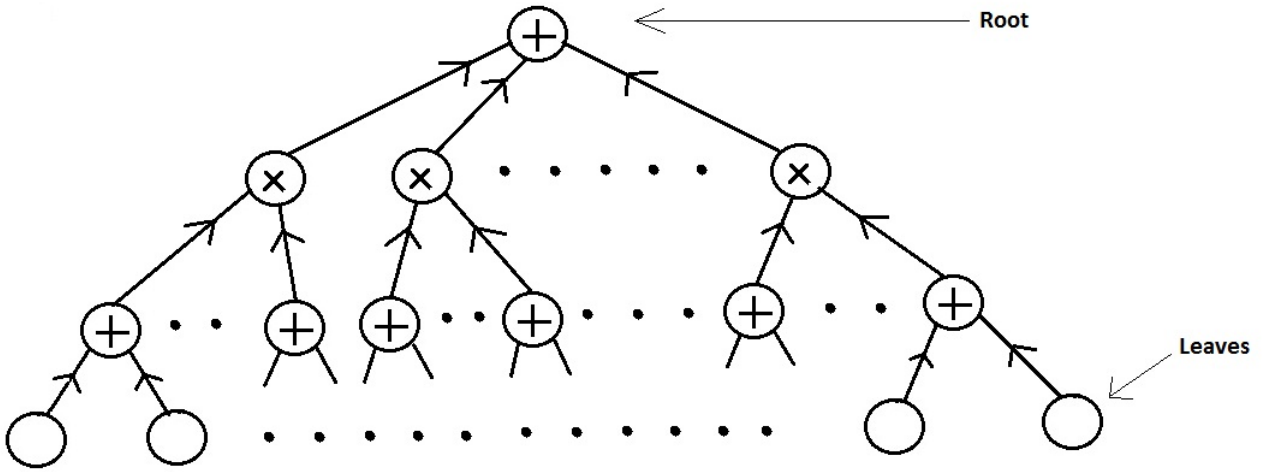

Figure 1.1: $\sum\prod\sum$ depth three circuit

theory identity testing played a major role in results such as IP=PSPACE[LFKN92, Sha92], MIP=NEXPTIME[BFL91] and the proof of the PCP theorem[AS98, ALM+98].

The difficulty in the PIT problem comes from the way the polynomial is given as an input to the algorithm. If the coefficients of the polynomial are given as input then the problem is trivial. In our case the polynomial is given as input in the form of an arithmetic circuit (definition given in section 2.1). The PIT problem is considered with respect to two models. The first one is the black box model in which we can only query the circuit at points over the underlying field and in turn the circuit returns the values of the polynomial at those points. It is clear that in the black box model we must generate a test set (also called a hitting set) for the circuit, namely a set of points, such that if the circuit vanishes on all points in the hitting set then it is computing the zero polynomial. The second setting is the white box model, where we can 'see' the circuit, equivalently the underlying directed acyclic graph of the circuit is given as input. Clearly the white box model is 'easier' of the two.

1.1 Motivation and Related Works

In this work we look at constant depth circuits particularly $\sum\prod\sum$ depth three circuits. This circuit has been formally defined in section 2.1. An example of a $\sum\prod\sum$ depth 3 circuit is shown in fig. 1.1. We can see that in a $\sum\prod\sum$ circuit the root is a sum gate, followed by product gates at level 2. The children of product gates are again sum gates at level 3. The

leaves are input variables or field constants. Similarly we have $\sum\prod\sum\prod$ depth four circuits where we have product gates at level 4. We say $\sum\prod\sum$ circuit or $\sum\prod\sum\prod$ circuit is a homogeneous circuit if every gate in that circuit computes a homogeneous polynomial. The motivation to look at these circuits comes from recent results by [AV08], [GKKS13], [Koi12] and [Tav13]. It follows from the work of [AV08], [Koi12] and [Tav13] that any poly(n) sized arithmetic circuit computing an n variate polynomial f of degree $d = n^{O(1)}$ can be computed by a depth 4 $\sum\prod\sum\prod$ homogeneous circuit of size $exp(O(\sqrt{d}\log n))$. Building on this depth reduction result [GKKS13, Tav13] showed that any poly(n) sized arithmetic circuit computing an n variate polynomial f of degree $d = n^{O(1)}$ can be computed by a depth three $\sum\prod\sum$ circuit of size $exp(O(\sqrt{d}\log n))$ but in this case the homogeneity of the circuit is lost. This implies proving non trivial lower bounds for depth three circuits implies non trivial lower bounds for general arithmetic circuits. It further follows from [AV08],[GKKS13] that a black-box poly-time PIT for depth three circuit implies a quasi-polynomial time blackbox PIT for general circuits. At present it seems we are quite far from a complete derandomization of depth three PIT. Neither whitebox nor blackbox quasi-polynomial time PIT is known for even multilinear depth three circuits (definitions given in section 2.1). The motivation to look at multilinear depth three circuit PIT is the hope that a strong technique to generate hitting sets for multilinear depth three circuits may give us some insight to general depth three circuit PIT. Finding an efficient algorithm for multilinear depth three PIT is also stated as an open problem in [SY10].

Set-multilinear depth three circuits first defined by [NW97] are a subclass of multilinear depth three circuits. We have defined set-multilinear depth three circuits formally in section 2.1. An example of a set-multilinear depth three circuit C in variables $X = \{x_1, x_2, ..., x_n\}$ is given below.

$$C(X) = (1 + x_1 + 3x_2 + 5x_3)(2 + 4x_4 + 3x_5 + x_6)$$

$$+(5 + 2x_1 + x_2 + x_3)(1 + x_4 + 3x_5 + 4x_6)$$

Observe that the variable set X has been partitioned in $C(X)$ into $X_1 = \{x_1, x_2, x_3\}$ and $X_2 = \{x_4, x_5, x_6\}$, such that a linear polynomial contains variables from exactly one of these sets. X_1 and X_2 are called colors of X. [ASS12] gave an $n^{O(\log k)}$ time blackbox identity testing for n-variate polynomials computed by set-multilinear depth three circuits with top fan-in k, where the top fan-in is the number of product gates at level two of the input circuit. Set-multilinear depth three circuits are also a subclass of 'Read-once Oblivious Algebraic Branching Programs' (ROABPs). ROABP has been defined and motivated in section 2.3. An O(wnd) time whitebox identity testing for degree d, n-variate polynomials computed by 'ROABPs' with

width w was given by [RS05]. Recently [AGKS14] gave a $(nw)^{O(\log n)}$ time blackbox identity test for n-variate polynomials computed by 'ROABPs' with width w. Since the set-multilinear depth three circuit model is a subclass of ROABPs, these identity tests hold for set-multilinear depth three circuits too. At this point it is natural to ask whether we can use the identity testing algorithms of ROABPs for multilinear depth three circuits. Indeed [OSV15] use the hitting set for ROABPs given by [AGKS14] to give an $2^{O(n^{\frac{2}{3}(1+\delta)})}$ size hitting set for multilinear depth three circuits of size 2^{n^δ}. A useful notion in this context is 'superposition of two or more set-multilinear depth three circuits', which is defined formally in chapter 3. We give an example of superposition of two set-multilinear depth three circuits below.

$$C(X,Y) = (1 + 3x_1 + 5y_2)(4 + x_2 + y_1) + (6 + 9x_1 + 4y_1)(2 + 5x_2 + 3y_2)$$

The variable sets X and Y are completely disjoint and are called the *base sets* of $C(X,Y)$. $C(X,Y)$ is a set-multilinear depth three circuit in X variables with colors $\{x_1\}$ and $\{x_2\}$, when projected on X variables (i.e after putting the Y variables to zero). Similarly $C(X,Y)$ is a set-multilinear depth three circuit in Y variables with colors $\{y_1\}$ and $\{y_2\}$, when projected on Y variables. A multilinear depth 3 circuit can be trivially viewed as a superpostion of n set-multilinear depth 3 circuits with single variable in every base set. [OSV15] view multilinear depth three circuits as a superposition of sub-exponentially many set-multilinear depth three circuits but with some leeway in each product gate. In a product gate some linear polynomials may have variables from two colors of the same base set, but the number of such linear polynomials is bounded. They achieve hitting set for multilinear depth three circuits by finding the sub-exponentially many base sets that satisfy this criteria and then use the hitting set given by [AGKS14] on each of these base sets. The main step of the algorithm in [OSV15] is to find the sub-exponentially many base sets in sub-exponential time. We show that it is at least NP Hard to find base sets when circuit is a superposition of more than two set-multilinear depth three circuits.

At this juncture we wondered whether we can use hitting sets for ROABPs to give hitting sets for superposition of a small (constantly many) number of set-multilinear depth three circuits. If we could find the base sets (as done in [OSV15]) efficiently then [AGKS14] could be applied immediately. It turns out that the problem of finding base sets from a given multilinear depth three circuit that is a superposition of three or more set-multilinear depth three circuits is an NP-hard problem, as we show in section 3.2. This rules out the possibility of finding the base sets. But, could it be possible that we can reduce reduce $n^{O(1)}$ size superposition

of constantly many set-multlinear depth three circuits to $n^{O(1)}$ size ROABPs? If so then we can use [AGKS14]'s result to give a quasi-polynomial time blackbox identity test for superposition of constantly many set-multlinear depth three circuits. However, we show that such a direct approach of reducing to ROABP does not work. Our main result is an exponential lower bound for ROABP, where the hard polynomial is computed by a poly-sized superposition of constantly many set-multilinear depth three circuits. It is here that we use explicit construction of expander graphs, to make the hard polynomial explicit. The polynomial against which the lower bound is shown for ROABP is designed from an explicit 3-regular expander graph and its 'hardness' is argued using a complexity measure known as 'evaluation dimension' (defined in section 2.4). The hard polynomial so constructed turns out to be a superposition of constantly many set-multilinear depth three circuits and simultaneously a sum of constantly many set-multilinear depth three circuits. This model has been defined in section 3.3. For such a model we give a quasi-polynomial time hitting set by extending the shift and rank concentration technique used in [ASS12].

1.2 Contributions of thesis

As a stepping stone towards PIT for multilinear depth three circuits, we study the model-superposition of two or more set-multilinear depth three circuits. This model has been described in detail in chapter 3. We state our initial observations for this model below.

Theorem 1.2.1 *Given a circuit C which is a superposition of 2 set-multilinear circuits C_1 and C_2 on unknown base sets X and Y respectively, we can perform whitebox PIT for C in $n^{O(\log n)}$ time where $n = |X| \cup |Y|$.*

Theorem 1.2.1 is proved by finding base sets X' and Y', such that C is a superposition of two set-multilinear circuits C_1' and C_2' on the base sets X' and Y' respectively. We will elaborate on this in section 3.1. However we also show that the same strategy of finding base sets from C is unlikely to work for superposition of more than two set-multilinear circuits. This is due to the following theorem.

Theorem 1.2.2 *Given a circuit C which is a superposition of t set-multilinear circuits $C_1, C_2, ..., C_t$ on unknown base sets $X_1, X_2, ..., X_t$ respectively, finding t base sets $X_1', X_2', ..., X_t'$ such that C is a superposition of t set-multilinear circuits $C_1', C_2', ..., C_t'$ respectively on base sets $X_1', X_2', ..., X_t'$ respectively is NP-Hard when $t > 2$.*

The proof of theorem 1.2.2 is given in section 3.2. The NP-Hardness result also leads us to the following result via an approximation algorithm to find base sets.

Theorem 1.2.3 *Given a circuit C which is a superposition of t set-multilinear circuits $C_1, C_2, ..., C_t$ on unknown base sets $X_1, X_2, ..., X_t$, we can perform whitebox PIT for C in $exp(O(n^{1-\frac{3}{t+1}}.\text{poly}(\log n) + \log t))$ time.*

We use an approximation algorithm given by [KMS98] to prove theorem 1.2.5 in section 3.2. It is easy to see that when t is a constant, theorem 1.2.5 yields a sub-exponential time whitebox PIT. We would like to note that theorem 1.2.3 is in a way superseded by the recent, independent work of [OSV15]. As mentioned in section 1.1, they gave a hitting set for general multilinear depth three circuits with running time roughly $exp(n^{2/3})$. We note in the passing that the time complexity of theorem 1.2.3 is better than $exp(n^{2/3})$ for $t < 9$.

We now state the main result of this thesis.

Theorem 1.2.4 *There exists explicit polynomials F_1 computable by a superposition of two set-multilinear depth three circuits with top fan-in just three and F_2 computable by a superposition of three set-multilinear depth three circuits with top fan-in just two such that every ROABP computing F_1 or F_2 has width $2^{\Omega(n)}$, where n is the number of variables in F_1 or F_2.*

We prove theorem theorem 1.2.4 for F_1 in subsection 4.2.1 and for F_2 in subsection 4.2.2. In both the cases the hard polynomials i.e F_1 and F_2 are designed using an explicit degree three expander graph. We observed that the hard polynomials F_1 and F_2 are superpositions of constantly many set-multilinear depth three circuits and simultaneously a sum of constantly many set-multilinear depth three circuits. Consider the following example:

$$C(X,Y) = (1 + 3x_1 + 5y_2)(4 + x_2 + y_1) + (9 + 6x_1 + 4y_2)(3 + 2x_2 + 5y_1)$$

$$+(6 + 9x_1 + 4y_1)(2 + 5x_2 + 3y_2) + (3 + 6x_1 + 9y_1)(5 + 8x_2 + 2y_2)$$

$C(X,Y)$ is a superposition of two set-multilinear depth three circuits with base sets $X = \{x_1\} \cup \{x_2\}$ and $Y = \{y_1\} \cup \{y_2\}$. But $C(X,Y)$ is also a sum of two set-multlinear depth three circuits with $\{x_1, y_2\}$, $\{x_2, y_1\}$ being the colors in the first set-multilinear depth three circuit (corresponding to the first two products) and $\{x_1, y_1\}$, $\{x_2, y_2\}$ being the colors in the second set-multilinear depth three circuit (corresponding to the last two products). We give a sub-exponential time hitting set in section 3.3, for such a model i.e a subclass of multilinear depth three circuits which is a superposition of constantly many set-multilinear depth three circuits and simultaneously a sum of constantly many set-multilinear depth three circuits.

Theorem 1.2.5 *Given a circuit C which is a superposition of m set-multilinear depth three circuits $C_1, C_2, ..., C_m$ on base sets $X_1, X_2, ..., X_m$ and simultaneously a sum of k set-multilinear depth three circuits $C'_1, C'_2, ..., C'_k$, we can find a hitting set for C in time $l^{km(\log l+1)} \cdot n^{O(m \log l)}$ time, where l is the bound on the top fan-in of circuit C.*

In another recent work, [GKST15] gave a $(w^{k2^k} nd)^{O(k)}$ time whitebox test for degree d, n-variate polynomials computed by a sum of k ROABPs each of width less than w. They also gave a $(wnd)^{k2^k \log(wnd)}$ time hitting set for the same model. Hence when k is a constant this yields a polynomial time white-box test and a quasi-polynomial time blackbox test for degree d, n-variate polynomials computed by a sum of k ROABPs, each of width $n^{O(1)}$. Observe that in both cases: whitebox and blackbox identity testing, dependence on k is doubly exponential. In contrast we have a singly exponential dependence on k as stated in theorem 1.2.5, but our model is a superposition of constantly many set-multilinear depth three circuits in addition to being a sum of set-multilinear depth three circuits.

1.3 Organization of thesis

In chapter 2 we have defined various models that compute polynomials. In this chapter we also talk about the connections between polynomial identity testing and lower bounds for arithmetic circuits. We also introduce tools like evaluation dimension and expander graphs in this chapter. We define the model: superposition of set-multilinear depth three circuits and then give results related to identity testing for this model in chapter 3. In chapter 4 we give lower bounds for Read-once Oblivious Algebraic Branching Programs computing polynomials that are superposition of set-multilinear depth three circuits. Finally in chapter 5 we discuss future directions and the possibility of extending this work.

Chapter 2

Preliminaries

Arithmetic circuits are algebraic analogs of boolean circuits. They compute polynomials over aribitrary fields. Below we define arithmetic circuits and look at special classes of circuits called multilinear depth three circuits and set-multilinear depth three circuits.

2.1 Arithmetic Circuits

An arithmetic circuit is a standard model to compute a polynomial. Arithmetic circuits take inputs $X = \{x_1, \ldots, x_n\}$ variables and perform additions and multiplications on them, to output a polynomial (or a set of polynomials) in X variables. We formally define arithmetic circuits below.

Definition 2.1.1 (Arithmetic Circuit) *An arithmetic circuit C over a field $\mathbb{F}$ and the set of variables X is a directed acyclic graph. The leaves of the graph (with indegree equal to zero) are the input nodes and are labelled by input variables or field elements. The other gates are labelled by $\times$ or $+$ and are referred to as product gates or sum gates respectively. For two gates u and v in C, if (u, v) is an edge in C, then u is called a child of v, and v is called a parent of u. Every gate in C computes a polynomial as follows:*

- *The leaf nodes compute an input variable or the field element that they are labelled with.*
- *If u is a sum gate, then u computes a sum of the polynomials computed by its children.*
- *Similarly if u is a product gate, then u computes a product of the polynomials computed by its children.*

Every gate of outdegree zero is called the output gate. The polynomials computed by the output gates are the polynomials computed by the circuit C. When C has a single output gate we call the output gate as the root of the circuit C.

From here on if not mentioned explicitly, we assume the circuit C has a single output gate. We study two natural measures of complexity associated with arithmetic circuits: 'size' and 'depth' of circuit. The size of a circuit captures 'the number of elementary operations: additions and multiplications required to compute a polynomial' whereas the depth captures, 'how fast we can compute the polynomial in parallel'. The most classical question along this line is, 'what is the size of the smallest circuit or the depth of the shortest circuit computing a specific polynomial'. We now formally define the size and depth of an arithmetic circuit.

Definition 2.1.2 (Size and Depth) *The size of a circuit C, denoted by $|C|$ is the number of nodes in C. The depth of a ciruit C denoted by depth(C) is the length of the longest path from the root of the ciruit to a leaf.*

A polynomial is called *homogeneous*, if all the monomials in the polynomial have the same degree. An arithmetic circuit is called a *homogeneous circuit* if all the gates in that circuit compute a homogeneous polynomial. Before we look at special classes of arithmetic circuits, we define arithmetic formulas.

Definition 2.1.3 (Arithmetic Formula) *An arithmetic formula ϕ is an arithmetic circuit where the outdegree of each node is 1.*

Similar to arithmetic circuits we can associate, complexity measures of size and depth with arithmetic formulas. The definitions for size and depth of an arithmetic formula ϕ are similar to that of an arithmetic circuit. At this point we would like to make a small remark below.

Remark: When polynomial sized circuits - circuits whose size are bounded by a polynomial in the number of variables; and constant depth circuits - circuits whose depth are bounded by a constant independent of $|X|$ are considered, we can reduce the circuit to an arithmetic formula whose size is still bounded by a polynomial in $|X|$ and the depth is constant. Hence for constant depth, polynomial sized arithmetic circuits and polynomial sized arithmetic formulas are equivalent.

Now we define special classes of circuits called depth three and depth four circuits.

Definition 2.1.4 (Depth three circuits) *A depth three circuit is an arithmetic circuit with depth equal to three. The root - 'output gate' is a sum gate at level 1. All children of the root are product gates at level 2, and all children of these product gates are sum gates at level 3. These are also called $\sum\prod\sum$ circuits.*

Definition 2.1.5 (Depth four circuits) *A depth four circuit is an arithmetic circuit with depth equal to four. The root- 'output gate' is a sum gate at level 1. All children of the root are product gates at level 2, all children of these product gates are sum gates at level 3, and all children of these sum gates are again product gates at level 4. These are also called* $\sum\prod\sum\prod$ *circuits.*

A polynomial is multilinear, if the individual degree of each variable is at most 1 in all the monomials. An arithmetic circuit is multilinear if all the gates in that circuit compute multilinear polynomials. Below we define 'multilinear depth three circuits'.

Definition 2.1.6 (Multilinear depth three circuits) *A circuit* $C = \sum_{i=1}^{k}\prod_{j=1}^{d} l_{ij}(X_j^i)$ *is called a multilinear depth 3 circuit in* X *variables where* $X_1^i, X_2^i, ..., X_d^i$ *is a disjoint partitioning of* X *variables observed in the* i*th product gate, and* l_{ij}*'s are linear polynomials in* X_j^i *variables computed by the sum gates at layer 3.*

An example of a multilinear depth 3 circuit C in the X variables with $X_1^1 = \{x_1, x_2, x_3\}$, $X_2^1 = \{x_4, x_5, x_6\}$, $X_1^2 = \{x_1, x_3, x_5\}$ and $X_2^2 = \{x_2, x_4, x_6\}$ is given below

$$C(X_1) = (1 + 3x_1 + x_2 + 2x_3)(3 + 6x_4 + 8x_5 + 7x_6)$$

$$+(4 + x_1 + 2x_3 + 6x_5)(7 + 3x_2 + x_4 + 8x_6)$$

'k' in the above definition is called the top fan-in of circuit C and it is equal to number of product gates in layer 2 of circuit C. Set-multilinear depth three circuits (as defined below) form a subclass of multilinear depth three circuits, and they are closely related to tensor ranks [Raz10].

Definition 2.1.7 (Set-Multilinear Depth 3 Circuit) *A circuit* $C = \sum_{i=1}^{k}\prod_{j=1}^{d} l_{ij}(X_j)$ *is called a set-multilinear depth 3 circuit in* X *variables where* X *is the disjoint union of* $X_1, X_2, ..., X_d$ *and* l_{ij}*'s are linear polynomials in variables* X_j*.* $X_1, X_2, ..., X_d$ *are called colors of set* X*. If* $|X_j| = 1$ *we say it is a singleton color. When* X_j *is a singleton color for all* $j \in [d]$ *we say* X *has singleton colors and* C *is set-multilinear depth 3 circuit with singleton colors.*

Here is an example of a set-multilinear depth three circuit C in variables X with colors $X_1 = \{x_1, x_2, x_3\}$ and $X_2 = \{x_4, x_5, x_6\}$

$$C(X) = (1 + x_1 + 3x_2 + 5x_3)(2 + 4x_4 + 3x_5 + x_6)$$

$$+(5 + 2x_1 + x_2 + x_3)(1 + x_4 + 3x_5 + 4x_6)$$

The sets $X_1, X_2, ..., X_d$ remain unchanged for all product gates in layer 2, i.e the disjoint partitioning of $|X|$ variables observed in every product gate is same. This is precisely the difference between set-multilinear depth 3 circuits and multilinear depth 3 circuits. A multilinear depth 3 circuits might have separate disjoint partitioning of X variables across product gates in layer 2.

A convention - In this report sometimes we have referred to set-multilinear (multilinear) depth 3 circuits as simply set-multilinear (multilinear) circuits.

2.2 Connections between polynomial identity testing and arithmetic circuit lower bounds

One of the goals of algebraic complexity theory is to separate the complexity classes VP and VNP. It was shown by Valiant in [Val79a] and [Val79b] that permanent is VNP complete. VP and VNP can be separated by proving a super polynomial lower bound on the size of any circuit computing the permanent. Kabanets and Impagliazzo showed that an existence of sub-exponential time algorithm for blackbox PIT implies either an arithmetic circuit lower bound for the permanent or a boolean circuit lower bound for NEXP in [KI04].

Theorem 2.2.1 ([KI04]) *If Permanent is computable by polynomial size arithmetic circuits over $\mathbb{Z}$ then either*

- *NEXP $\subseteq$ P/poly*
  or
- *There is a sub-exponential time algorithm for blackbox PIT*

In the same paper [KI04], Kabanets and Impagliazzo also showed that lower bounds for arithmetic circuits imply efficient deterministic algorithms for blackbox PIT.

Theorem 2.2.2 ([KI04]) *Let $\mathbb{F}$ be a large enough field (of size at least some polynomial in n). Under the assumption that there exists a multilinear m-variate polynomial $f(x_1, x_2, ..., x_m)$ cannot be computed by arithmetic circuits over $\mathbb{F}$ with size less than $s(m)$, where $s : \mathbb{N} \to \mathbb{N}$ is monotonically increasing function, there exists a deterministic blackbox PIT algorithm for arithmetic circuits in n variables, of polynomial degree over $\mathbb{F}$, that runs in time $\exp(s^{-1}(poly(r))^2)$ for size $r = r(n) \geq n$ circuits.*

To understand theorem 2.2.2 better we will work out an example. Say $s(m) = 2^m$, then $s^{-1}(t) = \log t$, hence we have a blackbox PIT algorithm for poly(n) sized circuits that runs in

$\exp((\log n)^2)$ time. [HS80, Aga05] showed the following connection between blackbox identity testing and arithmetic circuit lower bounds.

Theorem 2.2.3 ([HS80, Aga05]) *If there exists a blackbox identity testing algorithm for size s circuits, that runs in time $T(s)$, where $T : \mathbb{N} \to \mathbb{N}$ is a monotonically increasing function, then there exists an n variate polynomial F whose coefficients are computable in PSPACE such that every arithmetic circuit computing F has size at least $T^{-1}(c^n)$ where c is a constant greater than 1 and T^{-1} is the inverse function of T.*

Blackbox identity testing thus seems a promising approach to proving lower bounds.

2.3 Algebraic Branching Programs

Algebraic Branching Programs (ABPs) were first defined and used in [Nis91] to show exponential lower bounds on the size of non-commutative ABPs computing the non-commutative permanent and determinant polynomials.

Definition 2.3.1 (Algebraic Branching Program) *An Algebraic Branching Program(ABP) in the variables $X = \{x_1, x_2, ..., x_n\}$ is represented by a directed acyclic graph with a source vertex s and a sink vertex t. Between s and t we have $(d-1)$ sets or layers of vertices $V_1, V_2, ..., V_{d-1}$. The width of an ABP is the maximum number of vertices in any of the $(d-1)$ layers. The source vertex s has only outgoing edges. Similarly the sink vertex t has only incoming edges. All the outgoing edges from the source vertex s are incident to the vertices in the layer V_1 and all the edges incident to the sink vertex t originate from the vertices in V_{d-1}. All the other edges in an ABP are such that an edge starts from a vertex in V_i and is directed to a vertex in V_{i+1} where V_i belongs to the set $\{V_1, V_2, ..., V_{d-1}\}$. The edges in an ABP are labelled by polynomials over the base field. The weight of the path between any two vertices u and v in an ABP is computed by taking the product of the edge labels on the path from u to v. An ABP computes the sum of the weights of all the paths from s to t.*

In a 'non-commutative ABP', the product of the labels is in the order of the path, from source to sink. The ABP itself computes the sum of all such polynomials. Although the 'ABP' model might seem unnatural it is quite significant in algebraic complexity theory - it is directly linked to the complexity of iterated matrix multiplication. In particular derandomizing PIT for ABPs is important. Specifically, the results of [BOC92] imply that blackbox PIT, even for width-3 ABPs, would imply derandomization of PIT for general arithmetic circuits of logarithmic depth and in general, a quasi-polynomial time derandomization of PIT for polynomial sized arithmetic circuits. Further Saha, Saptharishi and Saxena [SSS09] showed that black-box derandomization

of PIT of width-2 ABPs implies derandomization of PIT for depth-3 circuits. We work with a restricted class of ABPs called 'Read-once Oblivious ABP'(ROABP) introduced in [FS13]. We define and motivate ROABP in the next subsection.

2.3.1 Read-Once Oblivious Algebraic Branching Program

Definition 2.3.2 (Read-Once Oblivious Algebraic Branching Program) *A Read-Once Oblivious Algebraic Branching Program(ROABP) has an associated permutation $\pi : [n] \to [n]$ of the variables X. In the case of an ROABP the number of variables is equal to one plus the number of layers, i.e $n = 1 + (d-1) = d$. All the outgoing edges from the source vertex s are labelled with univariate polynomials in the variable $x_{\pi(1)}$ and all the incoming edges to the sink vertex s are labelled with univariate polynomials in the variable $x_{\pi(n)}$. The labels associated with each edge e from a vertex in V_i to a vertex in V_{i+1} is an univariate polynomial in the variable $x_{\pi(i+1)}$.*

The importance of the ROABP model stems from its connection to the 'L vs RL' question in the boolean world. The PIT problem for ROABP is the algebraic analog of the 'L vs RL' question. Forbes and Shpilka [FS13] gave a quasi-polynomial time blackbox PIT algorithm for ROABPs, when the associated permutation π (also known as the ordering of the variables) is known. The hitting set given by [FS13] is very close in nature to Nisans [Nis92] pseudo-random generator (PRG) for read-once oblivious (boolean) branching programs, which are a non-uniform version of randomized log-space Turing machines.

2.4 Evaluation Dimension

A general framework for proving lower bounds for a model T against a polynomial f involves using a measure μ, that is a function mapping polynomials to real numbers. We show that μ for a polynomial computed by the model T is less than some quantity specific to T; for example in case of ROABPs we might show it is less than the width of ROABP or for multlinear depth three circuits we might show it is less than top fan-in of circuit, and for polynomial f we show it is at least some value, which is a function in the number of variables in f. This would yield a lower bound on the quantity specific to T. The measure we use in this work is called the evaluation dimension. This concept has been used before in [RY09] and [FS13].

Definition 2.4.1 (Evaluation Dimension) *The evaluation dimension of a polynomial $h \in \mathbb{F}[X]$ with respect to a set $S \subseteq [X]$ denoted as* $\text{Evaldim}_S[h(X)]$ *is defined as*

$$\dim(\text{span}\{h(X)|_{\forall x_j \in S \ x_j = \alpha_j} : \forall x_j \in S \ \alpha_j \in \mathbb{F}\})$$

We will be using the following claim about evaluation dimension in the proofs for lower bounds given in section 4.2

Claim 2.4.1 *Suppose $h_1(X), h_2(X), ..., h_m(X)$ are $\mathbb{F}$-linearly independent polynomials in the variables $X = \{x_1, x_2, ..., x_n\}$ where $m = 2^n$. If $Y = \{y_1, y_2, ..., y_n\}$ are n variables different from $\{x_1, x_2, ..., x_n\}$ then*

$$\text{Evaldim}_Y[\sum_{S_i \subseteq [n]} y_{S_i} h_i(X)] = m$$

where for $S \subseteq [n]$ $y_S = \prod_{j \in S} y_j$

Proof: We will prove the claim in 2 parts. First we will show that $\text{Evaldim}_Y[\sum_{S_i \subseteq [n]} y_{S_i} h_i(X)] \geq m$. Finally to complete the proof we will claim that $\text{Evaldim}_Y[\sum_{S_i \subseteq [n]} y_{S_i} h_i(X)] \leq m$. These two inequalities together would imply that $\text{Evaldim}_Y[\sum_{S_i \subseteq [n]} y_{S_i} h_i(X)] = m$

Let us prove the first part. Consider the $\mathbb{F}$-evaluation of $\{y_1, y_2, ..., y_n\}$ over the following set of points, $\forall S_i \subseteq [n]$ (if $j \in S_i$ put $y_j = 1$ else $y_j = 0$). It is easy to see that there are m such points. Evaluating over these m points and then taking appropriate linear combinations gives us m polynomials $h_1(X), h_2(X), ...$ and $h_m(X)$. These m polynomials are given to be $\mathbb{F}$-linearly independent.
$\Rightarrow \text{Evaldim}_Y[\sum_{S_i \subseteq [n]} y_{S_i} h_i(X)] \geq m$

For the second part observe that any $\mathbb{F}$-evaluation of the polynomial $\sum_{S_i \subseteq [n]} y_{S_i} h_i(X)$ over the Y variables is a linear combination of the m polynomials $h_1(X), h_2(X), ...$ and $h_m(X)$.
$\Rightarrow \text{Evaldim}_Y[\sum_{S_i \subseteq [n]} y_{S_i} h_i(X)] \leq m$. Hence $\text{Evaldim}_Y[\sum_{S_i \subseteq [n]} y_{S_i} h_i(X)] = m$ □

A Quick Comparison: Evaluation dimension has been used earlier (like in [RY09]) to prove a lower bound on multilinear depth 3 circuits. They show that evaluation dimension for a single product gate is 'low' with respect to a randomly chosen subset of variables with a high probability. Applying union bound, the existence of a set such that evaluation dimensions of all the product gates are low with respect to that set is established. Finally they use a polynomial with high evaluation dimension with respect to every set to show it cannot be computed by a poly-sized multilinear depth 3 circuit. In contrast, we would like the hard polynomial (against which an ROABP lower bound is shown) to be computed by a small multilinear depth three circuit, and hence our approach is to show that with respect to every subset of variables of a certain size, the evaluation dimension of the hard circuit is somewhat 'high'. This forces us to look at all the product gates at a time instead of a single product gate as done before,

because for a single product gate there always exists a set with respect to which the evaluation dimension is 1.

2.5 Expander Graphs

Expander graphs are sparse but well connected graphs. They have found applications in a lot of areas due to their 'pseudorandom' properties. Expander graphs are defined in terms of expansion property they possess. Below we define edge boundary and edge expansion of an undirected d-regular graph G.

Definition 2.5.1 ([HLW06]) *Let $G = (V, E)$ be an undirected d-regular graph with self loops and multiple edges. For $S, T \subseteq V$ let $E(S,T) = \{(u,v)|(u \in S), (v \in T), (u,v) \in E\}$. Then*
1. The 'Edge boundary' of a set S, denoted as $\partial S = E(S, \overline{S})$. This is the set of edges emanating from the set S to its compliment.
2. The 'Edge expansion' of G denoted $h(G)$ is defined as:

$$h(G) = min_{S:\ |S| \leq \frac{n}{2}} \frac{|\partial S|}{|S|}$$

We can have an alternate definition of expansion called 'vertex expansion' $\phi(G)$. Here we count the number of neighbouring vertices of vertex sets S rather than the number of outgoing edges. Expander graphs are defined with respect to edge expansion as follows.

Definition 2.5.2 (Expander graph) *A d-regular graph family $\{G_1, G_2, ...\}$ where G_i has i vertices is an 'Expander graph family' if for all G_i in the family $\{G_1, G_2, ...\}$, $h(G_i) > 0$.*

It was first shown by [Pin73] and independently by [KB93], that any random graph satisfies the properties of expander graphs with high probability. Explicit constructions of expander graphs are also known [Mar73],[GG81],[LPS88] and [Mar88]. They are non trivial and use algebraic and group-theoretic techniques. More combinatorial constructions are known using the zig-zag product [ORW02]. In subsection 2.5.1 we look at expander graphs from a different perspective by looking at the second eigenvalue of the adjacency matrix of an expander graph.

2.5.1 Spectral gap and its connections to edge expansion

Let G be a connected d-regular undirected graph with n vertices. The adjacency matrix of G is an $n \times n$ matrix such that (u, v) entry in G represents the number of edges in G between u and v. Let A_G be the adjacency matrix of G. Since A_G is real and symmetric, it has n real eigenvalues. Say $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ are the n eigenvalues of A_G. The eigenvalues of A_G are

referred to as the spectrum of G. The connection between the second eigenvalue 'λ_2' and the edge expansion $h(G)$ is given by the following theorem.

Theorem 2.5.1 (Cheeger's inequality) *Let G be a d-regular graph with spectrum $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. Then*

$$\frac{d-\lambda_2}{2} \leq h(G) \leq \sqrt{2d(d-\lambda_2)}$$

Cheeger [Che70] and independently Buser [Bus82] proved theorem 2.5.1 in the continuous case. In the discrete case, it was proved by Dodziuk [Dod84] and independently by Alon and Milman [AM85] and Alon [Alo86]. Since G is a d-regular graph we know $d = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq -d$. $(d-\lambda_2)$ is referred to as the 'spectral gap' for G. Spectral gap provides an estimate on the edge exapansion of G as stated in the above theorem. More the spectral gap, more the expansion.

2.5.2 Explicit construction of degree three expanders

Let G be a connected d-regular undirected graph with n vertices. Let A_G be the adjacency matrix of G and let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ be the n eigenvalues of A_G. Since G is a d-regular graph we know $d = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq -d$. We define $\lambda(G)$ as follows:

$$\lambda(G) = max_{|\lambda_i|<d} \ |\lambda_i|$$

Definition 2.5.3 (Explicit Expander Graph Family) *Let $G = \{G_1, G_2, ..., G_i, ...\}$ be a family of d-regular expander graphs such that the number of vertices in G_i: n_i, is bounded by a polynomial in i. G is*

- *Mildly Explicit if there exists an algorithm that takes input $i \in \mathbb{N}$ constructs G_i in time polynomial in the size of G_i.*
- *Strongly explicit if there exists an algorithm that takes input $i \in \mathbb{N}, j \in \{1, ..., n_i\}$ and $k \in \{1, ..., d\}$ and outputs the kth neighbour of the jth vertex in graph G_i in time polynomial in $(\log i + \log n_i + \log d)$ (size of the input).*

In this work we use mildly explicit 3-regular expander graphs. We prove two similar but equally important lower bounds in section 4.2. In both lower bounds we start with a 3-regular expander graph and reduce it to a polynomial that can be computed by a small multilinear depth three circuit but every ROABP computing it requires exponential width. [HLW06] mentions an explicit construction of 'a family of 3-regular p-vertex graphs' for every prime p. The vertices of the graph correspond to elements in $\mathbb{Z}_p$. Each vertex x in this graph is connected to $x+1$, $x-1$ and to its inverse x^{-1} (operations are $\mod p$ and inverse of 0 is defined as 0). The

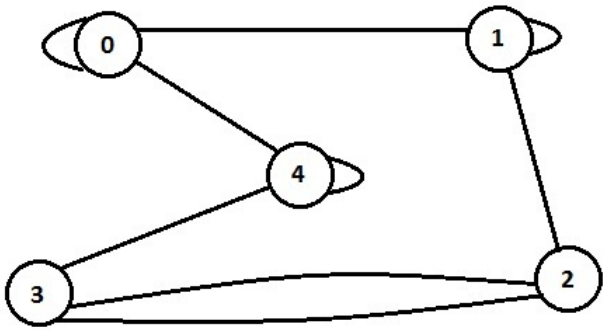

Figure 2.1: Degree three expander graph corresponding to $\mathbb{Z}_5$

expander graph corresponding to $\mathbb{Z}_5$ is shown in fig. 2.1. If G is a graph constructed as above it can be shown that $\lambda(G) < 1 - \frac{1}{10^4}$. For the proof we refer the readers to [HLW06], section 11.1.2. From theorem 2.5.1 we know that if G is a 3-regular exapnder graph with $\lambda(G) < 1 - \frac{1}{10^4}$, then $h(G) > \frac{2+\frac{1}{10^4}}{2}$.

2.5.3 Double Cover

In the reductions stated in section 4.2, we work with a bipartite 3-regular expander graph. We do this by taking the double cover of the explicit 3-regular expander graph mentioned in the previous section.

Definition 2.5.4 (Double Cover) *The double cover of a graph $G = (V, E)$ is the bipartite graph $H = (L \cup R, E_H)$ where $L = R = V$ and there are edges between $u_L \in L$ and $v_R \in R$, and $u_R \in R$ and $v_L \in L$ iff there is an edge between $u \in V$ and $v \in V$.*

We will illustrate double cover of a graph with an example. Let the given graph $G = (V, E)$ be as given in fig. 2.2. G is a 3-regular graph. The double cover of G is shown in fig. 2.3. As we can see H: the double cover of G, is bipartite and 3-regular. In claim 2.5.1 we show that if a d-regular graph G has a large spectral gap then the double cover of G also has a large spectral gap.

Claim 2.5.1 *For a given d-regular graph $G = (V, E)$ if $\lambda(G)$ is less than t then for the double cover H of G, $\lambda(H)$ is less than t.*

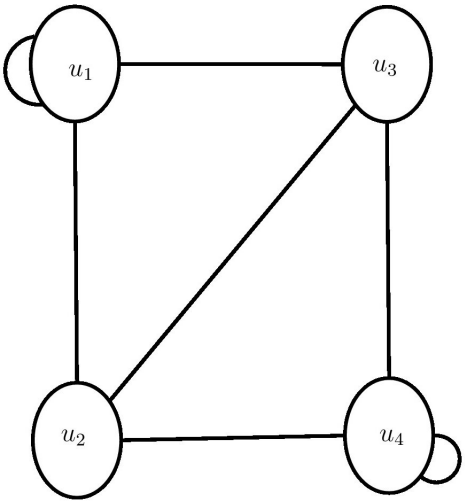

Figure 2.2: 3-regular graph G

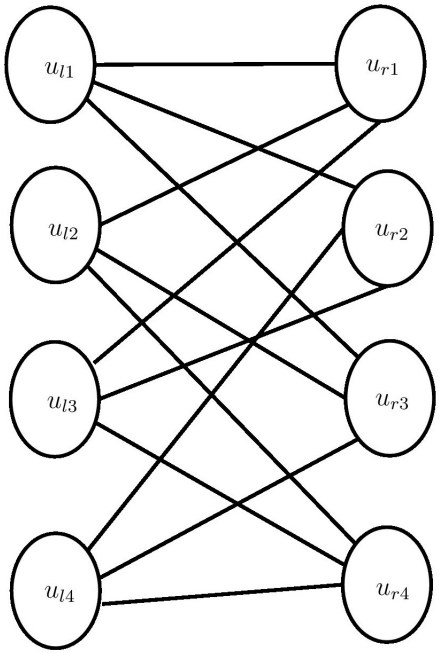

Figure 2.3: Bipartite double cover of the graph in fig. 2.2

Proof: Let $G = (V, E)$ be the given d-regular graph with $|V| = n$ and $H = (V_H, E_H)$ be its double cover, thus $|V_H| = 2n$. From the construction of double cover it is clear that H is a d-regular bipartite graph. Let A_G be the adjacency matrix of G and $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ be the n eigenvalues of A_G. Since G is a d-regular graph we know $\lambda_1 = d$. Given that, $\lambda(G) < t$. It is easy to see that if λ is an eigenvalue of G then λ and $-\lambda$ are eigenvalues of H. This implies $\lambda(H) < t$. □

From claim 2.5.1 we know the double cover H of a graph G constructed as stated in subsection 2.5.2 has $\lambda(H) < 1 - \frac{1}{10^4}$. Hence $h(H) > \frac{2+\frac{1}{10^4}}{2}$. We will use the following lemma about bipartite graphs in section 4.2 to reduce an expander graph to the required polynomial.

Lemma 2.5.1 *A 3-regular bipartite graph can be split into 3 edge disjoint perfect matchings.*

Lemma 2.5.1 follows directly from Hall's marriage theorem [Hal35].

Chapter 3

Superposition of set-multilinear depth 3 circuits

We study the model 'superposition of set-multilinear depth three circuits' in this chapter.

Definition 3.0.5 (Superposition of Set-Multilinear Depth Three Circuits) *A multilinear depth 3 circuit C is a superposition of t set-multilinear depth three circuits over disjoint sets of variables $X_1, X_2, ..., X_t$ if for every $i \in [t]$, C is a set-multilinear depth three circuit in X_i variables over the field $\mathbb{F}(X_1, .., X_{i-1}, X_{i+1}, .., X_t)$. $X_1, ..., X_t$ are called the base sets of C. Further, we assume that X_i has singleton colors for every $i \in [t]$.*

Here is an example of a depth 3 multilinear circuit that is a superposition of two set-multilinear depth 3 circuits C in the base sets X, Y with colors $\{x_1\}$ and $\{x_2\}$ for X and $\{y_1\}$ and $\{y_2\}$ for Y.

$$C(X,Y) = (1 + 3x_1 + 5y_2)(4 + x_2 + y_1)$$

$$+(6 + 9x_1 + 4y_1)(2 + 5x_2 + 3y_2)$$

Naturally a multilinear depth 3 circuit C over X variables can be trivially viewed as a superposition of n set-multilinear depth three circuits with each variable corresponding to a distinct base set. Before we give results for this model we define support of a monomial.

Definition 3.0.6 (Support of a monomial) *The support of monomial $\eta = x_1^{a_1} x_2^{a_2} ... x_n^{a_n}$ denoted as* supp*(η) is the number of variables x_i such that the degree corresponding to it, i.e a_i is non zero.*

We give results for the superposition of set-multilinear depth three circuit model below.

3.1 Whitebox PIT for superposition of two set-multilinear depth three circuits

[ASS12] gave a quasi-polynomial time hitting set for set-multilinear depth three circuits. Below we briefly explain the shift and rank concentration technique used by [ASS12] to give a quasi-polynomial time hitting set for set-multilinear depth three circuits.

Shift and Rank Concentration: We wish to check whether a polynomial computed by a set-multilinear depth three circuit is identically zero or not. Suppose the given polynomial is

$$C = \sum_{i=1}^{k} \prod_{j=1}^{d} l_{ij}(X_j)$$

in X variables, where X is the disjoint union of $X_1, X_2, ..., X_d$, $X_j = \{x_{1j}, x_{2j}, ..., x_{nj}\}$ and l_{ij}'s are linear polynomials in variables X_j. We view the polynomial C as a k component vector where the ith component is the polynomial computed by the ith product gate. A dot product with the all ones vector $(\bar{1})$, would give us the polynomial C. In shift and rank concentration, we shift each variable x_{ij} to $x_{ij} = x_{ij}+t_{ij}$, where t_{ij}'s are formal variables. Let $T_j = \{t_{1j}, t_{2j}, ..., t_{nj}\}$, $T = T_1 \uplus T_2 \uplus ... \uplus T_d$, $S \subseteq X$, $X_S = \prod_{x_{ij} \in S} x_{ij}$ and Z_{X_S} be the coefficient vector over $\mathbb{F}(T)$ corresponding to the monomial X_S . We use a map $\tau : t_{ij} \to t^{\omega_{ij}}$ such that

$$sp_{\mathbb{F}(t)}\{Z_{X_j} | supp(X_j) \leq \lceil \log k \rceil\} \;=\; sp_{\mathbb{F}(t)}\{Z_{X_j}\}$$

where $sp_{\mathbb{F}(t)}\{Z_{X_j}\}$ denotes the the span of the coefficient vectors over $\mathbb{F}(t)$ corresponding to the different monomials in the shifted monomial. [KS01],[ASS12] show that it is sufficient to try $n^{O(\log k)}$ many maps to find the desired one such that the ω_{ij}'s are bounded by a polynomial in n^2, the number of variables. After such a shift using the desired map, the polynomial C is non zero if and only if there a exists a monomial in the shifted polynomial with, support less than or equal to $\lceil \log k \rceil$ and has a non zero coefficient over $\mathbb{F}(t)$. Thus we can check whether the shifted polynomial has a monomial with support less than or equal to $\lceil \log k \rceil$, by projecting over all possible choices of $\lceil \log k \rceil$ variables and test if the shifted polynomial is non zero using [KS01] in $n^{O(\log k)}$ time.

Lemma 3.1.1 shows that in case of superposition of t set-multilinear depth three circuits if we know the base sets a priori then a quasi-polynomial time hitting set can be readily con-

structed using [ASS12]. However this would not be a complete PIT algorithm as in a real setting we would not have the knowledge of the base sets to begin with.

Lemma 3.1.1 *Given a circuit C which has top fan-in k and is a superposition of t set-multilinear depth three circuits $C_1, C_2, ..., C_t$ on base sets $X_1, X_2, ..., X_t$. With the knowledge of base sets we can find a hitting set for C in $tn^{O(t \log k)}$ time.*

Proof: We shift variables x_{ij} to $x_{ij} + r_i^{\omega_{ij}}$, where each r_i is a formal variable and ω_{ij}'s belong to $\mathbb{N}$ (the values of ω_{ij}'s would be fixed later). Observe that we are shifting variables in different base set by powers of different variables whereas the variables in the same base set, say X_i, are shifted by powers of the same variable r_i. We can do this because we have the knowledge of base sets. We view the polynomial C as a set-multilinear depth three polynomial in X_1 variables over $\mathbb{F}(X_2, ..., X_t)$. Since [ASS12] make no assumptions on the underlying field we can initially shift only the variables in X_1 (just for analysis), i.e $x_{1j} = x_{1j} + r_1^{\omega_{1j}}$ where the $\omega'_{1j}s$ (as explained in 'Shift and Rank Concentration') are chosen such that, the input polynomial is non zero if and only if a monomial with support less than $l = \lceil \log k + 1 \rceil$ (also known as low support monomial) in X_1 variables has non zero coefficient over the field $\mathbb{F}(r_1, X_2, ..., X_t)$. Any coefficient polynomial over $\mathbb{F}(r_1, X_2, ..., X_t)$, corresponding to a monomial in X_1 variables is again a superposition of $(t-1)$ set-multilinear depth three circuits on base sets $X_2, ..., X_t$ over $\mathbb{F}(r_1)$. Hence we repeat the same process on the coefficient polynomial of the low support monomial in X_1 variables. We shift the X_2 variables, i.e $x_{2j} = r_2^{\omega_{2j}}$ and ensure that the input polynomial is non zero if and only if a low support monomial in X_2 variables has non zero coefficient over the field $\mathbb{F}(r_1, r_2, X_3, ..., X_t)$. We keep repeating the same process for each of the base sets. Eventually we would have that, the polynomial is non zero if and only if there exists a monomial with support less than tl in the shifted polynomial that has non zero coefficient over $\mathbb{F}(r_1, r_2, ..., r_t)$. We can check whether a monomial with support less than tl has a non zero coefficient over $\mathbb{F}(r_1, r_2, ..., r_t)$ in $tn^{O(t \log k)}$ time by by projecting over all possible choices of l variables and test if the shifted polynomial is non zero . □

Consider circuit C which is a superposition of 2 set-multilinear circuits C_1 and C_2 defined over base sets X and Y respectively. The following lemma shows that we can find two base sets with singleton colors (not necessarily X and Y) such that restricted to any of the base sets, C is set-multilinear. Restricting a polynomial $F(X)$ to a set of of variables $X_1 \subseteq X$ implies substituting all the variables in $X \setminus X_1$ to 0.

Lemma 3.1.2 *Given a circuit C which is a superposition of 2 set-multilinear circuits C_1 and C_2 on base sets X and Y, we can find 2 base sets X' and Y' with singleton colors in polynomial*

time such that restricted to any of the base sets $X^{'}$ or $Y^{'}$, C is a set-multilinear depth three circuit.

Proof: We construct a graph G with vertices as variables. We add an edge between 2 variables x_1 and y_1 if they appear in the same linear polynomial in any of the product gates. Since the given circuit C is a superposition of two set-multilinear circuits, the graph G is a bipartite graph with all the edges between the variables in the set X and the variables in the set Y. We color the vertices of G such that, if (x_i, y_j) is an edge then x_i and y_j have different colors. Since G is bipartite we can color G greedily using two colors, say red and yellow in $O(n)$ time, where $n = |X| + |Y|$. All the vertices colored red form the base set $X^{'}$ and those colored yellow form the base set $Y^{'}$. From the way we color the graph G it is easy to see that, circuit C when restricted $X^{'}$ or $Y^{'}$ is a set-multilinear circuit. □

Once we have the knowledge of the base sets we can use lemma 3.1.1 to give us a hitting set for C in quasi-polynomial time. The algorithm is clearly whitebox since to find the base sets we look at the linear polynomials on the incoming edges to layer 2 product gates of circuit C. We state the result formally in the next theorem.

Theorem 3.1.1 *Given a circuit C which is a superposition of 2 set-multilinear circuits C_1 and C_2 on unknown base sets X and Y respectively, we can perform whitebox PIT for C in $n^{O(\log n)}$ time where $n = |X| + |Y|$.*

3.2 NP-hardness and approximation algorithm

We know that the problem of deciding whether a given graph G is t colorable is NP-Complete when $t > 2$. A proof of the case when $t = 3$ is shown in [**?**]. We reduce this problem to our problem of finding the base sets in C where C is a superposition of t set-multilinear circuits, where $t > 2$.

Theorem 3.2.1 *Given a circuit C which is a superposition of t set-multilinear circuits $C_1, C_2, ..., C_t$ on unknown base sets $X_1, X_2, ..., X_t$ respectively, finding t base sets $X_1^{'}, X_2^{'}, ..., X_t^{'}$ such that C is a superposition of t set-multilinear circuits $C_1^{'}, C_2^{'}, ..., C_t^{'}$ respectively on base sets $X_1^{'}, X_2^{'}, ..., X_t^{'}$ respectively is NP-Hard when $t > 2$.*

Proof: We will reduce the t-coloring problem to this problem. Let us suppose we are given a graph G(V,E). From G we construct a circuit C which is a superposition of t base sets with singleton colors if and only if G is t colorable. Let V=$\{u_1, ..., u_n\}$. We add a product gate p in C consisting of the product of n variables $(u_1)...(u_n)$. If there exists an edge between two

vertices u_1 and u_2 in G then we add a product gate p_{u_1,u_2} in C having a single linear polynomial $(u_1 + u_2)$.

Claim 3.2.1 *Circuit C constructed using the above reduction is a superposition of t set-multilinear depth three circuits if and only if graph G is t colorable.*

Proof: If G is t colorable then it is easy to see that C is a superposition of t set-multilinear depth three circuits. The t partitions of G correspond to t base sets in C. In the reverse direction, say C is a superposition of t set-multilinear depth three circuits. This implies C has t base sets. We claim these t base sets correspond to t partitions of G. Say two variables u_i and u_j belong to the same base set, then it implies u_i and u_j don't appear in the same linear polynomial. But this implies there is no edge between u_i and u_j in G else there would have been a product gate p_{u_i,u_j} having a single linear polynomial $(u_1 + u_2)$ in C. □

We know deciding whether a given graph G is t colorable is NP-Complete for $t > 2$. Hence from claim 3.2.1, theorem 3.2.1 follows. □

Although it is provably hard to find the base sets in polynomial time in C where C is a superposition of t base sets when $t > 2$, we can find $n^{1-\frac{3}{t+1}}$ base sets using [KMS98], such that restricted to any of the base sets C is a set-multilinear circuit. This immediately gives us a sub-exponential whitebox PIT when t is a constant. We state this formally in the next theorem.

Theorem 3.2.2 *Given a circuit C which is a superposition of t set-multilinear circuits $C_1, C_2, ..., C_t$ on unknown base sets $X_1, X_2, ..., X_t$, we can perform whitebox PIT for C in $exp(O(n^{1-\frac{3}{t+1}}.\text{poly}(\log n) + \log t))$ time.*

Proof: We draw a t partite graph G where the vertices are the variables from the circuit C. We draw an edge between two variables if they appear in the same linear polynomial in any of the product gates. It is easy to see that, if C is a superposition of t set-multilinear circuits, then the resulting graph G is t partite, the t base sets in C would correspond to the t partitions of vertices in G. Now we use a simple greedy algorithm given in [KMS98] to color a t colorable graph with less than $n^{1-\frac{3}{t+1}}$ colors, where n is the number of vertices in the graph G. This would essentially give us $n^{1-\frac{3}{t+1}}$ base sets such that restricted to any one of them C is a set-multilinear circuit. Using lemma 3.1.1 we get a whitebox PIT for C in $exp(O(n^{1-\frac{3}{t+1}}.\text{poly}(\log n) + \log t))$ time. □

3.3 Hitting sets for superposition of set-multilinear depth three circuits

In this section we use shift and rank concentration technique used in [ASS12] to give a quasi-polynomial time hitting set for a restricted class of superposition of set-multilinear depth three circuits. The model we consider is a multilinear depth 3 circuit that is both a superposition of m set-multilinear depth three circuits and simultaneously a sum of k set-multilinear depth three circuits, where m and k are constants. Consider for example the following polynomial

$$C(X,Y,Z) = (1+3x_1+5y_2+4z_1)(4+x_2+y_1+6z_2)+(9+6x_1+4y_2+z_1)(3+2x_2+5y_1+3z_2)$$

$$+(6+9x_1+4y_1+z_2)(2+5x_2+3y_2+2z_1)+(3+6x_1+9y_1+5z_2)(5+8x_2+2y_2+8z_1)$$

$C(X,Y,Z)$ is a superposition of three set-multilinear depth three circuits with base sets $X = \{x_1\} \cup \{x_2\}$, $Y = \{y_1\} \cup \{y_2\}$ and $Z = \{z_1\} \cup \{z_2\}$. But $C(X,Y,Z)$ is also a sum of two set-multlinear depth three circuits with $\{x_1, y_2, z_1\}$, $\{x_2, y_1, z_2\}$ being the colors in the first set-multilinear depth three circuit (corresponding to the first two products) and $\{x_1, y_1, z_2\}$, $\{x_2, y_2, z_1\}$ being the colors in the second set-multilinear depth three circuit (corresponding to the last two products). The motivation for studying this model comes from the lower bound result in chapter 4. In chapter 4 we give an exponential lower bound for ROABP where the hard polynomial is a polynomial of this kind; that is it is simultaneously a superposition of constantly many set-multilinear depth three circuits as well as sum of constantly many set-multilinear depth three circuits.We prove the following for this model.

Theorem 3.3.1 (Main Theorem) *Given a circuit C which is a superposition of m set-multilinear depth three circuits $C_1, C_2, ..., C_m$ on base sets $X_1, X_2, ..., X_m$, where $X_i = \{x_{i1}, x_{i2}, ..., x_{in}\}$, and simultaneously a sum of k set-multilinear depth three circuits $C_1', C_2', ..., C_k'$, we can find a hitting set for C in time $l^{km(\log l+1)} \cdot n^{O(m \log l)}$ time, where l is the bound on the top fan-in of circuit C.*

[GKST15] recently gave a $(wnd)^{k2^k \log(wnd)}$ time hitting set for n variate polynomials computed by sum of k ROABPs each of width less than w. Observe the doubly exponential dependence on k in their result. On the contrary in theorem 3.3.1 the dependence is singly exponential in k. Although it is important to note that the model considered in theorem 3.3.1 is weaker than the sum of ROABPs model. Sum of k set-multilinear depth three circuits reduces to sum of k ROABPS. Moreover our model is a superposition of m set-multilinear depth three circuits and simultaneously a sum of k set-multilinear depth three circuits. We will now prove theorem 3.3.1.

Proof: Let $X = \{X_1 \cup ... \cup X_m\}$. Hence we can represent C as follows.

$$C(X_1, X_2, ..., X_m) = \sum_{i=1}^{l} \prod_{j=1}^{n} (\alpha_{ij} + z_{i11}x_{11} + z_{i2\sigma_{i2}(j)}x_{2\sigma_{i2}(j)} + ... + z_{im\sigma_{im}(j)}x_{m\sigma_{im}(j)})$$

where for all $i \in [l]$ and $q \in \{2, ..., m\}$, σ_{iq} represents the permutation function $[n] \to [n]$ corresponding to the ith product gate and qth base set. Without loss of generality we can assume identity permutation corresponds to the first base set X_1 in all the product gates. Now C is also a sum of k set-multilinear circuits $C'_1, ..., C'_k$. Observe that the permutations with respect to a base set corresponding to a particular set-multilinear circuit say, C'_p are identical, i.e if i_1 and i_2 are two product gates corresponding to the same set-multilinear circuit C'_p then for all $b \in [m]$ $\sigma_{i_1 b}$ and $\sigma_{i_2 b}$ are identical. Thus we have just km many distinct permutation each corresponding to a combination of a base set and a set-multilinear circuit.

Proof Outline: Let $T_i = \{t_{i1}, t_{i2}, ..., t_{in}\}$, we have m such sets $T_1, T_2, ..., T_m$. Let $T = \{T_1 \cup ... \cup T_m\}$. We shift a variable x_{ij} to $(x_{ij} + t_{ij})$. For now consider each of the t_{ij} variables as formal variables, finally we will substitute $t_{ij} = t^{\omega_{ij}}$, where t is a fresh variable and w_{ij} is an appropriate small constant. We analyze the shift in m steps. In the ith step we analyze the shift of the X_i variables and show that there exists a monomial in $X_1 \cup X_2 \cup ... \cup X_i$ variables of support less than $i \log l$, where l is the top fan-in of circuit C, that has a non-zero coefficient polynomial over $\mathbb{F}(T_1 \cup ... \cup T_i \cup X_{i+1} \cup ... \cup X_m)$. Finally after substituting every t_{ij} to $t^{\omega_{ij}}$ we show that there exists a monomial of support less than $m \log l$ that has a non-zero coefficient over the field $\mathbb{F}(t)$. This would imply the original polynomial C is non-zero if and only if there exists a monomial in the shifted polynomial of support less than $m \log l$ that has a non-zero coefficient over the field $\mathbb{F}(t)$. Once we show this, finding a hitting set is easy: project over all possible choices of $(m \log l)$ variables and test if the shifted polynomial is non-zero over $\mathbb{F}(t])$ using sparse PIT [KS01]. We stress again that the algorithm shifts all variables simultaneously, only the analysis proceeds in steps.

We will explain step 1 and then generalize the argument to the rth step.

Step 1: We view C as a polynomial in X_1 variables with coefficients over $\mathbb{F}(X_2 \cup X_3 \cup ... \cup X_m)$. Thus C is a set-multilinear depth three circuit in X_1 variables. In step 1 we consider shifts of only the variables in base set X_1, i.e we shift x_{1j} to $x_{1j} + t_{1j}$. Since C is a set-multilinear depth three circuit in X_1 variables, from [ASS12] we know that, C is non-zero if and only if there exists a low support monomial in X_1 variables (support less than $\log l$) that has a non-zero

coefficient polynomial over $\mathbb{F}(T_1 \cup X_2 \cup X_3 \cup ... \cup X_m)$. Let $x_{11}...x_{1\log l}$ be such a monomial. We will call such monomials as non-zero monomials. The coefficient polynomial of the the monomial $x_{11}...x_{1logl}$ is as follows:

$$C_2(X_2, ..., X_m) = \sum_{i=1}^{l} \prod_{j=\log l+1}^{n} (1 + z_{i11}(t_{1j}) + z_{i2\sigma_{i2}(j)}x_{2\sigma_{i2}(j)} + ... + z_{im\sigma_{im}(j)}x_{m\sigma_{im}(j)})$$

The coefficient polynomial is again both, a superposition of $(m-1)$ set-multilinear depth three circuits on base sets $X_2, ..., X_m$ and sum of k set-multilinear depth three circuits over $\mathbb{F}(T_1)$. Hence in each step we shift the variables from a particular base set and obtain a coefficient polynomial corresponding to a non-zero monomial (in the variables from the base set that have already been shifted) that is both, a superposition of set-multilinear depth three circuits (but on lesser number of base sets) and sum of k set-multilinear depth three circuits. Thus in the $(r-1)$th step we would have a monomial in $X_1 \cup X_2 ... \cup X_{r-1}$ variables with support less than $(r-1)\log l$ that has a non-zero coefficient polynomial over $\mathbb{F}(T_1 \cup ... \cup T_{r-1} \cup X_r \cup ... \cup X_m)$. The coefficient polynomial we get in general would be as follows:

$$C_r(X_r, ..., X_m) = \sum_{i=1}^{l} \prod_{j=1}^{n} (1 + z_{i11}t_{1j} + z_{i2\sigma_{i2}(j)}t_{2\sigma_{i2}(j)} + ... + z_{i(r-1)\sigma_{i(r-1)}(j)}t_{(r-1)\sigma_{i(r-1)}(j)}$$

$$+ z_{ir\sigma_{ir}(j)}x_{r\sigma_{ir}(j)} + ... + z_{im\sigma_{im}(j)}x_{m\sigma_{im}(j)})$$

Step r: In step r we analyze the non-zero coefficient polynomial of the low support monomial in $X_1 \cup X_2 ... \cup X_{r-1}$ variables that we get from step $(r-1)$. These non-zero coefficient polynomial is over $\mathbb{F}(T_1 \cup ... \cup T_{r-1} \cup X_r \cup ... \cup X_m)$. We show that, after we shift the X_r variables by T_r, there exists a monomial in X_r variables with support less than $\log l$ in C_r such that, it has a non-zero coefficient polynomial over $\mathbb{F}(T_1 \cup ... \cup T_r \cup X_{r+1} \cup ... \cup X_m)$. This would imply after the rth step we have a monomial in $X_1 \cup X_2 \cup ... \cup X_r$ variables of support less than $r \log l$ in C, that has a non-zero coefficient polynomial over $\mathbb{F}(T_1 \cup ... \cup T_r \cup X_{r+1} \cup ... \cup X_m)$.

We rewrite the non-zero coefficient polynomial that we get from step $(r-1)$, such that, we can associate the identity permutation with the base set X_r.

$$C_r(X_r, ..., X_m) = \sum_{i=1}^{l} \prod_{j=1}^{n} (1 + z_{i1\pi_{i1}(j)}t_{1\pi_{i1}(j)} + z_{i2\pi_{i2}(j)}t_{2\pi_{i2}(j)} + ... + z_{i(r-1)\pi_{i(r-1)}(j)}t_{(r-1)\pi_{i(r-1)}(j)}$$

$$+ z_{irj}x_{rj} + ... + z_{im\pi_{im}(j)}x_{m\pi_{im}(j)})$$

Here again π_{ij} represents the permutation function $[n] \to [n]$ corresponding to the ith product gate and jth base set. We view $C_r(X_r, ..., X_m)$ as a polynomial in X_r variables over $\mathbb{F}[X_{r+1} \cup ... \cup X_m \cup T_1 \cup ... \cup T_{r-1}]$. For $J \subseteq [n]$, $X_{rJ} = \prod_{j \in J} x_{rj}$. We view the coefficient of a monomial X_{rJ} as a l component vector $Z_{X_{rJ}}$, where the ith component is the coefficient of the monomial in the ith product gate which is equal to

$$\prod_{j \in J} z_{irj} \prod_{j \in [n] \setminus J} (1 + z_{i1\pi_{i1}(j)} t_{1\pi_{i1}(j)} + ... + z_{i(r-1)\pi_{i(r-1)}(j)} t_{(r-1)\pi_{i(r-1)}(j)}$$

$$+ z_{i(r+1)\pi_{i(r+1)}(j)} x_{(r+1)\pi_{i3}(j)} + ... + z_{im\pi_{im}(j)} x_{m\pi_{im}(j)})$$

Pick a monomial whose support is exactly $\log l + 1$. Say we pick X_{rJ}, corresponding to $J = [\log l + 1]$. We consider the monomial X_{rJ} and all its subset monomials $X_{rJ'}$ corresponding to $J' \subseteq J$, there are exactly $2^{\log l + 1} > l$ many such monomials. Hence we have a linear dependency among the coefficient polynomials of these monomials, i.e

$$\sum_{J \subseteq [\log l + 1]} b_J Z_{X_{rJ}} = 0$$

where $\forall\ J \subseteq [\log l + 1],\ b_J \in \mathbb{F}[X_{r+1} \cup ... \cup X_m \cup T_1 \cup ... \cup T_{r-1}]$ and $\exists\ J \subseteq [\log l]$ such that $b_J \neq 0$. Now we shift the variables in X_r, i.e we shift $x_{rj} = x_{rj} + t_{rj}$. Thus we have

$$C'_r(X_r, ..., X_m) = \sum_{i=1}^{l} \prod_{j=1}^{n} (1 + z_{i1\pi_{i1}(j)} t_{1\pi_{i1}(j)} + z_{i2\pi_{i2}(j)} t_{2\pi_{i2}(j)} + ... + z_{i(r-1)\pi_{i(r-1)}(j)} t_{(r-1)\pi_{i(r-1)}(j)}$$

$$+ z_{irj}(x_{rj} + t_{rj}) + ... + z_{im\pi_{im}(j)} x_{m\pi_{im}(j)})$$

For $i \in [l]$ and $j \in [n]$ we let

$$\rho_{ij} = (z_{i1\pi_{i1}(j)} t_{1\pi_{i1}(j)} + z_{i2\pi_{i2}(j)} t_{2\pi_{i2}(j)} + ... + z_{i(r-1)\pi_{i(r-1)}(j)} t_{(r-1)\pi_{i(r-1)}(j)}$$

$$+ z_{i(r+1)\pi_{i(r+1)}(j)} x_{(r+1)\pi_{i(r+1)}(j)} + ... + z_{m\pi_{im}(j)} x_{m\pi_{im}(j)})$$

Hence we have

$$C'_r(X_r, ..., X_m) = \sum_{i=1}^{l} \prod_{j=1}^{n} (1 + \rho_{ij} + z_{rj}(x_{rj} + t_{rj}))$$

$$C'_r(X_r, ..., X_m) = \sum_{i=1}^{l} (\prod_{j=1}^{n} (1 + \rho_{ij} + z_{irj} t_{rj}))(\prod_{j=1}^{n} (1 + \frac{z_{irj} x_{2j}}{1 + \rho_{ij} + z_{irj} t_{rj}}))$$

Let

$$D_r(X_r, ..., X_m, T_1, .., T_r) = \sum_{i=1}^{l} \prod_{j=1}^{n} z'_{irj} x_{rj}$$

where

$$z'_{irj} = \frac{z_{irj}}{1 + \rho_{ij} + z_{irj} t_{rj}}$$

$$\Rightarrow z_{irj} = \frac{z'_{i2j}(1 + \rho_{ij})}{1 - z'_{irj} t_{rj}}$$

First observe that $C_r(X_r, ..., X_m)$ is non-zero if and only if $C'_r(X_r, ..., X_m)$ is non-zero. Recall we intend to show that $C'_r(X_r, ..., X_m)$ is non-zero if and only if there exists a low support monomial of X_r variables that has a non-zero coefficient polynomial over $\mathbb{F}(T_1 \cup ... \cup T_r \cup X_{r+1} \cup X_{r+2} \cup ... X_m)$. To show this it is sufficient to show, $D_r(X_r, ..., X_m, T_1, ..., T_r)$ has a low support monomial of X_r variables that has a non-zero coefficient polynomial over $\mathbb{F}(T_1 \cup ... \cup T_r \cup X_{r+1} \cup X_{r+2} \cup ... X_m)$. We view the coefficient of a monomial X_{rJ} in D_r, where $J \subseteq [n]$, as a l component vector $Z'_{X_{rJ}}$, where the ith component is the coefficient of the monomial in the ith product gate which is equal to $\prod_{j \in J} z'_{rj}$. Let $P = T_1 \cup ... \cup T_r \cup X_{r+1} \cup X_{r+2} \cup ... X_m$. We show the following for these set of coefficient vectors,

$$\text{span}_{\mathbb{F}(P)}\{Z'_{X_{rJ}} | J \subseteq [n]\} \;=\; \text{span}_{\mathbb{F}(P)}\{Z'_{X_{2J}} | J \subseteq [n], supp(X_{2J}) < l\}$$

This would imply $C_r(X_r, ..., X_m)$ is non-zero if and only if $D_r(X_r, ..., X_m, T_1, ..., T_r)$ has a low support monomial of X_r variables that has a non-zero coefficient polynomial over $\mathbb{F}(P)$. Recall

$$\sum_{J \subseteq [\log l + 1]} b_J Z_{X_{rJ}} = 0$$

We write the equation for the ith component of coefficient vectors

$$\sum_{J \subseteq [\log l + 1]} b_J \prod_{j \in J} z_{irj} \prod_{j \in [n] \setminus J} (1 + \rho_{ij}) = 0$$

$$\Rightarrow \sum_{J \subseteq [\log l + 1]} b_J \prod_{j \in J} z_{irj} \prod_{j \in [l] \setminus J} (1 + \rho_{ij}) = 0$$

$$\Rightarrow \sum_{J \subseteq [\log l + 1]} b_J \prod_{j \in J} \frac{z'_{irj}(1 + \rho_{ij})}{1 - z'_{irj} t_{irj}} \prod_{j \in [l] \setminus J} (1 + \rho_{ij}) = 0$$

Since $\prod_{j\in[\log l+1]}(1+\rho_{ij}) \neq 0$ we have

$$\sum_{J\subseteq[\log l+1]} b_J \prod_{j\in J} \frac{z'_{irj}}{1-z'_{irj}t_{irj}} = 0$$

Multiplying both sides by $\prod_{j\in[\log l+1]}(1-z'_{irj}t_{irj})$

$$\sum_{J\subseteq[\log l+1]} b_J \prod_{j\in J} z'_{irj} \prod_{j\in[\log l+1]\setminus J} (1-z'_{irj}t_{irj}) = 0$$

Since this is true for all the l components of coefficient vectors we have

$$\sum_{J\subseteq[\log l+1]} b_J Z'_{rJ} \prod_{j\in[\log l+1]\setminus J} (1-Z'_{rj}t_{rj}) = 0$$

$$\underbrace{(\sum_{J\subseteq[\log l+1]} b_J(-1)^{l-|J|} \prod_{j\in[l]\setminus J} t_{rj})}_{g_{[\log l+1]}(T_1,T_2,...,T_r,X_{r+1},...,X_m)} Z'_{r[\log l+1]} + \sum_{J\subset[\log l+1]} g_J(T_1,T_2,...,T_r,X_{r+1},...,X_m)Z'_{rJ} = 0$$

Since $\forall\ J\subseteq[\log l+1]$, $b_J \in \mathbb{F}(T_1\cup...\cup T_{r-1}\cup X_{r+1}\cup...\cup X_n)$, $g_{[\log l+1]}[T_1,T_2,...,T_r,X_{r+1},...,X_m]$ is non-zero, $Z'_{r[\log l+1]}$ is $\mathbb{F}(T_1,T_2,...,T_r,X_{r+1},...,X_m)$ linearly dependent on vectors $Z'_{rJ'}$ where $J'\subset[\log l+1]$. The set $[\log l+1]$ is just a representative case. By the same argument we have that Z'_{rJ}, where $J\subseteq[n]$ and $|J|=\log l+1$ is $\mathbb{F}(T_1,T_2,...,T_r,X_{r+1},...,X_m)$ linearly dependent on vectors $Z'_{rJ'}$ where $J'\subset J$. Also every $J\subseteq[n]$, Z'_{rJ} can be inductively expressed as $\mathbb{F}(T_1\cup T_2\cup...\cup T_r\cup X_{r+1},...,X_m)$ linear combinations of $Z'_{rJ'}$, where $J'\subseteq[n]$ and $|J'|<\log l+1$.

This would imply after the rth step if $C(X_1,X_2,...,X_m)$ computes a non-zero polynomial then we have a monomial in $X_1\cup...\cup X_r$ variables with support less than $r(\log l+1)$ such that it has a non-zero coefficient polynomial over $\mathbb{F}(T_1\cup...\cup T_r\cup X_{r+1}\cup...\cup X_n)$. Hence after m steps there exists a monomial in $X_1\cup...\cup X_m$ variables with support less than $m(\log l+1)$, such that it has a non-zero coefficient polynomial over $\mathbb{F}(T_1\cup...\cup T_m)$.

Now we apply a map ψ that maps t_{ij} to $t^{\omega_{ij}}$, where for all $i\in[m]$ and $j\in[n]$, $\omega_{ij}\in\mathbb{N}$ such that ω_{ij}'s are bounded by a polynomial in (mn: the number of variables) and the non-zero coefficient polynomial over $\mathbb{F}(T_1\cup...\cup T_m)$ corresponding to the monomial in $X_1\cup...\cup X_m$ variables with support less than $m(\log l+1)$, continues to be non-zero over $\mathbb{F}(t)$ after we apply this map. We claim that we can find such a map in time $l^{km(\log l+1)}\cdot n^{O(m\log l)}$. To show this we

look at the structure of $g_{[\log l+1]}(T_1, T_2, ..., T_r, X_{r+1}, ..., X_m)$ in the above argument. We claim the following:

Claim 3.3.1 *The number of distinct variables from* $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$ *in* $g_{[\log l+1]}(T_1, T_2, ..., T_r, X_{r+1}, ..., X_m)$ *is at most* $(r-1)k(\log l + 1)$.

Proof:

$$g_{[\log l+1]}(T_1, T_2, ..., T_r, X_{r+1}, ..., X_m) = \sum_{J \subseteq [\log l+1]} b_J (-1)^{l-|J|} \prod_{j \in [\log l+1] \setminus J} t_{rj}$$

Observe that variables from the set $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$ only appear in the b_J's. We know

$$\sum_{J \subseteq [\log l+1]} b_J Z_{X_{rJ}} = 0$$

b_J's don't contain variables from the set T_r. We study b_J's using Cramer's rule [**?**]. Let $J^{'} \subseteq [\log l + 1]$, using Cramer's rule we can express $b_{J'}$ as a ratio of two determinants, each over $\mathbb{F}(T_1 \cup T_2 \cup ... \cup T_{r-1} \cup X_{r+1} \cup ... \cup X_m)$ and of dimension at maximum $l \times l$. Let $Var(b_{J'})$ be the set of variables from $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$ appearing in $b_{J'}$. Let the coefficient vector Z_{rJ} of monomial x_{rJ} where $J \subseteq [\log l + 1]$, be one of the column vectors in both the determinants. The coefficient of monomial x_{rJ} where $J \subseteq [\log l + 1]$, in any product gate $j \in [l]$ has exactly $(r-1)(n - |J|)$ variables from $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$. These variables correspond to the $(r-1)$ variables from $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$ that appear along with x_{ri} where $i \in [n] \setminus J$, in the same linear factor in product gate j. Recall for $i \in [l]$ and $j \in [n]$ we let

$$\rho_{ij} = (z_{i1\pi_{i1}(j)} t_{1\pi_{i1}(j)} + z_{i2\pi_{i2}(j)} t_{2\pi_{i2}(j)} + ... + z_{i(r-1)\pi_{i(r-1)}(j)} t_{(r-1)\pi_{i(r-1)}(j)}$$

$$+ z_{i(r+1)\pi_{i(r+1)}(j)} x_{(r+1)\pi_{i(r+1)}(j)} + ... + z_{m\pi_{im}(j)} x_{m\pi_{im}(j)})$$

ρ_{ij} contains the $(r-1)$ variables from $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$ that appear along with x_{ri} where $i \in [n] \setminus J$, in the same linear factor in product gate j. Among these only $(r-1)(\log l + 1 - |J|)$ variables from $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$ that appear in ρ_{ij} where $i \in [\log l + 1] \setminus J$ might eventually appear in $Var(b_{J'})$, the rest get cancelled. The product gates corresponding to a single set-multilinear circuit in C_r would have the same variables from $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$ in the coefficient of monomial x_{rJ}, since these product gates have the same permutation function. There are k set-multilinear circuits, hence the number of variables from $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$ in the coefficient vector Z_{rJ} is at most $k(r-1)(\log l + 1)$. Since all the monomials we consider are subsets of $\prod_{i=1}^{\log l+1} x_{ri}$ the total number of variables in both the determinants used to express $b_{J'}$ is at most

$k(r-1)(\log l+1)$. This implies $|Var(b_{J'})| \leq k(r-1)(\log l+1)$. Let $J'' = \phi$, then $\forall J \subseteq [\log l+1]$, $Var(b_J) \subseteq Var(b_{J''})$. Hence

$$|\cup_{J\subseteq[\log l+1]} \quad Var(b_J)| \leq k(r-1)(\log l+1)$$

. □

Let $J \subseteq [\log l+1]$, from claim 3.3.1 we know that every monomial in b_J has at most $k(r-1)(\log l+1)$ variables from $\{T_1 \cup T_2 \cup ... \cup T_{r-1}\}$. The degree of each of these variables is at most l in a monomial. Now every monomial in b_J is multiplied with a monomial in T_r with support less than $(\log l+1)$. The variables in the monomial in T_r are a subset of $\{t_{r1}, ..., t_{\log l+1}\}$. Hence the total number of variables in $g_{[\log l+1]}(T_1, T_2, ..., T_r, X_{r+1}, ..., X_m)$ from $\{T_1 \cup T_2 \cup ... \cup T_r\}$ is at most $kr(\log l+1)$. Thus any monomial in $g_{[\log l+1]}(T_1, T_2, ..., T_r, X_{r+1}, ..., X_m)$ has at most $kr(\log l+1)$ variables from $\{T_1 \cup T_2 \cup ... \cup T_r\}$ with degree of each of these variables being at most l. This is even true at step m. Let $g_{[\log l+1]}[T_1, T_2, ..., T_m]$ correspond to $Z'_{m[\log l+1]}$ as in the above argument, then any monomial in $g_{[\log l+1]}(T_1, T_2, ..., T_m)$ has at most $km(\log l+1)$ distinct variables with degree of each of these variables being at most l. There are at most $l^{km(\log l+1)}$ such monomials. Hence the non-zeroness of $g_{[\log l+1]}(T_1, T_2, ..., T_m)$ is maintained by a univariate transformation $\psi : t_{ij} \to t^{\omega_{ij}}$ that maps these $l^{km(\log l+1)}$ many monomials to distinct weights. We construct the map ψ in time $l^{O(km(\log l+1))}$, using well known sparse PIT methods [KS01]. As states previously the set $[\log l+1]$ is just a representative case. By the same argument we have that Z'_{mJ}, where $J \subseteq [n]$ and $|J| = \log l+1$ is $\mathbb{F}(t)$ linearly dependent on vectors $Z'_{mJ'}$ where $J' \subset J$. Also every $J \subseteq [n]$, Z'_{mJ} can be inductively expressed as $\mathbb{F}(t)$ linear combinations of $Z'_{mJ'}$, where $J' \subseteq [n]$ and $|J'| < \log l+1$. This implies even after applying ψ we have a monomial in $X_1 \cup ... \cup X_r$ variables with support less than $m(\log l+1)$ such that it has a non-zero coefficient polynomial over $\mathbb{F}(t)$.

Once we know that there exists a monomial with support less than $m(\log l+1)$ in the shifted polynomial $C(X+T)$ iff $C(X)$ is non-zero, finding a hitting set is elementary. We project over all possible choices of $m \log l$ variables and test if $C(X+T)$ is non-zero using [KS01]. This can be done in $n^{O(m \log l)}$ time. Hence we can find a hitting set for C in $l^{km(\log l+1)} \cdot n^{O(m \log l)}$ time. If l is bounded by a polynomial in n then we can find a hitting set in $n^{O(m \log n(k+1))}$ time. □

Chapter 4

Lower bounds for ROABP's against multilinear depth 3 circuits

In this chapter we first show in section 4.1 that, any multilinear depth 3 circuit having two product gates and at most two variables in every linear factor in each product gate, can be computed by a polynomial sized ROABP. Further we show that if we either increase the number of product gates or the number of variables in every linear factor in each product gate, in the multilinear depth 3 circuit model to three then there exists a multilinear polynomial computed by such a model of polynomial size which requires exponential width ROABP to compute it.

4.1 Constructing a polynomial sized ROABP

Say we have a multilinear polynomial $F(X)$ which can be computed by a multilinear depth 3 circuit with top fan-in 2 and the linear polynomials computed by 'sum' gates at level 3 from the top, has at most 2 variables. Let $\sigma : [n] \to [n]$ be a permutation function. Then $F(X)$ can be expressed as

$$F(X) = \prod_{i=1; i \bmod 2=1} (1 + x_i + x_{i+1}) \; + \prod_{i=1; i \bmod 2=1} (1 + x_{\sigma(i)} + x_{\sigma(i+1)})$$

We have assumed that the coefficients of the variables x_i's are 1 and the constant term in every linear polynomial is also 1. This is without any loss of generality and our argument holds even otherwise.. We will show that $F(X)$ can be computed by a polynomial sized ROABP with constant width.

Proof Outline: First we show that $F(X)$ can be expressed as sum of two, width two ROABP's.

The two ROABPs correspond to the two product gates. Without loss of generality assume the first ROABP computes the polynomial computed by the first product gate i.e $\prod_{i=1;i\bmod 2=1}(1+x_i+x_{i+1})$ and the second ROABP computes the polynomial computed by the second product gate, i.e $\prod_{i=1;i\bmod 2=1}(1+x_{\sigma(i)}+x_{\sigma(i+1)})$. The variable ordering in the first ROABP is $\{x_1, x_2, ..., x_n\}$. The variable ordering in the second ROABP is 'σ' permutation of the variable ordering in the first ROABP, i.e if x_j is in jth position of the variable ordering in first ROABP, then $x_{\sigma(j)}$ is in the jth position of the variable ordering in second ROABP . We then express both the ROABPs in the same variable ordering - which is possible because $F(X)$ has only two product gates and two variables in every linear polynomial. We show that in this process the width of the first ROABP remains same whereas the width of the second ROABP increases only by a constant. Now we join both the ROABP's (since the variable ordering is same) to give a single ROABP computing $F(X)$.

We will call $\prod_{i=1;\ i \bmod 2=1}(1+x_i+x_{i+1})$ as P_1 and $\prod_{i=1;\ i \bmod 2=1}(1+x_{\sigma(i)}+x_{\sigma(i+1)})$ as P_2. Each linear polynomial $l=(1+x_j+x_k)$ in P_i where $i \in [2]$ and $x_j, x_k \in X$ can be expressed as a product of two matrices:

$$\begin{bmatrix} 1+x_j & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_k & 0 \end{bmatrix} = \begin{bmatrix} 1+x_j+x_k & 0 \\ 0 & 0 \end{bmatrix}$$

The product of two linear polynomials $l_1=(1+x_{j_1}+x_{k_1})$ and $l_2=(1+x_{j_2}+x_{k_2})$ in P_i can be computed as the product of four matrices as follows:

$$\begin{bmatrix} 1+x_{j_1} & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_{k_1} & 0 \end{bmatrix} \begin{bmatrix} 1+x_{j_2} & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_{k_2} & 0 \end{bmatrix}$$

Hence it is easy to see P_1 can be expressed by an ROABP with the variable ordering $(x_1, ..., x_n)$ and P_2 can be expressed by an ROABP with the variable ordering $(x_{\sigma(1)}, ..., x_{\sigma(n)})$. There are two important observations to be noted here:

1. The linear polynomials in P_1 and P_2 can commute and hence even P_1 could be expressed by ROABPs in different variable orderings for example $(x_3, x_4, x_1, x_2, ..., x_n)$ is also a valid variable ordering for an ROABP computing P_1. Thus the variable ordering depends on the way the linear polynomials are ordered in P_1.

2. The two variables in a particular linear polynomial in P_1 or P_2 can be exchanged because $l=(1+x_j+x_k)$ in P_i where $i \in [2]$ and $x_j, x_k \in X$ can be expressed as a product of two

matrices, in two ways:

$$\begin{bmatrix} 1+x_j & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_k & 0 \end{bmatrix} = \begin{bmatrix} 1+x_j+x_k & 0 \\ 0 & 0 \end{bmatrix}$$

and

$$\begin{bmatrix} 1+x_k & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_j & 0 \end{bmatrix} = \begin{bmatrix} 1+x_j+x_k & 0 \\ 0 & 0 \end{bmatrix}$$

Hence even $(x_2, x_1, x_3, x_4, ..., x_n, x_{n-1})$ is also a valid variable ordering for an ROABP computing P_1.

This shows that $F(X)$ can be expressed as sum of two ROABP's. Our next step would be to express the two ROABP's in the same variable ordering. To do this we would partition the linear polynomials in P_1 and P_2 into sets $L_{11}, L_{12}, \ldots, L_{1k}$ and $L_{21}, L_{22}, \ldots, L_{2k}$ such that the set of variables appearing in the linear polynomials in L_{1t} is equal to the set of variables appearing in the linear polynomials in L_{2t}, where $t \in [k]$, and is completely disjoint from the set of variables appearing in linear polynomials in L_{mr} where $m \in [2]$ and $r \in [k] \setminus t$. We give the partition procedure below and explain it with an example later.

Mark all the linear polynomials in P_1 and P_2 as unpicked. Initialize $t = 1$ and $i = 1$:

1. Pick an unpicked linear polynomial $l_p = (1 + x_i + x_{i+1})$ in P_1 and put it in L_{1t}. Mark l_p as picked. Store the value i in temp: temp=i.

2. Let the linear polynomial in which the variable x_{i+1} appears in P_2 be $l_q = (1 + x_{i+1} + x_j)$. Put l_q in L_{2t} and mark l_q as picked.

3. If j is equal to temp then increment t and start from step 1.

4. Else set $i = j$ and let the linear polynomial in which the variable x_i appears in P_1 be $l_r = (1 + x_i + x_{i+1})$. Put l_r in L_{1t} and mark l_r as picked.

5. Repeat from step 2.

Since permutation can be written as a product of cycles it is easy to see why the set of variables appearing in the linear polynomials in L_{1t} is equal to the set of variables appearing in the linear polynomials in L_{2t}, where $t \in [k]$, and is completely disjoint from the set of variables appearing in linear polynomials in L_{mr} where $m \in [2]$ and $r \in [k] \setminus \{t\}$. We will explain the partition

procedure with the following example. Let

$$C(X) = (1+x_1+x_2)(1+x_3+x_4)(1+x_5+x_6)(1+x_7+x_8)+(1+x_3+x_6)(1+x_7+x_1)(1+x_4+x_5)(1+x_2+x_8)$$

In step 1 we pick the linear polynomial $(1 + x_1 + x_2)$ and put it in L_{11}. In step two we pick the linear polynomial in which x_2 appears in P_2, i.e $(1 + x_2 + x_8)$ and put it in L_{21}. Since x_8 is not equal to x_1 we move onto step 4. In step four we pick the linear polynomial in which x_8 appears in P_1, i.e $(1 + x_7 + x_8)$ and put it in L_{11}. Similarly we pick $(1 + x_7 + x_1)$ from P_2 and put it in L_{22}. Since x_7 is paired with x_1 in P_2 we begin fresh from step 1 to construct L_{12} and L_{22}. Hence we get $L_{11} = \{(1 + x_1 + x_2), (1 + x_7 + x_8)\}$, $L_{21} = \{(1 + x_2 + x_8), (1 + x_7 + x_1)\}$, $L_{12} = \{(1 + x_3 + x_4), (1 + x_5 + x_6)\}$ and $L_{22} = \{(1 + x_4 + x_5), (1 + x_6 + x_3)\}$.

We express the two ROABPs in same variable ordering in k parts. In each part we pick the linear polynomials in L_{1t} and L_{2t} and order them such that they are in same variable ordering. Finally we combine these k parts to give a single ROABP of width 6. To order the linear polynomials in L_{1t} and L_{2t} in the same variable ordering, we arrange the linear polynomials in L_{1t} and L_{2t} one below the other, in the order they are picked during the partition process. In the example we gave above, we would arrange for L_{11} and L_{21} as follows.

$$\begin{bmatrix} a+x_1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_2 & 0 \end{bmatrix} \begin{bmatrix} a+x_8 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_7 & 0 \end{bmatrix}$$
$$\begin{bmatrix} a+x_2 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_8 & 0 \end{bmatrix} \begin{bmatrix} a+x_7 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ x_1 & 0 \end{bmatrix}$$

Observe that we arrange in such a way that only the first variable in first linear polynomial of L_{1t} (which is also the last variable in the last linear polynomial of L_{2t}) is not in the same order, the other variables are in the same variable ordering. We show the corresponding directed acyclic graph for these matrices when aligned as above in fig. 4.1. We have marked the input and output nodes corresponding to L_{11} and L_{21} in fig. 4.1. We order the first variable in first linear polynomial of L_{1t} by breaking the second ROABP in two parts as shown in fig. 4.2. The first part computes the polynomial in which x_1 does not appear and the second part brings the x_1 to the beginning and computes the polynomial in which x_1 appears and finally we add these two parts by adding an extra layer. It is easy to see that, this method can be done on all the k different sets of linear polynomials. Once we do this we get k different DAG's which are variable disjoint, and each has a consistent variable ordering. We are just left to connect the k graphs. We connect the input nodes of the graph corresponding to L_{1r}, L_{2r} to the output

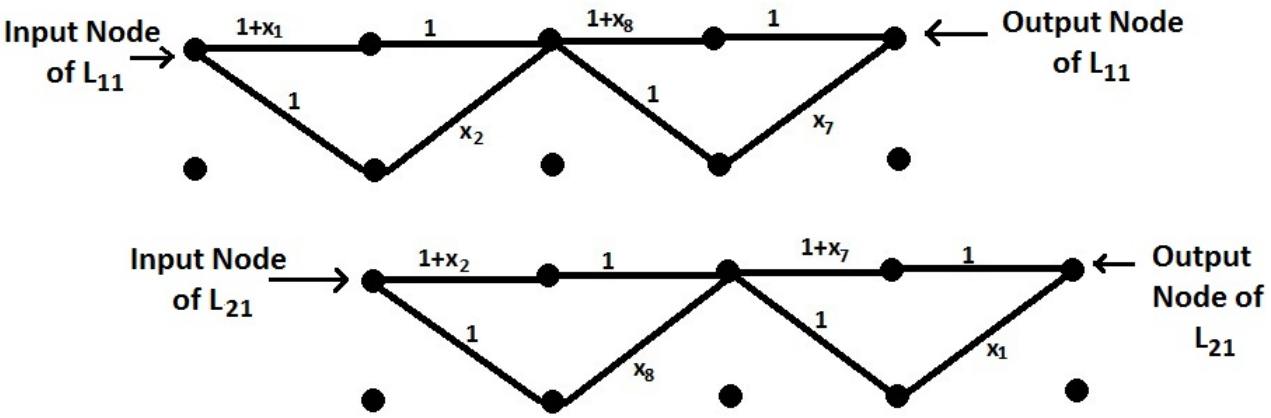

Figure 4.1: Directed acyclic graph corresponding to L_{11} and L_{21}

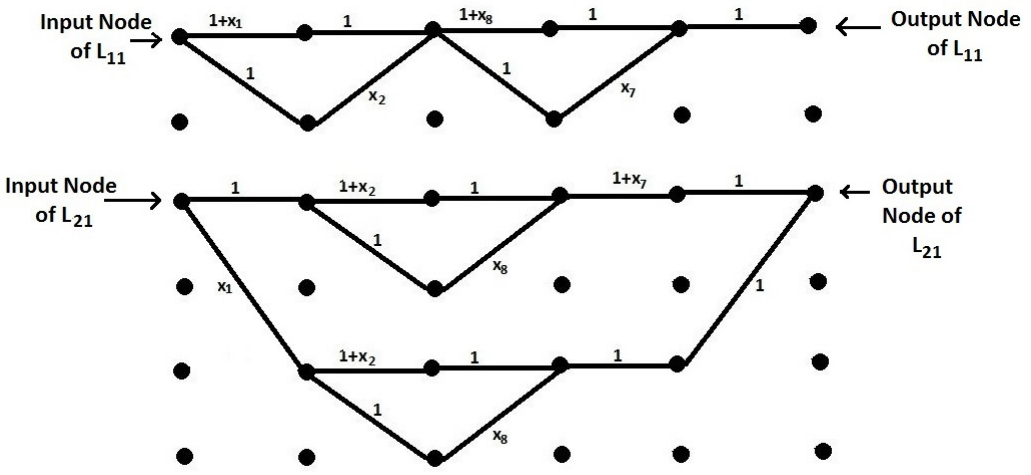

Figure 4.2: Directed acyclic graph corresponding to L_{11} and L_{21} in the same variable ordering

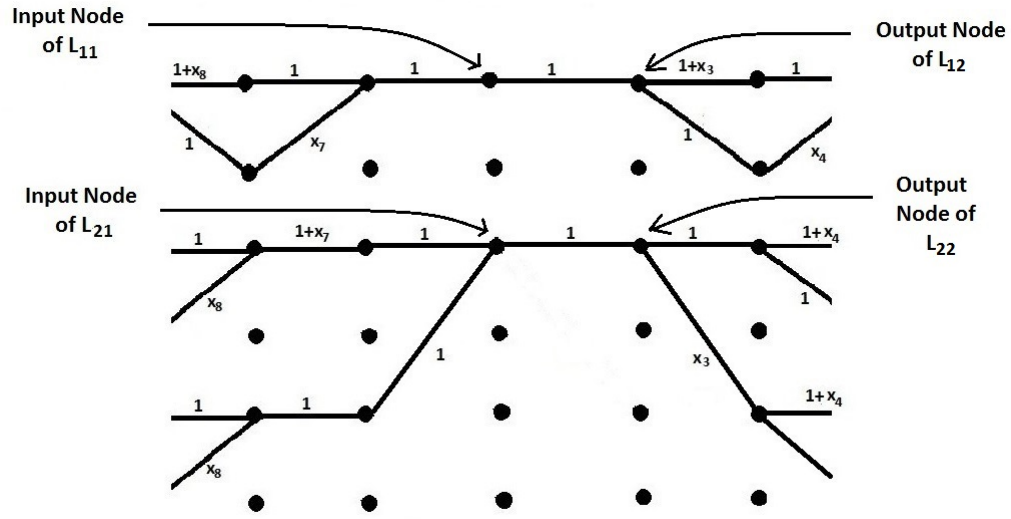

Figure 4.3: Connecting L_{11}, L_{21} and L_{12}, L_{22} in the same variable ordering

nodes of $L_{1(r+1)}, L_{2(r+1)}$ respectively, where $r \in [k-1]$. Figure 4.3 shows how to connect the graphs corresponding to L_{11}, L_{21} and L_{21}, L_{22} in the above example. We connect the right ends of the graph corresponding to L_{11}, L_{21} to the left ends of L_{12}, L_{22} such that the polynomial computed still remains the same. We connect the output node corresponding to L_{11} (first ROABP) to the input node corresponding to L_{21} by adding an edge of weight 1 between them. Similarly we connect the output node corresponding to L_{21} (second ROABP) to the input node corresponding to L_{22} by adding an edge of weight 1 between them. Finally we get a width six ROABP in the same variable ordering computing the original polynomial. The connection process is same for connecting all the k parts. Hence we see that any multilinear depth 3 circuit having two product gates and at most two variables per linear polynomial can be computed by a width six ROABP.

4.2 Lower Bounds for ROABPs

We saw in the previous section that a polynomial sized ROABP of constant width can be constructed for a polynomial $F(X)$ computed by a $\sum\prod\sum$ multlinear circuit C such that the top fan-in of C is 2 and the number of variables per linear polynomial in any product gate is at most 2. In this section we show that if we consider multilinear depth three circuits with three product gates and two variables per linear polynomial or two product gates and three variables per linear polynomial then there exists an explicit multilinear polynomial computed by such a circuit family, such that any ROABP computing it requires exponential width. In

subsection 4.2.1 we prove the lower bound for a polynomial $F_1(X,Y)$ computed by a multilinear depth 3 circuit C_1 such that the top fan-in of C_1 is 3 and the number of variables per linear polynomial is 2. In subsection 4.2.2 we prove the lower bound for a polynomial $F_2(X)$ computed by a multilinear depth 3 circuit C_2 such that the top fan-in of C_2 is 2 but the number of variables per linear polynomial is 3. In both subsection 4.2.1 and subsection 4.2.2 we construct the required polynomials from an explicit bipartite, 3-regular expander graph. Slightly weaker lower bound of $2^{\Omega(\sqrt{n})}$ on the width of any ROABP computing

$$F(X,Y) = \prod_{j=1}^{n}(1+x_j+y_j) + \prod_{j=1}^{n}(1+x_j+y_{j+1 \bmod n})$$

$$+\prod_{j=1}^{n}(1+x_j+y_{j+q \bmod n}) + \prod_{j=1}^{n}(1+y_j+x_{j+q \bmod n})$$

where $q = n - \frac{1.5\sqrt{n}}{16\epsilon}$ and $0 < \epsilon < \frac{1}{8}$, is given in section A.1. Also a $2^{\Omega(n)}$ lower bound on the witdth of any ROABP computing

$$F(X,Y) = \sum_{i=0}^{\frac{n}{2}-1}\prod_{j=1}^{n}(1+x_j+y_{(j+i) \bmod n})$$

is given in section A.2. The proofs in section A.1 and section A.2 do not use expander graphs, but in one case the lower bound is weaker whereas in the other case the top fan-in of the explicit multilinear depth 3 polynomial $F(X,Y)$ is $\Theta(n)$ (in contrast to the explicit polynomials used in next sections that have top fan-in either 3 or 2).

4.2.1 Lower Bounds for multilinear depth 3 circuits with 3 product gates and 2 base sets

We will construct a polynomial $F_1(X,Y)$ with the desired property from an explicit degree 3 expander. Suppose G is an n vertex 3-regular expander graph. Explicit construction of such an expander graph is given in subsection 2.5.2. Let $H = (L \cup R, E)$ be its double cover. We know from subsection 2.5.3 $h(H) > \frac{2+\frac{1}{10^4}}{2}$. With each vertex in L we associate a unique variable in X and similarly with each vertex in R we associate a unique variable in Y(X and Y represents variable sets, stated in section 2). An edge between x_i and y_j will correspond to a linear polynomial $(1+x_i+y_j)$ in $F_1(X,Y)$. From lemma 2.5.1 we know a 3-regular bipartite graph can be split into 3 edge disjoint perfect matchings. $F_1(X,Y)$ will have three product gates corresponding to the three edge disjoint perfect matchings. We will take the product of the

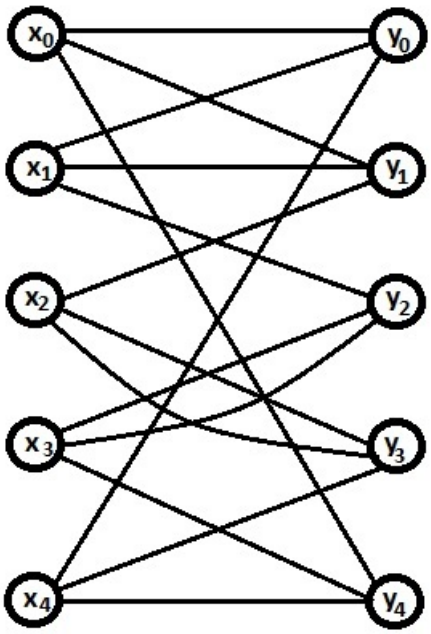

Figure 4.4: Double cover of $\mathbb{Z}_5$ shown in fig. 2.1

linear polynomials corresponding to the edges in a single matching and then take the sum of these products corresponding to the three edge disjoint perfect matchings. It is easy to see that each linear polynomial in every product (P_1, P_2 or P_3) has exactly two variables; one belongs to X and the other to Y. Hence polynomial $F_1(X, Y)$ is computed by C_1, a superposition of set-multilinear depth 3 circuit with 2 base sets and top fan-in 3. Let us explain the reduction with an example. Consider the double cover H of $\mathbb{Z}_5$ (fig. 2.1) shown in fig. 4.4. We split the edges in H into three edge disjoint perfect matchings as follows: $\{(x_0y_0), (x_1y_1), (x_2y_3), (x_3y_2), (x_4y_4)\}$, $\{(x_0y_1), (x_1y_2), (x_2y_3), (x_3, y_4), (x_4, y_0)\}$ and $\{(x_0y_4), (x_1y_0), (x_2y_1), (x_3, y_2), (x_4, y_3)\}$. Hence the polynomial we get from this graph by our reduction is

$$F_1(X,Y) = (1 + x_0 + y_0)(1 + x_1 + y_1)(1 + x_2 + y_3)(1 + x_3 + y_2)(1 + x_4 + y_4)$$

$$+(1 + x_0 + y_1)(1 + x_1 + y_2)(1 + x_2 + y_3)(1 + x_3 + y_4)(1 + x_4 + y_0)$$

$$+(1 + x_0 + y_4)(1 + x_1 + y_0)(1 + x_2 + y_1)(1 + x_3 + y_2)(1 + x_4 + y_3)$$

We will now compute the size of the circuit C_1. The bottom fan-in (fan-in of the sum gates at layer 3 from top) is 3, since there are 2 variables and a field constant per linear polynomial. The fan-in of every product gate is n. Since there are 3 product gates the total number of nodes

in C_1 is $4 + 3(n(1 + 3)) = 4 + 12n = O(n)$. Hence the size of the circuit C_1 is $O(n)$. We wish to prove a lower bound on the width of any ROABP that computes the polynomial $F_1(X, Y)$.

Theorem 4.2.1 (Main Theorem) *Any ROABP that computes the polynomial $F_1(X, Y)$ constructed as above has width $2^{\Omega(n)}$.*

Proof Outline: To prove the lower bound in theorem 4.2.1 we use evaluation dimension mentioned in section 2.4, as the measure. First we show that there exists a set of fixed size 's' such that evaluation dimension of the polynomial $F_1(X, Y)$ with respect to this set is always less than or equal to the width of the ROABP computing it. We then prove that for any set of size 's' the evaluation dimension of the polynomial $F_1(X, Y)$ with respect to it is at least $2^{\Omega(n)}$. This would imply any ROABP computing $F_1(X, Y)$ has width $2^{\Omega(n)}$.

Suppose R is a width-k ROABP that computes $F_1(X, Y)$. R has an associated variable ordering $(t_1, t_2, ..., t_{2n})$ of X and Y variables. Below we give an upper bound on the evaluation dimension of $F_1(X, Y)$ with respect to a set of size $\frac{n}{10}$ in terms of the width of R.

Lemma 4.2.1 *If R is a width-k ROABP that computes $F_1(X, Y)$ then there exists a set $S \subseteq X \cup Y$ of size $\frac{n}{10}$ such that* $\text{Evaldim}_S[F_1(X, Y)] \leq k$.

Proof: R can be viewed as the product of $2n$ matrices. Hence,

$$F_1(X, Y) = T_1 T_2 ... T_{2n}$$

where $T_1 \in \mathbb{F}[t_1]^{1\times k}$, $T_{2n} \in \mathbb{F}[t_{2n}]^{k\times 1}$ and $T_i \in \mathbb{F}[t_i]^{k\times k}$ for all $i \in [2, ..., 2n - 1]$. Fix $S = \{t_1, t_2, ..., t_{\frac{n}{10}}\}$. Consider any $\mathbb{F}$-evaluation of the S variables in $F_1(X, Y)$. Denote the resulting polynomial by $\mathbb{F}_{1_{eval}}(X, Y) \in \mathbb{F}[t_{n/10+1}, ..., t_{2n}]$. Hence,

$$F_{1_{eval}}(X, Y) = T_{eval} T_{\frac{n}{10}+1}, ..., T_{2n}$$

where $T_{eval} \in \mathbb{F}^{1\times k}$. Thus,

$$F_{1_{eval}}(X, Y) = T_{eval} T_p$$

where $T_p \in \mathbb{F}[t_{\frac{n}{10}+1}, ..., t_{2n}]^{1\times k}$. Thus $F_{1_{eval}}(X, Y)$ is a linear combination of some fixed k polynomials in the variables $\{t_{\frac{n}{10}+1}, ..., t_{2n}\}$. For any evaluation of the first $\frac{n}{10}$ variables these k polynomials remain fixed. Hence, evaluation dimension of $F_1(X, Y)$ with respect to S is less than equal to k, i.e $\text{Evaldim}_S[F_1(X, Y)] \leq k$. □

We will now prove a lower bound on the evaluation dimension of $F_1(X, Y)$ with respect to every

subset S of $X \cup Y$ variables where $|S| = \frac{n}{10}$.

Theorem 4.2.2 *For any set $S \subseteq X \cup Y$ with size equal to $\frac{n}{10}$, the* $\text{Evaldim}_S[F_1(X,Y)]$ *is at least $2^{\epsilon n}$ where ϵ is a constant greater than zero.*

Proof: Take any subset of $\frac{n}{10}$ variables from $X \cup Y$. Call this set S. With respect to set S we have three types of linear polynomials in $F_1(X,Y)$: untouched, partially touched and completely touched. A linear polynomials is untouched in $F_1(X,Y)$ if both the variables, one from X and the other from Y do not belong to S, whereas it is partially touched if exactly one of the two variables belongs to S and it is completely touched if both the variables belong to S. Sometimes we group the partially touched and completely touched linear polynomials and call them touched linear polynomials, i.e a linear polynomial is touched if at least one of the variables in the linear polynomial belongs to S. Here is an example.

Let $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3\}$ and $S = \{x_1, x_2, y_3\}$ then the linear polynomial $l_1 = (1 + x_3 + y_2)$ is an untouched linear polynomial since no variable in l_1 belongs to S whereas linear polynomials $l_2 = (1 + x_2 + y_3)$, $l_3 = (1 + x_1 + y_2)$ and $l_4 = (1 + x_3 + y_3)$ are touched linear polynomials since l_2, l_3 and l_4 contain at least one variable in S. Linear polynomial l_2 is a completely touched linear polynomial since both the variables in l_2, x_2 and y_3 belong to S. Linear polynomials l_3 and l_4 are partially touched linear polynomials since the variables x_1 in l_3 and y_3 in l_4 belong to S.

For $i \in [3]$, let A_i, B_i and C_i be the set of untouched, the set of partially touched and the set of completely touched linear polynomials in P_i respectively. To prove theorem 4.2.2 we first show that for any chosen subset S of size $\frac{n}{10}$, the size of the set of partially touched linear polynomials is at least ϵn in one of the product gates, i.e $\exists\ i$ such that, $|B_i| \geq \epsilon n$. We then show that if size of the set of partially touched linear polynomials is at least ϵn in one of the product gates then $\text{Evaldim}_S[F_1(X,Y)]$ is at least $2^{\epsilon n}$.

Lemma 4.2.2 *The size of the set containing the partially touched linear polynomials is greater than or equal to ϵn in at least one of the product gates.*

Proof: The objective of this lemma is to show that the number of partially touched linear polynomials in one of the product gates is at least ϵn, i.e $\exists i \in [3]$ such that, $|B_i| \geq \epsilon n$. Let T be such that, for all $i \in [3]$, $|B_i| \leq T$. We now look at H the bipartite expander graph from which the polynomial $F_1(X,Y)$ was constructed. Recall that in the construction we had labelled the vertices as variables and the edges as linear polynomials. Let A be the set of vertices in H

corresponding to the variables in S, thus $|A| = \frac{n}{10}$. Now using the expansion property of H we get

$$|E(A, \overline{A})| \geq h(H) \cdot |A| \geq \frac{2 + \frac{1}{10^4}}{2} \cdot \left(\frac{n}{10}\right)$$

Each edge in $E(A, \overline{A})$ connects a vertex in A to a vertex outside A. Since A corresponds to variables in S and $\overline{A}$ corresponds to variables not in S, each edge in $E(A, \overline{A})$ corresponds to a partially touched linear polynomial. Since the degree of H is 3 at least $\frac{|E(A,\overline{A})|}{3}$ of the edges correspond to unique partially touched linear polynomials. From our initial assumption the total number of such partially touched linear polynomials is at most $3T$; T from each product gate. Hence,

$$\Rightarrow \quad 3T \geq \frac{|E(A, \overline{A})|}{3} \geq \frac{2 + \frac{1}{10^4}}{6} \cdot \left(\frac{n}{10}\right)$$
$$\Rightarrow T \geq \epsilon n$$

where $\epsilon = 0.11$ greater than 0. □

Lemma 4.2.2 shows that size of either B_1, B_2 or B_3 is greater than equal to ϵn. We will use the next lemma to complete the proof of theorem 4.2.4.

Lemma 4.2.3 *If size of one of B_1, B_2 or B_3 is greater than or equal to ϵn then* $\text{Evaldim}_S[F_1(X, Y)]$ *is at least $2^{\epsilon n}$.*

Proof: For notational convenience we again let $T = \epsilon n$. Suppose $|B_i| \geq T$ where $i \in [3]$. We prove lemma 4.2.3 in two parts. In the first part we make appropriate substitutions of some variables such that product gates P_j and P_k compute the zero polynomial where $j, k \in [3]$ and $j \neq k \neq i$. We ensure that P_i remains non-zero after these substitutions. We then show that evaluation dimension with respect to S of the polynomial we achieve after these substitutions is less than or equal to the evaluation dimension of $F_1(X, Y)$ with respect to S. In the next part we show that the evaluation dimension with respect to S of the polynomial we achieve after these substitutions is at least $2^{\epsilon n}$. Thus part 1 and part 2 would together imply $\text{Evaldim}_S[F_1(X, Y)]$ is at least $2^{\epsilon n}$.

Part 1: We wish to perform substitutions such that P_j and P_k compute the zero polynomial but P_i still computes a non-zero polynomial. The easiest way to make any product gate compute the zero polynomial is by picking an untouched linear polynomial $l = (1 + x_r + y_m)$ in that product gate, where $x_r \in X$ and $y_m \in Y$, and substituting $x_r = -1 - y_m$. The untouched linear polynomial l is such that it does not appear in P_i. Hence after the substitution P_i is

non-zero. Also we want to ensure that the variable we perform substitution on, 'x_r' in the above example, appears in an untouched linear polynomial in all the product gates. This would imply the partially touched linear polynomials in P_i are not affected by these substitutions. This helps to simplify the proof in the second part. Now we show via a small argument that there are sufficient number of X variables that appear in an untouched linear polynomial in all the product gates.

Recall the size of S is equal to $\frac{n}{10}$. Hence the number of touched linear polynomials in all the product gates together is at most $\frac{3n}{10}$ and thus the number of variables appearing in these touched linear polynomials is at most $\frac{6n}{10}$. This implies at least $\frac{4n}{10}$ X variables appear in an untouched linear polynomial in all the product gates. Now $F_1(X,Y)$ is constructed from H such that, a linear polynomial l appears in two product gates if and only if there is a double edge between the endpoints of the edge corresponding to the linear polynomial l in H. Recall H is the double cover of G, a 3-regular expander graph. As mentioned in subsection 2.5.2, G belongs to a family of '3-regular p vertex graphs' for every prime p not equal to 2. Each vertex x in this graph is connected to $x+1$,$x-1$ and to its inverse x^{-1} (operations are $\mod p$ and inverse of 0 is 0). There exists a double edge in G if and only if any two of $x+1$,$x-1$ and x^{-1} are equal. If $x+1 = x-1 \mod p$, then $2 = 0 \mod p$. This cannot be true unless $p = 2$. Hence $x+1$ is not equal to $x-1$ when p is a prime not equal to 2. Moreover $x+1 = x^{-1} \mod p$ and $x-1 = x^{-1} \mod p$, each have at most two solutions. Hence there can be at most four double edges. Each double edge in G corresponds to two double edges in H, four of which corresponds to the X variables. Hence at least $\left(\frac{4n}{10} - 4\right)$ X variables are such that they appear in an untouched linear polynomial in all the product gates and the untouched linear polynomials in which they appear are unique to a product gate. We pick two of these variables x_{r_1} and x_{r_2} where $x_{r_1}, x_{r_2} \in X$, that appear in an untouched linear polynomial in all the product gates and the untouched linear polynomials in which they appear are unique to a product gate. Let the linear polynomials that x_{r_1} appears in P_j and x_{r_2} appears in P_k, where $j \neq k \neq i$ be, $l_j = (1 + x_{r_1} + y_{j_1})$ and $l_k = (1 + x_{r_2} + y_{k_1})$ where $y_{j_1}, y_{k_1} \in Y$. We go modulo l_j and l_k by substituting $x_{r_1} = -(1 + y_{j1})$ and $x_{r_2} = -(1 + y_{k1})$. For all $m \in [3]$ let $P_m^{'}$ correspond to P_m after these substitutions. Let $F_1^{'}(X,Y), l_j^{'}$ and $l_k^{'}$ correspond to $F_1(X,Y), l_j$ and l_k after these substitutions. These substitutions would ensure $l_j^{'} = 0$ and $l_k^{'} = 0$ which implies $P_j^{'} = P_k^{'} = 0$. Since the linear polynomials l_j and l_k corresponding to x_{r_1} and x_{r_2} appear only in P_j and P_k respectively (the untouched linear forms in which x_{r_1} and x_{r_2} appear are unique to a product gate), $P_i^{'} \neq 0$ .Thus we have $F_1^{'}(X,Y) = P_i^{'} \neq 0$. Below we prove that the evaluation dimension of $F_1^{'}(X,Y)$ with respect to S is at most the evaluation dimension of $F_1(X,Y)$ with respect to

S.

Claim 4.2.1 $\text{Evaldim}_S[F_1(X,Y)] \geq \text{Evaldim}_S[F_1^{'}(X,Y)]$

Proof: Let

$$E = \text{span}\{F_1(X,Y)|_{\forall x_m \in S \ x_m=\alpha_m} : \forall x_m \in S \ \alpha_m \in \mathbb{F}\}$$

$$E^{'} = \text{span}\{F_1^{'}(X,Y)|_{\forall x_m \in S \ x_m=\alpha_m} : \forall x_m \in S \ \alpha_m \in \mathbb{F}\}$$

and

$$\text{Evaldim}_S[F_1(X,Y)] = t$$

This implies every polynomial in E can be written as a linear combination of t linearly independent polynomials. First observe that the variables we have substituted, 'x_{r_1}' and 'x_{r_2}' are not in S. Once we make the substitutions, $x_{r_1} = -(1 + x_{j_1})$ and $x_{r_2} = -(1 + x_{k_1})$, two scenarios are possible. These t polynomials may still be linearly independent in which case $\text{Evaldim}_S[F_1^{'}(X,Y)] = t$. These t polynomials might become linearly dependent in which case every polynomial in $E^{'}$ can be written as a linear combination of some linearly independent subset of these t polynomials. □

We now go to the second part of the proof of lemma 4.2.3 where we show that $\text{Evaldim}_S[F_1^{'}(X,Y)] \geq 2^{\epsilon n}$.

Part 2: Observe that the substitutions we did in part 1 do not affect the partially touched linear polynomials in P_i. Hence $P_i^{'}$ has at least T partially touched linear polynomials. Each of these partially touched linear polynomials contains a field constant '1' and two variables; one of which belongs to S and the other does not. We pick exactly T of these partially touched linear polynomials. Choose one variable from each of these partially touched linear polynomials such that they belong to set S. Call them $Z_T = \{z_1, z_2, ..., z_T\}$, here $Z_T \subseteq X \cup Y$. Now we substitute some more variables. This is done in two steps. First we substitute all variables in S except those in Z_T to 1. In the second step, we focus on the linear polynomials in which the variables from Z_T appear in product gate $P_i^{'}$. Let us call the set of these linear polynomials as $L_T = \{l_1, l_2, ..., l_T\}$ such that $z_r \in Z_T$ appears in the linear polynomial l_r. For each $l_r \in L_T$, $l_r = (1 + z_r + u)$ where $z_r \in Z_T$ and $u \in X \cup Y$, we substitute $u = u - 1$. These substitutions would ensure each linear polynomial in L_T contains no field constants and exactly two variables; one belongs to S and the other does not. Call the polynomial we get after these substitutions as $F_1^{''}(X,Y)$ and the linear polynomial in which $z_r \in Z_T$ appears as $l_r^{''}$. Let $L_T^{''}$ correspond to L_T after these substitutions.

Claim 4.2.2 $\text{Evaldim}_S[F_1^{'}(X,Y)] \geq \text{Evaldim}_{Z_T}[F_1^{''}(X,Y)]$

Proof: We have made two types of substitutions to arrive at $F_1^{''}(X,Y)$ from $F_1^{'}(X,Y)$. In first set of substitutions we have put some of the variables in S as 1. It is easy to see that the evaluation dimension of the polynomial with respect to $Z_T \subseteq S$ after substituting some of the variables in S to 1 is always going to be at most the original evaluation dimension of the polynomial with respect to S. The second set of substitutions subtract '1' from some of the variables. It follows from claim 4.2.1 that this procedure does not increase the evaluation dimension. □

We will now show $\text{Evaldim}_{Z_T}[F_2(X)] \geq 2^{\epsilon n}$. Let $L^{''} = \{l_1^{''}, l_2^{''}, \ldots, l_N^{''}\}$ be the set of linear polynomials in $P_1^{''}$. Observe that the linear polynomials in $L^{''} \setminus L_T^{''}$ contain only variables from the set $(X \cup Y) \setminus Z_T$. Let $f((X \cup Y) \setminus Z_T) = \prod_{l'' \in (L'' \setminus L_T^{''})} l^{''}$. Hence we can write

$$F_1^{''}(X,Y) = (\prod_{l'' \in L_T} l^{''}) f((X \cup Y) \setminus Z_T)$$

$$F_1^{''}(X,Y) = \sum_{t=1; E_t \subseteq Z_T}^{t=2^T} z_{E_t} f_{E_t}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T)$$

where for $E_t \subseteq Z_T$, $z_{E_t} = \prod_{z_i \in E_t} z_i$ and $f_{E_t}((X \cup Y) \setminus Z_T)$ are just monomials in the variables $(X \cup Y \setminus Z_T)$ appearing in linear polynomials in $L_T^{''}$. If we show the set of polynomials $\{f_{E_1}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T), f_{E_2}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T), \ldots, f_{E_{2^T}}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T)\}$ where each polynomial corresponds to a subset of Z_T, are linearly independent then we can use claim 2.4.1 to claim that $\text{Evaldim}_{Z_T}[F_1^{''}(X,Y)] = 2^T$ since the variables in the set of polynomials $\{f_{E_1}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T), f_{E_2}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T), \ldots, f_{E_{2^T}}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T)\}$ are completely disjoint from the variables in Z_T.

Claim 4.2.3 *The set of polynomials* $\{f_{E_1}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T), f_{E_2}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T), \ldots, f_{E_{2^T}}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T)\}$ *where each polynomial corresponds to a subset of* Z_T*, are linearly independent.*

Proof: Assume for contradiction that there exists a linear dependence between the given polynomials. This implies there exists $\alpha_1, \ldots, \alpha_{2^T} \in \mathbb{F}$ such that not all α_i is equal to 0 where $i \in [2^T]$ and

$$\sum_{t=1; E_t \subseteq Z_T}^{t=2^T} \alpha_t f_{E_t}((X \cup Y) \setminus Z_T) f((X \cup Y) \setminus Z_T) = 0$$

$$\Rightarrow \sum_{t=1;E_t\subseteq Z_T}^{t=2^T} \alpha_t f_{E_t}((X \cup Y) \setminus Z_T) = 0$$

But $f_{E_1}, f_{E_2}, \ldots, f_{E_{2^T}}$ are distinct monomials which implies $\alpha_1 = \alpha_2 = \cdots = \alpha_{2^T} = 0$. Hence a contradiction. □

Hence $\text{Evaldim}_S[F_1(X,Y)] \geq \text{Evaldim}_S[F_1^{'}(X,Y)] \geq \text{Evaldim}_{Z_T}[F_1^{''}(X,Y)] = 2^T \geq 2^{\epsilon n}$. □

Combining lemma 4.2.2 and lemma 4.2.3 we can conclude for any set $S \subseteq X \cup Y$ with size equal to $\frac{n}{10}$ the $\text{Evaldim}_S[F_1(X,Y)]$ is at least $2^{\epsilon n}$. □

To complete the proof of theorem 4.2.1 we look at lemma 4.2.1 which tells us that there exists a choice of $\frac{n}{10}$ variables such that the evaluation dimension of $F_1(X,Y)$ is less than the width of the ROABP that computes $F_1(X,Y)$, but theorem 4.2.2 on the other hand states that evaluation dimension of $F_1(X,Y)$ with respect to any subset of X of size $\frac{n}{10}$ is at least $2^{\epsilon n}$. This implies that any ROABP that computes the polynomial $F_1(X,Y)$ has width equal to $2^{\Omega(n)}$.

4.2.2 Lower Bounds for multilinear depth 3 circuits with 2 product gates and 3 base sets

Like $F_1(X,Y)$ we will construct the polynomial $F_2(X)$ with the desired property from an explicit degree 3 bipartite expander graph. Suppose G is an N vertex 3-regular expander graph. Explicit construction of such an expander graph is given in subsection 2.5.2. Let $H = (L \cup R, E)$ be its double cover. We know from subsection 2.5.3 $h(H) > \frac{2+\frac{1}{10^4}}{2}$. Unlike the previous construction we associate a unique variable from the set X with every edge in the graph H. The number of variables is thus equal to $n = 3N$. The three variables, say x_i, x_j and x_k correspondng to each edge incident on a vertex u in H form a linear polynomial $(1 + x_i + x_j + x_k)$, i.e we have a linear polynomial corresponding to every vertex in H. Observe that this is complete contrast to the construction in subsection 4.2.1, where we associated the variables with vertices and the linear polynomials with edges. We will take the product of the linear polynomials corresponding to vertices in L and R separately and then add this two products. It is easy to see $F_2(X)$ can be computed by $\sum\prod\sum$ circuit C_2 with top fan-in just two. Again by lemma 2.5.1 we know a 3-regular bipartite graph can be split into three edge disjoint perfect matchings. Each linear polynomial in $F_2(X)$ contains three variables corresponding to edges from three edge disjoint perfect matchings. We can group the variables corresponding to edges in a single matching into a single base set. This implies the variables can be split into three distinct base sets. Hence C_2

is a superposition of set-multilinear depth three circuits with three base sets and the top fan-in of C_2 is two. Let us explain the reduction with an example. Consider the double cover H of $\mathbb{Z}_5$ graph shown in $fig.$ 4.4. If there is an edge $(x_i y_j)$ we label the edge by x_{ij} The polynomial we get from this graph by our reduction is

$$F_2(X) = (1+x_{00}+x_{01}+x_{04})(1+x_{10}+x_{11}+x_{13})(1+x_{21}+x_{23}+x_{23})(1+x_{32}+x_{32}+x_{34})(1+x_{40}+x_{43}+x_{44})$$

$$+(1+x_{00}+x_{10}+x_{41})(1+x_{01}+x_{11}+x_{21})(1+x_{12}+x_{32}+x_{32})(1+x_{23}+x_{23}+x_{43})(1+x_{04}+x_{34}+x_{44})$$

To compute the size of C_2 we will compute the number of nodes in C_2 which is equal to $1+2+2(N(1+4)) = 3+10N = O(n)$. Hence the size of the circuit C_2 is equal to $O(n)$. We prove the following for $F_2(X)$.

Theorem 4.2.3 (Main Theorem) *Any ROABP that computes the polynomial $F_2(X)$ constructed as above has width $2^{\Omega(n)}$.*

Proof Outline: The proof strategy for theorem 4.2.3 is similar to theorem 4.2.1. We use evaluation dimension mentioned in section 2.4 as the measure to prove the lower bound. First we show that there exists a set of fixed size 's' such that evaluation dimension of the polynomial $F_2(X)$ with respect to this set is always less than or equal to the width of the ROABP computing it. We then prove that for any set of size 's' the evaluation dimension of the polynomial $F_2(X)$ with respect to it is at least $2^{\Omega(n)}$. This would imply any ROABP computing $F_2(X)$ has width $2^{\Omega(n)}$.

Suppose R is a width-k ROABP that computes $F_2(X)$. R has an associated variable ordering $(t_1, t_2, ..., t_n)$ of X variables. Below we give an upper bound on the evaluation dimension of $F_2(X)$ with respect to a set of size $\frac{n}{10}$ in terms of the width of R.

Lemma 4.2.4 *If R is a width-k ROABP that computes $F_2(X)$ then there exists a set $S \subseteq X \cup Y$ of size $\frac{n}{10}$ such that* $\text{Evaldim}_S[F_2(X)] \leq k$.

The proof of lemma 4.2.4 is similar to lemma 4.2.1. From lemma 4.2.1 it follows that if we fix S to be the first $\frac{n}{10}$ variables in the ordering $(t_1, t_2, ..., t_n)$ then the evaluation dimension of the polynomial with respect to S is less than the width of the ROABP computing it. We will now prove a lower bound on the evaluation dimension of $F_2(X)$ with respect to every subset S of X variables where the $|S| = \frac{n}{10}$.

Theorem 4.2.4 *For any set $S \subseteq X$ with size equal to $\frac{n}{10}$ the* $\text{Evaldim}_S[F_2(X)]$ *is at least $2^{\epsilon n}$ where ϵ is a constant greater than 0.*

Proof: Take any subset of $\frac{n}{10}$ variables from X. Call this set S. With respect to set S we have three types of linear polynomials in $F_2(X)$: untouched, partially touched and completely touched linear polynomials. A linear polynomial is untouched in $F_2(X)$ if all the variables in the linear polynomial do not belong to S, whereas it is partially touched if either one or two of the three variables in the linear polynomial belongs to S and it is completely touched if all the variables in the linear polynomial belongs to S. Sometimes we group the partially touched and completely touched linear polynomials and call them touched linear polynomials, i.e a linear polynomial is touched if at least one of the variables in the linear polynomial belongs to S. Let us illustrate this with an example.

Let $X = \{x_1, x_2, ..., x_6\}$ and $S = \{x_1, x_2, x_3\}$ then the linear polynomial $l_1 = (1 + x_4 + x_5 + x_6)$ is an untouched linear polynomial since no variable in l_1 belongs to S whereas the linear polynomials $l_2 = (1 + x_1 + x_2 + x_3)$, $l_3 = (1 + x_1 + x_2 + x_4)$ and $l_4 = (1 + x_2 + x_4 + x_5)$ are touched linear polynomials since l_2, l_3 and l_4 contain at least one variable in S. Linear polynomial l_2 is a completely touched linear polynomial since all the variables in l_2, x_1, x_2 and x_3 belong to S. Linear polynomials l_3 and l_4 are partially touched linear polynomials since the variables x_1, x_2 in l_3 and the variable x_2 in l_4 belong to S.

For $i \in [2]$, let A_i be equal to the set of untouched linear polynomials, B_i be equal to the set of partially touched linear polynomials and C_i be equal to the set of completely touched linear polynomials in P_i. To prove theorem 4.2.4 we first show that for any chosen subset S of size $\frac{n}{10}$, the size of the set of partially touched linear polynomials is at least ϵn in one of the product gates, i.e $\exists\ i \in [2]$ such that $|B_i| \geq \epsilon n$. We then show that if size of the set of partially touched linear polynomials is at least ϵn in one of the product gates then $\text{Evaldim}_S[F_2(X)]$ is at least $2^{\epsilon n}$.

Lemma 4.2.5 *The size of the set containing the partially touched linear polynomials is greater than or equal to ϵn in at least one of the product gates.*

Proof: The objective of this lemma is to show that the number of partially touched linear polynomials in one of the product gates is at least ϵn, i.e $\exists i \in [2]$, such that $|B_i| \geq \epsilon n$. Let T be such that, for all $i \in [2]$, $|B_i| \leq T$. Observe that if a linear polynomial is partially touched then the number of variables that belong to S from that linear polynomial is at most 2, since the number of variables per linear polynomial is 3. This implies the number of variables that belong to S and appearing in a partially touched linear polynomial in any of the product gates is at most $4T$; $2T$ from each product gate. Hence at least $\frac{n}{10} - 4T$ variables in S, appear in a

completely touched linear polynomial in both the product gates. Since the number of variables per linear polynomial is 3, the number of completely touched linear polynomials in both the product gates together is at least $2 \cdot \left(\frac{n}{30} - \frac{4T}{3}\right)$, i.e $|C_1| + |C_2|$ is at least $\frac{n}{15} - \frac{8T}{3}$. Now we look at H the bipartite expander graph from which the polynomial $F_1(X, Y)$ was reduced. Recall that in the reduction we had labelled the edges as variables and the variables as linear polynomials. Let C be the set of vertices corresponding to the completely touched linear polynomials in both the product gates, thus $|C| = |C_1| + |C_2| \geq \frac{n}{15} - \frac{8T}{3}$. Now using the expansion property of H we get

$$|E(C, \overline{C})| \geq h(H) \cdot |C| \geq \frac{2 + \frac{1}{10^4}}{2} \cdot \left(\frac{n}{15} - \frac{8T}{3}\right)$$

Each edge in $E(C, \overline{C})$ connects a vertex in C to a vertex in $\overline{C}$. Since C contains those vertices which correspond to completely touched linear polynomials, the edges in $E(C, \overline{C})$ correspond to variables which are in S. This implies each edge in $E(C, \overline{C})$ connects a vertex which corresponds to a completely touched linear polynomial to a vertex which corresponds to a partially touched linear polynomial. Since edges in $E(C, \overline{C})$ correspond to variables in S, a vertex corresponding to a partially touched linear polynomial would have at most two edges incident to it from $E(C, \overline{C})$. Hence the number of vertices corresponding to partially touched linear polynomials is at least $\frac{E(C,\overline{C})}{2}$. But from our initial assumption we know that the number of partially touched linear polynomials is at most $2T$; T from each product gate. Thus the number of vertices corresponding to partially touched linear polynomials is at most $2T$. Hence we get

$$2T \geq \frac{|E(C, \overline{C})|}{2} \geq \frac{2 + \frac{1}{10^4}}{2} \cdot |C| \geq \frac{2 + \frac{1}{10^4}}{2} \cdot \left(\frac{n}{15} - \frac{8T}{3}\right)$$

$$\Rightarrow T \geq \epsilon n$$

where $\epsilon = 0.025$ greater than 0. □

Lemma 4.2.5 shows that size of either B_1 or B_2 is greater than equal to ϵn. We will use the next lemma to complete the proof of theorem 4.2.4.

Lemma 4.2.6 *If size B_1 or B_2 is greater than or equal to ϵn then* $\text{Evaldim}_S[F_2(X)]$ *is at least* $2^{\epsilon n}$.

Proof: For notational convenience we again let $T = \epsilon n$. Suppose $|B_i| \geq T$ where $i \in [2]$. We prove lemma 4.2.6 in two parts. In the first part we make appropriate substitutions of some variables such that product gate P_j computes the zero polynomial where $j \in [2]$ and $j \neq i$. We ensure that P_i remains non-zero after these substitutions. We then show that evaluation

dimension with respect to S of the polynomial we achieve after these substitutions is less than or equal to the evaluation dimension of $F_2(X)$ with respect to S. In the next part we show that the evaluation dimension with respect to S of the polynomial we achieve after these substitutions is at least $2^{\epsilon n}$. Thus part 1 and part 2 together would imply $\text{Evaldim}_S[F_2(X)]$ is at least $2^{\epsilon n}$.

Part 1: We wish to perform substitutions such that P_j computes the zero polynomial but P_i still computes a non-zero polynomial. The easiest way to make any product gate compute the zero polynomial is by picking an untouched linear polynomial $l = (1 + x_1 + x_2 + x_3)$ in that product gate, where $x_1, x_2, x_3 \in [X]$ and substituting $x_1 = -1 - x_2 - x_3$. Since no two vertices in H have all the three edges in common, the linear polynomial l is unique to a product gate, i.e if l is a linear factor of P_j then l is not a linear factor of P_i. Hence P_i computes a non-zero polynomial after this substitution. Moreover we want to ensure that the variable we perform substitution on, 'x_1' in the above example, appears in an untouched linear polynomial in both the product gates. This would imply the partially touched linear polynomials in P_i are not affected by these substitutions. This helps to simplify the proof in the second part. Now we show via a small argument that there are sufficient number of variables that appear in an untouched linear polynomial in both the product gates.

Recall the size of S is equal to $\frac{n}{10}$. Hence the number of touched linear polynomials in either of the product gates is at most $\frac{2n}{10}$; $\frac{n}{10}$ from each product gate. Thus the number of variables appearing in these touched linear polynomials is at most $\frac{6n}{10}$. This implies at least $\frac{4n}{10}$ variables appear in an untouched linear polynomial in both the product gates. We pick one such variable x_r that appears in an untouched linear polynomial in both product gates. Let the linear polynomials x_r appears in P_i and P_j where $j \neq i$ be $l_i = (1 + x_r + x_{i1} + x_{i2})$ and $l_j = (1 + x_r + x_{j1} + x_{j2})$. We go modulo l_j by substituting $x_r = -(1 + x_{j1} + x_{j2})$. For all $m \in [2]$, let $l_m^{'}$,$P_m^{'}$, and $F_2^{'}(X)$ correspond to l_m,P_m and $F_2(X)$ after this substitution respectively. This substitution would ensure $l_j^{'} = 0$ which implies $P_j^{'} = 0$ and $l_i^{'} = (x_{i1} + x_{i2} - x_{j1} - x_{j2}) \neq 0$ which implies $P_i^{'} \neq 0$. Hence we have $F_2^{'}(X) = P_i^{'} \neq 0$. Below we prove that the evaluation dimension of $F_2^{'}(X)$ with respect to S is at most the evaluation dimension of $F_2(X)$ with respect to S.

Claim 4.2.4 $\text{Evaldim}_S[F_2(X)] \geq \text{Evaldim}_S[F_2^{'}(X)]$

Proof: Let

$$E = \text{span}\{F_2(X)|_{\forall x_m \in S \ x_m = \alpha_m} : \forall x_m \in S \ \alpha_m \in \mathbb{F}\}$$

$$E^{'} = \text{span}\{F_2^{'}(X)|_{\forall x_m \in S \ x_m = \alpha_m} : \forall x_m \in S \ \alpha_m \in \mathbb{F}\}$$

and

$$\text{Evaldim}_S[F_2(X)] = t$$

This implies every polynomial in E can be written as a linear combination of t linearly independent polynomials. After substituting $x_r = -(1 + x_{j1} + x_{j2})$ two scenarios are possible. These t polynomials may still be linearly independent in which case $\text{Evaldim}_S[F_2^{'}(X)] = t$. These t polynomials might become linearly dependent in which case every polynomial in $E^{'}$ can be written as a linear combination of some linearly independent subset of these t polynomials. □

We now go to the second part of the proof of lemma 4.2.6 where we show that $\text{Evaldim}_S[F_2^{'}(X)] \geq 2^{\epsilon n}$.

Part 2: Observe that the substitutions we did in part 1 do not affect the partially touched linear polynomials in P_i. Hence $P_i^{'}$ has at least T partially touched linear polynomials. Each of these partially touched linear polynomials contains a field constant '1' and three variables; at least one of which does not belong to S. We pick exactly T of these partially touched linear polynomials. Choose one variable from each of these partially touched linear polynomials such that they belong to set S. Call them $X_T = \{x_{j_1}, x_{j_2}, ..., x_{j_T}\}$. Substitute all the variables in S except those in X_T to 1. Once we have made these substitutions we focus on the linear polynomials in which the X_T variables appear in product gate $P_i^{'}$. Let us call the set of these linear polynomials as $L_T = \{l_1, l_2, ..., l_T\}$ such that $x_{j_r} \in X_T$ appears in the linear polynomial l_r. For each $l_r \in L_T$ if $l_r = (1 + x_{j_r} + x_u + x_v)$ where $x_r \in X_T$ and $x_u, x_v \in X \setminus S$ we substitute $x_u = -1$ else if $l_r = (2 + x_{j_r} + x_u)$ where $x_r \in X_T$ and $x_u \in X \setminus S$ we substitute $x_u = x_u - 2$. This substitution would ensure each linear polynomial in L_T contains no field constants and exactly two variables; one belongs to S and the other does not. Call the polynomial we get after these substitutions as $F_2^{''}(X)$ and the linear polynomial in which $x_{j_r} \in X_T$ appears as $l_r^{''}$. Let $L_T^{''}$ correspond to L_T and $P_i^{''}$ correspond to $P_i^{'}$ after these substitutions.

Claim 4.2.5 $\text{Evaldim}_S[F_2^{'}(X)] \geq \text{Evaldim}_{X_T}[F_2^{''}(X)]$

Proof: We have made two types of substitutions to arrive at $F_2^{''}(X)$ from $F_2^{'}(X)$. In the first set of substitutions we have put some of the variables in S as 1. It is easy to see that the evaluation dimension of the polynomial after substituting some of the variables in S to 1 is always going to be at most the original evaluation dimension of the polynomial with respect to S. The second set of substitutions eliminates the extra variable and the field constant so that each linear polynomial in $L_T^{''}$ has only two variables and no field constants. It follows from

claim 4.2.4 that this procedure does not increase the evaluation dimension. □

We will now show $\text{Evaldim}_{X_T}[F_2^{''}(X)] \geq 2^{\epsilon n}$. Let $L^{''} = \{l_1^{''}, l_2^{''}, \ldots, l_N^{''}\}$ be the set of linear polynomials in $P_1^{''}$. Observe that the linear polynomials in $L^{''} \setminus L_T^{''}$ contain only variables from the set $X \setminus X_T$. Let $f(X \setminus X_T) = \prod_{l'' \in (L'' \setminus L_T'')} l^{''}$. Hence we can write

$$F_2^{''}(X) = (\prod_{l'' \in L_T''} l^{''}) f(X \setminus X_T)$$

$$F_2^{''}(X) = \sum_{t=1; E_t \subseteq X_T}^{t=2^T} x_{E_t} f_{E_t}(X \setminus X_T) f(X \setminus X_T)$$

where for $E_t \subseteq X_T$, $x_{E_t} = \prod_{x_i \in E_t} x_i$ and $f_{E_t}(X \setminus X_T)$ are just monomials in the variables $(X \setminus X_T)$ appearing in linear polynomials in $L_T^{''}$. If we show the set of polynomials $\{f_{E_1}(X \setminus X_T)f(X \setminus X_T), f_{E_2}(X \setminus X_T)f(X \setminus X_T), \ldots, f_{E_{2^T}}(X \setminus X_T)f(X \setminus X_T)\}$ where each polynomial corresponds to a subset of X_T, are linearly independent then we can use claim 2.4.1 to claim that $\text{Evaldim}_{X_T}[F_1^{''}(X, Y)] = 2^T$, since the set of polynomials $\{f_{E_1}(X \setminus X_T)f(X \setminus X_T), f_{E_2}(X \setminus X_T)f(X \setminus X_T), \ldots, f_{E_{2^T}}(X \setminus X_T)f(X \setminus X_T)\}$ are defined on variables completely disjoint from X_T.

Claim 4.2.6 *The set of polynomials* $\{f_{E_1}(X \setminus X_T)f(X \setminus X_T), f_{E_2}(X \setminus X_T)f(X \setminus X_T), \ldots, f_{E_{2^T}}(X \setminus X_T)f(X \setminus X_T)\}$ *where each polynomial corresponds to a subset of* X_T*, are linearly independent.*

Proof: Assume for contradiction that there exists a linear dependence between the given polynomials. This implies there exists $\alpha_1, \ldots, \alpha_{2^T} \in \mathbb{F}$ such that not all α_i is equal to 0 where $i \in [2^T]$ and

$$\sum_{t=1; E_t \subseteq X_T}^{t=2^T} \alpha_t f_{E_t}(X \setminus X_T) f(X \setminus X_T) = 0$$

$$\Rightarrow \sum_{t=1; E_t \subseteq X_T}^{t=2^T} \alpha_t f_{E_t}(X \setminus X_T) = 0$$

But $f_{E_1}, f_{E_2}, \ldots, f_{E_{2^T}}$ are distinct monomials which implies $\alpha_1 = \alpha_2 = \cdots = \alpha_{2^T} = 0$. Hence a contradiction. □

Hence $\text{Evaldim}_S[F_2(X)] \geq \text{Evaldim}_S[F_2^{'}(X)] \geq \text{Evaldim}_{X_T}[F_2^{''}(X)] = 2^T \geq 2^{\epsilon n}$. □

Combining lemma 4.2.5 and lemma 4.2.6 we can conclude for any set $S \subseteq X \cup Y$ with size

equal to $\frac{n}{10}$ the $\text{Evaldim}_S[F_2(X)]$ is at least $2^{\epsilon n}$. $\square$

To complete the proof of theorem 4.2.3 we look at lemma 4.2.4 which tells us that there exists a choice of $\frac{n}{10}$ variables such that the evaluation dimension of $F_S(X)$ is less than the width of the ROABP that computes $F(X)$, but theorem 4.2.4 on the other hand states that evaluation dimension of $F(X, Y)$ with respect to any subset of X of size $\frac{n}{10}$ is at least $2^{\frac{\epsilon n}{2}}$. This implies that any ROABP that computes the polynomial $F_2(X)$ has width equal to $2^{\Omega(n)}$.

Chapter 5

Future Work

At this point we have many interesting questions to resolve. Firstly we would like to show a complete separation between ROABPs and multlinear depth three circuits. As part of this work we have partly resolved this. In section 4.2 we show an explicit polynomial which is computed by a polynomial sized multlinear depth three circuit but every ROABP computing it requires exponential sized width. We wish to show a similar result in the opposite direction, i.e show an explicit polynomial that can be computed by a polynomial sized ROABP but every multilinear depth three circuit computing it requires exponential size. We plan to use 'projected shifted partial derivative' as the measure, first used in [KLSS14], to show an exponential lower bound for homogeneous depth four circuits. Precisely they show that any homogeneous depth four circuit computing the Nisan-Wigderson polynomial (a polynomial in VNP) requires size $n^{\Omega(\sqrt{n})}$. In [KS14] they have analyzed the shifted partial derivative for 'Iterated Matrix Multiplication (IMM)' (a polynomial in VP) and proved that any homogeneous depth four circuit computing IMM requires size $n^{\Omega(\sqrt{n})}$. To prove this they show a lower bound on 'projected shifted partial derivative' for IMM. We need a slightly stronger lower bound on 'projected shifted partial derivative' for IMM than that shown in [KS14], to show a complete separation between ROABPs and multlinear depth three circuits.

An another interesting direction is to consider 'sum of constantly many ROABPs'. Can we show an explicit polynomial which is computed by a polynomial sized multlinear depth three circuit that if computed by a sum of constantly many ROABPs, at least one ROABP among them has super-polynomial width. [GKST15] gave a quasi-polynomial time hitting set for sum of constantly many ROABPs.

We could also look at the algorithmic perspective and ask for PIT for the superposition model.

Can we give quasi-polynomial time hitting set for superposition of 2 set-multilinear circuits.

Appendix A

ROABP lower bound without expanders

In this appendix we are including two self-contained ROABP lower bound proofs that does not rely on explicit constructions of expander graphs. These bounds are weaker in certain senses - one yields an exp($\sqrt{n}$) lower bound as opposed to exp(n) lower bound achieved earlier, while the other uses a multilinear depth three circuit with $O(n)$ top fan-in unlike in chapter 4 where the target hard polynomial has fanin either 2 or 3. We are including the full proofs of these bounds in this appendix for completion (just to make the point that weaker ROABP lower bounds can be proved without using expander graphs).

A.1 Exp($\sqrt{n}$) lower bound against ROABPs

In this section we give a $2^{\Omega(\sqrt{n})}$ lower bound against ROABP for a polynomial computable by superposition of two set-multilinear circuits with singleton colors and having four product gates, without using the expander graphs. The polynomial is the superposition of the following two polynomials.

$$C_1(X) = \sum_{i=1}^{4} \prod_{j=1}^{n} (1 + x_j)$$

i.e the top fan in of $C_1(X)$ is equal to four and the partition $X_{11}, X_{12}, ..., X_{1n}$ are singletons containing $x_1, x_2, ..., x_n$ respectively. $C_2(Y)$ is defined similarly i.e

$$C_2(Y) = \sum_{i=1}^{4} \prod_{j=1}^{n} (1 + y_j)$$

$F(X,Y)$ represents the superposition of $C_1(X)$ and $C_2(Y)$.

$$F(X,Y) = \prod_{j=1}^{n}(1+x_j+y_j) + \prod_{j=1}^{n}(1+x_j+y_{j+1 \mod n})$$

$$+\prod_{j=1}^{n}(1+x_j+y_{j+q \mod n}) + \prod_{j=1}^{n}(1+y_j+x_{j+q \mod n})$$

where $q = n - \frac{1.5\sqrt{n}}{16\epsilon}$ and $0 < \epsilon < \frac{1}{8}$. [1]

We follow the following convention in this section, $n \equiv n \mod n$ and $n+1 \equiv 1 \mod n$. $F(X,Y)$ has four product gates represented by P_1, P_2, P_3 and P_4, i.e

$$P_1 = \prod_{j=1}^{n}(1+x_j+y_j)$$

$$P_2 = \prod_{j=1}^{n}(1+x_j+y_{j+1 \mod n})$$

$$P_3 = \prod_{j=1}^{n}(1+x_j+y_{j+q \mod n})$$

$$P_4 = \prod_{j=1}^{n}(1+y_j+x_{j+q \mod n})$$

We wish to prove a lower bound on the width of any ROABP that computes the polynomial $F(X,Y)$.

Theorem A.1.1 (Main Theorem) *Any ROABP that computes the polynomial*

$$F(X,Y) = \prod_{j=1}^{n}(1+x_j+y_j) + \prod_{j=1}^{n}(1+x_j+y_{j+1 \bmod n})$$

$$+\prod_{j=1}^{n}(1+x_j+y_{j+q \bmod n}) + \prod_{j=1}^{n}(1+y_j+x_{j+q \bmod n})$$

where $q = n - \frac{1.5\sqrt{n}}{16\epsilon}$ *and* $0 > \epsilon < \frac{1}{8}$ *has width* $2^{\Omega(\sqrt{n})}$.

[1] We are avoiding ceil\floor notations for simplicity of exposition. Assume that q is an integer.

We follow a similar proof strategy that we used in the proofs of theorem 4.2.1 and theorem 4.2.3. Suppose R is a width-k ROABP that computes $F(X,Y)$. R has an associated variable ordering $(t_1, t_2, ..., t_{2n})$ of X and Y variables. First we give an upper bound on the $\text{Evaldim}_S[F(X,Y)]$ in terms of the width of R.

Lemma A.1.1 *If R is a width-k ROABP that computes $F(X,Y)$ then there exists a set $S \subseteq X \cup Y$ of size $\frac{n}{4}$ such that $\text{Evaldim}_S[F(X,Y)] \leq k$*

The proof for lemma A.1.1 is similar to lemma 4.2.1. We will now prove a lower bound on the evaluation dimension of $F(X,Y)$ with respect to every subset S of the $X \cup Y$ variables where the $|S| = \frac{n}{4}$.

Theorem A.1.2 *For any set $S \subseteq X \cup Y$ with size equal to $\frac{n}{4}$ the $\text{Evaldim}_S[F(X,Y)]$ is at least $2^{\frac{\epsilon\sqrt{n}}{2}}$ where $\epsilon < \frac{1}{8}$.*

Take any subset of $\frac{n}{4}$ variables from $X \cup Y$. Call this set S. With respect to set S we have three types of linear polynomials in $F(X,Y)$: untouched, partially touched and completely touched linear polynomials. A linear polynomial is untouched in $F(X,Y)$ if both the variables one from the set X and the other from set Y do not belong to the set S, whereas it is partially touched if exactly one of the two variables belongs to the set S and its completely touched if both the variables in the linear polynomial belongs to the set S. Let A_i=the set of untouched linear polynomials in P_i, B_i=the set of partially touched linear polynomials in P_i and C_i=the set of completely touched linear polynomials in P_i.

Lemma A.1.2 *The size of the set containing the partially touched linear polynomials is greater than or equal to $\epsilon\sqrt{n}$ in at least one of the product gates.*

Proof: With respect to set S we have three types of linear polynomials in $F(X,Y)$: untouched, partially touched and completely touched linear polynomials. A linear polynomial is untouched in $F(X,Y)$ if both the variables one from the set X and the other from set Y do not belong to the set S, whereas it is partially touched if exactly one of the two variables belongs to the set S and its completely touched if both the variables in the linear polynomial belongs to the set S. Let A_i=the set of untouched linear polynomials in P_i, B_i=the set of partially touched linear polynomials in P_i and C_i=the set of completely touched linear polynomials in P_i.

We are to show in this lemma that the number of partially touched linear polynomials in one of the product gates is at least $\epsilon\sqrt{n}$, i.e $\exists i \in [4]\ |B_i| \geq \epsilon\sqrt{n}$. Let $T = \epsilon\sqrt{n}$, we assume for contradiction for all $i \in [4]\ |B_i| < T$. Since $|S| = \frac{n}{4}$, we have at least $\frac{n}{8}$ variables which are

either completely X variables or completely Y variables.
Case 1: Assume the $\frac{n}{8}$ variables are completely X variables.
Of these $\frac{n}{8}$ X variables at least $(\frac{n}{8} - T)$ variables appear in the linear polynomials in set C_1. We want to see how many of these $(\frac{n}{8} - T)$ X variables are continuous chunks, where a continuous chunk is a set of variables that are sequential based on the ordering $(x_1, ..., x_n)$. For example (x_1, x_2, x_3) is a continuous chunk whereas (x_1, x_2, x_3, x_6) is not a continuous chunk. Holes are the set of variables that appear between the continuous chunks. For example say (x_1, x_2, x_3, x_6) appear in the linear polynomials in set C_1 whereas (x_4, x_5) does not, then (x_4, x_5) is a hole . It is easy to see that the number of holes is equal to the number of continuous chunks of variables (that appear in the linear polynomials in set C_1).

Claim A.1.1 *Number of holes between the continuous chunks of variables that appear in the linear polynomials in set C_1 is less than $2T$.*

Proof: Let δ denote the number of holes that exist between the continuous chunks of variables (that appear in the linear polynomials in the set C_1). If δ holes exist then there exists δ distinct $i's$ in $[n-1]$ such that, x_i does not appear in in any of the linear polynomials in the set C_1 and x_{i+1} appears in one of the linear polynomials in the set C_1.

SubClaim A.1.1 *For all $i \in [n-1]$ x_i does not appear in any of the linear polynomials in C_1 and x_{i+1} appears in one of the linear polynomials in $C_1 \Rightarrow x_i$ appears in one of the linear polynomials in B_1 or B_2.*

Proof: There are 2 possibilities $x_i \in S$ or $x_i \notin S$. $x_i \in S \Rightarrow x_i$ appears in one of the linear polynomials in B_1 since x_i does not appear in any of the linear polynomials in C_1. x_{i+1} appears in one of the linear polynomials in C_1 implies $y_{i+1} \in S$. Thus $x_i \notin S \Rightarrow x_i$ appears in one of the linear polynomials in B_2 □

Using subClaim A.1.1 we conclude that the number of distinct $i's$ in $[n-1]$ such that x_i does not appear in any of the linear polynomials in C_1 and x_{i+1} appears in one of the linear polynomials in C_1 is less than $2T$. This implies δ is less than $2T$. □

As the number of holes is equal to the number of continuous chunks of variables (that appear in the linear polynomials in set C_1), we infer that the number of continuous chunks of variables (appearing in the linear polynomials in set C_1) is equal to δ which is less than $2T$. Let these continuous chunks of variables be $d_1, d_2, ...d_\delta$.

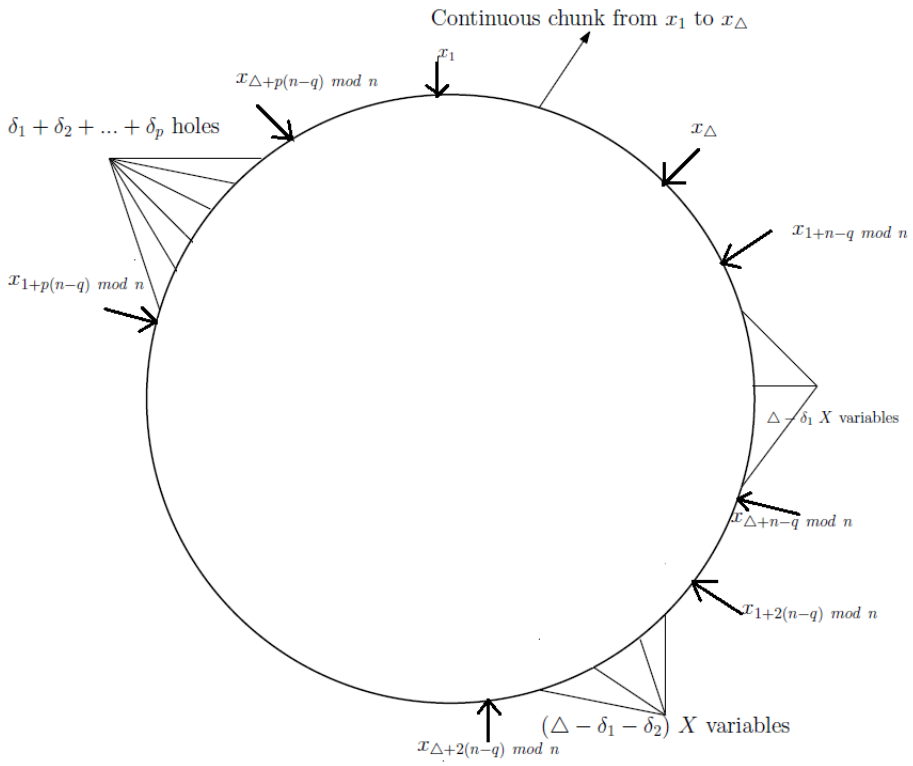

Figure A.1: Depiction of the X variables appearing in the linear polynomials in C_1

Since the number of x variables appearing in C_1 is at least $\frac{n}{8} - T$,

$$\sum_{i=1}^{\delta<2T} |d_i| \geq \frac{n}{8} - T$$

Let d_{max} represent the continuous chunk of variables (that appear in the linear polynomials in the set C_1) with maximum size, i.e $|d_{max}| = max_{i\in[\delta]}|d_i|$. Then

$$d_{max} \geq \frac{\frac{n}{8} - T}{2T} := \triangle$$

The continuous chunk d_{max} has at least $\triangle$ variables that are sequential based on the ordering $(x_1, x_2, ..., x_n)$. W.l.o.g we can assume the $\triangle$ variables at the end in d_{max} to be $x_1, x_2, ..., x_\triangle$. Since $x_1, x_2, ..., x_\triangle$ appear in the linear polynomials in set C_1 we know $y_1, y_2, ..., y_\triangle \in S$. In P_3 the variable y_i appears with $x_{(i+n-q) \ \bmod n}$. We wish that there is no overlap between the

continous chunk of variables: $x_1, ..., x_\delta$ and $x_{(1+n-q) \mod n}, ..., x_{(\triangle+n-q) \mod n}$. Observe that

$$\triangle = \frac{\frac{n}{8} - \epsilon\sqrt{n}}{2\epsilon\sqrt{n}} = \frac{\sqrt{n}}{16\epsilon} - \frac{1}{2} \leq \frac{1.5\sqrt{n}}{16\epsilon} = n - q$$

$$\Rightarrow (1 + (i+1)n - q) \mod n > \triangle + i(n-q) \quad \forall i \in \mathbb{N} \setminus \{0\}$$

i.e $x_{(1+n-q) \mod n}$ appears after $x_\triangle$ in the sequential ordering $(x_1, ..., x_n)$. In general $x_{(1+(i+1)(n-q)) \mod n}$ appears after $x_{\triangle+i(n-q)}$ in the sequential ordering $(x_1, ..., x_n)$ as shown in figure 1.

In P_3 $y_1, y_2, ..., y_\triangle$ appear with $x_{(1+n-q) \mod n}, x_{(2+n-q) \mod n}, ..., x_{(\triangle+n-q) \mod n}$ respectively. Some of these x variables among $x_{(1+n-q) \mod n}, x_{(2+n-q) \mod n}, ..., x_{(\triangle+n-q) \mod n}$ might appear in the linear polynomials in C_1. We want to determine the number of X variables among $x_{(1+n-q) \mod n}, x_{(2+n-q) \mod n}, ..., x_{(\triangle+n-q) \mod n}$ that appear in the linear polynomials in C_1. Let the number of X variables between $x_{(1+n-q) \mod n}$ to $x_{(\triangle+n-q) \mod n}$ appearing in the linear polynomials in C_1 be $(\triangle - \delta_1)$. Note that if x_i appears in C_1 then $y_i \in S$

Now consider the Y variables $y_{(1+n-q) \mod n}, ..., y_{(\triangle+n-q) \mod n}$. Among these atleast $\triangle - \delta$ variables are in S. The X variables that appear with these Y variables $y_{(1+n-q) \mod n}, ..., y_{(\triangle+n-q) \mod n}$ in P_3 are $x_{(1+2(n-q)) \mod n}, ..., x_{(\triangle+2(n-q)) \mod n}$. We only consider the $(\triangle - \delta_1)$ X variables that appear with those Y variables in P_3 which we know are definitely in S. Let the number of variables among these ($\triangle$ - δ_1) X variables between $x_{(1+2(n-q)) \mod n}$ to $x_{(\triangle+2(n-q)) \mod n}$ appearing in the linear polynomials in C_1 be $(\triangle - \delta_1 - \delta_2)$. We can visualize this process using figure 1 which depicts the X variables on a circle and shows the variables that appear in the linear polynomials in C_1 between each interval we take into consideration.

We repeat the above process p times where $p = 8\epsilon\sqrt{n}$ and finally have $(\triangle - (\delta_1 + \delta_2 + ... + \delta_p))$ many X variables between $x_{(1+p(n-q)) \mod n}$ to $x_{(\triangle+p(n-q)) \mod n}$ that appear in the linear polynomials in C_1.

Note that

$$\triangle + p(n-q) = \frac{\sqrt{n}}{16\epsilon} - \frac{1}{2} + 8\epsilon\sqrt{n} \times \frac{1.5\sqrt{n}}{16\epsilon} < n$$

This implies

$$(\triangle + p(n-q)) \mod n < n$$

Claim A.1.2 *The number of X variables such that for all $i \in [\triangle]$ and $j \in [p]$ when $y_i \in S$ then $x_{(i+j(n-q)) \mod n}$ does not appear in any of the linear forms in C_1 is less than $2T$.*

Proof: For all $i \in [\triangle]$ and $j \in [p]$ there are 2 possibilities $x_{(i+j(n-q)) \mod n} \in S$ or $x_{(i+j(n-q)) \mod n} \notin S$. $x_{(i+j(n-q)) \mod n} \in S \Rightarrow x_{(i+j(n-q)) \mod n}$ appears in one of the linear forms in B_1 since $x_{(i+j(n-q)) \mod n}$ does not appear in any of the linear forms in C_1. $x_{(i+j(n-q)) \mod n} \notin S \Rightarrow x_{(i+j(n-q)) \mod n}$ appears in one of the linear forms in B_3 since $y_i \in S$. Since the sizes of B_1 and B_3 are less than T, the claim is proved. □

Using claim A.1.2, we claim $\sum_{i=1}^{p} \delta_i \leq 2T$. Hence at least $(\triangle - 2T)$ variables among $x_{(1+i(n-q)) \mod n}$ to $x_{(\triangle+(i+1)(n-q)) \mod n}$ appear in the linear forms in C_1, for every $i \in [0, ..., p]$. This implies at least $(\triangle - 2T)(p+1)$ X variables appear in the linear forms in C_1 and thus belong to S. But

$$\begin{aligned}(\triangle - 2T)(p+1) &= (\frac{\sqrt{n}}{16\epsilon} - \frac{1}{2} - 2\epsilon\sqrt{n})(8\epsilon\sqrt{n}+1) \\ &\approx n(\frac{8\epsilon}{16\epsilon} - 16\epsilon^2) \\ &> \frac{n}{4} = |S| \quad (\text{for } \epsilon < \frac{1}{8})\end{aligned}$$

This gives us a contradiction. So our initial assumption is wrong.
Case 2: Assume the $\frac{n}{8}$ variables are completely Y variables.
We can handle case 2 similarly by again using the relationship between the X and Y variables in P_1 and P_2 to ascertain the number holes between the continuous chunks of Y variables that appear in the linear forms in C_1 is less than $2T$ (similar to claim A.1.1). Then we use the relationship between the X and Y variables in P_1 and P_4 to arrive at a contradiction as we did in case 1. □

lemma A.1.2 shows that the size of at least one of the sets among B_1, B_2, B_3 and B_4 is greater than equal to $\epsilon\sqrt{n}$. We will use the next lemma to complete the proof of theorem A.2.1.

Lemma A.1.3 *If the size of any one of the sets among B_1, B_2, B_3 and B_4 is greater than or equal to $\epsilon\sqrt{n}$ then* $\text{Evaldim}_S(F(X,Y))$ *is at least* $2^{\frac{\epsilon\sqrt{n}}{2}}$.

Proof: Assume $|B_1| \geq T$ where $T = \epsilon\sqrt{n}$ (the other cases will be similar). Since $|B_1| \geq T$ we have at least T variables in S appearing in the linear forms in B_1. At least $\frac{T}{2}$ of these variables are entirely X variables or Y variables. W.l.o.g assume $\frac{T}{2}$ of these variables are entirely X variables.

Let these $\frac{T}{2}$ X variables be $X_j = \{x_{j_1}, x_{j_2}, ..., x_{j_{\frac{T}{2}}}\}$. $F_{X_j}(X,Y)$ represents the polynomial where the other $\frac{n}{4} - \frac{T}{2}$ variables in S are substituted as 1. It is easy to see that $\text{Evaldim}_{X_j}[F_{X_j}(X,Y)] \leq \text{Evaldim}_S[F(X,Y)]$.

Claim A.1.3 $\text{Evaldim}_{X_j}[F_{X_j}(X,Y)] \geq 2^{\frac{T}{2}}$

Proof: For all $i \in [4]$ P_i looks like

$$(x_{j_1} + ...) \times (x_{j_2} + ...) \times .... \times (x_{j_{\frac{T}{2}}} + ...) f_i(X,Y)$$

Along with each of the $x_{j_1}, x_{j_2}, ..., x_{j_{\frac{T}{2}}}$ variables one of the Y variables may appear. Also notice that $x_{j_1}, x_{j_2}, ..., x_{j_{\frac{T}{2}}}$ variables do not appear in $f_i(X,Y)$.

$$\Rightarrow \forall i \in [4] \; P_i = (\sum_{S_t \subseteq [\frac{T}{2}]} x_{S_t} \times h_{it}) f_i(X,Y)$$

Here $x_{S_t} = \prod_{k \in S_t} x_{j_k}$ and h_{it} is a polynomial in the Y variables that are summed with x_{j_k} where $k \in [\frac{T}{2}] \setminus S_t$

After we pick $\frac{n}{4}$ variables we are left with with $\frac{7n}{4}$ variables that are not in S. From these $\frac{7n}{4}$ variables atleast $\frac{3n}{4}$ are X variables. These $\frac{3n}{4}$ X variables appear in all the 4 polynomials $f_1(X,Y), f_2(X,Y), f_3(X,Y)$ and $f_4(X,Y)$. Pick 3 variables from these $\frac{3n}{4}$ X variables. Let them be x_{i_1}, x_{i_2} and x_{i_3}. Substitute

$$x_{i_1} = -y_{(i_1+1) \mod (n)}, x_{i_2} = -y_{(i_2+q) \mod (n)}$$

$$x_{i_3} = -y_{(i_3+n-q) \mod (n)}$$

After these substitutions we have $f_2(X,Y) = f_3(X,Y) = f_4(X,Y) = 0$ and $f_1(X,Y) \neq 0$. Let $F'_{X_j}(X,Y)$ be the polynomial after we have made the above substitutions to $F_{X_j}(X,Y)$.

$$F'_{X_j}(X,Y) = (\sum_{S_t \subseteq [\frac{T}{2}]} x_{S_t} \times h_{1t}) f_1(X,Y)$$

Let $h^{\sim}_{1t} = h_{1t} f_1(X,Y)$. Hence

$$F'_{X_j}(X,Y) = (\sum_{S_t \subseteq [\frac{T}{2}]} x_{S_t} \times h^{\sim}_{1t})$$

Below we show that of evaluation dimension of $F'_{X_j}(X,Y)$ with respect to X_j is at most evaluation dimension of $F_{X_j}(X,Y)$ with respect to X_j.

Claim A.1.4 *$Evaldim_{X_j}[F_{X_j}(X,Y)] \geq Evaldim_{X_j}[F'_{X_j}(X,Y)]$*

The proof of claim A.1.4 is similar to claim 4.2.1. Since $\text{Evaldim}_{X_j}[F_{X_j}(X,Y)] \geq \text{Evaldim}_{X_j}[F'_{X_j}(X,Y)]$ it is sufficient to show $\text{Evaldim}_{X_j}[F'_{X_j}(X,Y)] \geq 2^{\frac{T}{2}}$.

SubClaim A.1.2 *The polynomials $h^{\sim}_{1t}$ corresponding to $S_t \subseteq [\frac{T}{2}]$ as defined above are all linearly independent.*

Proof: Let us say for contradiction that there exists a linear dependence between the polynomials $h^{\sim}_{1t}$ corresponding to $S_t \subseteq [\frac{T}{2}]$

$$\Rightarrow \sum_{S_t \subseteq [\frac{T}{2}]} \alpha_t h^{\sim}_{1t} = 0$$

$$\Rightarrow \sum_{S_t \subseteq [\frac{T}{2}]} \alpha_t h_{1t} f_1(X,Y) = 0$$

$$\Rightarrow \sum_{S_t \subseteq [\frac{T}{2}]} \alpha_t h_{1t} = 0$$

$x_{j_1}, x_{j_2}, ..., x_{j_{\frac{T}{2}}}$ appear in the partially set linear forms in the product gate P_1. Hence the Y variables $y_{j_1}, y_{j_2}, ..., y_{j_{\frac{T}{2}}}$ does not belong to the set S. This implies in $F'_{X_j}(X,Y)$ the linear forms containing the variables $x_{j_1}, x_{j_2}, ..., x_{j_{\frac{T}{2}}}$ are $(x_{j_1} + y_{j_1} + 1), (x_{j_2} + y_{j_2} + 1), ..., (x_{j_{\frac{T}{2}}} + y_{j_{\frac{T}{2}}} + 1)$ respectively. Observe that each h_{1t} is a product of some of these Y variables, i.e
$h_{1t} = \prod_{i \in [\frac{T}{2}] \setminus S_t} (1 + y_{j_i})$
Now substitute $y_{j_i} = y_{j_i} - 1$. Let h'_{1t} represent h_{1t} after substitution.
$\Rightarrow \ h'_{1t} = \prod_{i \in [\frac{T}{2}] \setminus S_t} (y_{j_i})$
$\Rightarrow h'_{1t}s$ are distinct monomials.
$\Rightarrow$ for all $S_{j \subseteq [\frac{T}{2}]}$ $\alpha_j = 0$, a contradiction
$\Rightarrow$ The polynomials $h^{\sim}_{1t}$ corresponding to $S_j \subseteq [\frac{T}{2}]$ are all linearly independent. □

Since the polynomials $h^{\sim}_{1t}$ corresponding to $S_t \subseteq [\frac{T}{2}]$ are all linearly dependent we can use claim 2.4.1 to finish the proof of this claim. Hence $\text{Evaldim}_{X_j}[F_{X_j}(X,Y)] \geq 2^{\frac{T}{2}}$. □

As stated earlier $\text{Evaldim}_S[F(X,Y)] \geq \text{Evaldim}_{X_j}[F_{X_j}(X,Y)]$. Hence $\text{Evaldim}_S[F(X,Y)] \geq$

$2^{\frac{T}{2}} = 2^{\frac{\epsilon\sqrt{n}}{2}}$. □

Combining lemma A.1.2 and lemma A.1.3 we can conclude for any set $S \subseteq X \cup Y$ with size equal to $\frac{n}{4}$ the $\text{Evaldim}_S(F(X,Y))$ is at least $2^{\frac{\epsilon\sqrt{n}}{2}}$. This concludes the proof for theorem A.2.1. To complete the proof of theorem A.1.1 we look at lemma A.1.1 which tells us that there exists a choice of $\frac{n}{4}$ variables such that the evaluation dimension of $F(X,Y)$ is less than the width of the ROABP that computes $F(X,Y)$, but theorem A.2.1 on the other hand states that evaluation dimension of $F(X,Y)$ with respect to any subset of $X \cup Y$ of size $\frac{n}{4}$ is at least $2^{\frac{\epsilon\sqrt{n}}{2}}$. This implies that any ROABP that computes the polynomial $F(X,Y)$ has width equal to $2^{\Omega(\sqrt{n})}$.

A.2 Exponential lower bound against ROABPs for multilinear depth three circuit with $O(n)$ top fan-in

In this section, we show a $2^{\Omega(n)}$ lower bound on the width of any ROABP that computes the polynomial

$$F(X,Y) = \sum_{i=0}^{\frac{n}{2}-1} \prod_{j=1}^{n} (1 + x_j + y_{(j+i) \bmod n})$$

Theorem A.2.1 (Main Theorem) *Any ROABP that computes the polynomial*

$$F(X,Y) = \sum_{i=0}^{\frac{n}{2}-1} \prod_{j=1}^{n} (1 + x_j + y_{(j+i) \bmod n})$$

has width $2^{\Omega(n)}$.

The proof strategy remains essentially same that of theorem A.1.1, theorem 4.2.1 and theorem 4.2.3. Suppose R is a width k ROABP that computes $F(X,Y)$. R has an associated variable ordering $\{t_1, ..., t_{2n}\}$ of the X and Y variables. Let $T = \{t_1, ..., t_{\frac{n}{2}}\}$. From lemma 4.2.1 we know $\text{Evaldim}_T[F(X,Y)] < k$. We will now show that for every choice of $\frac{n}{2}$ variables, the evaluation dimension of $F(X,Y)$ with respect to these $\frac{n}{2}$ varibales is $2^{\Omega(n)}$. This would imply a $2^{\Omega(n)}$ lower bound on the width of any ROABP that computes the polynomial $F(X,Y)$.

Theorem A.2.2 *For any set $S \subseteq X \cup Y$ with size equal to $\frac{n}{2}$ the* $\text{Evaldim}_S[F(X,Y)]$ *is at least* $2^{\frac{n}{8}}$.

Choose any set $S \subseteq X \cup Y$ such that $|S| = \frac{n}{2}$. Let S consists of a X variables and b Y variables.

$$\Rightarrow \ a + b = \frac{n}{2} \tag{A.1}$$

We denote by P_i the ith product gate $\prod_{j=1}^{n}(1 + x_j + y_{j+i})$ of $F(X, Y)$. After choosing S we have three types of linear forms in each product gate: A_i =the set of untouched linear forms in P_i, B_i=the set of partially touched linear forms in P_i, C_i=the set of completely touched linear forms in P_i, where $i \in \{0\} \cup [\frac{n}{2} - 1]$.

Lemma A.2.1 *The size of the set containing the partially touched linear forms is greater than or equal to $T = \frac{n}{4}$ in at least one of the product gates.*

Proof: Let us assume for contradiction that for all $i \in \{0\} \cup [\frac{n}{2} - 1]$ $|B_i| < T$. This implies in each product gate at least $(\frac{n}{2} - T)$ variables appear in linear forms which are completely touched. Since a linear form contains exactly two variables, we have for all product gates $i \in \{0\} \cup [\frac{n}{2} - 1]$, $|A_i| > \frac{n}{4} - \frac{T}{2}$. Observe that for all $i, j \in \{0\} \cup [\frac{n}{2} - 1]$, $A_i \cap A_j = \phi$. Recall S consists of 'a' X variables and 'b Y variables. Hence the maximum possible number of completely touched linear forms is at most ab. Thus we have

$$ab > \sum_{i=1}^{n}(|A_i|) > \frac{n}{2}(\frac{n}{4} - \frac{T}{2})$$

Applying AM>GM inequality we have,

$$\frac{(a+b)^2}{4} > \frac{n}{2}(\frac{n}{4} - \frac{T}{2})$$

$$\frac{nT}{4} > \frac{n^2}{16}$$

$$T > \frac{n}{4}$$

□

lemma A.2.1 shows that the size of at least one of the sets among $\{B_0, B_1, ...B_{\frac{n}{2}-1}\}$ is greater than equal to $\frac{n}{4}$. We will use the next lemma to complete the proof of theorem A.2.2.

Lemma A.2.2 *If the size of any one of the sets among $\{B_0, B_1, ...B_{\frac{n}{2}-1}\}$ is greater than or equal to $\frac{n}{4}$ then* $\text{Evaldim}_S(F(X, Y))$ *is at least $2^{\frac{n}{8}}$.*

Proof: Assume $|B_i| \geq T$ where $i \in \{0, ..., (\frac{n}{2} - 1)\}$ and $T = \frac{n}{4}$. Since $|B_i| \geq T$ we have at least T variables in S appearing in the linear forms in B_i. At least $\frac{T}{2}$ of these variables are

entirely X variables or Y variables. W.l.o.g assume $\frac{T}{2}$ of these variables are entirely X variables.

Let these $\frac{T}{2}$ X variables be $X_j = \{x_{j_1}, x_{j_2}, ..., x_{j_{\frac{T}{2}}}\}$. $F_{X_j}(X,Y)$ represents the polynomial where the other $\frac{n}{4} - \frac{T}{2}$ variables in S are substituted as 1. It is easy to see that $\text{Evaldim}_{X_j}[F_{X_j}(X,Y)] \leq \text{Evaldim}_S[F(X,Y)]$.

Claim A.2.1 $\text{Evaldim}_{X_j}[F_{X_j}(X,Y)] \geq 2^{\frac{T}{2}}$

Proof: For all $i \in \{0\} \cup [\frac{n}{2} - 1]$ P_i looks like

$$(x_{j_1} + ...) \times (x_{j_2} + ...) \times .... \times (x_{j_{\frac{T}{2}}} + ...) f_i(X,Y)$$

Along with each of the $x_{j_1}, x_{j_2}, ..., x_{j_{\frac{T}{2}}}$ variables one of the Y variables may appear. Also notice that $x_{j_1}, x_{j_2}, ..., x_{j_{\frac{T}{2}}}$ variables do not appear in $f_i(X,Y)$.

$$\Rightarrow \forall i \in \{0\} \cup [\frac{n}{2} - 1]\ P_i = (\sum_{S_t \subseteq [\frac{T}{2}]} x_{S_t} \times h_{it}) f_i(X,Y)$$

Here $x_{S_t} = \prod_{k \in S_t} x_{j_k}$ and h_{it} is a polynomial in the Y variables that are summed with x_{j_k} where $k \in [\frac{T}{2}] \setminus S_t$

After we pick $\frac{n}{2}$ variables we are left with with $\frac{3n}{2}$ variables that are not in S. From these $\frac{3n}{2}$ variables at least $\frac{n}{2}$ are X variables. These $\frac{n}{2}$ X variables appear in all the $\frac{n}{2}$ polynomials $f_1(X,Y), f_2(X,Y), ..., f_{\frac{n}{2}}(X,Y)$. Pick $\frac{n}{2} - 1$ variables from these $\frac{n}{2}$ X variables, that are not in S. Let them be $\{x_{r_0}, x_{r_1}, ..., x_{r_{i-1}}, x_{r_{i+1}}, ...,$
$x_{r_{\frac{n}{2}-1}}\}$ such that the variable x_{r_w} corresponds to the wth product gate . For all $w \in \{0, ..., (\frac{n}{2} - 1)\} \setminus \{i\}$, substitute

$$x_{r_w} = -y_{((r_w + w) \bmod (n))} - 1$$

After these substitutions we have For all $w \in \{0, ..., (\frac{n}{2} - 1)\} \setminus \{i\}$, $f_w(X,Y) = 0$ and $f_i(X,Y) \neq 0$. Let $F'_{X_j}(X,Y)$ be the polynomial after we have made the above substitutions to $F_{X_j}(X,Y)$.

$$F'_{X_j}(X,Y) = (\sum_{S_t \subseteq [\frac{T}{2}]} x_{S_t} \times h_{it}) f_i(X,Y)$$

Let $h^{\sim}_{it} = h_{it} f_i(X,Y)$. Hence

$$F'_{X_j}(X,Y) = (\sum_{S_t \subseteq [\frac{T}{2}]} x_{S_t} \times h^{\sim}_{it})$$

Below we show that of evaluation dimension of $F'_{X_j}(X,Y)$ with respect to X_j is at most evaluation dimension of $F_{X_j}(X,Y)$ with respect to X_j.

Claim A.2.2 *$Evaldim_{X_j}[F_{X_j}(X,Y)] \geq Evaldim_{X_j}[F'_{X_j}(X,Y)]$*

The proof of claim A.2.2 is similar to claim 4.2.1. Since $\text{Evaldim}_{X_j}[F_{X_j}(X,Y)] \geq \text{Evaldim}_{X_j}[F'_{X_j}(X,Y)]$ it is sufficient to show $\text{Evaldim}_{X_j}[F'_{X_j}(X,Y)] \geq 2^{\frac{T}{2}}$.

SubClaim A.2.1 *The polynomials $h^{\sim}_{it}$ corresponding to $S_t \subseteq [\frac{T}{2}]$ as defined above are all linearly independent.*

For the proof of subClaim A.2.1 refer to the proof of subClaim A.1.2. Since the polynomials $h^{\sim}_{1t}$ corresponding to $S_t \subseteq [\frac{T}{2}]$ are all linearly dependent we can use claim 2.4.1 to finish the proof of this claim. Hence $\text{Evaldim}_{X_j}[F_{X_j}(X,Y)] \geq 2^{\frac{T}{2}}$. □

As stated earlier $\text{Evaldim}_S[F(X,Y)] \geq \text{Evaldim}_{X_j}[F_{X_j}(X,Y)]$. Hence $\text{Evaldim}_S[F(X,Y)] \geq 2^{\frac{T}{2}} = 2^{\frac{n}{8}}$. □

Combining lemma A.2.1 and lemma A.2.2 we can conclude for any set $S \subseteq X \cup Y$ with size equal to $\frac{n}{4}$ the $\text{Evaldim}_S(F(X,Y))$ is at least $2^{\frac{\epsilon\sqrt{n}}{2}}$. This concludes the proof for theorem A.2.2. To complete the proof of theorem A.2.1 we know there exists a choice of $\frac{n}{2}$ variables such that the evaluation dimension of $F(X,Y)$ is less than the width of the ROABP that computes $F(X,Y)$, but theorem A.2.2 on the other hand states that evaluation dimension of $F(X,Y)$ with respect to any subset of $X \cup Y$ of size $\frac{n}{2}$ is at least $2^{\frac{n}{8}}$. This implies that any ROABP that computes the polynomial $F(X,Y)$ has width equal to $2^{\Omega(n)}$.

Bibliography

[Aga05] M. Agarwal. Proving lower bounds via pseudo-random generators. In *Proc. 25th Annual Conference on Foundations of Software Technology and Theoretical Computer Science*. lncs, 2005. 1, 12

[AGKS14] M. Agarwal, R. Gurjar, A. Korwar, and N. Saxena. Hitting-sets for roabp and sum of set-multilinear circuits. *Electronic Colloquium on Computational Complexity*, 2014. i, ii, 4, 5

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. 2

[Alo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 1986. 16

[AM85] N. Alon and V. D. Milman. isoperimetric inequalities for graphs and superconcentrators. *J. Combin. Theory Ser.*, 1985. 16

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM*, 45(1):70–122, 1998. 2

[ASS12] M. Agarwal, C. Saha, and N. Saxena. Quasi-polynomial hitting set for set-depth-delta formulas. In *Proc. 45th Annual ACM Symposium on the Theory of Computing*. ACM, 2012. i, ii, 3, 5, 21, 22, 25, 26

[AV08] M. Agarwal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *Proc. 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 67–75, 2008. 3

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1991. 2

[BOC92] M. Ben-Or and R. Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing*, 21(1):54–58, 1992. 12

[Bus82] P. Buser. A note on the isopoermitric constant. *Annual Science Ecole Norm.*, 4(15):213–230, 1982. 16

[Che70] J. Cheeger. Lower bound for the smallest eigenvalue of the laplacian. *Problems in analysis*, pages 195–199, 1970. 16

[CRS95] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness-optimal unique element isolation with applications to perfect matching and related problems. *SIAM Journal on Computing*, 24(5):1036–1050, 1995. 1

[Dod84] J. Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *SIAM Journal on Computing*, 284(2):787–794, 1984. 16

[FS13] M. Forbes and A. Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *Proc. 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 243–252, 2013. i, 13

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *J. Comput. Syst. Sci.*, 22(3):407–420, 1981. 15

[GKKS13] A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi. Arithmetic circuits: A chasm at depth four. In *Proc. 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 578–587, 2013. 3

[GKST15] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read once abps. *Computational Complexity*, 2015. 7, 25, 55

[Hal35] Philip Hall. On representatives of subsets. *J. London Math. Soc.*, 10(1):26–30, 1935. 19

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006. 15, 16, 17

[HS80] J. Heintz and C. P. Schnorr. Testing polynomials which are easy to compute. In *Proc. 12th Annual ACM Symposium on the Theory of Computing*. ACM, 1980. 1, 12

[KB93] A. N. Kolmogorov and Ya. M. Barzdi. On the realization of networks in three-dimensional space. *Selected works of Kolmogorov*, 3, 1993. 15

[KI04] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 2004. 1, 11

[KLSS14] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. In *Proc. 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 61–70, 2014. 55

[KMS98] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998. 5, 6, 24

[Koi12] P. Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012. 3

[KS01] Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proc. 33rd Annual ACM Symposium on the Theory of Computing*, pages 216–223, 2001. 21, 26, 32

[KS14] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *Proc. 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 364–373, 2014. 55

[KUW86] R. Karp, E. Upfal, and A. Wigderson. Constructing a perfect matching is in random nc. *Combinatorica*, 6(2):35–48, 1986. 1

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992. 2

[Lov79] L. Lovasz. On determinants matchings and random algorithms. *Fundamentals of Computing Theory*, 1979. 1

[LPS88] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988. 15

[Mar73] G. A. Margulis. Explicit constructions of expanders. *Problemy Peredaci Informacii*, 9(7):71–80, 1973. 15

[Mar88] G. A. Margulis. Explicit group-theoretic constructions and combinatorial schemes and their applications in the construction of expanders and concentrators. *Problems of Information Transmission*, 24(1):39–46, 1988. 15

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. 1

[Nis91] N. Nisan. Lower bounds for non commutative computation. In *Proc. 23rd Annual ACM Symposium on the Theory of Computing*. ACM, 1991. 12

[Nis92] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992. 13

[NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. 3

[ORW02] S. Vadhan O. Reingold and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 2(155):157–187, 2002. 15

[OSV15] Rafael Mendes De Oliveira, Amir Shpilka, and Ben Lee Volk. Subexponential size hitting sets for bounded depth multilinear formulas. *Computational Complexity*, 2015. i, ii, 4, 6

[Pin73] M. Pinsker. On the complexity of a concentrator. *International Telegrafic Conference*, 1973. 15

[Raz10] Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. In *Proc. 42nd Annual ACM Symposium on the Theory of Computing*, pages 659–666, 2010. 10

[RS05] R. Raz and A. Shpilka. Deterministic polynomial identity testing in non-commutative models. In *Computational Complexity*, pages 1–19, 2005. 4

[RY09] R. Raz and A. Yehudayof. Lower bounds and separations for constant depth multilinear circuits. In *Computational Complexity*, pages 128–139, 2009. 13, 14

[Sha92] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992. 2

[SSS09] C. Saha, R. Saptharishi, and N. Saxena. The power of depth 2 circuits over algebras. In *Proc. 29th Annual Conference on Foundations of Software Technology and Theoretical Computer Science*. lncs, 2009. 12

[SV85] Sven Skyum and Leslie G. Valiant. A complexity theory based on boolean algebra. *J. ACM*, 32(2):484–502, 1985. 1

[SY10] A. Shpilka and A. Yehudayof. Arithmetic circuits:a survey of recent results and open questions. Technical report, 2010. Available at "http://www.cs.technion.ac.il/ shpilka/publications/SY10.pdf". 3

[Tav13] S. Tavenas. Improved bounds for reduction to depth 4 and depth 3. *In Mathematical Foundations of Computer Science*, page 813824, 2013. 3

[Val79a] L. G. Valiant. Completeness classes in algebra. In *Proc. 11th Annual ACM Symposium on the Theory of Computing*, pages 249–261, 1979. 1, 11

[Val79b] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979. 1, 11