Equivalence test for the trace iterated matrix multiplication polynomial

A THESIS

SUBMITTED FOR THE DEGREE OF

Master of Technology (Research)

IN THE

Faculty Of Engineering

BY

Janaky Murthy

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

April, 2020

Declaration of Originality

I, **Janaky Murthy**, with SR No. **04-04-00-10-22-17-1-14923** hereby declare that the material presented in the thesis titled

Equivalence test for the trace iterated matrix multiplication polynomial

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2017-19**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Advisor Signature

DEDICATED TO

amma, appa, sankar and meenu

for showering me with so much love...

Acknowledgements

First of all, I would like to thank the universe for whatever I have experienced till now. I am very grateful to my advisor Chandan Saha for his support, guidance and patience. This thesis would not have been possible without him. I really admire his clear, structured way of thinking. I am also inspired by his excellent teaching and presentation skills. I also want to thank Vineet Nair for being a helpful and patient mentor. He guided me in picking up the required background subjects and also suggested valuable improvements to the thesis. I would also like to thank him for his contribution to Chapter 6 of the thesis. I want to thank all my lab mates - Vineet, Nikhil, Arpita, Anuj and Bhargav for their wonderful company. I thank all the faculty members at IISc who taught me various courses. The courses helped in strengthening my basics in the relevant areas.

I want to thank my Amma and Appa for everything I have. They have always encouraged me, right from my childhood to follow my heart and be responsible. I also want to thank my brother Sankar and my sister Meenu for being a bundle of joy in my life (sarcasm intended). Without naming anyone, I am also thankful to all my friends at IISc and elsewhere who have always stood by me. I apologize to all those friends who expected a special mention of their name :P I have you all in my heart!

My M.Tech Research Program at IISc was a great learning experience for me - both academically and about my own self. I made many mistakes on the way and I have tried to learn from them. Our campus is extremely beautiful and is the first place where I had to be on my own away from my family. Hence I have a special love for this place. I am sure that the memories and the experience will stay with me forever :)

Abstract

An m-variate polynomial f is *affine equivalent* to an n-variate polynomial g if $m \geq n$ and there is a rank n matrix $A \in \mathbb{F}^{n \times m}$ and $\mathbf{b} \in \mathbb{F}^n$ such that $f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{b})$. Given blackbox access to f and g (i.e membership query access) the affine equivalence test problem is to determine whether f is affine equivalent to g, and if yes then output a rank n matrix $A \in \mathbb{F}^{n \times m}$ and $\mathbf{b} \in \mathbb{F}^n$ such that $f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{b})$. This problem is at least as hard as graph isomorphism and algebra isomorphism even when the coefficients of f and g are given explicitly (Agarwal and Saxena, STACS 2006), and has been studied in literature by fixing g to be some interesting family of polynomials. In this work, we fix g to be the trace of the product of d, $w \times w$ symbolic matrices $X_1, \ldots, X_d$. We call this polynomial Tr-IMM$_{w,d}$. Kayal, Nair, Saha and Tavenas (CCC 2017) gave an efficient (i.e $(mwd)^{O(1)}$ time) randomized algorithm for the affine equivalence test of the iterated matrix multiplication polynomial IMM$_{w,d}$, which is the $(1,1)$-th entry of the product of d $w \times w$ symbolic matrices. Although the definitions of Tr-IMM$_{w,d}$ and IMM$_{w,d}$ are closely related and their circuit complexities are very similar, it is not clear whether an efficient affine equivalence test algorithm for IMM$_{w,d}$ implies the same for Tr-IMM$_{w,d}$. In this thesis, we take a step towards showing that equivalence test for Tr-IMM$_{w,d}$ and IMM$_{w,d}$ have different complexity. We show that equivalence test for Tr-IMM$_{w,d}$ reduces in randomized polynomial time to equivalence test for the determinant (DET), under mild conditions on the underlying field. If the converse is also true then equivalence tests for Tr-IMM$_{w,d}$ and DET are randomized polynomial time equivalent. It would then follow from the work of Gupta, Garg, Kayal and Saha (ICALP 2019) that equivalence test for Tr-IMM$_{w,d}$ over $\mathbb{Q}$ is at least as hard as Integer Factoring. This would then be in sharp contrast with the complexity of equivalence test for IMM$_{w,d}$ over $\mathbb{Q}$ which can be solved efficiently in randomized polynomial time (by Kayal, Nair, Saha and Tavenas (CCC 2017)).

Recent Update: Soon after the thesis is written, we (together with Vineet Nair) have succeeded in showing the converse direction. So, the above conclusion is indeed true!

Contents

List of Figures

Chapter 1

Introduction

Finding efficient algorithms for problems with an algebraic flavour arises often in a wide variety of theoretical and practical problems. Examples of such algorithms includes the randomized algorithm for perfect matching [25, 30, 9], Discrete Fourier Transforms [7], matrix multiplication [29] etc. Developing such algorithms is the central objective of *computational algebra* which is a subarea of ***Algebraic Complexity Theory (ACT)***. These algorithms mostly involve arithmetic operations like $'+', '\times'$ and $'\div'$. This motivates us to look at ***Arithmetic Circuit Complexity*** which is the other subarea of ACT. Arithmetic Circuit Complexity seeks to understand the complexity of computing polynomials using various circuit models like arithmetic circuits, algebraic branching programs or arithmetic formulas. There are many natural and fundamental structural as well as algorithmic questions related to polynomials being computed by these circuit models which we elaborate upon below.

A natural model to compute polynomials are ***Arithmetic Circuits*** which are algebraic analogues of boolean circuits. An arithmetic circuit (see Definition 2.15) takes formal variables as input and computes a polynomial in these variables using addition and multiplication operations. The *size* of the circuit is the total number of operations required by the circuit to compute the polynomial. Figure 1 depicts an arithmetic circuit computing the polynomial $x_1^2 + 5x_2$. Analogous to analyzing the size and depth complexity of boolean circuit families computing boolean function families, in arithmetic circuit complexity we analyze the size and depth complexity of arithmetic circuit fam-

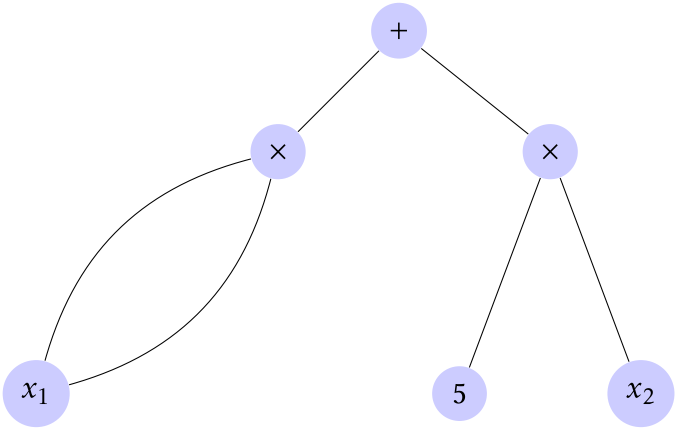


Figure 1.1: Arithmetic Circuit computing $x_1^2 + 5x_2$

ilies computing polynomial families[1]. In an effort to categorize the polynomial families based on the size of the circuit needed to compute them, Valiant [33] defined the classes VP and VNP similar to the non-uniform P and NP respectively. VP is the class of (low degree) polynomial families (see Definition 2.16) that can be computed by polynomial sized arithmetic circuits. The symbolic determinant polynomial family[2] (DET_n) and the trace of iterated matrix multiplication polynomial[3] ($\text{Tr-IMM}_{w,d}$) is in VP. Loosely speaking, VNP is the class of polynomial families where coefficient of any monomial of a given polynomial f in the family is efficiently computable (see Definition 2.17 for a rigorous definition of VNP). The symbolic permanent polynomial family denoted as PERM_m is in VNP. Although, it is clear that VP is contained in VNP, whether VP $\stackrel{?}{=}$ VNP is the central open question of ACT. Valiant conjectured that this is not the case (***Valiant's Conjecture***) [34].

An ***Arithmetic Formula*** is an arithmetic circuit whose underlying directed acyclic graph is a tree. The complexity class VF is equal to the set of (low degree) polynomial families that can be computed by polynomial sized arithmetic formulas. Clearly the class VF is contained in VP. Another important complexity class is VBP which consists of polynomial families that can be computed by Arithmetic Branching Programs (ABP) of polynomially bounded size. An ***Arithmetic Branching Program*** of width w and depth d (i.e. of size wd) computes a polynomial that can be written as the $(1, 1)$-th

[1] A polynomial family $\{f_i\}_{i \in \mathbb{N}}$ is a sequence of polynomials

[2] DET_n is the determinant of a $n \times n$ symbolic matrix.

[3] see Definition 1.3

entry of the product of d matrices $\prod_{i=1}^{d} X_i$ where each X_i is a $w \times w$ matrix. The entries of the X_i's are linear polynomials in the variables. The polynomial families $\mathrm{DET}_n, \text{Tr-}\mathrm{IMM}_{w,d}$ are in VBP. In fact these polynomial families are ***complete*** for the class VBP. In this regard we define the notion of *projections* which are the algebraic counterparts of *reductions* in the boolean world. A polynomial $f(\mathbf{x})$ is a ***projection*** of a polynomial $g(\mathbf{x})$ if f can be obtained by substituting each variable of g by a variable of f or a field constant. A family of polynomials $(f_n)_{n\geq 1}$ is a ***p-projection*** of $(g_n)_{n\geq 1}$ if each f_n is a projection of $g_{p(n)}$ and $p(n)$ is polynomially bounded. The PERM_m polynomial family is complete for VNP under p-projections. With this notion of *complete* polynomials, we can ask the following question: Is a complete polynomial of one complexity class a *p-projection* of a complete polynomial of another complexity class and vice-versa. This helps us to compare two complexity classes. To separate VP and VNP we need to show that permanent polynomial does not have *small* sized arithmetic circuits. In this regard, the following containment is well known: VF $\subseteq$ VBP $\subseteq$ VP $\subseteq$ VNP; separating any two of these classes requires establishing strong ***lower bounds*** [34]. As we do not know whether VBP = VP, we also do not know if DET_n is complete for VP under p-projections. However it can be shown that the DET_n polynomial is complete for VP under *quasi-polynomial projections* [33].

Given an arbitrary n-variate, degree d polynomial f, Baur and Strassen [4] showed an $\Omega(n\log(d))$ lower bound for general circuits and it is an open problem to improve this bound. Lower bound for restricted circuit classes has also been looked into. [16] showed an $\Omega(n^2)$ lower bound on formula size involving n variables. Agarwal and Vinay [3] show that a sufficiently strong exponential lower bound on depth 4 circuit imply an exponential lower bound on general circuits. Koiran [23] and Tavenas [31] improved this result and showed that if f can be computed by a general circuit of size s then there is a *depth-4* circuit of size $\exp(O(\sqrt{d}\log(s)))$ that computes f.

Polynomial Identity Testing (PIT) is another interesting problem which asks if all the coefficients of a given polynomial f is 0. PIT has applications in various interesting problems such as efficient parallel algorithms for perfect matching [25]. If the polynomial f is given explicitly as list of coefficients then the problem is trivial. However, when f is given in some compact representation, say arithmetic circuit or as a blackbox, then it is not clear if one can efficiently perform identity testing. Although an efficient randomized algorithm via Schwartz-Zippel Lemma (see Claim 2.1) is known, an efficient deterministic algorithm for both blackbox and whitebox PIT is still open. In an attempt to understand the hardness of derandomizing PIT, Kabanets and Impagliazzo [15] showed that derandomizing blackbox PIT implies certain lower bounds in arithmetic or boolean world (PERM_m has small sized circuits or NEXP $\not\subseteq$ P/Poly). They also showed that a super-polynomial lower bound for permanent implies an efficient blackbox PIT for polynomial sized circuits. Hence PIT is closely related to the lower bound problem. Agrawal and Vinay [3] showed that an efficient blackbox PIT for depth 4 circuits implies an efficient blackbox PIT for general circuits. However we do not know how to derandomize PIT even for depth 3 circuits.

The ***Circuit Reconstruction*** problem asks to efficiently learn a circuit from a complexity class $\mathcal{C}$ computing a polynomial f, given blackbox access to f. It is the algebraic analogue of learning boolean circuits using membership queries and is closely related to PIT. In fact it is not always straightforward to devise an efficient reconstruction algorithm for a circuit class $\mathcal{C}$ even when we are given an efficient blackbox PIT for $\mathcal{C}$ because of the following reason: A blackbox PIT algorithm for a circuit class $\mathcal{C}$ gives a set of points $\mathcal{H}$ known as the hitting set such that for any n variate, degree d polynomial $f \in \mathcal{C}$, there exists a point $h \in \mathcal{H}$ such that $f(h) \neq 0$. So assuming $\mathcal{C}$ is closed under subtractions, any two circuits in $\mathcal{C}$ will compute the same polynomial f if and only if they agree on all points in the hitting set. Thus the hitting set gives us a way of distinguishing between two circuits evaluating different polynomials. However it is a non-trivial problem to reconstruct a circuit computing f even when we are given its values on hitting set.

In this thesis, we look at another important problem called ***Polynomial Equivalence Testing***. Two n-variate degree d polynomials f, g are said to be equivalent if there is an invertible $n \times n$ matrix A such that $f(\mathbf{x}) = g(A\mathbf{x})$. Given blackbox access to f and g we wish to efficiently compute an A (if exists) such that $f(\mathbf{x}) = g(A\mathbf{x})$. Consider the simpler case when the polynomials f and g are given as list of coefficients of monomials. Then, a straight forward approach to the equivalence testing problem is to treat the entries of the matrix A as variables. Then $f(\mathbf{x}) = g(A\mathbf{x})$ gives us a system of polynomial equations in the variables of A. However solving a system of polynomial equations is known to be NP-hard over $\mathbb{C}$, finite fields and is not even known to be decidable over $\mathbb{Q}$. Can we hope equivalence testing to be easier than solving system of polynomial equations? It turns out that when f and g are given as list of coefficients, equivalence testing *can not* be NP-hard unless PH collapses [32, 28]. While this is a strong evidence that equivalence testing is not NP-hard, we also have evidences which suggests that it is not in P either. For instance, [2] shows that equivalence testing of degree 3 homogeneous polynomial is as hard as Graph Isomorphism and $\mathbb{F}$-algebra isomorphism. In particular, the graph isomorphism problem reduces to testing algebra isomorphism which further reduces to equivalence testing of cubic polynomials. A natural way to attack this problem is by looking at some special cases. For example, let us consider the case when both the polynomials f, g have some constant degree d. For $d = 2$, we have efficient equivalence testing algorithms over $\mathbb{C}$, finite fields and it is randomized polynomial time equivalent to integer factoring over $\mathbb{Q}$. However equivalence testing of f, g when $d = 3$ is not even known to be decidable over $\mathbb{Q}$. There is even a cryptographic encryption scheme that assumes "non-easiness" of equivalence testing of constant degree polynomials [27]. Another special case of this problem is to fix one of the polynomial to belong to some interersting polynomial family. Many results are known for some well known polynomial families (like the $\mathrm{PERM}_m, \mathrm{DET}_n$ etc...) in this direction which we have elaborated in Section 1.2. In this thesis, we fix g to be a polynomial coming from $\text{Tr-IMM}_{w,d}$ polynomial family (see Definiton 1.3) and f to be an n-variate, degree d polynomial given as a blackbox. In the upcoming sections, we will motivate and formally state this problem and present our results.

1.1 Motivation

In this section we explain why the problem of polynomial equivalence for the $\text{Tr-IMM}_{w,d}$ polynomial is interesting to us.

Why Equivalence Testing for a fixed family of polynomials? The polynomial equivalence problem is a very natural algebraic problem. [18] motivates a more general problem, namely the *affine projection* problem. An m-variate polynomial f is said to be an ***affine projection*** of an n-variate polynomial g if there exists an $A \in \mathbb{F}^{n \times m}$ and $\mathbf{b} \in \mathbb{F}^n$ such that $f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{b})$. They show that this problem is NP-hard in general. Infact, the central question of ACT - the PERM_m vs DET_n

problem (aka whether the permanent polynomial has small sized arithmetic circuits) is a special case of the affine projection problem. A natural relaxation of this problem is to impose conditions on the matrix A. When we require A to be full rank, the affine projection problem is known as the ***affine equivalence problem***. In [18], they show that the affine equivalence problem reduces to the ***equivalence problem*** and also give an efficient affine equivalence test for some special polynomial families like Perm_m, Det_n (over $\mathbb{C}$) etc. A motivation for equivalence testing for special polynomial families naturally arises from ***Geometric Complexity Theory***. GCT is a program introduced by Ketan Mulmuley and Milind Sohoni [26] to resolve the VP vs VNP problem (more ambitiously P vs NP) using tools from algebraic geometry and representation theory. We refer the reader to Chapter 4 of Joschua Grochow's PhD thesis [13] for a better insight. The survey by [1] also provides a concise introduction to GCT. To begin, a problem in arithmetic complexity is translated to a problem in algebraic geometry, which is then attacked using the tools representation theory. When translated to algebraic geometry the VP vs VNP problem can be stated as follows: Is the padded permanent $\mathrm{PERM}_{m,n}$ [1] polynomial in the *orbit closure* of the $n \times n$ determinant polynomial where $n = 2^{m^{o(1)}}$? For any n-variate polynomial f, the *orbit* of f consists of those polynomials g such that $g(\mathbf{x}) = f(A\mathbf{x})$ where A is invertible. In other words, the orbit of f consists of all polynomials that are equivalent to f. The orbit closure is obtained by taking the Zariski closure of the orbit. A natural algorithmic question arising at this point is the following: Given a polynomial f can we efficiently check if f is in the orbit-closure of the determinant. A starting point would be to understand if we can efficiently check if f is in the orbit of determinant, which is exactly the equivalence testing problem for determinant.

Why Tr-IMM$_{w,d}$ polynomial? The Tr-IMM$_{w,d}$ polynomial is a complete polynomial family for the class VBP just like the DET and $\mathrm{IMM}_{w,d}$[2]. In the same spirit as above, it is natural to ask if equivalence testing of Tr-IMM, $\mathrm{IMM}_{w,d}$, DET can be solved efficiently. For instance, $\mathrm{IMM}_{w,d}$ equivalence test over $\mathbb{Q}$ can be solved efficiently [20] whereas DET equivalence testing over $\mathbb{Q}$ is at least INTFACT hard. In this sense, the complexity of equivalence tests for $\mathrm{IMM}_{w,d}$ and DET are different and dependent on the underlying field. Could it be the case that the complexity of equivalence test for Tr-IMM$_{w,d}$

[1] $\mathrm{PERM}_{m,n} = z^{n-m}\mathrm{PERM}_m$ where z is a fresh variable.

[2] $\mathrm{IMM}_{w,d}$ polynomial is obtained by considering only the $(1,1)$-th entry of the product of the matrices $Q_1 \ldots Q_d$.

and DET are more closely tied to each other (independent of the underlying field)? In this thesis, we take a step towards answering this question. We show that the equivalence test for Tr-$\text{IMM}_{w,d}$ reduces to equivalence test for DET (see also the "Recent Update" at the end of this chapter). It is worth noting that an efficient equivalence testing algorithm for one complete polynomial family need not imply an efficient equivalence test for another complete polynomial family.

1.2 Related Work

Efficient equivalence testing algorithms for other interesting polynomial families have also been looked into. [18] gave an efficient randomized equivalence testing algorithm for the PERM_m polynomial, Power Symmetric polynomial, Sum of Products polynomial, Elementary symmetric polynomial over any field ($\mathbb{Q}$, $\mathbb{C}$, finite fields). In [18], they also gave an efficient equivalence testing algorithm for the DET_n polynomial over $\mathbb{C}$. Recently, [11] gives an efficient randomized algorithm for determinant equivalence testing over finite fields[1]. They also give an efficient randomized reduction from integer factoring to determinant equivalence testing over $\mathbb{Q}$. [20] gives an efficient randomized equivalence test for the $\text{IMM}_{w,d}$ polynomial over $\mathbb{Q}, \mathbb{C}$ and finite fields. Recall that the $\text{IMM}_{w,d}$ polynomial is the $(1,1)$-th entry of the matrix product given in Definition 1.3.

We would like to acknowledge that the question whether or not it is possible to extend [20]'s algorithm to Tr-$\text{IMM}_{w,d}$ was asked by Avi Wigderson to Vineet Nair at CCC'17 after the presentation of [20]'s work. Christian Ikenymeyer also pointed out at the same venue that the Tr-$\text{IMM}_{w,d}$ polynomial is more interesting to mathematicians compared to the $\text{IMM}_{w,d}$. Keeping the "Recent Update" in mind, we answer Avi's query by showing that such an extension from equivalence test of $\text{IMM}_{w,d}$ to equivalence test of Tr-$\text{IMM}_{w,d}$ is not possible irrespective of the underlying field (unless INTFACT is easy).

1.3 Problem Statement

We use $\mathsf{GL}(n,\mathbb{F})$ to denote the set of $n \times n$ invertible matrices over $\mathbb{F}$. We first define the notion of equivalence of two polynomials which is the general question of our interest.

[1] with some conditions on the characteristic of the field

Definition 1.1 (Affine equivalence) *Given an m-variate polynomial f and an n-variate polynomial g (where $m \leq n$), f is said to be an **affine equivalent** to g if there exists a full rank matrix $A \in \mathbb{F}^{n\times m}$ and $\mathbf{b} \in \mathbb{F}^n$ such that $f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{b})$.*

[18] showed that the problem of testing *affine equivalence* of two polynomial reduces to the problem of testing the *equivalence* of two polynomials which is defined as follows.

Definition 1.2 (Equivalent Polynomials) *Two n-variate polynomials f and g are said to be **equivalent** if there exists an $A \in \mathsf{GL}(n,\mathbb{F})$ such that $f(\mathbf{x}) = g(A\mathbf{x})$.*

Example 1.1 *Let $f(x_1,x_2) = x_1 + x_2^2$ and $g(x_1,x_2) = x_1 + x_2 + x_2^2$.Then $f(x_1,x_2) = g(A\cdot(x_1,x_2))$ where $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.*

The problem of testing the equivalence of two polynomials f and g is known to be at least as hard as graph isomorphism and algebra isomorphism [2]. However, the converse is not known.

Definition 1.3 (Tr-IMM$_{w,d}$ polynomial) *The Tr-IMM$_{w,d}$ polynomial is the trace of the matrix product $Q_1 \dots Q_d$ where for all $i \in [d]$, Q_i is a symbolic matrix whose entries are distinct formal variables.*

$$\text{Tr-IMM}_{w,d} = \text{Trace}(Q_1 \dots Q_d) \ .$$

Example 1.2

$$\text{Tr-IMM}_{2,3} = \text{Trace}(Q_1 \cdot Q_2 \cdot Q_3)$$

where

$$Q_1 = \begin{bmatrix} x_{11}^{(1)} & x_{12}^{(1)} \\ x_{21}^{(1)} & x_{22}^{(1)} \end{bmatrix}; Q_2 = \begin{bmatrix} x_{11}^{(2)} & x_{12}^{(2)} \\ x_{21}^{(2)} & x_{22}^{(2)} \end{bmatrix}; Q_3 = \begin{bmatrix} x_{11}^{(3)} & x_{12}^{(3)} \\ x_{21}^{(3)} & x_{22}^{(3)} \end{bmatrix}$$

The total number of variables in Tr-IMM$_{w,d}$ is $n = w^2 d$. We now precisely state our problem.

Problem Statement: Given an n-variate, degree-d polynomial f as a blackbox, find an efficient algorithm that outputs an $A \in \mathsf{GL}(n, \mathbb{F})$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$ if such an A exists; otherwise it outputs "No such A exists".

As mentioned earlier, an efficient equivalence test for the $\text{Tr-IMM}_{w,d}$ polynomial implies an efficient affine equivalence test as well.

1.4 Results

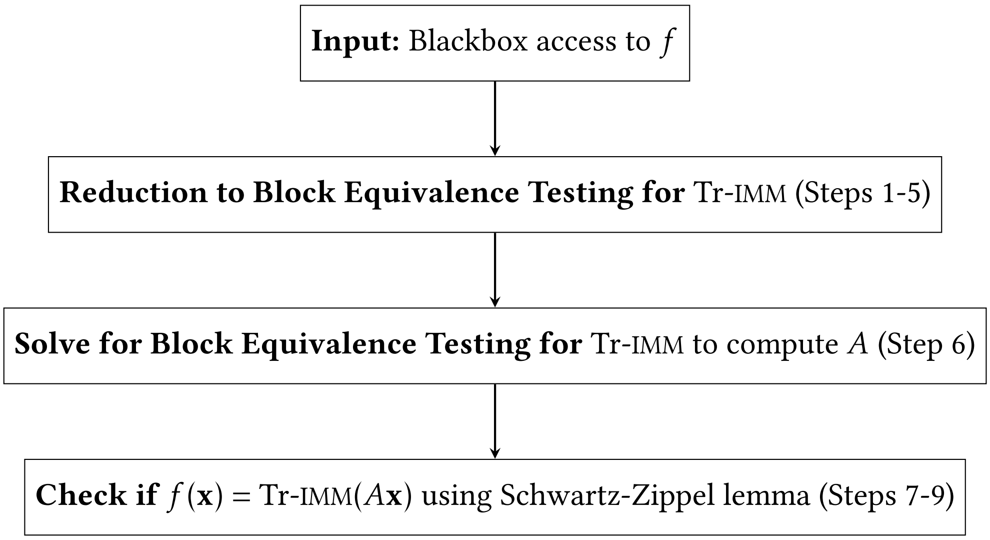
Let DETEQ denote an algorithm that takes as input blackbox access to a n^2-variate, degree n polynomial f and outputs an $n^2 \times n^2$ invertible matrix A such that $f = \text{DET}_n(A\mathbf{x})$ if such an A exists; else it outputs 'No such A exists'. We now state our main result which says that we have an efficient randomized algorithm for checking *affine equivalence* for $\text{Tr-IMM}_{w,d}$ where $w > 1, d > 2$, given oracle access to DETEQ.

Theorem 1.1 *Given an n-variate, degree-d polynomial f as a blackbox, and oracle access to* DETEQ, *there is a randomized algorithm with running time $poly(n, \beta)$ (where β is the bit length of coefficients of f) that outputs with probability $1 - o(1)$ a full rank matrix $A \in \mathbb{F}^{n \times m}$ and $\mathbf{b} \in \mathbb{F}^m$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x} + \mathbf{b})$ (where $w > 1, d > 2$) if such an $A, \mathbf{b}$ exists; otherwise it outputs "No such $A, \mathbf{b}$ exists".*

As affine equivalence testing reduces to equivalence testing [18], it is sufficient to give an efficient randomized algorithm for equivalence testing for $\text{Tr-IMM}_{w,d}$ (Theorem 1.2).

Theorem 1.2 *Given an n-variate, degree-d polynomial f as a blackbox, and oracle access to* DETEQ, *there is a randomized algorithm (Algorithm 1) with running time $poly(n, \beta)$ (where β is the bit length of coefficients of f) that outputs with probability $1 - o(1)$ an $A \in \mathsf{GL}(n, \mathbb{F})$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$ (where $w > 1, d > 2$) if such an A exists; otherwise it outputs "No such A exists".*

Remark: In the remainder of this thesis we will only consider the problem of equivalence testing for $\text{Tr-IMM}_{w,d}$ when $w > 1$ and $d > 2$. When $w = 1$, the problem can be solved using the blackbox



Input: Blackbox access to f

Reduction to Block Equivalence Testing for Tr-IMM (Steps 1-5)

Solve for Block Equivalence Testing for Tr-IMM to compute A (Step 6)

Check if $f(\mathbf{x}) = \text{Tr-IMM}(A\mathbf{x})$ using Schwartz-Zippel lemma (Steps 7-9)

Figure 1.2: High level approach of Algorithm 1

factorization algorithm of Kaltofen-Trager[17]. For reasons elaborated in Chapter 3, our reduction to DETEQ does not hold when $d = 2$.

Algorithm and Proof Strategy: Algorithm 1 follows and extends the approach in the algorithm for equivalence test for the $\text{IMM}_{w,d}$ polynomial given in [20]. The high level idea of the algorithm is as follows: We assume that f and $\text{Tr-IMM}_{w,d}$ are equivalent, i.e $\exists\ A \in \text{GL}(n, \mathbb{F})$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$. We want to devise an algorithm that finds such an A. Our assumption is valid because if f and $\text{Tr-IMM}_{w,d}$ were not equivalent to begin with, then for any A returned by the algorithm, $f(\mathbf{x}) \neq \text{Tr-IMM}_{w,d}(A\mathbf{x})$ which can be checked in randomized polynomial time using Schwartz-Zippel lemma. Figure 1.2 shows the high level approach of our algorithm. In Steps 1-5, we reduce our problem to a simpler problem called *Block Equivalence Test* for Tr-IMM (explained in Chapter 6). We then solve for Block Equivalence in Step 6 to retrieve A. Each step of this algorithm is elaborated in the subsequent chapters. However, we give a brief description of each step of Algorithm 1 below.

Step 1: Associated with every n-variate polynomial f, there is the *Lie algebra* $\mathfrak{g}_f$ of f (Chapter 2), which is a vector space consisting of $n \times n$ matrices satisfying certain constraints. If $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$ then their corresponding Lie algebra are conjugates of each other, i.e $\mathfrak{g}_f = A^{-1} \cdot \mathfrak{g}_{\text{Tr-IMM}} \cdot A$ (Fact 2.2). It turns out that the elements of $\mathfrak{g}_{\text{Tr-IMM}}$ are block-diagonal (Chapter 3). The key idea is to simultaneously *block-diagonalize* the basis elements of $\mathfrak{g}_f$ in order to reconstruct A. For this purpose we compute a basis of the lie algebra of f.

Step 2: Block diagonalizing the basis elements of $\mathfrak{g}_f$ is equivalent to computing the irreducible invariant subspaces of the space $\mathfrak{g}_f$ (See Section 2.4) which we accomplish using Algorithm 3.

Steps 3, 4: We exploit the relation between the *irreducible invariant subspaces* of $\mathfrak{g}_f$ and the *layer spaces* of f to compute a basis for the later (Algorithm 4) and re-order them appropriately.

Step 5: By using a suitable linear map on the basis of the *layer spaces* of $\mathfrak{g}_f$, the problem is reduced to Block Equivalence testing for Tr-IMM.

Step 6: In Chapter 6, we give an efficient algorithm (Algorithm 6) for Block Equivalence testing for Tr-IMM given oracle access to DETEQ. We use this algorithm to finally compute A.

Steps 7-11: We apply Schwartz-Zippel Lemma to check if f is indeed equivalent to Tr-IMM$_{w,d}$ to begin with.

Algorithm 1 Equivalence testing for Tr-IMM$_{w,d}$

```
INPUT: Blackbox access to $n$ variate, degree $d$ polynomial $f$
OUTPUT: An $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$ if such an $A$ exists.
 1: Compute a basis $B$ of the Lie algebra $\mathfrak{g}_f$ of the polynomial $f$ (refer to Algorithmic Preliminary 1).
 2: Using Algorithm 3 and the basis $B$ compute a basis $B'$ for the irreducible invariant subspaces of $\mathfrak{g}_f$.
 3: Using Algorithm 4 and the basis $B'$ compute a permutation of the layer spaces corresponding to $X_1, \ldots, X_d$.
 4: Reorder the layer spaces in order using Claim 5.1.
 5: Given the layer spaces in correct order, reduce the problem to Block Equivalence Testing using Claim 5.2.
 6: Solve for Block Equivalence testing using Algorithm 6 and DETEQ orcale to compute $A$.
 7: Pick a random point $\mathbf{a} \in S^n$ where $S \subseteq \mathbb{F}$ and $|S| \geq \mathrm{poly}(n)$.
 8: if $f(\mathbf{a}) = \text{Tr-IMM}_{w,d}(A\mathbf{a})$ then
 9:     Output $A$.
10: else
11:     Output ‘No such $A$ exists’.
```

Symmetry Characterization: In Chapter 7, we give a proof of the well known fact that the Tr-IMM polynomial is characterized by its symmetries (see Definition 7.1).

1.4.1 Comparison with [20]

We point out some important differences of our work from [20]. The major differences arise due to the difference in the structure of the Lie algebras of Tr-$\text{IMM}_{w,d}$ and $\text{IMM}_{w,d}$. The Lie algebra of Tr-$\text{IMM}_{w,d}$ is block diagonal, whereas the $\text{IMM}_{w,d}$ has corner spaces in addition to the block diagonal structure (Chapter 3). Consequently the irreducible invariant subspaces of the corresponding Lie algebras are slightly different (Chapter 4). In step 3 our algorithm outputs a permutation of the layer spaces corresponding to $X_1, \ldots, X_d$ whereas the algorithm in [20] outputs a permutation of $X_2, \ldots, X_{d-1}$. In step 4, the reordering procedure uses the notion of *evaluation dimension* (see Chapter 5). The evaluation dimension parameters in our algorithm also turns out to be different from [20] (Chapter 5). It will be clear from the respective chapters that these differences are due to the block diagonal structure of $\mathfrak{g}_{\text{Tr-IMM}}$.

However, the main difference is the following: Block Equivalence testing for $\text{IMM}_{w,d}$ reduces to set-multilinear ABP reconstruction which can be performed efficiently over any field. But, Block-Equivalence testing Tr-$\text{IMM}_{w,d}$ polynomial does not reduce to set-multilinear ABP reconstruction. Instead, we reduce to DETEQ which has an efficient algorithm over $\mathbb{C}$ ([18]) and finite fields ([11]). Also, Block Equivalence testing of Tr-$\text{IMM}_{w,d}$ *cannot* reduce to set-multilinear ABP reconstruction over $\mathbb{Q}$ unless INTFACT is easy (see "Recent Update" at the end of this chapter).

1.5 Organization of the thesis

In Chapter 2 we set up notations and state important definitions and other preliminaries. Then, in Chapter 3 we discuss the structure of Lie Algebra of $\mathfrak{g}_{\text{Tr-IMM}}$ which will be used in our algorithm. Chapter 4 elaborates on step 2 of Algorithm 1 in which we compute a basis of the irreducible invariant subspaces of $\mathfrak{g}_f$. In Chapter 5 we extract the matrix A from the irreducible invariant subspaces of $\mathfrak{g}_f$ by exploiting the relation between these spaces and the layer spaces of f and reduce the problem to Block Equivalence testing for Tr-IMM. In Chapter 6, we give an efficient randomized algorithm for

Block Equivalence testing for Tr-IMM polynomial given oracle access to DETEQ. Finally in Chapter 7 we provide an alternate proof for symmetry characterization of the Tr-IMM$_{w,d}$ polynomial.

Recent Update: In [11], they show that DETEQ is randomized polynomial time Turing reducible to another well known problem known as Full-Matrix Algebra Isomorphism (FMAI) and vice-versa. In our work we show that equivalence testing for Tr-IMM is randomized polynomial time Turing reducible to DETEQ. Improving on both these results, Vineet Nair and us have shown that these three problems are randomized Turing reducible to each other.

Chapter 2

Preliminaries

> In this chapter we develop the notations used in the thesis and cover some background concepts to understand the work done as part of the thesis.

We fix some **notations and conventions** used in the thesis. We denote ***variables*** with small letters like x, y, z, x_1, y_2 and so on. ***Variable sets*** are denoted with a bold face font like $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{x}_i, \mathbf{z}_i$. We use bold face font for ***vectors*** as well, however we typically denote them as $\mathbf{u}, \mathbf{v}, \mathbf{w}$. We denote ***numeric matrices*** using A, B, P, M, R etc while ***symbolic matrices*** are denoted by X, X_i, Q_i, Y_i, Z_i etc. Calligraphic font like $\mathcal{U}, \mathcal{V}, \mathcal{W}$ is used to indicate ***vector spaces***. We now begin with some common definitions from linear algebra.

2.1 Linear Algebra

Definition 2.1 (Vector Space) *A Vector Space $\mathcal{V}$ over a field $\mathbb{F}$ is a set with two operations: vector addition $'+' : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$ and scalar multiplication $'\cdot' : \mathbb{F} \times \mathcal{V} \to \mathcal{V}$ satisfying the following properties:*

1. *$(\mathcal{V}, +)$ is an abelian group.*

2. *$1_{\mathbb{F}}.\mathbf{v} = \mathbf{v}$ for all $\mathbf{v} \in \mathcal{V}$.*

3. *$a \cdot (b \cdot \mathbf{v}) = (a \cdot b) \cdot \mathbf{v}$ for all $a, b \in \mathbb{F}$ and $\mathbf{v} \in \mathcal{V}$.*

4. *$(a + b) \cdot \mathbf{v} = a \cdot \mathbf{v} + b \cdot \mathbf{v}$ for all $a, b \in \mathbb{F}$ and $\mathbf{v} \in \mathcal{V}$.*

5. *$a \cdot (\mathbf{u}+\mathbf{v}) = a \cdot \mathbf{u} + a \cdot \mathbf{v}$ for all $a \in \mathbb{F}$ and $\mathbf{u}, \mathbf{v} \in \mathcal{V}$.*

The elements of $\mathcal{V}$ are called *vectors* and the elements of the underlying field $\mathbb{F}$ are called *scalars.*

Definition 2.2 (Subspace) *Let $\mathcal{V}$ be a vector space and $\mathcal{U} \subseteq \mathcal{V}$. Then $\mathcal{U}$ is called a subspace of $\mathcal{V}$ iff it satisfies the following properties:*

1. *$\mathbf{0} \in \mathcal{U}$.*

2. *Closed under vector addition: $\mathbf{u}+\mathbf{v} \in \mathcal{U}$ for all $\mathbf{u}, \mathbf{v} \in \mathcal{U}$.*

3. *Closed under scalar multiplication: $a \cdot \mathbf{u} \in \mathcal{U}$ for all $a \in \mathbb{F}$ and $\mathbf{u} \in \mathcal{U}$.*

It can be easily verified that a subspace of a vector space is also a vector space where the vector addition and scalar multiplication in $\mathcal{U}$ is the vector addition and scalar multiplication of $\mathcal{V}$ respectively restricted to $\mathcal{U}$.

Definition 2.3 (Span) *The span $\mathcal{U}$ of the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{V}$ is denoted as $\operatorname{span}_{\mathbb{F}}(\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}) := \{a_1\mathbf{v}_1 + \ldots + a_n\mathbf{v}_n | a_1, \ldots, a_n \in \mathbb{F}\}$. It is easy to check that $\operatorname{span}(\{\mathbf{v}_1, \ldots, \mathbf{v}_n\})$ is a subspace of $\mathcal{V}$.*

Definition 2.4 (Linear Independence) *Let $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{V}$. The set of vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ is said to be linearly independent over $\mathbb{F}$ if there exists no non-zero tuple $(a_1, \ldots, a_n) \in \mathbb{F}^n$ satisfying $a_1\mathbf{v}_1 + \ldots + a_n\mathbf{v}_n = 0$. Otherwise they are said to be linearly dependent.*

Definition 2.5 (Basis) *A basis of a vector space $\mathcal{V}$ is a set of linearly independent vectors that spans $\mathcal{V}$.*

It can be verified that there exists a bijection between any two basis of $\mathcal{V}$. Thus the cardinality of a basis of $\mathcal{V}$ is independent of the choice of the basis.

Definition 2.6 (Dimension) *If the cardinality of a basis of $\mathcal{V}$ is n for some $n \in \mathbb{N}$, then $\mathcal{V}$ is a called a finite dimensional vector space with dimension denoted as $\dim(\mathcal{V}) := n$.*

Now we define the notion of a coordinate subspace which will be used in Chapter 4.

Definition 2.7 (Coordinate Subspace) *Let e_i be a unit vector in $\mathbb{F}^n$ whose i-th coordinate is 1 and the other coordinates are 0. A coordinate subspace of $\mathbb{F}^n$ is a space spanned by a subset of the unit vectors $\{e_1, \dots, e_n\}$.*

In Chapter 4 we compute the irreducible invariant subspaces of the Lie Algebra of the group of symmetries of a polynomial f (See Section 2.4 for the definition of Lie Algebra). The related definitions are given below. For a matrix $M \in \mathbb{F}^{n\times n}$ and a subspace $\mathcal{U} \subseteq \mathbb{F}^n$, $M\mathcal{U} \overset{\text{def}}{=} \{M \cdot \mathbf{v} \mid \mathbf{v} \in \mathcal{U}\}$. It is easy to observe that $M\mathcal{U}$ is a subspace of $\mathbb{F}^n$.

Definition 2.8 (Invariant Subspace) *Let $M \in \mathbb{F}^{n\times n}$ be an $n \times n$ matrix. Then a subspace $\mathcal{U} \subseteq \mathbb{F}^n$ is an invariant subspace of M if $M\mathcal{U} \subseteq \mathcal{U}$. Further, let $\mathcal{L} \overset{\text{def}}{=} \text{span}_{\mathbb{F}}\{M_1, M_2, \dots, M_k\}$, where $M_i \in \mathbb{F}^{n\times n}$ for all $i \in k$. Then $\mathcal{U}$ is an invariant subspace of $\mathcal{L}$ if $M_i\ \mathcal{U} \subseteq \mathcal{U}$ for all $i \in [k]$.*

Definition 2.9 (Irreducible Invariant Subspace) *An invariant subspace $\mathcal{U} \subseteq \mathbb{F}^n$ of a vector space $\mathcal{L}$ over $\mathbb{F}$ spanned by matrices in $\mathbb{F}^{n\times n}$ is said to be irreducible if there are no invariant subspaces $\mathcal{U}_1$ and $\mathcal{U}_2$ of $\mathcal{L}$ such that $\mathcal{U} = \mathcal{U}_1 \oplus \mathcal{U}_2$ and $\mathcal{U}_1, \mathcal{U}_2$ are not equal to either $\{\mathbf{0}\}$ or $\mathbb{F}^n$.*

Definition 2.10 (Null Space) *Let $M \in \mathbb{F}^{n\times n}$ be an $n \times n$ matrix. Then the null space of M denoted as* $\text{null}(M) := \{\mathbf{v} \in \mathbb{F}^n : M\mathbf{v} = \mathbf{0}\}$.

It can be easily verified that the null space is a subspace of $\mathbb{F}^n$.

Definition 2.11 (Characteristic polynomial of a matrix) *Let M be an $n \times n$ matrix. The characteristic polynomial of M denoted as $h_A(x) := \det(xI_n - M)$. Here $\det(xI_n - M)$ denotes the determinant of the matrix $xI_n - M$.*

Algorithm 2 in Section 2.6 gives an efficient method to compute the closure of a vector $\mathbf{v}$ under the action of a vector space $\mathcal{L}$ spanned by matrices in $\mathbb{F}^{n\times n}$ which is defined as follows.

Definition 2.12 (Closure of a vector) *The closure of a vector $\mathbf{v} \in \mathbb{F}^n$ under the action of a vector space $\mathcal{L}$ over $\mathbb{F}$ spanned by matrices in $\mathbb{F}^{n\times n}$ is the smallest invariant subspace of $\mathcal{L}$ containing $\mathbf{v}$.*

2.2 The Tr-IMM$_{w,d}$ polynomial

This section contains the definitions related to the Tr-IMM$_{w,d}$ polynomial. Recall that the Tr-IMM$_{w,d}$ polynomial is defined as the trace of the product of $Q_1 \cdot Q_2 \ldots Q_d$, where $Q_1, \ldots Q_d$ are $w \times w$ symbolic matrices whose entries are distinct variables. We denote the set of variables in Q_k as $\mathbf{x}_k$ and assume the following ordering on variables: $\mathbf{x}_1 > \mathbf{x}_2 > \ldots > \mathbf{x}_d$. Within a given $\mathbf{x}_i$, the variables are ordered in column major fashion. We either index the variables as $\{x_{11}^{(1)}, x_{12}^{(1)}, \ldots, x_{ww}^{(1)}, \ldots, x_{1w}^{(d)}, x_{2w}^{(d)}, \ldots, x_{ww}^{(d)}\}$ or $\{x_1, x_2, \ldots, x_n\}$ which should be clear from the context. We drop the subscripts and use Tr-IMM to mean Tr-IMM$_{w,d}$ when w and d are clear from the context. Definition 2.13 gives an alternate graph theoretic definition of the Tr-IMM polynomial.

Definition 2.13 *Consider the directed acyclic graph* G$_{\text{Tr-IMM}}$ *given in Figure 2.1. It has $d+1$ layers of vertices with w vertices in each layer. There is an outgoing edge from every vertex in layer k to every vertex in layer $k+1$. The edge from vertex i of layer k to vertex j of layer $k+1$ is labelled as $x_{ij}^{(k)}$. Call the i-th vertex in layer 1 and $d+1$ as s_i and t_i respectively and let $\gamma : s_i \longrightarrow t_i$ denote a path from vertex s_i to vertex t_i. The weight of such a path γ denoted $wt(\gamma)$ is equal to the product of edge labels on that path. The polynomial* Tr-IMM *is defined as:*

$$\sum_{\gamma_1 : s_1 \to t_1} wt(\gamma_1) + \sum_{\gamma_2 : s_2 \to t_2} wt(\gamma_2) + \ldots + \sum_{\gamma_w : s_w \to t_w} wt(\gamma_w) \ .$$

Identifying the label of the edge from vertex i of layer k to vertex j of layer $k+1$ with the (i,j)-th entry of Q_k, it can be easily shown that the Tr-IMM$_{w,d}$ as defined above is equal to Trace$(Q_1 \ldots Q_d)$.

Each monomial in Tr-IMM$_{w,d}$ corresponds to a path in G$_{\text{Tr-IMM}}$. This inspires the definition of a *path monomial* which will turn out to be a useful terminology.

Definition 2.14 (Path Monomial) *A path monomial is a monomial that appears in the* Tr-IMM *polynomial. Every path monomial corresponds to a s_i - t_i path in the* G$_{\text{Tr-IMM}}$ *graph.*

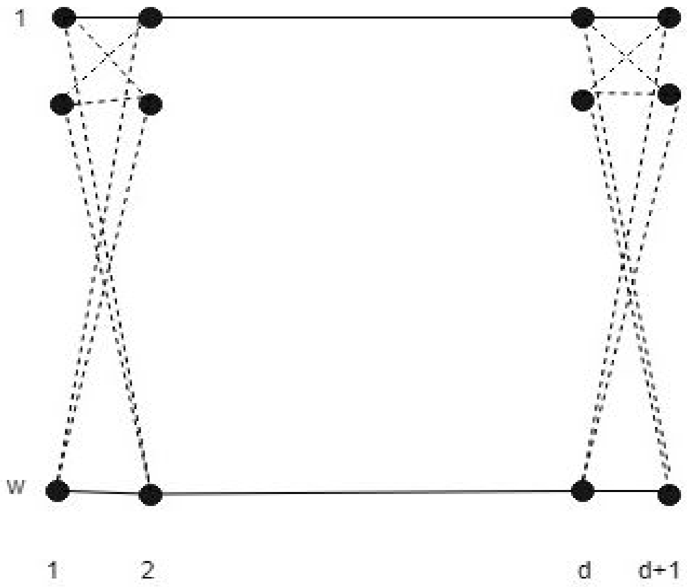


Figure 2.1: The Graph $G_{\text{Tr-IMM}}$

2.3 Algebraic Models of Computation

In Chapter 1, we gave a brief overview of the various algebraic models of computation. In this section, we formally define some of these models and the related complexity classes. Arithmetic circuits are the most natural models to compute polynomials.

Definition 2.15 (Arithmetic Circuits) *An arithmetic circuit is a directed acyclic graph. The nodes having out degree 0 are called the output nodes. The nodes of in degree 0 (input nodes) are labelled with either formal variables or field constants. All other nodes are labelled with either + or ×. Each node of the circuit computes a polynomial in a natural way which is illustrated in Figure 1.1. The set of polynomials computed by the circuit are the polynomials computed by the output nodes. The size of the arithmetic circuit is the total number of node it contains.*

Valiant [33] defined the complexity classes VP and VNP which are arithmetic analogues of non-uniform P and NP respectively.

Definition 2.16 (The Class VP) *A family of polynomials $\{f_n\}_{n\geq 1}$ over a field $\mathbb{F}$ is called **p-bounded** if for each polynomial f_n in the family, the following conditions are satisfied:*

- *the number of variables in* f_n *is* $\mathrm{poly}(n)$.

- *the degree of* f_n *is* $\mathrm{poly}(n)$.

- *there exists a* $\mathrm{poly}(n)$ *sized arithmetic circuit* C_n *that computes* f_n.

The class VP$_{\mathbb{F}}$ *consists of all p-bounded polynomial families over* $\mathbb{F}$.

The symbolic determinant polynomial family, denoted as DET_n, the iterated matrix multiplication polynomial $\mathrm{IMM}_{w,d}$ and the trace of matrix product polynomial $\text{Tr-IMM}_{w,d}$ are in VP.

Definition 2.17 (The Class VNP) *A polynomial family* (f_n) *is said to be* ***p-definable*** *if there exists a polynomial family* $\{g_n\}_{n\geq 1}$ *in VP and a polynomially bounded function* p *such that*

$$f_n(\mathbf{x}) = \sum_{\mathbf{y}\in\{0,1\}^{p(n)}} g_{p(n)}(\mathbf{x},\mathbf{y}) \ .$$

The class VNP$_{\mathbb{F}}$ *consists of all p-definable polynomial families over* $\mathbb{F}$.

The symbolic permanent polynomial family denoted as PERM_m.
While VP is contained in VNP, it is a longstanding open problem to show the separation between these classes (Conjecture 2.1)

Conjecture 2.1 (Valiant's Conjecture [34]) $\mathrm{VP}_{\mathbb{F}} \neq \mathrm{VNP}_{\mathbb{F}}$ *whenever* $\mathbb{F} \neq \mathbb{F}_2$

2.3.1 Algebraic Branching Programs

Algebraic Branching Programs (ABPs) are another well studied model for computing polynomial which are defined as follows.

Definition 2.18 (Algebraic Branching Program) *An Algebraic Branching Program (ABP) is a layered directed acyclic graph with a unique source vertex* s *and a sink vertex* t. *All edges from layer* i *to layer* $i+1$ *are labelled by a linear polynomial. Let* $wt(\gamma)$ *denote the product of edge labels on a path* γ *from* $s \rightsquigarrow t$. *Then the polynomial* f *computed by the ABP is given by:*

$$f = \sum_{\gamma : s\rightsquigarrow t} wt(\gamma)$$

The size of the ABP is the number of edges it contains.

Now we present an alternate equivalent definition of ABPs which we will use often in our thesis.

Definition 2.19 (Algebraic Branching Programs (ABP)) *Let $X_1, \ldots, X_d$ be $w \times w$ symbolic matrices whose entries are affine forms in the $\mathbf{x} \in \mathbb{F}^n$ variables. An Algebraic Branching Program A is the $(1,1)$-th entry of the product $X_1 \cdot X_2 \ldots X_d$.*

Definition 2.20 (The Class VBP) ***The class $VBP_{\mathbb{F}}$*** *consists of all polynomial families that can be computed by ABPs of polynomially bounded size.*

[20] defined the notion of *layer spaces* of an ABP A. Observing that the Tr-IMM$_{w,d}$ polynomial can be written as sum of w ABPs, we state a natural extension of their definition better suited for the purposes of our problem in Definiton 2.21. Our Algorithm crucially uses the relation between the irreducible invariant subspaces of $\mathfrak{g}_f$ and the layer spaces of f which is explained in detail in Chapter 5.

Definition 2.21 (Layer spaces) *Let f be the polynomial computed by the* $\mathrm{Trace}(\prod_{i=1}^{d} X_i)$ *where X_i is a symbolic matrix whose entries are linear forms in the $\mathbf{x}$ variables. Let $\mathcal{X}_i \subseteq \mathbb{F}^n$ denote the space spanned by the linear forms in X_i* [1]. *Then $\mathcal{X}_1, \ldots, \mathcal{X}_d$ are called the layer spaces corresponding to $X_1, \ldots, X_d$ respectively.*

In Step 1 of Algorithm 6, we will reconstruct a set-multilinear ABP computing a polynomial f. We state the related definitions below.

Definition 2.22 (Set-Multilinear ABP) *Let A be an ABP computed by the $(1,1)$-th entry of the matrix product $X_1 \cdot X_2 \ldots X_d$ and let $\mathbf{x}_i$ denote the variables appearing in X_i for all $i \in [d]$. The ABP A is said to set-multilinear whenever $i \neq j$ implies $\mathbf{x}_i \cap \mathbf{x}_j = \emptyset$ for all $i, j \in [d]$.*

The notion of a ***set-multilinear polynomial*** will be used at various places in this thesis. We say that a polynomial f is set-multilinear in the variable sets $\mathbf{x}_1, \ldots, \mathbf{x}_d$ if each monomial of f contains at most one variable from each of the $\mathbf{x}_i$ for $i \in [d]$.

[1] We can associate every linear form $\sum_{i=1}^{n} a_i x_i$ with a vector $(a_1, \ldots, a_n) \in \mathbb{F}^n$.

2.4 Lie Algebra

In Chapter 3, we analyze the structure of the Lie Algebra of the group of symmetries of the Tr-IMM$_{w,d}$ polynomial. We state some useful definitions related to this.

Definition 2.23 (Group of symmetries of a polynomial f) *Let $f(\mathbf{x})$ be an n-variate, degree d polynomial over $\mathbb{F}$. The group of symmetries of f denoted as $\mathcal{G}_f$ is the set of all invertible $n \times n$ matrices $A \in \mathsf{GL}(n, \mathbb{F})$ such that $f(A\mathbf{x}) = f(\mathbf{x})$.*

We work out the group of symmetries for the power symmetric polynomial as an illustrative example.

Example 2.1 (Group of symmetries of the Power Symmetric Polynomial) *Let $f(\mathbf{x}) = x_1^2 + x_2^2$. Consider $A = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \in \mathcal{G}_f$. i.e, it satisfies the following.*

$$\begin{aligned} & f(A\mathbf{x}) = f(\mathbf{x}) \\ \implies & (a_1^2 + a_3^2)x_1^2 + (a_2^2 + a_4^2)x_2^2 + 2(a_1a_2 + a_3a_4) = x_1^2 + x_2^2 \,. \end{aligned}$$

Comparing the coefficients, we get the following system of equations.

$$\begin{aligned} a_1^2 + a_3^2 &= 1 \\ a_2^2 + a_4^2 &= 1 \\ a_1a_2 + a_3a_4 &= 0 \,. \end{aligned}$$

Solving for a's we obtain $A = \begin{bmatrix} a_1 & \pm\sqrt{1-a_1^2} \\ \pm\sqrt{1-a_1^2} & -a_1 \end{bmatrix}$ or $A = \begin{bmatrix} a_1 & \mp\sqrt{1-a_1^2} \\ \pm\sqrt{1-a_1^2} & a_1 \end{bmatrix}$

Observe that $\mathcal{G}_f$ along with the usual matrix multiplication as the group operation forms a group. In fact they are ***Lie Groups***. We illustrate the notion of Lie Groups with an example and refer the reader to standard texts like [14, 21] for a more formal treatment. Consider the set of all transformations $G = \{T_\theta\}$ that rotates a point in $\mathbb{R}^2$ by an angle θ. The set G along with the composition operation defined as $T_{\theta 1} \cdot T_{\theta 2} = T_{\theta 1 + \theta 2}$ forms a group. The inverse of an element is given by $T_\theta^{-1} = T_{-\theta}$. Observe that any element in the group is described by the ***continuous parameter*** $\theta \in \mathbb{R}$. Further the group multiplication operations and the inverse maps are ***smooth maps***. (Informally put, smooth maps are those which are continuous and infinitely differentiable) Such continuous groups whose group multiplication operations and the inverse operations is smooth are known as ***Lie Groups***.

The group of symmetries of the determinant polynomial plays an important role in Algorithm 6. We state the following well known fact without proof.

Fact 2.1 (Group of Symmetries of the determinant polynomial) *Let X be an $n \times n$ symbolic matrix and* $\det(X)$ *be the determinant polynomial of X. If Y is an $n \times n$ matrix with* $\det(Y) = \det(X)$*, then exactly one of the following holds:*

1. *$Y = A \cdot X \cdot B$ where A and B are $n \times n$ matrices with $AB = I_n$.*

2. *$Y = A \cdot X^T \cdot B$ where A and B are $n \times n$ matrices with $AB = I_n$.*

Consider continuous, smooth curves in plane ($f : \mathbb{R} \rightarrow \mathbb{R}$). The *tangent* to the curve at any point (say $(1, f(1))$) is a first order linear approximation of f. Analogously, the elements of Lie Group (which is continuous and has smooth multiplication and inverse maps) can be approximated by the first order linear approximation around the identity $\mathbb{I}_n$. These are what are referred to as the ***Lie Algebra*** of the associated ***Lie Group***. Various equivalent definitions of the Lie Algebra are used in literature and standard texts ([14, 21, 13]). In particular, [19] worked with the following definition for Lie Algebra of the group of symmetries of a polynomial f.

Definition 2.24 (Lie Algebra $\mathfrak{g}_f$ of the group of symmetries of a polynomial f) *Let f be an n-variate polynomial. Then the Lie Algebra of the group of symmetries $\mathcal{G}_f$ of f is denoted as $\mathfrak{g}_f$, consists of all $n \times n$ matrices E satisfying the following equation.*

$$f((I_n + \epsilon E)\mathbf{x}) = f(\mathbf{x})\ .$$

where ϵ is a formal variable with $\epsilon^2 = 0$.

In fact it can be shown that the above definition of Lie Algebra is equivalent to the following definition 2.25 which we use in this thesis and was also used in [18]. We abuse terminology and say Lie Algebra of a polynomial f to mean the Lie Algebra of the group of symmetries of a polynomial f.

Definition 2.25 (Lie Algebra $\mathfrak{g}_f$ of a polynomial f) *Let f be a n-variate polynomial. Then the Lie Algebra of f denoted as $\mathfrak{g}_f$ of the polynomial f is the set of $n \times n$ matrices $E = (e_{ij})_{i,j \in [n]} \in \mathbb{F}^{n \times n}$ such that the following holds:*

$$\sum_{i,j \in [n]} e_{ij} \cdot x_j \cdot \frac{\partial f}{\partial x_i} = 0 \ .$$

Equivalence of Definition 2.24 and Definition 2.25: We show that, if f is a monomial then $f((I_n + \epsilon E)\mathbf{x}) - f(\mathbf{x}) = \epsilon(\sum_{i,j \in [n]} e_{ij} \cdot x_j \cdot \frac{\partial f}{\partial x_i})$. Due to linearity of derivatives, the result will hold for any polynomial f. Let $f = x_1 \cdot x_2 \ldots x_n$. Then the i-th element of $\epsilon E\mathbf{x}$ is $\epsilon(\sum_{j \in [n]} e_{ij} x_j)$. Applying Taylor expansion, we get

$$\begin{aligned} & f(\mathbf{x} + \epsilon E\mathbf{x}) \\ =& f(\mathbf{x}) + \frac{1}{1!} \cdot \left(\sum_{i \in [n]} \left(\epsilon(\sum_{j \in [n]} e_{ij} x_j) \right) \frac{\partial f}{\partial x_i} \right) + \underbrace{\ldots}_{0 \text{ as } \epsilon^2 = 0} \end{aligned}$$

Thus we have $f((I_n + \epsilon E)\mathbf{x}) - f(\mathbf{x}) = \epsilon(\sum_{i,j \in [n]} e_{ij} \cdot x_j \cdot \frac{\partial f}{\partial x_i}) = 0$.

Geometrically, the Lie Algebra of an n-variate polynomial f is the subspace of $\mathbb{F}^{n \times n}$ obtained by translating the tangent of the algebraic set $\{A \in \mathbb{F}^{n \times n} : f(\mathbf{x}) = f(A\mathbf{x})\}$ at $A = I_n$. We now work out the Lie Algebra of the power symmetric polynomial using Definition 2.25.

Example 2.2 (Lie Algebra of the Power Symmetric Polynomial) *Let $f(\mathbf{x}) = x_1^2 + x_2^2$. Consider*

$E = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \in \mathfrak{g}_f$ and hence satisfying the following equation.

$$
\begin{aligned}
&\sum_{i,j\in[n]} e_{ij}x_j \frac{\partial f}{\partial x_i} = 0 \\
&\implies 2e_{11}x_1^2 + 2(e_{12}+e_{21})x_1x_2 + 2e_{22}x_2^2 = 0 \\
&\implies e_{11} = 0, e_{12}+e_{21} = 0, e_{22} = 0 \; .
\end{aligned}
$$

Hence any $E \in \mathfrak{g}_f$ is of the form $E = \begin{bmatrix} 0 & k \\ -k & 0 \end{bmatrix}$ for some $k \in \mathbb{F}$.

Recall the ordering $x_1, \ldots, x_n$ on the variables of Tr-IMM$_{w,d}$ that was defined in Section 2.2. The i-th row and the j-th column of a matrix $E = (e_{ij})_{n\times n}$ in $\mathfrak{g}_{\text{Tr-IMM}}$ is indexed by the variable $x_i, x_j \in \mathbf{x}$ respectively. The following fact says that the Lie Algebra of two equivalent polynomials f and g are conjugates of each other (see [18, 20] for proof). As discussed in Section 1.4, exploiting this relation is the starting point of our algorithm.

Fact 2.2 *Let f and g be n-variate polynomial such that $f(\mathbf{x}) = g(A\mathbf{x})$ where $A \in \mathrm{GL}(n)$. Then the Lie Algebra of f is a conjugate of the Lie Algebra of g via A, i.e.*

$$\mathfrak{g}_f = A^{-1}\mathfrak{g}_g A := \{A^{-1}MA : M \in \mathfrak{g}_g\} \; .$$

In Chapter 3, we will see that the elements of $\mathfrak{g}_{\text{Tr-IMM}}$ have a block-diagonal structure. The following definition explains the structure of a block-diagonal matrix.

Definition 2.26 (Block-diagonal matrix) *Figure 2.2 depicts a block-diagonal matrix. The (i, j)-th entry of a block-diagonal matrix is 0 whenever $x_i \in \mathbf{x}_k$, $x_j \in \mathbf{x}_l$ and $k \neq l$.*

In Section 1.4, we mentioned that block diagonalizing a matrix M is equivalent to computing the basis of the invariant subspaces of M which we elaborate below.

Block diagonalizing a matrix M: Let $M \in \mathbb{F}^{n\times n}$. We wish to *block-diagonalize* M. In other words, we want to compute a basis $\beta = \beta_1 \uplus \ldots \uplus \beta_d$ such that M is block-diagonal with respect to the basis β, i.e. $[M]_\beta$ is block-diagonal. Let $\mathcal{U}_i$ be the space spanned by β_i. As $[M]_\beta$ is block-diagonal, $M \cdot \mathcal{U}_i \subseteq \mathcal{U}_i$ implying that $\mathcal{U}_i$ is an *invariant subspace* of M. Hence computing the basis β that block-diagonalizes M is equivalent to computing the basis of the invariant subspaces $\mathcal{U}_i$ for $i \in [d]$, such that $\mathbb{F}^n = \mathcal{U}_1 \oplus \ldots \oplus \mathcal{U}_d$.

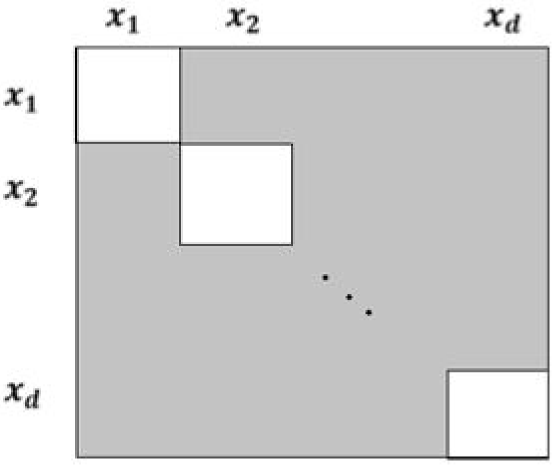


Figure 2.2: Block-Diagonal Matrix

2.5 Technical Lemmas and Facts

In this section, we state some technical lemmas and facts that we will refer to at various places in our thesis.

Lemma 2.1 (Schwartz-Zippel Lemma [35]) *Let f be an n-variate, degree d polynomial over $\mathbb{F}$. Let $S \subseteq F$. Let $a_1, \ldots, a_n$ be chosen from S independently and uniformly at random. Then,*

$$Pr[f(a_1, \ldots, a_n) = 0] \le \frac{d}{|S|} .$$

We now define the Sylvester matrix and the resultant of two uni-variate polynomials f and g. In Lemma 2.2 we show that f and g are co-prime if and only if their resultant is 0.

Definition 2.27 (Sylvester matrix of polynomials f and g) *Let $f(x) = f_n x^n + \ldots + f_0$ and $g(x) = g_m x^m + \ldots + g_0$ be degree n and degree m polynomials respectively. The Sylvester matrix of f and g is a $(n+m) \times (n+m)$ matrix denoted by $\mathrm{Syl}_x(f,g)$. Consider the rows and columns of the matrix to be indexed by the set $[1, n+m]$. Then the (i,j)-th entry can be obtained as follows:*

Case 1: $j \le m$

$$\mathrm{Syl}_x(f,g)[i,j] = \begin{cases} f_{n+j-i} & \textit{if } 0 \le n+j-i \le n \\ 0 & \textit{otherwise} \end{cases}$$

Case 2: $j > m$

$$\mathrm{Syl}_x(f,g)[i,j] = \begin{cases} g_{j-i} & \textit{if } 0 \le j-i \le m \\ 0 & \textit{otherwise} \end{cases}$$

To illustrate the above definition, consider $f = f_3x^3 + f_2x^2 + f_1x + f_0$ and $g = g_2x^2 + g_1x + g_0$. Then $\mathrm{Syl}_x(f,g)$ (a 5×5 matrix) is given by:

$$\begin{bmatrix} f_3 & 0 & g_2 & 0 & 0 \\ f_2 & f_3 & g_1 & g_2 & 0 \\ f_1 & f_2 & g_0 & g_1 & g_2 \\ f_0 & f_1 & 0 & g_0 & g_1 \\ 0 & f_0 & 0 & 0 & g_0 \end{bmatrix}$$

Definition 2.28 (Resultant of polynomials f and g) *Let f and g be univariate polynomials of degree m and n respectively. Then the resultant of f and g denoted by $\mathrm{Res}_x(f,g)$ is defined as the determinant of the Sylvester matrix of f and g, $\mathrm{Syl}_x(f,g)$, i.e. $\mathrm{Res}_x(f,g) = \det(\mathrm{Syl}_x(f,g))$.*

Claim 2.1 *Let f and g be two non-zero univariate polynomials of degree n and m respectively. Then $f(x)$ and $g(x)$ share a non constant factor if and only if there are non-zero univariate polynomials $s(x)$ and $t(x)$ of degrees at most $m-1$ and $n-1$ respectively such that $f(x)s(x) + g(x)t(x) = 0$.*

Proof: Let f and g share a non constant common factor $h(x)$. Then $f(x) = f'(x)h(x)$ and $g(x) =$

$g'(x)h(x)$ and $\deg(f') \leq n-1$ and $\deg(g') \leq m-1$ as $\deg(h(x)) \geq 1$. Hence $f(x)g'(x)+g(x)(-f'(x)) = 0$. To show the other direction let us assume that $f(x)s(x)+g(x)t(x) = 0$ with $\deg(s) \leq m-1$ and $\deg(t) \leq n-1$, and f and g are co-prime. Since f and g are co-prime there exists polynomial p and q such that $f(x)p(x)+g(x)q(x) = 1$. Then

$$\begin{aligned} s(x) &= 1 \cdot s(x) \\ &= (f(x)p(x)+g(x)q(x)) \cdot s(x) \\ &= f(x)p(x)s(x)+g(x)q(x)s(x) \\ &= p(x)(-g(x)t(x))+g(x)q(x)s(x) \\ &= g(x)(-p(x)t(x)+q(x)s(x)) \ . \end{aligned}$$

Since $s(x)$ is a non-zero polynomial $deg(s) \geq deg(g) = m$ which is a contradiction. □

Lemma 2.2 *Two univariate polynomials f and g of degrees n and m respectively have a non constant common factor if and only if* $\mathrm{Res}(f,g) = 0$.

Proof: Let f and g have a non constant common factor. Then by Claim 2.1 there are polynomials s and t of degree $m-1$ and $n-1$ respectively satisfying $fs+gt = 0$. Let $f(x) = \sum_{i=0}^{n} f_i x^i$; $g(x) = \sum_{i=0}^{m} g_i x^i$; $s(x) = \sum_{i=0}^{m-1} s_i x^i$; $t(x) = \sum_{i=0}^{n-1} t_i x^i$. Then we have

$$\begin{aligned} & f(x)s(x)+g(x)t(x) = 0 \\ \implies & \sum_{i=0}^{m+n-1} \left(\sum_{k=0}^{i} \Big(f_{i-k}s_k + g_{i-k}t_k \Big) \right) x^i = 0 \ . \end{aligned}$$

Treat the coefficients of s and t to be formal variables. Equating the coefficients of each x^i in $fs+gt$ to zero we get a system of $(m+n)$ homogeneous linear equations in the s_i and t_j variables. The coefficient matrix of such a system of linear equations is the same as the sylvester matrix of f and g and hence the system has a non-trivial solution if and only if $\det(\mathrm{Syl}_x(f,g)) = 0$. □

Fact 2.3 and Claim 2.2 will turn out to be useful in Chapter 5.

Fact 2.3 *Let $\{p_i(\mathbf{x}) : i \in [m]\}$ and $\{q_j(\mathbf{y}) : j \in [n]\}$ be two sets of linearly independent polynomials over $\mathbb{F}$ and let $\mathbf{x}$ and $\mathbf{y}$ be disjoint. Then the set of polynomials $\{p_i q_j(\mathbf{xy}) : i \in [m], j \in [n]\}$ is also linearly independent over $\mathbb{F}$.*

Proof: Suppose $\{p_i(\mathbf{x}) : i \in [m]\}$, $\{q_j(\mathbf{y}) : j \in [n]\}$ are linearly independent sets of polynomials but $\{p_i q_j(\mathbf{xy}) : i \in [m], j \in [n]\}$ is a linearly dependent set of polynomial.

$$\sum_{\substack{i \in [m] \\ j \in [n]}} a_{ij} p_i(\mathbf{x}) q_j(\mathbf{y}) = 0$$

$$\implies \Big(\sum_{j \in [n]} a_{1j} \cdot q_j \Big) p_1 + \ldots + \Big(\sum_{j \in [n]} a_m j \cdot q_j \Big) p_m = 0.$$

$$\implies \sum_{j \in [n]} a_{ij} \cdot q_j = 0 \;; \forall i \in [m] \ (\because p_1, \ldots, p_m \text{ are linearly independent})$$

$$\implies a_{ij} = 0 \;; \forall i \in [m], j \in [n] \ (\because q_1, \ldots, q_n \text{ are linearly independent})$$

Hence the set of polynomials $\{p_i q_j(\mathbf{xy}) : i \in [m], j \in [n]\}$ are also linearly independent. □

Claim 2.2 *Let $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ be n-variate, degree d polynomials and $\mathcal{U} = \mathrm{span}_{\mathbb{F}}(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ where $\dim(\mathcal{U}) = m - r$ for some $r \in [0, m-1]$. Further, let*

$$M := (f_j(\mathbf{b}_i))_{i,j \in [m]} = \begin{bmatrix} f_1(\mathbf{b}_1) & \ldots & f_m(\mathbf{b}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{b}_m) & \ldots & f_m(\mathbf{b}_m) \end{bmatrix}_{m \times m}$$

where $\mathbf{b}_1, \ldots, \mathbf{b}_m$ are chosen independently and uniformly at random from $S^n \subset \mathbb{F}^n$ where $|S| = d \cdot m \cdot poly(n)$. Then with probability at least $1 - \frac{1}{poly(n)}$, $\mathrm{rank}(M) = m - r$.

Proof: Without loss of generality, let $f_1, \ldots, f_{m-r}$ be a basis of $\mathcal{U}$. Hence $\mathrm{rank}(M)$ is at most $m - r$.

Now we show that, with high probability rank(M) is at least $m-r$. Define M_{m-r} as follows:

$$M_{m-r} := \begin{bmatrix} f_1(\mathbf{b}_1) & \dots & f_{m-r}(\mathbf{b}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{b}_{m-r}) & \dots & f_{m-r}(\mathbf{b}_{m-r}) \end{bmatrix}_{m-r \times m-r}$$

To show that rank of M is at least $m-r$ it is sufficient to show that $\det(M_{m-r}) \neq 0$ with high probability. Let Y be the symbolic matrix

$$Y := \begin{bmatrix} f_1(\mathbf{y}_1) & \dots & f_{m-r}(\mathbf{y}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{y}_{m-r}) & \dots & f_{m-r}(\mathbf{y}_{m-r}) \end{bmatrix}_{m-r \times m-r}$$

where $\mathbf{y}_i$ is a fresh set of n variables for each $i \in [m-r]$ and the $\mathbf{y}_1, \dots, \mathbf{y}_{m-1}$ are pairwise disjoint sets of variables. Since $\{f_1, \dots, f_{m-r}\}$ are linearly independent, $\det(Y)$ is a non-zero polynomial in $\mathbf{y}_1, \dots, \mathbf{y}_{m-r}$ variables. Further $\deg(\det(Y)) \leq dm$. Assign $\mathbf{b}_1, \dots, \mathbf{b}_{m-r}$ for the variable sets $\mathbf{y}_1, \dots, \mathbf{y}_{m-r}$ respectively independently and uniformly at random from S^n. Then by Schwartz-Zippel Lemma (Lemma 2.1) the probability that $\det(M_{m-r}) \neq 0$ is at least $1 - \frac{dm}{d \cdot m \cdot poly(n)} = 1 - \frac{1}{poly(n)}$. □

2.6 Algorthmic Preliminaries

In this section, we state some well known algorithms without proof.

1. **Given blackbox access to an n variate, degree d polynomial f, a basis for the Lie Algebra of a polynomial f can be computed.** This has been elaborated in [18] where they give an efficient randomized algorithm to compute a basis for the Lie Algebra of an n variate, degree d polynomial f that has running time $poly(n, d, \beta)$ where β is the bit length of the coefficients.

2. **Univariate Polynomial Factorization:** [24] gave an efficient deterministic algorithm ($O(poly(n, \beta))$ time), where β is bound on the bit length of the polynomial) to factorize a univariate polynomial of degree n over $\mathbb{Q}$. There are efficient randomized algorithms ($O(poly(n, \log q))$ time) such as [5],[6] to factorize a degree n univariate polynomial over

some finite field $\mathbb{F}_q$. .

3. **Basis of the null spaces:** We can efficiently compute a basis for the null space of any matrix $A \in \mathbb{F}^{n \times n}$. Let A' be the row reduced echeolon form of A. Then null(A) = null(A'). A basis for null(A') can be found by analyzing the equations $A'\mathbf{x} = 0$. It is easy to see that solving this system of equations by gaussian elimination will take $O(n^3)$ time.

4. **Computing the closure of a vector:** Let $\mathbf{v} \in \mathbb{F}^n$ and $\mathcal{L}$ be the vector space over $\mathbb{F}$ spanned by matrices $\{M_1, \ldots, M_k\} \in \mathbb{F}^{n \times n}$. By definition span($\mathbf{v}$) $\in$ closure($\mathbf{v}$). If $\mathbf{u} \in$ closure($\mathbf{v}$) then $M_i \cdot \mathbf{u} \in$ closure($\mathbf{v}$) for each $i \in [k]$. Algorithm 2 computes the closure of a vector $\mathbf{v}$ under the action of $\mathcal{L}$ using this idea. In the *while* loop (steps 4-6), the algorithm executes steps 5 and 6 only if $\mathcal{V}^{(i-1)}$ is a strict subspace of $\mathcal{V}^{(i)}$. Since dim(closure($\mathbf{v}$) $\leq n$, the algorithm terminates after at most n iterations of the *while* loop.

Algorithm 2 Computing the closure of a vector $\mathbf{v}$ under the action of a space $\mathcal{L}$

```
INPUT: $\mathbf{v} \in \mathbb{F}^n$ and a basis $\{M_1, \ldots, M_k\}$ of $\mathcal{L}$.
OUTPUT: Basis of the closure($\mathbf{v}$) under the action of $\mathcal{L}$.
1: $\mathcal{V}^{(0)} = \{\mathbf{v}\}$. Then $T_0 = \{\mathbf{v}\}$ is a basis of $\mathcal{V}^{(0)}$.
2: $\mathcal{V}^{(1)} = \mathrm{span}_{\mathbb{F}}(T_0 \cup \mathcal{L} \cdot T_0)$ and $T_1$ be a basis of $\mathcal{V}^{(1)}$.
3: i = 1
4: while $\mathcal{V}^{(i)} \neq \mathcal{V}^{(i-1)}$ do
5:     i = i+1
6:     $\mathcal{V}^{(i)} = \mathrm{span}_{\mathbb{F}}(T_{i-1} \cup \mathcal{L} \cdot T_{i-1})$ and $T_i$ be a basis of $\mathcal{V}^{(i)}$.
7: Output $T_i$.
```

5. **Checking if two n dimensional vector spaces $\mathcal{V}_1$ and $\mathcal{V}_2$ are equal:** Let V_1 be the matrix whose columns are the basis vectors of V_1. For $\mathcal{V}_1$ to be equal to $\mathcal{V}_2$ we require that for each basis vector $\mathbf{v}$ of $\mathcal{V}_2$, the system of linear equation $V_1\mathbf{x} = \mathbf{v}$ has a solution. Solving a system of linear equations can be done efficiently ($O(\dim(\mathcal{V}_1)^3)$ time).

6. **Set Multilinear ABP Reconstruction:** Recall from Chapter 1 that an ABP is the $(1,1)$-th entry of $w \times w$ matrices $X_1, \ldots, X_d$ whose entries are linear forms in the variables. Let $\mathbf{x}_i$ denote the variables appearing in X_i. An ABP is said to set-multilinear whenever $i \neq j$ implies $\mathbf{x}_i \cap \mathbf{x}_j = \emptyset$ for all $i, j \in [d]$. The *set-multilinear ABP reconstruction* problem is the following: Given

a polynomial f, (re)construct a set-multilinear ABP that computes it. [22] give an efficient algorithm for the same.

Chapter 3

The Lie Algebra of Tr-IMM polynomial

> In this chapter we discuss the structure of the Lie Algebra of the group of symmetries of the Tr-IMM polynomial. The equivalence test algorithm in Theorem 1.2 crucially exploits this structure.

The Lie Algebra of Tr-IMM$_{w,d}$ denoted as $\mathfrak{g}_{\text{Tr-IMM}}$ has already been analysed in [12]. However, in this chapter we present an alternate analysis of the structure of matrices of $\mathfrak{g}_{\text{Tr-IMM}}$, which would be used by the equivalence test algorithm of Theorem 1.2. Recall that the rows and columns of matrices in $\mathfrak{g}_{\text{Tr-IMM}}$ are indexed by variables of Tr-IMM, and that they are ordered using the variable ordering among these $\mathbf{x}$ variables (see Section 2.2).

Recall Definition 2.25 of Lie Algebra from Chapter 2. We are interested in understanding the Lie Algebra $\mathfrak{g}_{\text{Tr-IMM}}$ of Tr-IMM polynomial. The space $\mathfrak{g}_{\text{Tr-IMM}}$ consists of matrices $E = (e_{ij})_{n\times n}$ that satisfies the following condition.

$$\sum_{i,j\in[n]} e_{ij} \cdot x_j \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i} = 0 \quad . \tag{3.1}$$

Recall the notion of block-diagonal matrices from Definition 2.26. Consider the following categorization of the entries of E:

- **Diagonal Entries:** These are the entries of E that appear in the diagonal of E. The rows and columns of each diagonal entry is indexed by the variable x_i for some $i \in [n]$. Hence diagonal entries are of the form e_{ii} where $i \in [n]$.

- **Block-Diagonal Entries:** These are entries of E that lie on some block of E but are not on the diagonal. More precisely, these are entries of the form e_{ij} where the row and column of e_{ij} is indexed by the variables $x_i \in \mathbf{x}_l$ and $x_j \in \mathbf{x}_l$ respectively with $x_i \neq x_j$ and $l \in [d]$.
- **Corner Entries:** These are entries of E that do not lie on any block of E. The row and column of a corner entry e_{ij} is indexed by variables $x_i \in \mathbf{x}_l$ and $x_j \in \mathbf{x}_k$ respectively with $l \neq k$ and $l, k \in [d]$.

We now re-write Equation 3.1 as follows:

$$\underbrace{\sum_{i\in[n]} e_{ii} \cdot x_i \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i}}_{(a)} + \underbrace{\sum_{\substack{x_i \neq x_j \\ x_i, x_j \in \mathbf{x}_l}} e_{ij} \cdot x_j \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i}}_{(b)} + \underbrace{\sum_{\substack{l \neq k \\ x_i \in \mathbf{x}_l \; x_j \in \mathbf{x}_k;}} e_{ij} \cdot x_j \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i}}_{(c)} = 0. \tag{3.2}$$

Note that the terms (a),(b) and (c) correspond to the Diagonal, Block-Diagonal and Corner entries of E respectively. We now show that these terms do not share any monomials in common.

Observation 3.1 *The terms (a), (b) and (c) in Equation 3.2 are pairwise monomial disjoint.*

Proof: A monomial in (a) or (b) has exactly one variable from each of the $\mathbf{x}_k$ for some $k \in [d]$. But any monomial in (c) has two variables from $\mathbf{x}_l$ and none from $\mathbf{x}_k$ where $l \neq k$ and $l, k \in [d]$. Hence (c) is monomial disjoint from (a) and (b). Recall from Definition 2.14 that a path monomial is a monomial that appears in the Tr-IMM polynomial. Clearly, any monomial in (a) is a path monomial. Whereas, no monomial in (b) can be a path monomial due to the fact that any two path monomial must differ by at least two variables. Hence the terms in (a) and (b) are also monomial disjoint. □

We will use the monomial disjointness of (a), (b) and (c) to decompose $\mathfrak{g}_{\text{Tr-IMM}}$ into simpler subspaces. We begin by showing that the matrices in $\mathfrak{g}_{\text{Tr-IMM}}$ are block-diagonal.

Lemma 3.1 *Let $E \in \mathfrak{g}_{\text{Tr-IMM}}$. Then, E is block-diagonal. Further $\mathfrak{g}_{\text{Tr-IMM}} = \mathcal{W}_b \oplus \mathcal{W}_d$ where:*

- *$\mathcal{W}_b$ (Block Diagonal Space) is the subspace that consists of $n \times n$ block-diagonal matrices $E = (e_{ij})_{i,j\in[n]}$ whose diagonal entries are 0 and*

$$\sum_{i,j\in[n]} e_{ij} \cdot x_j \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i} = 0 \ .$$

- *$\mathcal{W}_d$ (Diagonal Space) is the subspace that consists of $n \times n$ diagonal matrices $D = (e_{ii})_{i\in[n]}$ and*

$$\sum_{i\in[n]} e_{ii} \cdot x_i \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i} = 0 \ .$$

Proof: Let $E = (e_{ij})_{n\times n} \in \mathfrak{g}_{\text{Tr-IMM}}$ and hence satisfies Equation 3.1. Using Observation 3.1, Equation 3.1 can be split into the following three monomial disjoint equations.

$$\sum_{i\in[n]} e_{ii} \cdot x_i \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i} = 0; \ e_{ij} \neq 0 \implies x_i = x_j \tag{3.3}$$

$$\sum_{i,j\in[n]} e_{ij} \cdot x_j \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i} = 0; \ e_{ij} \neq 0 \implies x_i, x_j \in \mathbf{x}_l, x_i \neq x_j \tag{3.4}$$

$$\sum_{i,j\in[n]} e_{ij} \cdot x_j \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i} = 0; \ e_{ij} \neq 0 \implies x_i \in \mathbf{x}_l, x_j \in \mathbf{x}_k, l \neq k. \tag{3.5}$$

Let $\mathcal{W}_d, \mathcal{W}_b, \mathcal{W}_c$ denote the spaces containing $n \times n$ matrices satisfying Equations 3.3, 3.4, 3.5 respectively. Clearly $\mathfrak{g}_{\text{Tr-IMM}} = \mathcal{W}_d + \mathcal{W}_b + \mathcal{W}_c$. Further as $\mathcal{W}_d \cap \mathcal{W}_b = \mathcal{W}_c \cap (\mathcal{W}_b + \mathcal{W}_d) = 0_n$ we infer that, $\mathfrak{g}_{\text{Tr-IMM}} = \mathcal{W}_d \oplus \mathcal{W}_b \oplus \mathcal{W}_c$. Claim 3.1 shows that $\mathcal{W}_c = \{\mathbf{0}\}$ implying $\mathfrak{g}_{\text{Tr-IMM}} = \mathcal{W}_d \oplus \mathcal{W}_b$.

□

Claim 3.1 *Let $\mathcal{W}_c$ denote the space spanned by matrices satisfying Equation 3.5. Then $\mathcal{W}_c = \{\mathbf{0}\}$.*

Proof: Let $E = (e_{ij})_{n\times n} \in \mathcal{W}_c$ and hence satisfies Equation 3.5. Every monomial in $x_j \frac{\partial \text{Tr-IMM}}{\partial x_i}$ contains two variables from $\mathbf{x}_k$ and none from $\mathbf{x}_l$. So, the terms $x_j \frac{\partial \text{Tr-IMM}}{\partial x_i}$ and $x_{j'} \frac{\partial \text{Tr-IMM}}{\partial x_{i'}}$ are monomial disjoint whenever $(l,k) \neq (l',k')$. Hence, we can write separate equations corresponding to each such pair (l, k). Fix some (l, k) where $l \neq k$. Then we have:

$$\sum_{x_i \in \mathbf{x}_l, x_j \in \mathbf{x}_k} e_{ij} \cdot x_j \frac{\partial \text{Tr-IMM}}{\partial x_i} = 0$$

Collecting terms of the variable x_i together we have,

$$\sum_{x_i \in \mathbf{x}_l} L_{x_i}^{(k,l)}(\mathbf{x}_k) \frac{\partial \text{Tr-IMM}}{\partial x_i} = 0 \tag{3.6}$$

where $L_{x_i}^{(k,l)}(\mathbf{x}_k)$ is a linear form in the $\mathbf{x}_k$ variables. We now show that this linear form is identically 0 which concludes the proof.

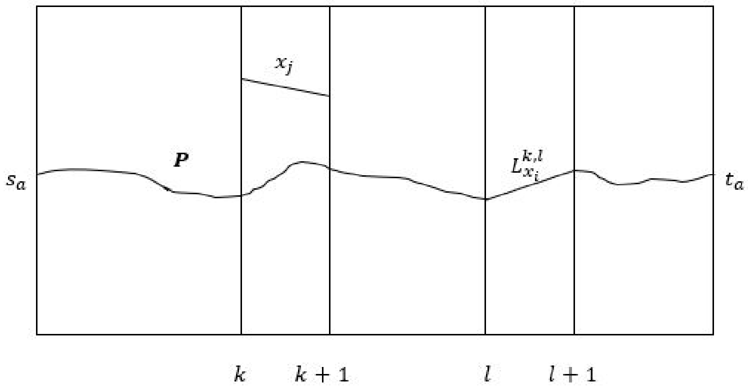


Figure 3.1: The Graph $\text{G}'_{\text{Tr-IMM}}$

Consider the graph $\text{G}_{\text{Tr-IMM}}$. Modify this graph as follows: for every $x_i \in \mathbf{x}_l$ replace the label of x_i by $L_{x_i}^{(k,l)}(\mathbf{x}_k)$. We denote this graph as $\text{G}'_{\text{Tr-IMM}}$ (Figure 3.1). The polynomial computed by $\text{G}'_{\text{Tr-IMM}}$ is the LHS of Equation 3.6 (which is 0). Suppose for contradiction $L_{x_i}^{(k,l)}(\mathbf{x}_k) \neq 0$. Then there must exist at least one $x_j \in \mathbf{x}_k$ such that the corresponding coefficient $e_{ij} \neq 0$. Consider some $(s_a - t_a)$ path P that includes x_i and excludes x_j. Such a path always exists. Now, set all the variables to 0 except the variables appearing in path P and x_j. Under this assignment the polynomial computed by $\text{G}'_{\text{Tr-IMM}}$ is non-zero as the linear form $L_{x_i}^{(k,l)}(\mathbf{x}_k) \neq 0$. But, $\text{G}'_{\text{Tr-IMM}}$ computes a zero polynomial which is a contradiction. □

Remark: Observe that Lemma 3.1 is not true for Tr-IMM$_{w,d}$ where $w = 1$ or $d = 2$. In particular when $d = 2$, the elements of the Lie Algebra of Tr-IMM$_{w,2}$ are *not block diagonal.* Algorithm 1 crucially exploits the block diagonal structure of the Lie Algebra of Tr-IMM$_{w,d}$ and hence it does not extend to the case when $d = 2$.

We elaborate on the spaces $\mathcal{W}_b$ and $\mathcal{W}_d$ in Lemma 3.2 and Lemma 3.3 respectively. These lemmas in particular show that these spaces contain matrices of certain kind which help us to mirror the approach in [20] to give an equivalence test for Tr-IMM.

3.1 The structure of $\mathcal{W}_b$

We introduce few terminologies before explaining the structure of $\mathcal{W}_b$. Recall from Definition 2.13 that we can associate a graph $G_{\text{Tr-IMM}}$ with the Tr-IMM polynomial. The edges from layer i to layer $i+1$ forms the *i-th interface* of the graph. The graph $G_{\text{Tr-IMM}}$ has d interfaces. Roll it into a cylinder such that the vertex s_i coincides with the vertex t_i for each $i \in [w]$ (Figure 3.2). This cylinder also has d interfaces. Any path monomial is a closed path [1] in the cylinder and vice-versa.

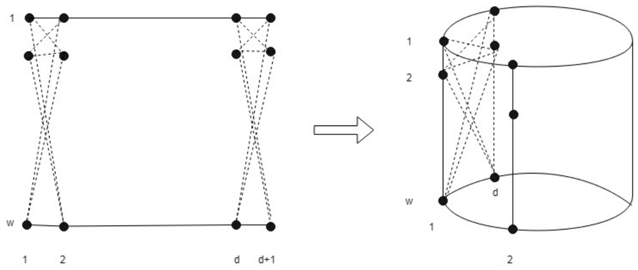


Figure 3.2: Rolling $G_{\text{Tr-IMM}}$ into a cylinder

Consider $B = (b_{ij})_{n\times n} \in \mathcal{W}_b$. It satisfies Equation 3.4 (the e_{ij}'s in the equation are replaced by the corresponding b_{ij}'s). We will now see which terms in this equation have non-zero coefficients. Consider a term $x_j \cdot \frac{\partial \text{Tr-IMM}}{x_i}$ in Equation 3.4. Let $x_i, x_j \in \mathbf{x}_k$ for some $k \in [d]$. We say that x_i and x_j are *parallel* if and x_i and x_j do not share any common end point in the graph $G_{\text{Tr-IMM}}$, i.e. $x_i = x_{pq}^{(k)}$, $x_j = x_{rs}^{(k)}$ and $r \neq p; s \neq q$. We now make the following observation which can be easily verified.

[1] a closed path is a sequence of vertices $v_1, v_2, \ldots, v_n, v_1$ such that (v_i, v_{i+1}) is an edge.

Observation 3.2 *The term $x_j \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i}$ where $x_i, x_j \in \mathbf{x}_k$ are parallel edges does not share a monomial with any other term in Equation 3.4.*

Observation 3.2 implies that Equation 3.4 does not contain terms of the form $x_j \cdot \frac{\partial \text{Tr-IMM}}{\partial x_i}$ where x_i and x_j are parallel. We now look at other terms where x_i and x_j are not parallel.

Monomials broken at k-th interface: A monomial broken at the k-th interface is of the form $x_{i_1 i_2} \cdot x_{i_2 i_3} \dots x_{i_k i_{k+1}} \cdot x_{i'_{k+1} i_{k+2}} \dots x_{i_d i_1}$ $\forall k \in [d-1]$ where $i_{k+1} \neq i'_{k+1}$. A monomial broken at the d-th interface has the form $x_{i_1 i_2} \cdot x_{i_2 i_3} \dots x_{i_d i'_1}$ where $i_1 \neq i'_1$. Any monomial in the term $x^{(k)}_{ps} \cdot \frac{\partial \text{Tr-IMM}}{\partial x^{(k)}_{pq}}$ or $x^{(k+1)}_{qt} \cdot \frac{\partial \text{Tr-IMM}}{\partial x^{(k+1)}_{st}}$ is a monomial broken at k-th interface (Figure 3.3). Infact it can be verified that any monomial broken at the k-th interface is contained in one of the terms $x^{(k)}_{ps} \cdot \frac{\partial \text{Tr-IMM}}{\partial x^{(k)}_{pq}}$ or $x^{(k+1)}_{qt} \cdot \frac{\partial \text{Tr-IMM}}{\partial x^{(k+1)}_{st}}$ where $p, q, s, t \in [w]$. From the above discussion, we make the following observation.

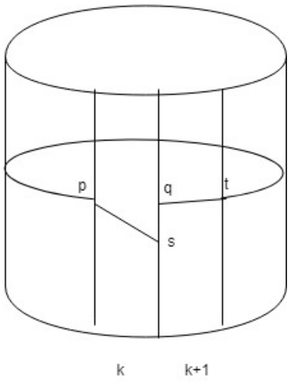


Figure 3.3: Monomial broken at k-th interface.

Observation 3.3 *Let M_k denote the set of all monomials broken at k-th interface, where $k \in [d]$. Then $i \neq j$ implies, M_i and M_j are disjoint.*

The following lemma gives a partial characterization of the space $\mathcal{W}_b$. It is interesting to note that a similar lemma gave a complete characterization of the block diagonal space in case of $\text{IMM}_{w,d}$ ([20]).

Lemma 3.2 *Let $\mathcal{Z}_w$ be the space of $w \times w$ matrices with diagonal entries 0. Let $\mathcal{S}$ be the space spanned by $t \times t$ matrices of the form*

$$\begin{bmatrix} -Z^T \otimes I_w & 0 \\ 0 & I_w \otimes Z \end{bmatrix}_{t \times t}$$

where $t = 2w^2$ and $Z \in \mathcal{Z}_w$. Then, $\mathcal{B}'_1 \oplus \ldots \oplus \mathcal{B}'_d \subset \mathcal{W}_b$ where $\mathcal{B}'_k \cong \mathcal{S}$ for $k \in [d]$.

Proof: Consider $B = (b_{ij})_{n\times n} \in \mathcal{W}_b$ satisfying Equation 3.4. From Observation 3.2 and Observation 3.3 it is clear that Equation 3.4 consists only of terms corresponding to the monomials broken at each of the d interfaces and we can split Equation 3.4 into d equations corresponding to the monomials broken at each of the interface. For some $k \in [d]$, the equation for the k-th interface is:

$$\sum_{\substack{p,q,r\in[w]\\ q\neq r}} b^{(k)}_{(pq,pr)} \cdot x^{(k)}_{pr} \cdot \frac{\partial \text{Tr-IMM}}{\partial x^{(k)}_{pq}} + \sum_{\substack{p,q,r\in[w]\\ p\neq r}} b^{(k+1)}_{(pq,rq)} \cdot x^{(k+1)}_{rq} \cdot \frac{\partial \text{Tr-IMM}}{\partial x^{(k+1)}_{pq}} = 0 \ . \tag{3.7}$$

We now associate subspaces $\mathcal{B}_1, \ldots, \mathcal{B}_d$ with each of these d equations and show that $\mathcal{W}_b = \mathcal{B}_1 \oplus \ldots \oplus \mathcal{B}_d$. Let $\mathcal{B}_k$ denote the space of $n \times n$ matrices B_k whose

- $(x^{(k)}_{pq}, x^{(k)}_{pr})$-th entry is $b^{(k)}_{(pq,pr)}$.
- $(x^{(k+1)}_{pq}, x^{(k+1)}_{rq})$-th entry is $b^{(k+1)}_{(pq,rq)}$.
- the remaining entries are 0.

Further, B_k satisfies Equation 3.7. For any $B \in \mathcal{W}_b, \exists\ B_1 \in \mathcal{B}_1, \ldots. B_d \in \mathcal{B}_d$ such that $B = B_1 + \ldots + B_d$. So $\mathcal{W}_b = \mathcal{B}_1 + \ldots + \mathcal{B}_d$. Further, each B_i for $i \in [d]$, controls different entries of B. Hence $\mathcal{W}_b = \mathcal{B}_1 \oplus \ldots \oplus \mathcal{B}_d$.

Consider the spaces $\mathcal{B}'_1, \ldots, \mathcal{B}'_d$ as defined in the statement of this lemma. To show that $\mathcal{B}'_1 \oplus \ldots \oplus \mathcal{B}'_d \subseteq \mathcal{W}_b$, it is sufficient to show that $\mathcal{B}'_k \subseteq \mathcal{B}_k$ for each $k \in [d]$. Equation 3.7 can be re-written as:

$$\sum_{p,q\in[w]} l^{(k)}_{pq} \frac{\partial \text{Tr-IMM}}{\partial x^{(k)}_{pq}} + \sum_{p,q\in[w]} l^{(k+1)}_{pq} \frac{\partial \text{Tr-IMM}}{\partial x^{(k+1)}_{pq}} = 0$$

$$\text{where } l^{(k)}_{pq} = \sum_{\substack{r\in[w]\\ r\neq q}} b^{(k)}_{(pq,pr)} \cdot x^{(k)}_{pr} \text{and } l^{(k+1)}_{pq} = \sum_{\substack{r\in[w]\\ r\neq p}} b^{(k+1)}_{(pq,rq)} \cdot x^{(k+1)}_{rq}. \tag{3.8}$$

Let $Q_k = (x^{(k)}_{ij})_{i,j\in[w]}$, $Q'_k = (l^{(k)}_{ij})_{i,j\in[w]}$ and $Q_{k+1} = (x^{(k+1)}_{ij})_{i,j\in[w]}$, $Q'_{k+1} = (l^{(k+1)}_{ij})_{i,j\in[w]}$. Equation 3.8

is equivalent to:

$$\mathrm{Trace}(Q_1 \dots Q_{k-1} \cdot Q'_k \cdot Q_{k+1} \dots Q_d \ + Q_1 \dots Q_k \cdot Q'_{k+1} \cdot Q_{k+2} \dots Q_d) = 0 \ . \tag{3.9}$$

A sufficient condition for Equation 3.9 to hold is

$$Q_1 \dots Q_{k-1} \cdot Q'_k \cdot Q_{k+1} \dots Q_d + Q_1 \dots Q_k \cdot Q'_{k+1} \cdot Q_{k+2} \dots Q_d = 0$$

$$\iff Q'_k \cdot Q_{k+1} + Q_k \cdot Q'_{k+1} = 0 \ . \tag{3.10}$$

The (i,j)-th entry of Q'_k is a linear combination of the i-th row of Q_k. Similarly, the (i,j)-th entry of Q'_{k+1} is a linear combination of the j-th column of Q_{k+1}. So, $\exists\ Z_1, Z_2 \in \mathbb{F}^{w \times w}$ such that

$$Q'_{k+1} = Z_1 \cdot Q_{k+1} \text{ and } Q'_k = Q_k \cdot Z_2.$$

Since, the coefficient of $x^{(k)}_{ij}$ in the linear form $l^{(k)}_{ij}$ is 0, the diagonal entries of Z_2 must be 0. By similar argument, the diagonal entries of Z_1 must be 0. So, $Z_1, Z_2 \in \mathcal{Z}_w$. Substituting $Q'_{k+1} = Z_1 \cdot Q_{k+1}$ and $Q'_k = Q_k \cdot Z_2$ in Equation 3.10, we have $Z_1 = -Z_2 = Z$.

Similarly, given a $Z \in \mathcal{Z}_w$ we can construct B'_k that satisfies Equation 3.10 as follows. Let $Z = (z_{ij})_{i,j \in [w]}$. Then

$$l^{(k)}_{pq} = \sum_{r \in [w]} z_{rq} x^{(k)}_{pr} \text{ and } l^{(k+1)}_{pq} = \sum_{r \in [w]} z_{pr} x^{(k+1)}_{rq} \ .$$

But

$$l^{(k)}_{pq} = \sum_{\substack{r \in [w] \\ r \neq q}} b^{(k)}_{pq,pr} \cdot x^{(k)}_{pr} \text{ and } l^{(k+1)}_{pq} = \sum_{\substack{r \in [w] \\ r \neq p}} b^{(k+1)}_{pq,rq} \cdot x^{(k+1)}_{rq} \ .$$

Comparing the coefficients, we get $b^{(k)}_{(pq,pr)} = z_{rq}$ and $b^{(k+1)}_{(pq,rq)} = z_{pr}$. All other entries of B'_k is 0. The sub matrix of B'_k indexed by the variables in $\mathbf{x}_k \uplus \mathbf{x}_{k+1}$ is:

$$\begin{bmatrix} -Z^T \otimes I_w & 0 \\ 0 & I_w \otimes Z \end{bmatrix}_{t \times t}$$

where $t = 2w^2$ and $Z \in \mathcal{Z}_w$. For each $k \in [d]$, define $\mathcal{B}'_k$ to be the space containing $n \times n$ matrices B'_k as discussed above. Clearly, $\mathcal{B}'_k \cong \mathcal{S}$. Any $B'_k \in \mathcal{B}'_k$ satisfies Equation 3.10 and hence Equation 3.9. So $\mathcal{B}'_k \subset \mathcal{B}_k$. Hence, $\mathcal{B}'_1 \oplus \ldots \oplus \mathcal{B}'_d \subset \mathcal{W}_b$. □

3.2 The structure of $\mathcal{W}_d$

Lemma 3.3 gives a partial characterization of the space $\mathcal{W}_d$ which will in turn aid us in the proof of Lemma 3.4.

Lemma 3.3 *Let $\mathcal{Y}_w$ be the space of $w \times w$ diagonal matrices. Let $\mathcal{D}_k$ be the space of $n \times n$ diagonal matrices which is isomorphic to*

$$\text{span}_{\mathbb{F}}\left(\begin{bmatrix} -Y \otimes I_w & 0 \\ 0 & I_w \otimes Y \end{bmatrix}_{t \times t}\right)$$

where $t = 2w^2, Y \in \mathcal{Y}_w$. Then, $\mathcal{D}_1 \oplus \mathcal{D}_2 \oplus \ldots \oplus \mathcal{D}_w \subseteq \mathcal{W}_d$

Proof: Let $D = (d_{ij})_{n \times n} \in \mathcal{W}_d$. By definition, it satisfies the following equation:

$$\sum_{k=1}^{d} \sum_{i,j \in [w]} d_{ij}^{(k)} x_{ij}^{(k)} \frac{\partial \text{Tr-IMM}}{\partial x_{ij}^{(k)}} = 0 \ . \tag{3.11}$$

Let, $Q_k = (x_{ij}^{(k)})_{i,j \in [w]}$ and $Q'_k = (d_{ij}^{(k)} x_{ij}^{(k)})_{i,j \in [w]}$, $\forall k \in [d]$. Then, Equation 3.11 can be re-written as follows:

$$\sum_{k=1}^{d} \text{Trace}(Q_1 \ldots Q_{k-1} \cdot Q'_k \cdot Q_{k+1} \ldots Q_d) = 0 \ . \tag{3.12}$$

A sufficient condition for Equation 3.12 to hold is:

$$\sum_{k=1}^{d} Q_1 \ldots Q_{k-1} \cdot Q'_k \cdot Q_{k+1} \ldots Q_d = 0 \ . \tag{3.13}$$

Define $d+1 := 1$ and $1-1 := d$. Let $D_k \in \mathcal{W}_d$ be a matrix such that $Q'_1, \ldots, Q'_{k-1}, Q'_{k+2}, \ldots, Q'_d$ are all 0 in Equation 3.13. Then, it will satisfy the following equation:

$$Q_1 \ldots Q_{k-1} \cdot Q'_k \cdot Q_{k+1} \ldots Q_d + Q_1 \ldots Q_k \cdot Q'_{k+1} \cdot Q_{k+2} \ldots Q_d = 0$$

$$\implies Q'_k \cdot Q_{k+1} + Q_k \cdot Q'_{k+1} = 0 \ . \tag{3.14}$$

By comparing the coefficients of the $(i,j)^{th}$ entry of $Q'_k \cdot Q_{k+1}$ and $Q_k \cdot Q'_{k+1}$, we observe that the i^{th} row of Q'_k must be equal to the negative of the j^{th} column of Q'_{k+1} for any $i, j \in [w]$. So, Q'_k and Q'_{k+1} must have the following structure:

$$Q'_k = \begin{bmatrix} -\alpha_1 x_{11}^{(k)} & \ldots & -\alpha_w x_{1w}^{(k)} \\ \vdots & \ddots & \vdots \\ -\alpha_1 x_{w1}^{(k)} & \ldots & -\alpha_w x_{ww}^{(k)} \end{bmatrix}$$

$$Q'_{k+1} = \begin{bmatrix} \alpha_1 x_{11}^{(k+1)} & \ldots & \alpha_1 x_{1w}^{(k+1)} \\ \vdots & \ddots & \vdots \\ \alpha_w x_{w1}^{(k+1)} & \ldots & \alpha_w x_{ww}^{(k+1)} \end{bmatrix}$$

Clearly Q'_k and Q'_{k+1} can be written as $Q'_k = -Q_k Y$, $Q'_{k+1} = Y Q_{k+1}$ where $Y \in \mathcal{Y}_w$ is a diagonal matrix with diagonal entries $\alpha_1, \ldots, \alpha_w$. The sub matrix of the corresponding D_k which is indexed by the variables in $\mathbf{x}_k \uplus \mathbf{x}_{k+1}$ is:

$$\begin{bmatrix} -Y \otimes I_w & 0 \\ 0 & I_w \otimes Y \end{bmatrix}_{t \times t} ; \ t = 2w^2 \ .$$

All the other entries of D_k will be 0. Conversely, given $Y \in \mathcal{Y}_w$ we can construct a $D_k \in \mathcal{D}_k$ that satisfies Equation 3.13. This implies

$$\mathcal{D}_k \cong \text{span}\left(\begin{bmatrix} -Y \otimes I_w & 0 \\ 0 & I_w \otimes Y \end{bmatrix}_{t \times t} \middle| t = 2w^2; Y \in \mathcal{Y}_w\right).$$

□

3.3 Characteristic polynomial of a random element in $\mathfrak{g}_{\text{Tr-IMM}}$

Lemma 3.5 says that the characteristic polynomial of a random element $L \in \mathfrak{g}_{\text{Tr-IMM}}$ is square free with high probability. This is crucially used in the block-diagonalization of an element $\mathfrak{g}_f$. Lemma 3.4 is used in the proof of Lemma 3.5 .

Lemma 3.4 *There is a diagonal matrix $D \in \mathfrak{g}_{\text{Tr-IMM}}$ with distinct entries.*

Proof: The argument is same as in [20]. We present the proof here for completeness. From Lemma 3.3, we know that $\forall k \in [d]$, $\exists D_k \in \mathcal{D}_k$ where

$$\mathcal{D}_k \cong \begin{bmatrix} -Y_k \otimes I_w & 0 \\ 0 & I_w \otimes Y_k \end{bmatrix}$$

and $Y_k \in \mathcal{Y}_w$. Treat the entries of the Y_k as distinct formal variables $\mathbf{y_k} = (y_i^{(k)})_{i \in [w]}$. Let $D = D_1 + \ldots + D_d$. By Lemma 3.3, $D \in \mathcal{W}_d$. The $(x_{ij}^{(k)}, x_{ij}^{(k)})$-th entry of D is $-y_j^{(k)} + y_i^{(k-1)}$. Each entry of D is a distinct linear form in $\mathbf{y}$ variables. There are n such linear forms $l_1(\mathbf{y}), \ldots, l_n(\mathbf{y})$. Suppose we assign values to these $\mathbf{y}$s uniformly at random from a set $S \subset \mathbb{F}$. (All the probabilities given below is over the randomness of y values).

For $i \neq j$ and $i, j \in [n]$

$$Pr[l_i(\mathbf{y}) = l_j(\mathbf{y})] = Pr[l_i(\mathbf{y}) - l_j(\mathbf{y}) = 0] \leq \frac{1}{|S|}.$$

Let A denote the event that all the n linear forms have distinct values after assigning values to the

y variables.

$$\begin{aligned}Pr[A] &= \bigwedge_{i\neq j;i,j\in[n]} Pr[l_i(\mathbf{y}) \neq l_j(\mathbf{y})]\\ &= 1 - \bigvee_{i\neq j;i,j\in[n]} Pr[l_i(\mathbf{y}) = l_j(\mathbf{y})]\\ &\geq 1 - \frac{n^2}{|S|}\end{aligned}$$

If we take $|S| > n^2$, then $Pr[A] > 0$. So, there exists a $D \in \mathcal{W}_d$ with all entries distinct.

□

Lemma 3.5 *Let $L_1, \ldots, L_t$ be a basis of $\mathfrak{g}_{\text{Tr-IMM}}$ and $L = \sum_{i=1}^{t} r_i L_i$ where $r_i \in_R S; S \subset \mathbb{F}; |S| = 2n^3$. Then the characteristic polynomial of L is square free with probability at least $1 - \frac{1}{poly(n)}$.*

Proof: Let $h_r(x)$ be the characteristic polynomial of L. If $h_r(x)$ is not square free, then $h_r(x)$ and $\frac{\partial h_r}{\partial x}$ would share a common factor. Hence to show that $h_r(x)$ is square free, it is sufficient to show that $h_r(x)$ and $\frac{\partial h_r}{\partial x}$ are co-prime. $h_r(x)$ and $\frac{\partial h_r}{\partial x}$ are co-prime if and only if $Res_x(h_r(x), \frac{\partial h_r}{\partial x}) \neq 0$. Treat the $\mathbf{r}$'s as formal variables. Then $h_r(x)$ is a polynomial in x and the $\mathbf{r}$ variables.

Observation 3.4 *The $Res_x(h_r(x), \frac{\partial h_r}{\partial x})$ is not an identically zero polynomial in the* **r** *variables with degree at most $2n^2$.*

Proof: $Syl_x(h_r(x), \frac{\partial h_r}{\partial x})$ is a $(2n-1)\times(2n-1)$ matrix whose entries are polynomials in the **r** variables of degree at most n. Hence,$Res_x(h_r(x), \frac{\partial h_r}{\partial x})$ is a polynomial in **r** variables with degree at most $n \times (2n-1) \leq 2n^2$. If $Res_x(h_r(x), \frac{\partial h_r}{\partial x})$ is a identically zero polynomial then $h_r(x)$ is not square free for every setting of the **r** variables. Set **r** such that L is a diagonal matrix with distinct diagonal entries (Such an L exists by Lemma 3.4). For such an L, $h_r(x)$ is square free, which is a contradiction. □

Since $Res_x(h_r(x), \frac{\partial h_r}{\partial x})$ is a not an identically zero polynomial in **r**, we can apply Schwartz-Zippel lemma. The probability that $Res_x(h_r(x), \frac{\partial h_r}{\partial x}) = 0$ where each r_i is chosen independently and uniformly at random from S is at least $1 - \frac{2n^2}{2n^3} = 1 - \frac{1}{poly(n)}$. □

Chapter 4

Irreducible Invariant Subspaces of $\mathfrak{g}_f$

In this chapter we characterize the irreducible invariant subspaces of $\mathfrak{g}_{\text{Tr-IMM}}$ and use this to compute a basis for the irreducible invariant subspaces of a polynomial f equivalent to Tr-IMM$_{w,d}$.

4.1 Irreducible invariant subspaces of $\mathfrak{g}_{\text{Tr-IMM}}$

Recall that $\mathcal{U}$ is an invariant subspace of $\mathfrak{g}_{\text{Tr-IMM}}$ if for all $M \in \mathfrak{g}_{\text{Tr-IMM}}$, $M\mathcal{U} \subseteq \mathcal{U}$. We can associate an unit vector $e_i \in \mathbb{F}^n$ to each variable $x_i \in \mathbf{x}$ based on the ordering of the $\mathbf{x}$ variables defined in Section 2.2. Let $\mathcal{U}_k$ be the coordinate subspace (See Definiton 2.7) spanned by the unit vectors corresponding to the variables in $\mathbf{x_k}$ for all $k \in [d]$. In Lemma 4.1 we show that $\mathcal{U}_1, \ldots, \mathcal{U}_d$ are the only irreducible invariant subspaces of $\mathfrak{g}_{\text{Tr-IMM}}$.

Claim 4.1 *Let $\mathcal{U}$ be an invariant subspace of $\mathfrak{g}_{\text{Tr-IMM}}$. Then $\mathcal{U}$ is a coordinate subspace.*

Proof: For a vector $\mathbf{u} = (u_1, \ldots, u_n) \in \mathcal{U}$. Let $S_{\mathbf{u}}$ be the set of all non zero coordinates of $\mathbf{u}$, i.e. $S_{\mathbf{u}} := \{j : u_j \neq 0 \text{ and } j \in [n]\}$ and $E_{\mathbf{u}} := \{e_j : j \in S_{\mathbf{u}}\}$ be the corresponding unit vectors. Further suppose $E := \cup_{\mathbf{u} \in \mathcal{U}} E_{\mathbf{u}}$. To show that $\mathcal{U}$ is a coordinate subspace,it is sufficient to show that span(E) = $\mathcal{U}$. Clearly, $\mathcal{U} \subseteq$ span(E). To complete the proof we show that $e_j \in \mathcal{U}$ whenever $j \in S_{\mathbf{u}}$ for some $\mathbf{u} \in \mathcal{U}$.

By Lemma 3.4, there exists a diagonal matrix $D \in \mathfrak{g}_{\text{Tr-IMM}}$ with distinct entries $\lambda_1, \ldots, \lambda_n$. Hence, $D^i\mathbf{u} = \mathbf{u_i} = (\lambda_1^i u_1, \ldots, \lambda_n^i u_n) \in \mathcal{U}$ for all $i \in [n]$ and $D^0\mathbf{u} = \mathbf{u}_0 = \mathbf{u}$. Denote $|S_{\mathbf{u}}|$ as m. We use the fact that since $\lambda_1, \ldots, \lambda_n$ are distinct to argue that the vectors $\mathbf{u_0}, \ldots, \mathbf{u_{m-1}}$ are linearly independent as follows: Let $c_0, \ldots, c_{m-1} \in \mathbb{F}$ be such that $c_0\mathbf{u_0} + \ldots + c_{m-1}\mathbf{u_{m-1}} = \mathbf{0}$ and not all $c_0, \ldots, c_{m-1}$ are 0. In particular for each coordinate $j \in [1, n-1]$

$$u_j(c_0 + c_1\lambda_j + \ldots + c_{m-1}\lambda_j^{m-1}) = 0$$

Notice that each λ_j, $j \in S_{\mathbf{u}}$, is a solution of the equation $c_0 + c_1\lambda + \ldots + c_{m-1}\lambda^{m-1} = 0$. But a non zero degree $m-1$ univariate polynomial in λ can have at most $m-1$ roots, implying that $c_0 = c_1 = \ldots = c_{m-1} = 0$.

Since $\mathbf{u}_0, \ldots, \mathbf{u}_{m-1} \in \text{span}(E_{\mathbf{u}})$ and $\dim(\text{span}(\{\mathbf{u}_0, \ldots, \mathbf{u}_{m-1}\})) = \dim(\text{span}(E_{\mathbf{u}})) = m$, we infer that $\text{span}(\{\mathbf{u}_0, \ldots, \mathbf{u}_{m-1}\}) = \text{span}(E_{\mathbf{u}}))$. Hence $e_j \in \mathcal{U}$ where $j \in S_{\mathbf{u}}$ for some $\mathbf{u} \in \mathcal{U}$. □

Lemma 4.1 *The only irreducible invariant subspaces of $\mathfrak{g}_{\text{Tr-IMM}}$ are $\mathcal{U}_1, \ldots, \mathcal{U}_d$.*

Proof: Let $M \in \mathfrak{g}_{\text{Tr-IMM}}$. Since M is block-diagonal, $M \cdot \mathcal{U}_k \subseteq \mathcal{U}_k$ for $k \in [d]$. Hence $\mathcal{U}_1, \ldots, \mathcal{U}_d$ are invariant subspaces. We will now show that $\mathcal{U}_k$ is irreducible for $k \in [d]$. Let $\mathcal{U} \subset \mathcal{U}_k$ for some $k \in [d]$. We want to show that $\mathcal{U} = \mathcal{U}_k$, i.e. $\forall\ y \in \mathbf{x}_k, e_y \in \mathcal{U}$. Let 1_w be the all 1 $w \times w$ matrix and $\tilde{1}_w = 1_w - I_w$. Consider $L \in \mathfrak{g}_{\text{Tr-IMM}}$ such that

$$L = B_{k-1} + D_{k-1} + B_k$$

where B_k, B_{k-1}, D_{k-1} are defined as follows:

- $B_k \in \mathcal{B}'_k$ and whose sub-matrix indexed by variables $\mathbf{x}_k \uplus \mathbf{x}_{k+1}$ looks like :

$$\begin{bmatrix} \tilde{1}_w \otimes I_w & 0 \\ 0 & I_w \otimes -\tilde{1}_w \end{bmatrix}.$$

- $B_{k-1} \in \mathcal{B}'_{k-1}$ and whose sub-matrix indexed by variables $\mathbf{x}_{k-1} \uplus \mathbf{x}_k$ looks like :

$$\begin{bmatrix} -\tilde{1}_w \otimes I_w & 0 \\ 0 & I_w \otimes \tilde{1}_w \end{bmatrix}.$$

- $D_{k-1} \in \mathcal{D}'_{k-1}$ and whose sub-matrix indexed by variables $\mathbf{x}_{k-1} \uplus \mathbf{x}_k$ looks like :

$$\begin{bmatrix} -I_w \otimes I_w & 0 \\ 0 & I_w \otimes I_w \end{bmatrix} .$$

The sub matrix of L indexed by the variables from $\mathbf{x}_k$ is the $w^2 \times w^2$ matrix

$$L_k = \tilde{1}_w \otimes I_w + I_w \otimes \tilde{1}_w + I_w \otimes I_w \quad .$$

Figure 4.1 depicts the structure of L_k. Superscripts of the variables are dropped from Figure 4.1 for clarity. It has w^2 blocks each of size $w \times w$. The diagonal blocks are 1_w and the remaining blocks are I_w.

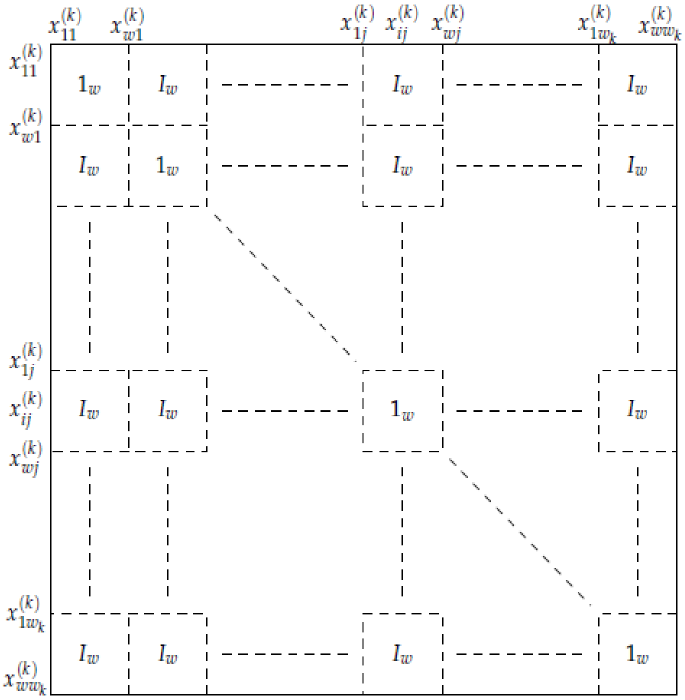


Figure 4.1: Structure of L_k [1]

[1] Image taken from [20].

Let $\mathbf{u} \in \mathcal{U}$ and $y = x_{ij}^{(k)}$ be such that the row indexed by y is non-zero in $\mathbf{u}$. Then from the proof of Claim 4.1 it follows that $e_{\mathbf{y}} \in \mathcal{U}$. Since $\mathcal{U}$ is invariant $Le_y \in \mathcal{U}$. In Figure 4.1, Le_y is the column of L_k indexed by the variable $x_{ij}^{(k)}$. Looking at the structure of L_k in Figure 4.1, we make the following observation.

Observation 4.1 *Let $C := \{x_{1j}^{(k)}, x_{2j}^{(k)}, \ldots, x_{wj}^{(k)}\}$ and $R := \{x_{i1}^{(k)}, x_{i2}^{(k)}, \ldots, x_{iw}^{(k)}\}$. The entries of Le_y corresponding to the variable indexed by $y' \in C \cup R$ are 1 (due to the presence of 1_w matrices and I_w matrices respectively), implying $e_{y'} \in \mathcal{U}$ for all $y' \in C \cup R$.*

We apply Observation 4.1 repeatedly to imply that $e_y \in \mathcal{U}$ for all $y \in \mathbf{x}_k$. Hence $\mathcal{U} = \mathcal{U}_k$. We now show that $\mathcal{U}_1, \ldots, \mathcal{U}_d$ are the only irreducible invariant subspaces of $\mathfrak{g}_{\text{Tr-IMM}}$. Let $\mathcal{V}$ be an irreducible invariant subspace of $\mathfrak{g}_{\text{Tr-IMM}}$ and hence an coordinate space (follows from Claim 4.1). Let $e_x \in \mathcal{V}$ where $x \in \mathbf{x}_k$ for some $k \in [d]$. Observation 4.1 implies that closure$(e_x) = \mathcal{U}_k$. So $\mathcal{U}_k \subseteq V$. Hence $\mathcal{V}$ is a direct sum of the $\mathcal{U}_k$'s such that $e_x \in \mathcal{V}$ for some $x \in \mathbf{x}_k$. But $\mathcal{V}$ is irreducible, implying that it is equal to one of these irreducible invariant subspaces. □

4.2 Irreducible Invariant Subspaces of $\mathfrak{g}_f$

Let f be equivalent to Tr-IMM, i.e. $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$ where $A \in \mathsf{GL}(n, \mathbb{F})$. The irreducible invariant subspaces of $\mathfrak{g}_f$ are related to the irreducible invariant subspaces of $\mathfrak{g}_{\text{Tr-IMM}}$ as given by Corollary 4.1.

Observation 4.2 *$\mathcal{U}$ is an irreducible invariant subspace of $\mathfrak{g}_{\text{Tr-IMM}}$ if and only if $A^{-1}\mathcal{U}$ is an irreducible invariant subspace of $\mathfrak{g}_f$.*

Proof: If $f(\mathbf{x}) = \text{Tr-IMM}(A\mathbf{x})$ then $\mathfrak{g}_f = A^{-1}\mathfrak{g}_{\text{Tr-IMM}}A$ (from Fact 2.2). Let $\mathcal{U}$ be an irreducible invariant subspace of $\mathfrak{g}_{\text{Tr-IMM}}$. Then, for any $\mathbf{u} \in \mathcal{U}$ and $M \in \mathfrak{g}_{\text{Tr-IMM}}$, $M\mathbf{u} \in \mathcal{U}$. Let $\mathbf{u}' \in A^{-1}\mathcal{U}$, i.e. $\mathbf{u}' = A^{-1}\mathbf{u}$ for some $\mathbf{u} \in \mathcal{U}$. Let $M' = A^{-1}MA \in \mathfrak{g}_f$ where $M \in \mathfrak{g}_{\text{Tr-IMM}}$. Then

$$M'\mathbf{u}' = (A^{-1}MA)(A^{-1}\mathbf{u})$$
$$\implies M'\mathbf{u}' = A^{-1}(M\mathbf{u}) \in A^{-1}\mathcal{U} \ .$$

Hence $A^{-1}\mathcal{U}$ is an invariant subspace of $\mathfrak{g}_f$. Now we show that $A^{-1}\mathcal{U}$ is irreducible. Suppose $A^{-1}\mathcal{U}$ is not irreducible. Then $A^{-1}\mathcal{U} = \mathcal{V}_1 \oplus \mathcal{V}_2$ where $\mathcal{V}_1$ and $\mathcal{V}_2$ are invariant subspaces of $\mathfrak{g}_f$. But this would imply $\mathcal{U} = A \cdot \mathcal{V}_1 \oplus A \cdot \mathcal{V}_2$ which is a contradiction. The other direction can proved in a similar fashion. □

Corollary 4.1 follows from Observation 4.2 and Lemma 4.1.

Corollary 4.1 *The only irreducible invariant subspaces of g_f are $A^{-1}\mathcal{U}_1, \ldots, A^{-1}\mathcal{U}_d$.*

We exploit the relation between the invariant subspaces of $\mathfrak{g}_{\text{Tr-IMM}}$ and $\mathfrak{g}_f$ to give an efficient randomized algorithm that computes a bases of the irreducible invariant subspaces of $\mathfrak{g}_f$ when given a basis of $\mathfrak{g}_f$.

4.2.1 Computing a basis for the irreducible invariant subspaces of $\mathfrak{g}_f$

Algorithm 3 outputs the irreducible invariant subspaces of $\mathfrak{g}_f$ given a basis $M_1, \ldots, M_t$ of $\mathfrak{g}_f$. This is the same algorithm used in [20] to compute the irreducible invariant subspaces of the Lie Algebra of a polynomial equivalent to $\text{IMM}_{w,d}$ polynomial. After stating the algorithm, we analyze it by tracing its steps.

Algorithm 3 Computing the irreducible invariant subspaces of $\mathfrak{g}_f$

```
INPUT: A basis $\{M_1, \ldots, M_t\}$ of $\mathfrak{g}_f$.
OUTPUT: A basis of the irreducible invariant subspaces of $\mathfrak{g}_f$.
Pick a random element $R' \in \mathfrak{g}_f$ as follows: for each $i \in [t]$, pick $r_i \in_R S$ independently and
  uniformly at random, where $S \subseteq \mathbb{F}$ and $|S| = 2n^3$. Define $R' := \sum_{i \in [t]} r_i \cdot M_i$ .
Compute $h(x)$– the characteristic polynomial of $R'$.
if $h(x)$ is not square-free then
    Output ‘Fail ’.
Factor $h(x)$ into irreducible factors $g_1(x), \ldots, g_s(x)$ over the field $\mathbb{F}$.
Find a basis of the null spaces $\mathcal{N}_1', \ldots, \mathcal{N}_s'$ of $g_1(R'), \ldots, g_s(R')$ respectively.
For every $i \in [s]$, pick an arbitary vector $\mathbf{v} \in \mathcal{N}_i'$ and compute the closure($\mathbf{v}$) under the action of
  $\mathfrak{g}_f$ using Algorithm 2.
Let $\mathcal{V}_1, \ldots, \mathcal{V}_s$ be the list of the closure spaces. Remove duplicates from this list by comparing all
  possible pairs of spaces from the list and get the pruned list $\mathcal{V}_1, \ldots, \mathcal{V}_d$.
Output the list $\{\mathcal{V}_1, \ldots, \mathcal{V}_d\}$.
```

Steps 1-4: Let R' be a random element of $\mathfrak{g}_f$ as chosen in step 1 and $R = A \cdot R' \cdot A^{-1}$ be the corresponding random element in $\mathfrak{g}_{\text{Tr-IMM}}$. Since the elements of $\mathfrak{g}_{\text{Tr-IMM}}$ is block-diagonal (Lemma 3.1), R is a block-diagonal matrix with individual blocks $R_1, \ldots, R_d$ (Figure 4.2). Further, since R and R' are similar matrices, they have the same characteristic polynomial denoted as $h(x)$ in step 2. By Lemma 3.5, $h(x)$ is square free with high probability. This ensures that the check at step 3 succeeds with high probability.

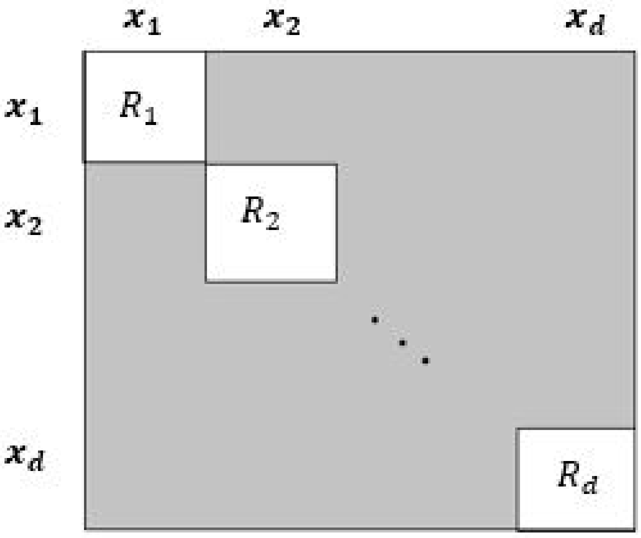


Figure 4.2: A random matrix $R \in \mathfrak{g}_{\text{Tr-IMM}}$

Step 5 Since R is block-diagonal, its characteristic polynomial $h(x)$ can be written as product of the characteristic polynomial of $R_1, \ldots, R_d$. Let $h_k(x)$ be the characteristic polynomial of R_k where $k \in [d]$. Then $h(x) = \prod_{k \in [d]} h_k(x)$. Let $g_1(x), \ldots, g_s(x)$ be the distinct irreducible factors of $h(x)$, i.e. $h(x) = \prod_{i=1}^{s} g_i(x)$. Note that each $g_i(x)$ is a factor of some $h_k(x)$ where $k \in [d]$. Factoring the univariate polynomial $h(x)$ into irreducible factors can be performed efficiently (See Algorithmic Preliminary 2)

Step 6 Let $\mathcal{N}_i$ be the null space of $g_i(R)$ and $\mathcal{N}_i'$ be the null space of $g_i(R')$. In step 6 we compute a basis of the null spaces $\mathcal{N}_1', \ldots, \mathcal{N}_s'$ by solving the system of linear equations $g_i(R') \cdot \mathbf{x} = 0$ which can be done efficiently (See Algorithmic Preliminary 3). It can be easily verified that $\mathcal{N}_i = A\mathcal{N}_i'$.

Claim 4.2 explains the relation between these null spaces and the irreducible invariant subspaces of $\mathfrak{g}_f$. The algorithm exploits this relation in step 7 to compute these irreducible invariant subspaces. The proof of Claim 4.2 is provided at the end of the chapter.

Claim 4.2 *Let $\mathcal{N}_i$ be the null space of $g_i(R)$ and $\mathcal{N}_i'$ be the null space of $g_i(R')$ where $g_i(x)$ is an irreducible factor of the characteristic polynomial $h_k(x)$ of some R_k, $k \in [d]$. Then $\mathcal{N}_i \subseteq \mathcal{U}_k$ and $\mathcal{N}_i' \subseteq A^{-1}\mathcal{U}_k$.*

Step 7 In step 7, we compute the closure of an arbitrary vector $\mathbf{v} \in \mathcal{N}_i'$ for all $i \in [s]$. This can be computed efficiently. (See Algorithmic Preliminary 4). In Lemma 4.2 which is proved at the end of the chapter, we show that the closure of such a vector is an irreducible invariant subspace of $\mathfrak{g}_f$.

Lemma 4.2 *Let $\mathcal{N}_i'$ be the null space of $g_i(R')$ where $g_i(x)$ is an irreducible factor of the characteristic polynomial $h_k(x)$ of some R_k, $k \in [d]$ and $\mathbf{v} \in \mathcal{N}_i'$. Then the irreducible invariant subspace $A^{-1}\mathcal{U}_k$ of $\mathfrak{g}_f$ is equal to the* closure($\mathbf{v}$) *under the action of $\mathfrak{g}_f$.*

Steps 8-9: In step 8, we remove the duplicate spaces from the list and output the pruned list in step 9.

Remark: If f is not equivalent to Tr-IMM, then there does not exist an $A \in GL(n)$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$. We perform a few additional checks after step 8 to handle this case. If the length of the pruned list is not d then output 'Fail'. If $\mathbb{F}^n \neq \mathcal{V}_1 \oplus \ldots \oplus \mathcal{V}_d$ then output 'Fail'.

Proofs of some Lemmas and Claims

Claim 4.2 (Restated) *Let $\mathcal{N}_i$ be the null space of $g_i(R)$ and $\mathcal{N}_i'$ be the null space of $g_i(R')$ where $g_i(x)$ is an irreducible factor of the characteristic polynomial $h_k(x)$ of some R_k, $k \in [d]$. Then $\mathcal{N}_i \subseteq \mathcal{U}_k$ and $\mathcal{N}_i' \subseteq A^{-1}\mathcal{U}_k$.*

Proof:

Observe that $\mathcal{N}_i \subseteq \mathcal{U}_k$ implies that $\mathcal{N}_i' \subseteq A^{-1}\mathcal{U}_k$. So, it is sufficient to show that $\mathcal{N}_i \subseteq \mathcal{U}_k$. Let $\mathbf{v} \in \mathcal{N}_i$. We need to show that $\mathbf{v} \in \mathcal{U}_k$. Let $\mathbf{v}_i$ be the rows of $\mathbf{v}$ restricted to the variables indexed by $\mathbf{x}_i$.

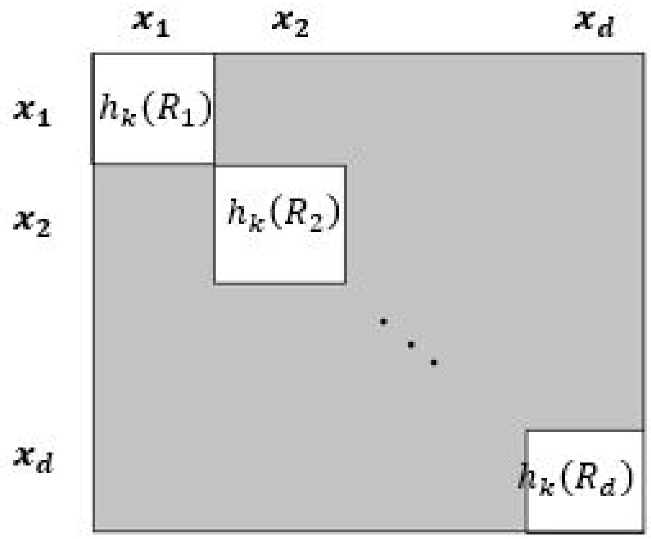


Figure 4.3: A random matrix $R \in \mathfrak{g}_{\text{Tr-IMM}}$

Consider the matrix $h_k(R)$ (Figure 4.3). It is a block-diagonal matrix with blocks $h_k(R_1),\ldots,h_k(R_d)$. Since $\mathbf{v} \in \mathcal{N}_i$ and $g_i(x)$ is a factor of $h_k(x)$,

$$\begin{aligned} g_i(R)\cdot\mathbf{v} &= 0 \\ \implies h_k(R)\cdot\mathbf{v} &= 0 \\ \implies h_k(R_i)\cdot\mathbf{v_i} &= 0;\ \forall i \in [d]\ . \end{aligned}$$

Further $h_j(R_j).\mathbf{v_j} = 0$ for all $j \in [d]$ as $h_j(x)$ is the characteristic polynomial of R_j. Since $h_k(x)$ and $h_j(x)$ are co-prime for $k \neq j$, by Bezout's lemma there exists polynomial $p(x)$ and $q(x) \in \mathbb{F}[\mathbf{x}]$ such that

$$\begin{aligned} p(x)h_k(x) + q(x)h_j(x) &= 1 \\ \implies p(R_j)h_k(R_j) + q(R_j)h_j(R_j) &= I_n \\ \implies p(R_j)\underbrace{h_k(R_j)\cdot\mathbf{v_j}}_{\mathbf{0}} + q(R_j)\underbrace{h_j(R_j)\cdot\mathbf{v_j}}_{\mathbf{0}} &= \mathbf{v_j} \end{aligned}$$

$$\implies \mathbf{v_j} = 0$$

$\mathbf{v_j} = 0$ for $j \neq k$ implies $\mathbf{v} \in \mathcal{U}_k$. Hence, $\mathcal{N}_i \subseteq \mathcal{U}_k$. □

Lemma 4.2 (Restated) *Let $\mathcal{N}_i'$ be the null space of $g_i(R')$ where $g_i(x)$ is an irreducible factor of the characteristic polynomial $h_k(x)$ of some R_k, $k \in [d]$ and $\mathbf{v} \in \mathcal{N}_i'$. Then the irreducible invariant subspace $A^{-1}\mathcal{U}_k$ of $\mathfrak{g}_f$ is equal to the* closure($\mathbf{v}$) *under the action of $\mathfrak{g}_f$.*

Proof: From Claim 4.2 $\mathcal{N}_i' \subseteq A^{-1}\mathcal{U}_k$. Since $A^{-1}\mathcal{U}_k$ is irreducible, $A^{-1}\mathcal{U}_k$ is the smallest invariant subspace of $\mathfrak{g}_f$ containing $\mathcal{N}_i'$. Hence for any vector $\mathbf{v} \in \mathcal{N}_i'$, $A^{-1}\mathcal{U}_k = \text{closure}(\mathbf{b})$ under the action of $\mathfrak{g}_f$. □

Chapter 5

Computing the layer spaces of f and reduction to Block Equivalence testing

In this chapter we exploit the relation between the irreducible invariant subspaces of a polynomial equivalent to f (computed in Chapter 4) and the layer spaces of f to compute a basis for the later. We will also see that the matrix A such that $f(\mathbf{x}) = \text{Tr-IMM}(A\mathbf{x})$ can be computed from these layer spaces (reordered appropriately) by reducing the problem to Block Equivalence testing for Tr-IMM polynomial.

Let $f = \text{Tr-IMM}_{w,d}(A\mathbf{x})$. Then there exists $w \times w$ matrices $X_1, \ldots, X_d$ whose entries are linear forms in the $\mathbf{x}$ variables such that $f(\mathbf{x}) = \text{Trace}(\prod_{i=1}^{d} X_i)$. Since A is invertible, the entries of $X_1, \ldots, X_d$ are $\mathbb{F}$-linearly independent. Recall the definition of Layer Spaces from Definition 2.21. Let $\mathcal{X}_1, \ldots, \mathcal{X}_d$ be the layer spaces corresponding to $X_1, \ldots, X_d$ respectively. Algorithm 3 outputs the irreducible invariable subspaces $\mathcal{V}_1, \ldots, \mathcal{V}_d$ of $\mathfrak{g}_f$ where $\mathcal{V}_i = A^{-1}\mathcal{U}_{\sigma(i)}$ for all $i \in [d]$ and σ is a permutation on $[d]$. Recall that $\dim(\mathcal{V}_i) = w^2$ for $i \in [d]$. The invariant subspace $\mathcal{V}_i$ can be represented by a $n \times w^2$ matrix V_i where the k-th column of V_i is the k-th basis vector of $\mathcal{V}_i$. Define $V := V_1|V_2|\ldots|V_{d-1}|V_d$ to be the matrix obtained by concatenating the matrices $V_1, \ldots, V_d$ in that order. We have already checked if $\mathbb{F}^n = \mathcal{V}_1 \oplus \ldots \oplus \mathcal{V}_d$ in Algorithm 3. Since $\mathbb{F}^n = \mathcal{V}_1 \oplus \ldots \oplus \mathcal{V}_d$, the basis vectors of the $\mathcal{V}_1, \ldots, \mathcal{V}_d$ are linearly independent. Hence V^{-1} exists. In this chapter we present an algorithm (Algorithm 4) to compute a basis for the layer spaces of f given any basis of the irreducible invariant subspaces of $\mathfrak{g}_f$ (as computed in Algorithm 3). We will also see that these layer spaces are *unique*.

Algorithm 4 Computing the layer spaces of f

```
INPUT: $n \times n$ matrix $V$ as described previously.
OUTPUT: $\mathcal{Y}_1, \ldots, \mathcal{Y}_d$– the layer spaces of $f$ .
  Compute $V^{-1}$.
  $\mathcal{Y}_i$ := space spanned by the rows $(i-1)w^2$ to $iw^2$ for all $i \in [d]$.
  Output $\mathcal{Y}_1, \ldots, \mathcal{Y}_d$ in that order.
```

The following lemma establishes the correctness of Algorithm 4.

Lemma 5.1 *If* $f = \text{Trace}(X_1 \ldots X_d)$ *and* $\mathcal{Y}_1, \ldots, \mathcal{Y}_d$ *is the output of Algorithm 4 then there exists a permutation* σ *on* $[d]$ *such that* $\mathcal{Y}_i = \mathcal{X}_{\sigma(i)}$*,* $\forall\ i \in [d]$*.*

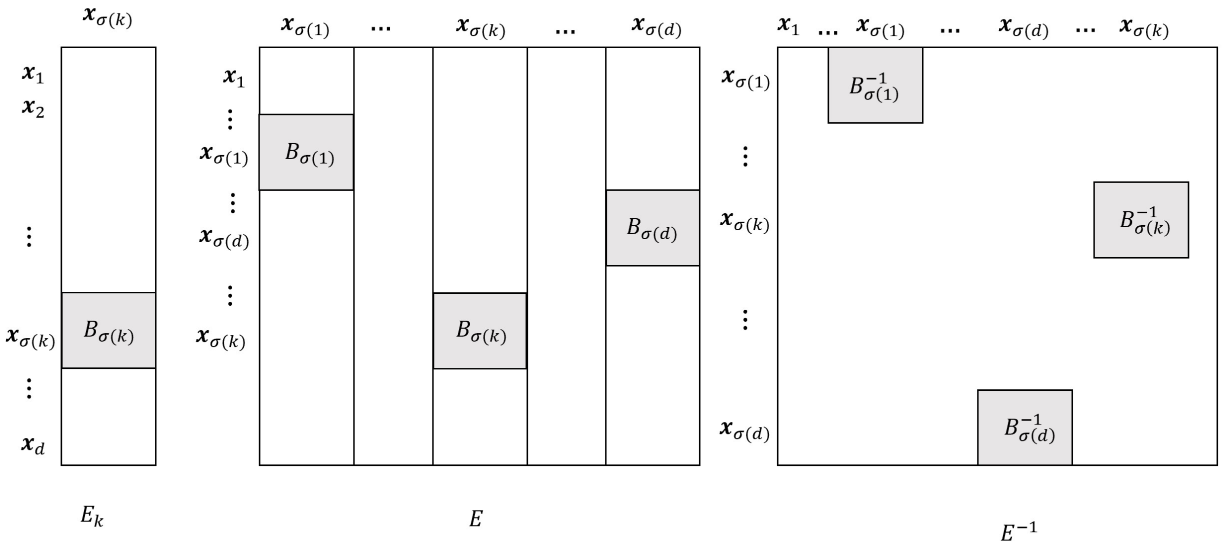


Figure 5.1: The matrices $E_k.E, E^{-1}$

Proof: Recall that $\mathcal{U}_{\sigma(k)}$ is the space spanned by coordinate vectors corresponding to the variables in $\mathbf{x}_{\sigma(k)}$. Since $\mathcal{V}_k = A^{-1}\mathcal{U}_{\sigma(k)}$, V_k can be written as $V_k = A^{-1}E_k$ where E_k is the following $n \times w^2$ matrix (see Figure 5.1): its rows are indexed by the $\mathbf{x}$ variables and the columns are indexed by $\mathbf{x}_{\sigma(k)}$ variables. Further, the only non-zero entries of E_k corresponds to the entries whose rows are indexed by $\mathbf{x}_{\sigma(k)}$ variables. Define $E := E_1|\ldots|E_d$ as the concatenation of the matrices $E_1, \ldots, E_d$. Figure 5.1 depicts the structure of E and E^{-1}. The rows of E are indexed by the variables from $\mathbf{x}_1, \ldots, \mathbf{x}_d$ in the usual order while the columns are indexed by the variables $\mathbf{x}_{\sigma(1)}, \ldots, \mathbf{x}_{\sigma(d)}$ in that order.The only non-zero entries of E are confined to the blocks $B_{\sigma(1)}, \dots, B_{\sigma(d)}$ where $B_{\sigma(k)}$ is the block whose rows and columns are indexed by the variables in $\mathbf{x}_{\sigma(k)}$. The columns of E^{-1} are indexed by the variables from $\mathbf{x}_1, \dots, \mathbf{x}_d$ in the usual order while the rows are indexed by the variables $\mathbf{x}_{\sigma(1)}, \dots, \mathbf{x}_{\sigma(d)}$ in that order. The non-zero entries of E^{-1} are confined to the blocks $B^{-1}_{\sigma(1)}, \dots, B^{-1}_{\sigma(d)}$ where $B^{-1}_{\sigma(k)}$ is the block whose rows and columns are indexed by the variables in $\mathbf{x}_{\sigma(k)}$. As $V = A^{-1}E$, we have $V^{-1} = E^{-1}A$ (See Figure 5.2). Looking at the structure of V^{-1} from Figure 5.2 we infer that the span of rows $(i-1)w^2$ to iw^2 is equal to $\mathcal{X}_{\sigma(i)}$ for each $i \in [d]$.

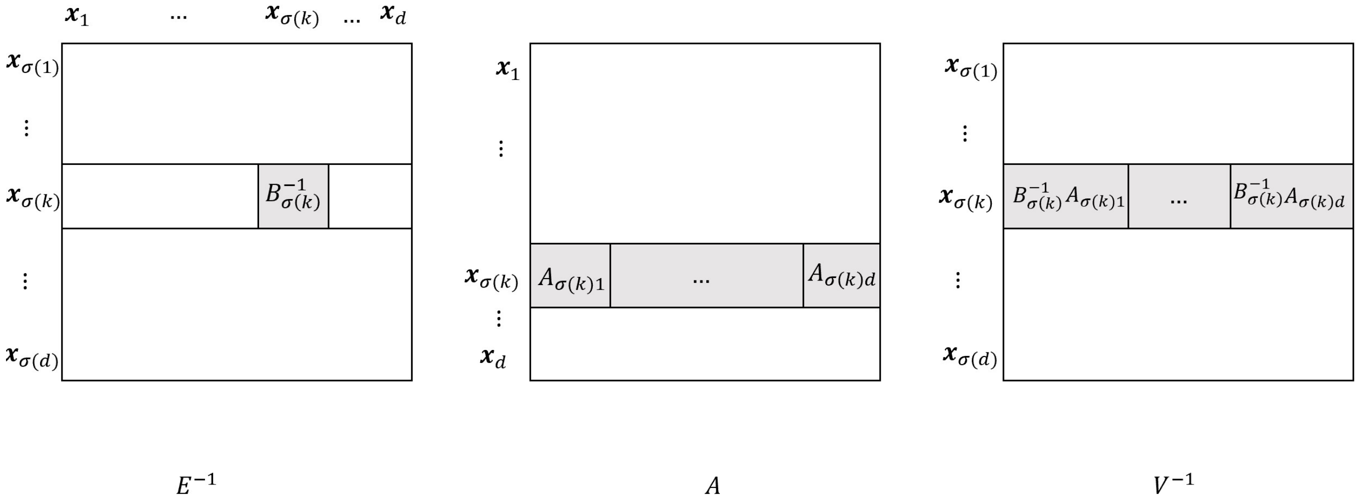


Figure 5.2: The matrix $V^{-1} = E^{-1}A$

□

Remark: (Uniqueness of the layer spaces) Recall from Chapter 4 that for a polynomial f equivalent to Tr-IMM, the irreducible invariant subspaces of $\mathcal{V}_1, \dots, \mathcal{V}_d$ of $\mathfrak{g}_f$ are unique. Correctness of Algorithm 4 implies that the layer spaces of f is also unique in the following sense: Suppose $f(\mathbf{x}) = \mathrm{Trace}(X_1 \cdot X_2 \dots X_d) = \mathrm{Trace}(X'_1 \cdot X'_2 \dots X'_d)$ where the entries of X_i and X'_i are linearly independent linear forms in the $\mathbf{x}$ variables. Let $\mathcal{X}_i$ and $\mathcal{X}'_i$ denote the layer spaces of X_i and X'_i respectively for each $i \in [d]$. Then $\{\mathcal{X}_1, \dots, \mathcal{X}_d\} = \{\mathcal{X}'_1, \dots, \mathcal{X}'_d\}$

5.1 Reordering the layer spaces

Given some permutation of the layer spaces $\mathcal{X}_{\sigma(1)},\ldots,\mathcal{X}_{\sigma(d)}$, we wish to recover σ. Claim 5.1 says that this can be done in randomized polynomial time. Before stating the claim, we define the notion of *evaluation dimension* which will turn out to be very useful in our proof. This definition is given in [10] and was also used in [20] to reorder the layer spaces of a polynomial equivalent to $\mathrm{IMM}_{w,d}$.

Definition 5.1 (Evaluation Dimension) *Let $f(\mathbf{x})$ be an n-variate polynomial and $\mathbf{x}' \subseteq \mathbf{x}$. Let $f(\mathbf{x})_{\mathbf{x}'=\alpha}$ denote the partial evaluation of f when $\mathbf{x}'$ is substituted with $\alpha \in \mathbb{F}^{|\mathbf{x}'|}$. The evaluation dimension of f with respect to $\mathbf{x}'$ is defined as:*

$$\mathrm{Evaldim}_{\mathbf{x}'}(f) := \dim(\mathrm{span}_{\mathbb{F}}(\{f(\mathbf{x})_{\mathbf{x}'=\alpha} : \alpha \in \mathbb{F}^{|\mathbf{x}'|}\})) \ .$$

The following example illustrates the above definition.

Example 5.1 *Let $f(x_1,x_2) = x_1 + 2x_1x_2$ be a polynomial over $\mathbb{Q}$. Substituting $x_2 = a$ for some $a \in \mathbb{Q}$, we have $f(\mathbf{x})_{\{x_2\}=\{a\}} = (2a+1)x_1$. Then $\mathrm{Evaldim}_{\{x_2\}}(f) = \dim(\mathrm{span}_{\mathbb{Q}}(\{(2a+1)x_1 : a \in \mathbb{Q}\})) = 1$.*

Claim 5.1 *There is a randomized polynomial time algorithm that takes as input the bases of the layer spaces $\mathcal{X}_{\sigma(1)},\ldots,\mathcal{X}_{\sigma(d)}$ and outputs σ' which is a rotation of the permutation σ with probability atleast $1 - \frac{1}{poly(n)}$.*

Proof: We first present the high level idea of the proof. We define a linear map μ that maps the $\mathbf{x}$ variables to a fresh set of $\mathbf{z}$ variables and hence maps the polynomial $f(\mathbf{x})$ to a polynomial $h(\mathbf{z})$ which can be represented as:

$$h(\mathbf{z}) = \mathrm{Trace}(Z_1 \cdot Z_2 \ldots Z_d)$$

such that $h(\mathbf{z})$ is a set-multilinear polynomial in the variable sets $\mathbf{z}_1,\ldots,\mathbf{z}_d$ where $\mathbf{z}_i$ denotes the set of variables appearing in the matrix Z_i for each $i \in [d]$. Further the entries of $Z_1,\ldots,Z_d$ are $\mathbb{F}$-linearly independent linear forms in the $\mathbf{z}$ variables. We now work with this new polynomial h to compute σ.

Step 1 Compute the linear map μ: Let $\mathbf{z}_k$ be a fresh set of $\dim(\mathcal{X})_{\sigma(k)} (= w^2)$ variables and $\mathbf{z} := \mathbf{z}_1 \uplus \ldots \uplus \mathbf{z}_d$. Let μ be a map that take each variable in $\mathbf{x}$ to a linear form in the $\mathbf{z}$ variables satisfying the following condition: The i-th basis vector of $\mathcal{X}_{\sigma(k)}$ maps to the i-th variable of $\mathbf{z}_k$. Replacing the variables in $f(\mathbf{x})$ by their image under the linear map μ, we obtain a new polynomial $h(\mathbf{z})$ which can be written as follows:

$$h(\mathbf{z}) = \mathrm{Trace}(Z_1 \cdot Z_2 \ldots Z_d)$$

where Z_i is a $w \times w$ matrix for $i \in [d]$. The entries of each Z_i for $i \in [d]$ are $\mathbb{F}$-linearly independent linear forms in Z. Further the linear forms in Z_k contains variables only from $\mathbf{z}_{\sigma^{-1}(k)}$.

Step 2 Compute σ^{-1} incrementally: Let $\mathbf{y}_j = \mathbf{z}_{\sigma^{-1}(1)} \uplus \ldots \uplus \mathbf{z}_{\sigma^{-1}(j)}$ for $j \in [d]$. The following observation (proved later) helps in computing σ^{-1} incrementally.

Observation 5.1 *Let $j \in [1, d-1]$, $k \in [2, d]$ and $\mathbf{z}_k \not\subset \mathbf{y}_j$. If $k = \sigma^{-1}(j+1)$ then* $\mathrm{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h) = w^2$*; else* $\mathrm{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h) > w^2$*. Further, there is an efficient randomized procedure to compute* $\mathrm{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h)$.

We can construct σ^{-1} incrementally using Observation 5.1 once we correctly identify $\mathbf{y}_1$. If we choose $\mathbf{y} = \mathbf{z}_{\sigma^{-1}(1)}$, Observation 5.1 yields σ^{-1}. Since $\mathrm{Trace}(Z_1 \cdot Z_2 \ldots Z_{n-1} \cdot Z_n) = \mathrm{Trace}(Z_n \cdot Z_1 \ldots Z_{n-2} \cdot Z_{n-1})$, Observation 5.1 can still be applied in the case when $\mathbf{y}_1$ is chosen to be $\mathbf{z}_{\sigma^{-1}(k)}$ for some $k \in [2, d]$. However, the resulting permutation σ' will be some rotation of the permutation σ. □

We now need to establish that $\mathrm{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h)$ can be computed efficiently (Observation 5.1). Fact 2.3 and Claim 2.2 from Chapter 2 will be used in the proof of Observation 5.1.

Proof of Observation 5.1 Recall that $h(z) = \mathrm{Trace}(Z_1 \ldots Z_d)$ and $|\mathbf{z}_k| = w^2$ for all $k \in [d]$. Let $\mathcal{V} := \mathrm{span}_{\mathbb{F}}(h(\mathbf{z})|_{\mathbf{y}_j \uplus \mathbf{z}_k = \alpha} : \alpha \in \mathbb{F}^{|\mathbf{y}_j \uplus \mathbf{z}_k|})$ denote the space spanned by polynomials obtained by the partial evaluation of h when $\mathbf{y}_j \uplus \mathbf{z}_k$ is substituted by some $\alpha \in \mathbb{F}^{|\mathbf{y}_j \uplus \mathbf{z}_k|}$. Then $\mathrm{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h) = \dim(\mathcal{V})$. We now compute $\dim(\mathcal{V})$ in both cases ($k = \sigma^{-1}(j+1)$ and $k \neq \sigma^{-1}(j+1)$).

Case 1: $\mathbf{k = \sigma^{-1}(j+1)}$

Let $G = (g_{ij})_{w\times w} = Z_{j+2}Z_{j+3}\dots Z_d$ and $G^{'} = (g^{'}_{ij})_{w\times w} = Z_1Z_2\dots Z_{j+1}$. Then

$$\begin{aligned} h(z) &= \text{Trace}(G^{'}G) \\ &= \sum_{i=1}^{w}\sum_{k=1}^{w} g^{'}_{ik}g_{ki} \end{aligned}$$

$h(z)$ will evaluate to g_{ij} when the following system of equations is satisfied:

$$\begin{aligned} &g^{'}_{pq} = 0 \text{ for all } p \neq j \text{ and } p \in [w] \\ &g^{'}_{ji} = 1 \\ &g^{'}_{jq} = 0 \text{ for all } q \neq i \text{ and } q \in [w] \end{aligned}$$

Since the linear forms in $Z_1, \dots, Z_d$ are linearly independent, there exists an assignment to the variables in $\mathbf{y}_j \uplus \mathbf{z}_{j+1}$ such that the above system is consistent. Further every partial evaluation of h at $\mathbf{y}_j \uplus \mathbf{z}_{j+1}$ can be expressed as $\mathbb{F}$-linear combination of $\{g_{ij} : i, j \in [w]\}$. Hence $\{g_{ij} : i, j \in [w]\}$ spans the space $\mathcal{V}$. From Fact 2.3 it follows that $\{g_{ij} | i, j \in [w]\}$ are linearly independent. Hence $\text{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h) = w^2 = |\mathbf{z}_k|$.

Case 2: $\mathbf{k \neq \sigma^{-1}(j+1)}$

We rewrite $h(\mathbf{z})$ in the following form:

$$\begin{aligned} h(\mathbf{z}) &= \text{Trace}(\underbrace{Z_1 \dots Z_j}_{G'=(g'_{ij})_{w\times w}} \cdot \underbrace{Z_{j+1} \dots Z_{\sigma(k-1)}}_{P=(p_{ij})_{w\times w}} \cdot \underbrace{Z_{\sigma(k)}}_{Z=(z_{ij})_{w\times w}} \cdot \underbrace{Z_{\sigma(k+1)} \dots Z_d}_{G=(g_{ij})_{w\times w}}) \\ &= \sum_{i\in[w]} \sum_{k,l.m\in[w]} g'_{ik}p_{km}z_{ml}g_{li} \end{aligned}$$

Similar to the reasoning in Case 1, there exists an assignment to $\mathbf{y}_j \uplus \mathbf{z}_k$ such that $h(\mathbf{z})$ evaluates to $p_{ij}g_{mn}$ for all $i, j, m, n \in [w]$ and every partial evaluation of h at $\mathbf{y}_j \uplus \mathbf{z}_k$ can be expressed as $\mathbb{F}$-linear combination of $\{p_{ij}g_{mn} : i, j, m, n \in [w]\}$. Hence $\{p_{ij}g_{mn} : i, j, m, n \in [w]\}$ spans the space $\mathcal{V}$. From Fact 2.3 it follows that $\{p_{ij} | i, j \in [w]\}$, $\{g_{mn} | m, n \in [w]\}$ and hence $\{p_{ij}g_{mn} : i, j, m, n \in [w]\}$ are linearly independent. Therefore, $\text{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h) = w^4 > |\mathbf{z}_k|$.

An efficient randomized procedure to compute $\mathrm{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h)$ Let $S^{|\mathbf{y}_j \uplus \mathbf{z}_k|} \subset \mathbb{F}^{|\mathbf{y}_j \uplus \mathbf{z}_k|}$ and $|S| = poly(n)$. Choose points $\mathbf{a}_1, \ldots, \mathbf{a}_{n^2}$ from $S^{|\mathbf{y}_j \uplus \mathbf{z}_k|}$ independently and uniformly at random and output the dimension of the space spanned by $h(\mathbf{a}_1), \ldots, h(\mathbf{a}_{n^2})$. Let $\mathrm{Evaldim}_{\mathbf{y}_j \uplus \mathbf{z}_k}(h) = e$. Observe that $h(\mathbf{z})$ can be expressed as follows:

$$h(\mathbf{z}) = \sum_{i \in [e]} f_i(\mathbf{y}_j \uplus \mathbf{z}_k) \cdot q_i \tag{5.1}$$

where f_i and q_i are variable disjoint and the $q_1, \ldots, q_e$ form a basis of the space $\mathcal{V}$ (which was defined in the proof of Observation 5.1). Further $\{f_i | i \in [e]\}$, $\{q_i | i \in [e]\}$ are linearly independent sets of polynomials. Hence, to show that $h(\mathbf{a}_1), \ldots, h(\mathbf{a}_e)$ is linearly independent, it is sufficient to show that the following matrix is full rank:

$$M := \begin{bmatrix} f_1(\mathbf{a}_1) & \ldots & f_e(\mathbf{a}_1) \\ \vdots & \vdots & \vdots \\ f_1(\mathbf{a}_e) & \ldots & f_e(\mathbf{a}_e) \end{bmatrix}_{e \times e}$$

Claim 2.2 implies that the matrix M is full rank with high probability. $h(\mathbf{a}_1), \ldots, h(\mathbf{a}_e)$ is linearly independent implies that the dimension of the space spanned by the polynomials $h(\mathbf{a}_1), \ldots, h(\mathbf{a}_{n^2})$ is at least e. Equation 5.1 implies that the dimension of the space spanned by the polynomials $h(\mathbf{a}_1), \ldots, h(\mathbf{a}_{n^2})$ is at most e. Hence, with high probability $\dim(\mathrm{span}_{\mathbb{F}}(h(\mathbf{a}_1), \ldots, h(\mathbf{a}_{n^2}))) = e$.

5.2 Reduction to Block Equivalence testing

Now that we have the basis of the layer spaces in correct order, we show that computing the matrix A for which $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$ reduces to Block Equivalence testing (defined in Chapter 6.

Claim 5.2 *Given the basis of the spaces $\mathcal{X}_1, \ldots, \mathcal{X}_d$ in order , the width w and an efficient algorithm for Block Equivalence testing for* Tr-IMM$_{w,d}$, *we can find an invertible $n \times n$ matrix A in polynomial time such that* $\text{Tr-IMM}_{w,d}(\mathbf{x}) = f(A\mathbf{x})$

Proof: Compute a linear map $\mathbf{x} \mapsto \hat{A}\mathbf{x}$ such that the basis vectors of $\mathcal{X}_k$ maps to distinct variable in $\mathbf{x}_k$. This can be computed efficiently. Let $h(\mathbf{x}) = f(\hat{A}\mathbf{x})$. Then $h(\mathbf{x}) = \mathrm{Trace}(X_1 \cdot X_2 \ldots X_d)$, and the entries of X_i are linear forms defined by the linear map $\mathbf{x} \mapsto \hat{A}\mathbf{x}$. Since we mapped each basis vector of $\mathcal{X}_k$ to distinct variables, $\mathbf{x}_i$ and $\mathbf{x}_j$ are disjoint whenever $i \neq j$ and $i, j \in [d]$, i.e. $h(\mathbf{x})$ is a sum of set-multilinear ABPs. As $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$, we infer that $h(\mathbf{x}) = \text{Tr-IMM}_{w,d}(V\mathbf{x})$ where $V = A \cdot \hat{A}$ is a block-diagonal matrix. We compute V using Algorithm 6 and obtain A using $A = V\hat{A}^{-1}$. □

In the proof of Claim 5.2, we saw that it is sufficient to compute a V such that $h(\mathbf{x}) = \text{Tr-IMM}_{w,d}(V\mathbf{x})$ where V is a block-diagonal matrix. We call this problem as *Block Equivalence Testing*. In the next chapter, we will discuss about *Block Equivalence Testing* in more detail.

Chapter 6

Block Equivalence Testing

> In this chapter we formally define the Block Equivalence Testing problem for the Tr-$\mathrm{IMM}_{w,d}$ polynomial and show that it reduces to Equivalence Testing for the DET_n polynomial.

In Chapter 5 we saw that the Equivalence testing problem for Tr-$\mathrm{IMM}_{w,d}$ polynomial [1] reduces to the Block Equivalence testing problem which we precisely state below.

Block Equivalence testing for Tr-$\mathrm{IMM}_{w,d}$ **polynomial:** Given black-box access to an n variate, degree d polynomial f check if there is an invertible *block-diagonal* matrix $V \in \mathbb{F}^{n\times n}$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(V\mathbf{x})$. If such a V exists, then output V; else output *"no such V exists"*.

An alternate equivalent restatement of the above problem is as follows :

Block Equivalence testing for Tr-$\mathrm{IMM}_{w,d}$ **polynomial: (Alternate version)** Given black-box access to an n variate, degree d polynomial f, check if there are matrices $X_1,\ldots,X_d$ such that $f(\mathbf{x}) = \mathrm{Trace}(X_1\cdot X_2\ldots X_d)$ where X_i is a $w\times w$ matrix whose entries are linearly independent linear forms in the variable set $\mathbf{x}_i$ for each $i\in[d]$. Further the variable sets are disjoint, i.e. $\mathbf{x}_i\cap\mathbf{x}_j=\emptyset$ whenever $i\neq j$. Output $X_1,\ldots,X_d$ if such matrices exists; else output *"no such matrices exist"*.

Recall the definition of set-multilinear ABP [2] from Definition 2.22. The following claim, which we

[1] where $w\geq 2$ and $d>2$

[2] The size of the ABP is the sum of widths of each layer. In our work, we consider ABPs having the same width w' at each layer.

state without proof (see [22]) gives us the width of smallest set-multilinear ABP computing any polynomial.

Claim 6.1 (Smallest width set-multilinear ABP computing f) *Let $f(\mathbf{x})$ be an n-variate, degree d polynomial which is set-multilinear in the variable sets $\mathbf{x}_1,\dots,\mathbf{x}_d$. Further, let w' be the width of the smallest size uniform width set-multilinear ABP with the variable ordering $\mathbf{x}_1,\dots,\mathbf{x}_d$ computing f. Then* $\mathrm{Evaldim}_{\mathbf{y}_i}(f) = w'$ *where $\mathbf{y}_i = \mathbf{x}_1 \uplus \dots \uplus \mathbf{x}_i$ for each $i \in [d]$.*

Recall from the proof of Claim 5.1 that if f is equivalent to Tr-$\mathrm{IMM}_{w,d}$ then $\mathrm{Evaldim}_{\mathbf{y_i}}(f(\mathbf{x})) = w^2$ where $\mathbf{y}_i = \mathbf{x}_1 \uplus \dots \uplus \mathbf{x}_i$ for all $i \in [d]$. Hence the smallest width uniform width set-multilinear ABP computing f has width w^2. The following observation which can be easily verified specifies a width w^2 set-multilinear ABP that computes f.

Observation 6.1 (A width w^2 ABP computing f) *Let $X_1,\dots,X_d$ be $w \times w$ be full rank linear matrices in the variable sets $\mathbf{x}_1,\dots,\mathbf{x}_d$ respectively. Consider a a set-multilinear polynomial $f = \mathrm{Trace}(X_1 \cdot X_2 \dots X_d)$. Then there exists a set-multilinear ABP A of width w^2 computing f which is given by $A = Y_1 \cdot Y_2 \dots Y_d$ with $Y_1 = [X_{11},\dots,X_{1w}]$, $Y_i = I_w \otimes X_i$ for all $i \in [2, d-1]$ and $X_d = [X_{d1}^T,\dots,X_{dw}^T]^T$ where X_{1i}, X_{dj} denotes the i-th row and j-th column of X_1 and X_d respectively, i.e.*

$$Y_1 = [X_{11},\dots,X_{1w}]_{1\times w^2}\ ; \quad Y_d = \begin{bmatrix} X_{d1} \\ \vdots \\ X_{dw} \end{bmatrix}_{w^2 \times 1}$$

and for $i \in [2, d-1]$,

$$X_i = \begin{bmatrix} x_{11}^{(i)} & \dots & x_{1w}^{(i)} \\ \vdots & & \vdots \\ x_{w1}^{(i)} & \dots & x_{ww}^{(i)} \end{bmatrix}_{w\times w} ; Y_i = \begin{bmatrix} X_i & & \\ & \ddots & \\ & & X_i \end{bmatrix}_{w^2 \times w^2}$$

In the remainder of this chapter we will use X_i and Y_i for all $i \in [d]$ to denote the matrices as defined in Observation 6.1.

6.1 An efficient algorithm for Block Equivalence testing

In [13], they give an efficient algorithm for Block Equivalence testing for Tr-IMM over the field of complex numbers $\mathbb{C}$ (an algebraically closed field). We now present an efficient randomized algorithm (Algorithm 6) for Block Equivalence testing for Tr-IMM which uses oracle access to DETEQ. Recall from Chapter 1 that DETEQ is an algorithm for determinant equivalence test.

Theorem 6.1 *Let f be an n-variate, degree d polynomial which is set-multilinear in the variable sets $\mathbf{x}_1, \ldots, \mathbf{x}_d$. Given blackbox access to f, the variable sets $\mathbf{x}_1, \ldots, \mathbf{x}_d$ and an oracle access to* DETEQ*, there is a randomized algorithm (Algorithm 6) with running time $poly(n, \beta)$ (where β is the bit length of coefficients of f) that outputs with probability at least $1 - \frac{1}{poly(n)}$ a block-diagonal matrix $V \in \mathsf{GL}(n, \mathbb{F})$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(V\mathbf{x})$ if such an V exists; otherwise it outputs "No such V exists".*

We now elaborate each step of Algorithm 6 and argue its correctness.

Step 1: Set Multilinear ABP computing f: Without loss of generality, assume that f is block-equivalent to $\text{Tr-IMM}_{w,d}$ polynomial (If it were not, we could test this at the end by evaluating at random points using Schwartz-Zippel lemma). Hence, f is computable by the ABP $Y_1 \cdot Y_2 \ldots Y_d$ as given by Observation 6.1. Further this is the smallest width ABP computing f. Hence, using the set-multilinear ABP reconstruction algorithm (See Algorithmic Preliminary 6), we can compute an ABP $A = \widehat{Y_1} \cdot \widehat{Y_2} \ldots \widehat{Y_{d-1}} \cdot \widehat{Y_d}$ where,

$$\begin{aligned}
\widehat{Y_1} &= Y_1 \cdot T_1 \\
\widehat{Y_i} &= T_{i-1}^{-1} \cdot Y_i \cdot T_i, \forall i \in [2, d-1] \\
\widehat{Y_d} &= T_{d-1}^{-1} \cdot Y_d
\end{aligned} \tag{6.1}$$

and $T_1, \ldots, T_{d-1}$ are $w^2 \times w^2$ invertible matrices. Observe that if the ABP computed in Step 1 is exactly $Y_1 \cdot Y_2 \ldots Y_{d-1} \cdot Y_d$ then we can recover $X_1, \ldots, X_d$ trivially and we are done. However we have $\widehat{Y_i}$ instead of Y_i at the end of Step 1. In the next step, we exploit this relation between X_i and $\widehat{Y_i}$ for all $i \in [2, d-1]$.

Algorithm 5 Block Equivalence testing for Tr-IMM$_{w,d}$

INPUT: Blackbox access to an n variate, degree d polynomial f and the variable sets $\mathbf{x}_1,\ldots,\mathbf{x}_d$ such that f is a set-multilinear polynomial in these variable sets.
OUTPUT: A block-diagonal matrix $V \in \mathsf{GL}(n,\mathbb{F})$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(V\mathbf{x})$ (if such a V exists).

```
▷ Step 1 - Set Multilinear ABP reconstruction of $f$
Compute a width $w^2$ set-multilinear ABP $A = \widehat{Y_1} \cdot \widehat{Y_2} \ldots \widehat{Y_{d-1}}\widehat{Y_d}$ computing $f$.

▷ Step 2 - Computing blackbox access to $c_i \cdot \det(X_i)$
for all $i \in [2, d-1]$ do
    Compute $\det(\widehat{Y_i})$.
    Factorize $\det(\widehat{Y_i})$ to obtain access to $c_i \cdot \det(X_i)$ where $c_i \in \mathbb{F}$.

▷ Step 3 - Try to obtain $X_i$ from $c_i \cdot \det(X_i)$ using Determinant Equivalence test
for all $i \in [2, d-1]$ do
    Using Determinant Equivalence Test, compute a $w \times w$ matrix $\widehat{X_i}$ such that $c_i \cdot \det(X_i) = \det(\widehat{X_i})$.

▷ Step 4 - Construct $Z_i$
for all $i \in [2, d-1]$ do
    Compute the $w \times w$ matrix $Z_i = I_w \otimes \widehat{X_k}$.

▷ Distinguish between the matrix and its transpose
for all $i \in [2, d-1]$ do
    if there are $w^2 \times w^2$ matrices $\widehat{T_{i-1}}$ and $\widehat{S_i}$ such that $\widehat{T_{i-1}} \cdot \widehat{Y_i} = Z_i \cdot \widehat{S_i}$ then
        Compute such $\widehat{T_{i-1}}$ and $\widehat{S_i}$.
    else
        Compute $\widehat{T_{i-1}}$ and $\widehat{S_i}$ such that $\widehat{T_{i-1}} \cdot \widehat{Y_i} = Z_i^T \cdot \widehat{S_i}$.

▷ Step 6 - Block Diagonalize $\widehat{Y_{d-1}}$
Compute $w^2 \times w^2$ matrices $Y_1', \ldots, Y_d'$ where
    $Y_1' := \widehat{Y_1} \cdot \widehat{T_1^{-1}}$
    $Y_i' := \widehat{T_{i-1}} \cdot \widehat{Y_i} \cdot \widehat{T_i^{-1}}; \forall i \in [2, d-2]$
    $Y_{d-1}' := \widehat{T_{d-2}} \cdot \widehat{Y_{d-1}} \cdot \widehat{S_{d-1}^{-1}}$
    $Y_d' := \widehat{S_{d-1}^{-1}} \cdot \widehat{Y_d}$ .
```

Algorithm 6 Block Equivalence testing for Tr-IMM$_{w,d}$ (continued...)

```

▹ Step 7 - Compute $X'_2, \dots, X'_{d-1}$
Compute $X'_{d-1}$ := first $w \times w$ block of the block-diagonal matrix $Y'_{d-1}$
for all $i \in [2, d-2]$ do
    Let $\alpha_i$ be some non-zero entry (say $(p,q)$-th entry) of $D_i D_{i+1}^{-1}$ where $D_i$ and $D_{i+1}$ are $w \times w$
    matrices as defined in Claim 6.3. Then compute $X'_i := (p,q)$-th block of $Y'_i$.

▹ Step 8 - Compute $X'_1$ and $X'_d$
Compute $X'_d$ using the proof of Claim 6.4.
Compute $X'_1$ using the proof of Claim 6.5.

▹ at this point Trace$(X'_1 \cdot X'_2 \dots X'_d) = f$ and
▹ the matrices $X'_1, \dots, X'_d$ are variable disjoint.
▹ Step 9 - Compute $V$
Compute the matrix $V$ by looking at the coefficients of the linear forms in the entries of $X'_i$ for
all $i \in [d]$.

▹ Step 10 - Output the result
Pick a random point $\mathbf{a} \in S^n$ where $S \subseteq \mathbb{F}$ and $|S| \geq$ poly$(n)$.
if $f(\mathbf{a}) =$ Tr-IMM$_{w,d}(V\mathbf{a})$ then
    Output $V$.
else
    Output 'No such $V$ exists'.
```

Step 2 - Computing blackbox access to $c_i \cdot \det(X_i)$: In this step, we compute blackbox access to $c_i \cdot \det(X_i)$ from $\widehat{Y_i}$ for all $i \in [2, d-1]$ as follows: First we compute $\det(\widehat{Y_i})$. This can be computed efficiently [8]. Since $Y_i = I_w \otimes X_i$ and $\widehat{Y_i} = T_{i-1}^{-1} \cdot Y_i \cdot T_i$, we have $\det(\widehat{Y_i}) = d_i \cdot \det(X_i)^w$ where $d_i = \det(T_{i-1}^{-1}) \cdot \det(T_i)$. By applying am efficient polynomial factorization algorithm [17] we can obtain blackbox access to $c_i \cdot \det(X_i)$ for some $c_i \in \mathbb{F}$.

Step 3 - *Try* to obtain X_i from $c_i \cdot \det(X_i)$ using Determinant Equivalence Test: From Step 2, we have blackbox access to a polynomial computing $c_i \cdot \det(X_i)$ for all $i \in [2, d-1]$. Let $\widehat{X_i}$ be a $w \times w$ matrix such that $\det(\widehat{X_i}) = c_i \cdot \det(X_i)$. Then *exactly* one of the following holds (due to Fact 2.1).

$$\begin{aligned}
&\exists A_i, B_i \text{ such that } X_i = A_i \cdot \widehat{X_i} \cdot B_i \ (\textbf{Case 3-1}) \\
&\exists \tilde{A}_i, \tilde{B}_i \text{ such that } X_i = \tilde{A}_i \cdot \widehat{X_i}^T \cdot \tilde{B}_i \ (\textbf{Case 3-2}) \ .
\end{aligned}$$

where $A_i, B_i, \tilde{A}_i, \tilde{B}_i$ are $w \times w$ invertible numeric matrices with $\det(A_i B_i) = c_i$ (for Case 3-1) or $\det(\tilde{A}_i \tilde{B}_i) = c_i$ (for Case 3-2). Using equivalence test for the determinant polynomial ([18, 11]) we can compute $\widehat{X_i}$ satisfying exactly one of the above cases for each $i \in [2, d-1]$.

Step 4 - Construct Z_i: In Step 4 we compute the $w^2 \times w^2$ block-diagonal matrix Z_i with $\widehat{X_i}$ along its diagonal blocks.

Step 5 - Find out which case in Step 3 succeeded: For some $i \in [2, d-1]$, suppose Case 3-1 is true at Step 3, i.e $X_i = A_i \cdot \widehat{X_i} \cdot B_i$ for some invertible $w \times w$ numeric matrices. Then,

$$\begin{aligned}
\widehat{Y_i} &= T_{i-1}^{-1} \cdot Y_i \cdot T_i \\
&= T_{i-1}^{-1} \cdot (I_w \otimes X_i) \cdot T_i \\
&= T_{i-1}^{-1} \cdot \left(I_w \otimes (A_i \cdot \widehat{X_i} \cdot B_i)\right) \cdot T_i \\
&= T_{i-1}^{-1} \cdot ((I_w \otimes A_i) \cdot (I_w \otimes \widehat{X_i}) \cdot (I_w \otimes B_i)) \cdot T_i \\
&= \underbrace{T_{i-1}^{-1} \cdot (I_w \otimes A_i)}_{\widehat{T}_{i-1}^{-1}} \cdot Z_i \cdot \underbrace{(I_w \otimes B_i) \cdot T_i}_{\widehat{S}_i} \ .
\end{aligned}$$

Similarly if Case 3-2 succeeds then $\widehat{Y_i} = \widehat{T_{i-1}^{-1}} \cdot Z_i^T \cdot \widehat{S_i}$. Claim 6.2 says that exactly one of these equations is satisfied. To compute these matrices, treat the entries of $\widehat{T_{i-1}}$ and $\widehat{S_i}$ as formal variables and compare the entries of $\widehat{Y_i} \cdot \widehat{T_{i-1}}$ with $Z_i \cdot \widehat{S_i}$ or $Z_i^T \cdot \widehat{S_i}$. This will give us a system of linear equations in the variables of $\widehat{T_{i-1}}$ and $\widehat{S_i}$ which can be solved using guassian elimination in poly(w^2) time.

Claim 6.2 *In Step 5 of Algorithm 6 exactly one of the cases succeeds.*

Proof: Recall that $\forall i \in [2, d-1]$:

$$\begin{aligned} Z_i &= I_w \otimes \widehat{X_i} \\ \widehat{Y_i} &= T_{i-1}^{-1} \cdot Y_i \cdot T_i \\ Y_i &= I_w \otimes X_i \end{aligned}$$

where X_i is a $w \times w$ symbolic matrix whose entries are linearly independent linear forms. In Step 3, we computed $w \times w$ matrix $\widehat{X_i}$ which satisfies exactly one of the following equations: $X_i = A_i \cdot \widehat{X_i} \cdot B_i$ or $X_i = \tilde{A}_i \cdot \widehat{X_i}^T \cdot \tilde{B}_i$ for some invertible $w \times w$ numeric matrices $A_i, B_i, \tilde{A}_i, \tilde{B}_i$. Accordingly we have the following two cases:

$$\begin{aligned} Y_i &= I_w \otimes A_i \widehat{X_i} B_i \\ \widehat{Y_i} &= T_{i-1}^{-1} \cdot (I_w \otimes A_i \widehat{X_i} B_i) \cdot T_i \\ \widehat{Y_i} &= \underbrace{T_{i-1}^{-1} \cdot (I_w \otimes A_i) Z_i (I_w \otimes B_i) \cdot T_i}_{(a)} \end{aligned} \quad \text{(or)} \quad \begin{aligned} Y_i &= I_w \otimes \tilde{A}_i \widehat{X_i}^T \tilde{B}_i \\ \widehat{Y_i} &= T_{i-1}^{-1} \cdot (I_w \otimes \tilde{A}_i \widehat{X_i}^T \tilde{B}_i) \cdot T_i \\ \widehat{Y_i} &= \underbrace{T_{i-1}^{-1} \cdot (I_w \otimes \tilde{A}_i) Z_i^T (I_w \otimes \tilde{B}_i) \cdot T_i}_{(b)} \end{aligned}$$

We claim that both (a) and (b) can not hold simultaneously. Suppose for contradiction let (a) and (b) be true simultaneously, which implies there exists $w^2 \times w^2$ invertible matrices P, Q such that $P Z_i Q = Z_i^T$. But this contradicts Claim 6.6 (given at the end of the chapter). Hence exactly one of the cases in Step 5 of Algorithm 6 succeeds. □

Remark: From this point in the algorithm, for the sake of brevity we assume that $\widehat{X_i}$ was computed according to Case 3-1. A similar analysis holds if Case 3-2 was true.

The following claim says that $\widehat{T_{i-1}}, \widehat{S_i}$ computed in Step 5 are unique up to multiplication by certain matrices. The proof uses Claim 6.7 which is proved at the end of the chapter.

Claim 6.3 (Uniqueness of $\widehat{T_{i-1}}$ and $\widehat{S_i}$) *For any $i \in [2, d-1]$, the $\widehat{T_{i-1}}, \widehat{S_i}$ computed in Step 5 of Algorithm 6 is of the following form:*

$$\widehat{T_{i-1}^{-1}} = T_{i-1}^{-1} \cdot (I_w \otimes A_i) \cdot (D_i^{-1} \otimes I_w)$$
$$\widehat{S_i} = (D_i \otimes I_w) \cdot (I_w \otimes B_i) \cdot T_i \ \ .$$

where D_i is a $w \times w$ invertible matrix.

Proof: The $\widehat{T_i}, \widehat{S_i}$ computed in Step 5 satisfies

$$\widehat{T_i} \cdot \widehat{Y_i} = Z_i \cdot \widehat{S_i}$$

Substituting $\widehat{Y_i} = T_{i-1}^{-1} \cdot (I_w \otimes A_i) \cdot Z_i \cdot (I_w \otimes B_i) \cdot T_i$ in the above equation, we get

$$\widehat{T_{i-1}} \cdot T_{i-1}^{-1} \cdot (I_w \otimes A_i) \cdot Z_i = Z_i \cdot \widehat{S_i} \cdot \widehat{T_i^{-1}} \cdot (I_w \otimes B_i^{-1})$$

By Claim 6.7, there exists $w \times w$ invertible matrix D_i such that

$$\widehat{T_{i-1}} \cdot T_{i-1}^{-1} \cdot (I_w \otimes A_i) = \widehat{S_i} \cdot \widehat{T_i} \cdot (I_w \otimes B_i^{-1}) = D_i \otimes I_w$$

implying,

$$\widehat{T_{i-1}^{-1}} = T_{i-1}^{-1} \cdot (I_w \otimes A_i) \cdot (D_i^{-1} \otimes I_w)$$
$$\widehat{S_i} = (D_i \otimes I_w) \cdot (I_w \otimes B_i) \cdot T_i \ \ .$$

□

Step 6 - Block Diagonalize $\widehat{Y_{d-1}}$: We compute the matrices $Y_1', \ldots, Y_d'$ from $\widehat{Y_1}, \ldots, \widehat{Y_d}$ in step 6. Note that the ABP $Y_1' \cdot Y_2' \ldots Y_d'$ still computes the original polynomial f as the intermediate matrix multiplications cancels out. We also observe that the matrix Y_{d-1}' is a block-diagonal matrix.

Observation 6.2 *Y'_{d-1} computed in Step 6 of Algorithm 6 is a block-diagonal matrix.*

Proof: The Y'_{d-1} computed in Step 6 is given by:

$$Y'_{d-1} = \widehat{T}_{d-2} \cdot \widehat{Y}_{d-1} \cdot \widehat{S}_{d-1}^{-1}$$

Substituting $\widehat{Y}_{d-1} = T_{d-2}^{-1} \cdot (I_w \otimes X_{d-1}) \cdot T_{d-1}$, $\widehat{T}_{d-2} = (D_{d-1} \otimes I_w) \cdot (I_w \otimes A_{d-1}^{-1}) \cdot (T_{d-2})$ and $\widehat{S}_{d-1}^{-1} = (T_{d-1}^{-1}) \cdot (I_w \otimes B_{d-1}^{-1}) \cdot (D_{d-1}^{-1} \otimes I_w)$ in the above equation, we have:

$$\begin{aligned} Y'_{d-1} &= (D_{d-1} \otimes I_w) \cdot \Big(I_w \otimes (A_{d-1}^{-1} X_{d-1} B_{d-1}^{-1})\Big) \cdot (D_{d-1}^{-1} \otimes I_w) \\ &= (D_{d-1} D_{d-1}^{-1} \otimes I_w)(I_w \otimes A_{d-1}^{-1} \cdot X_{d-1} \cdot B_{d-1}^{-1}) \\ &= I_w \otimes (A_{d-1}^{-1} \cdot X_{d-1} \cdot B_{d-1}^{-1}) \ . \end{aligned}$$

Clearly Y'_{d-1} is a block-diagonal matrix whose diagonal blocks are $A_{d-1}^{-1} \cdot X_{d-1} \cdot B_{d-1}^{-1}$. □

Now, we describe the structure of the matrices $Y'_2, \ldots, Y'_{d-2}$.

Observation 6.3 *The matrices Y'_i computed in Step 6 of Algorithm 6 can be written as $Y'_i = (D_i D_{i+1}^{-1} \otimes I_w) \cdot (I_w \otimes A_i^{-1} X_i A_{i+1})$ where $i \in [2, d-2]$ and D_i is a $w \times w$ matrix as defined in Claim 6.1.*

Proof: For any $i \in [2, d-2]$,

$$Y'_i = \widehat{T}_{i-1} \cdot \widehat{Y}_i \cdot \widehat{T}_i^{-1}$$

Substituting the appropriate expansions of $\widehat{T}_{i-1}$, $\widehat{Y}_i$ and $\widehat{T}_i^{-1}$ in the above equation, we have:

$$\begin{aligned} Y'_i &= \Big((D_i \otimes I_w)(I_w \otimes A_i^{-1}) T_{i-1}\Big) \cdot \Big(T_{i-1}^{-1}(I_w \otimes X_i) T_i\Big) \cdot \Big(T_i^{-1}(I_w \otimes A_{i+1})(D_{i+1}^{-1} \otimes I_w)\Big) \\ &= (D_i D_{i+1}^{-1} \otimes I_w) \cdot (I_w \otimes A_i^{-1} X_i A_{i+1}) \end{aligned}$$

□

In Steps 7-8, we compute $w \times w$ matrices $X'_1, \ldots, X'_d$ such that $f = \text{Trace}(X'_1 \ldots X'_d)$.

Step 7 - Computing $X_2',\ldots,X_{d-1}'$: In Step 7, we compute the $w \times w$ matrices $X_2',\ldots,X_{d-1}'$ by defining them to be suitable sub-matrices of the $w^2 \times w^2$ matrices $Y_1',\ldots,Y_{d-1}'$. Let α_i be some non-zero entry of $D_iD_{i+1}^{-1}$. Then the corresponding $w \times w$ block in Y_i' is $\alpha_k \cdot A_i^{-1}X_iA_{i+1}$. In Step 7, we define X_i' to be the block $\alpha_i \cdot A_i^{-1}X_iA_{i+1}$ of Y_i' for all $i \in [2, d-2]$ and X_{d-1}' to be the first $w \times w$ block of Y_{d-1}'. Hence the product $X_2' \cdot X_3' \ldots X_{d-1}'$ is given by:

$$X_2' \cdot X_3' \ldots X_{d-1}' = \alpha \cdot A_2^{-1}X_2 \cdot X_3 \ldots X_{d-1} \cdot B_{d-1}^{-1}$$

where $\alpha = \alpha_2 \cdot \alpha_3 \ldots \alpha_{d-2}$.

Step 8 - Computing X_1' and X_d':

Claim 6.4 (Computing X_d') *Given the matrices $X_2',\ldots,X_{d-1}'$ computed at Step 7 of Algorithm 6, we can efficiently compute the $w \times w$ matrix $X_d' = B_{d-1}X_dA$ where A is a $w \times w$ numeric matrix which is full rank with high probability.*

Proof: Define $\overline{X_1} := \alpha^{-1}X_1A_2$ and $\overline{X_d} := B_{d-1}X_d$. We first make the following observation.

Observation 6.4 *For any $i \in [w]$, the $(i,1)$-th entry of the product $\overline{X_d} \cdot \overline{X_1}$ can be computed efficiently.*

Proof: Fix some $i \in [w]$. Choose an assignment of the variables $\mathbf{x}_2,\ldots,\mathbf{x}_{d-1}$ that sets the $(1,i)$-th entry of the product $X_2'X_3'\ldots X_{d-1}'$ to 1 and the remaining entries to 0. It can be easily verified that, under this assignment the function f (if it was block-equivalent to Tr-IMM to begin with) will compute the $(i,1)$-th entry of $\overline{X_d} \cdot \overline{X_1}$. The assignment discussed above exists and can be computed as follows: Similar to the graph $\mathrm{G}_{\text{Tr-IMM}}$ we defined for the Tr-IMM polynomial, we associate a layered graph G corresponding to the matrices $X_2',\ldots,X_{d-1}'$. Then the $(1,i)$-th entry of the product $X_2'X_3'\ldots X_{d-1}'$ is the sum of all path weights from the 1st vertex u of first layer of G (corresponding to the matrix X_2') to the i-th vertex v of last layer (corresponding to the matrix X_{d-1}'). The $(1,i)$-th entry of the product is 1 only if all the edge labels that lie on any path from u to v is non-zero. Since the linear forms in X_i' is linear independent for each $i \in [w]$, we can apply an invertible linear transformation that maps the entries of X_i' to distinct formal variables. We can now trivially compute an assignment that sets the $(1,i)$-th entry of the product to a suitable non-zero number and the remaining entries to 0. □

Denote the first column of $\overline{X_1}$ by $\mathbf{c}_1 = [g_1(\mathbf{x}_1),\ldots,g_w(\mathbf{x}_1)]^T$ and the first column of $\overline{X_d}\cdot\overline{X_1}$ by $\mathbf{c}_d = [f_1(\mathbf{x}_1,\mathbf{x}_d),\ldots,f_w(\mathbf{x}_1,\mathbf{x}_d)]^T$. Clearly $\mathbf{c}_d = \overline{X_d}\mathbf{c}_1$. Choose w points $\mathbf{a}_1,\ldots,\mathbf{a}_w \in_r S^{w^2}$ uniformly at random where $S \subset \mathbb{F}$ and $|S| \geq \text{poly}(w^2)$. For each $i \in [w]$, compute $\mathbf{c}_{di} := [f_1(\mathbf{a}_i,\mathbf{x}_d),\ldots,f_w(\mathbf{a}_i,\mathbf{x}_d)]^T = \overline{X_d}\mathbf{c}_1(\mathbf{a}_i)$ where $\mathbf{c}_1(\mathbf{a}_i)$ denotes the evaluation of column vector $\mathbf{c}_1$ at point $\mathbf{a}_i$. Define X'_d to be the $w\times w$ matrix obtained by stacking these column vectors, i.e. $X'_d = [\mathbf{c}_{d1},\mathbf{c}_{d2},\ldots,\mathbf{c}_{dw}]$. Clearly, $X'_d = \overline{X_d}A = B_{d-1}X_dA$ where A is the $w\times w$ matrix $[\mathbf{c}_1(\mathbf{a}_1),\mathbf{c}_1(\mathbf{a}_2),\ldots,\mathbf{c}_1(\mathbf{a}_w)]$. As the linear forms in $\overline{X_1}$ are linearly independent, by Claim 2.2 the matrix A is full-rank with high probability. □

Claim 6.5 (Computing X'_1) *Given the matrices $X'_2,\ldots,X'_{d-1}$ computed in Step 7 of Algorithm 6 and the matrix X'_d computed according to Claim 6.4, we can efficiently compute the $w\times w$ matrix $X'_1 = \alpha^{-1}A^{-1}X_1A_2$.*

Proof: Fix $i,j \in [w]$. To compute the (i,j)-th entry of X'_1 choose an assignment of the $\mathbf{x}_2,\ldots,\mathbf{x}_d$ variables that sets the (j,i)-th entry of $X'_2X'_3\ldots X'_d$ to 1 and the remaining entries to 0. Such an assignment exists and can be efficiently computed by an argument similar to Observation 6.4. It can be easily verified that f computes the (i,j)-th entry of X'_1 where $X'_1 = \alpha^{-1}A^{-1}X_1A_2$. □

The following corollary easily follows from Claim 6.4 and Claim 6.5.

Corollary 6.1 *At the end of Step 8 of Algorithm 6, the following holds:* $\text{Trace}(X'_1\cdot X'_2\ldots X'_d) = \text{Trace}(X_1\cdot X_2\ldots X_d) = f$.

Step 9,10 - Computing V and output the result: Since $\text{Trace}(X'_1\cdot X'_2\ldots X'_d) = f$ and the X_i's are variable disjoint we can compute V by looking at the coefficients of the entries of X'_i. Then in Step 10, we use Schwartz-Zippel Lemma to check if f was indeed block-equivalent to Tr-IMM$_{w,d}$ to begin with and output the result accordingly.

6.2 Additional Claims and Observations

In this section we establish few helpful claims and observations that has been used in the proofs of Claim 6.2 and Claim 6.3.

Claim 6.6 *Let $X = (x_{ij})_{w\times w}$ be a symbolic matrix and $Y = X \otimes I_w$. Then there does not exist invertible matrices $P, Q \in \mathsf{GL}(w^2, \mathbb{F})$ such that $PY = Y^T Q$. In fact, the claim still holds even when X is a $w \times w$ symbolic matrix whose entries are linearly independent linear forms.*

Proof: For contradiction, suppose there exists $w^2 \times w^2$ invertible matrices P and Q such that $PY = Y^T Q$. The matrices P, Q looks as shown below.

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1w} \\ \vdots & \vdots & \vdots & \vdots \\ P_{w1} & P_{w2} & \dots & P_{ww} \end{bmatrix}_{w^2\times w^2} , Q = \begin{bmatrix} Q_{11} & Q_{12} & \dots & Q_{1w} \\ \vdots & \vdots & \vdots & \vdots \\ Q_{w1} & Q_{w2} & \dots & Q_{ww} \end{bmatrix}_{w^2\times w^2}$$

where each P_{ij} and Q_{ij} is a $w \times w$ matrix for all $i, j \in [w]$. So $PY = Y^T Q$ implies

$$\begin{bmatrix} P_{11}X & P_{12}X & \dots & P_{1w}X \\ \vdots & \vdots & \vdots & \vdots \\ P_{w1}X & P_{w2}X & \dots & P_{ww}X \end{bmatrix}_{w^2\times w^2} = \begin{bmatrix} X^T Q_{11} & X^T Q_{12} & \dots & X^T Q_{1w} \\ \vdots & \vdots & \vdots & \vdots \\ X^T Q_{w1} & X^T Q_{w2} & \dots & X^t Q_{ww} \end{bmatrix}_{w^2\times w^2}$$

In particular, for each $i, j \in [w]$ $P_{ij}X = X^T Q_{ij}$. Due to Observation 6.5, the first row of P_{ij} is 0 for any arbitrary $i, j \in [w]$ implying the first row of P is 0. So P is not invertible which is a contradiction. This claim still holds even when X is a $w\times w$ symbolic matrix whose entries are linearly independent linear form because we can apply a invertible linear transformation that maps each entry of X to a distinct formal variable. □

Observation 6.5 *Let $X = (x_{ij})_{w\times w}$ be a symbolic matrix with distinct entries. Let $P = (p_{ij})_{w\times w}$ and $Q = (q_{ij})_{w\times w}$ be $w \times w$ numeric matrices such that $PX = X^T Q$, then the first row of the matrix P has all entries 0.*

Proof: The $(1,2)$-th entry of PX is : $\sum_{j=1}^{w} p_{1j}x_{j2}$. The $(1,2)$-th entry of $X^T P$ is: $\sum_{j=1}^{w} q_{j1}x_{j1}$. Clearly the $(1,2)$-th entry of PX is variable disjoint from the $(1,2)$-th entry of $X^T P$ which means their corresponding coefficients must be 0. So the first row of P has all entries 0. □

Claim 6.7 *Let $X = (x_{ij})_{w\times w}$ be a symbolic matrix and $Y = I_w \otimes X$. Let $P, Q \in \mathsf{GL}(w^2, \mathbb{F})$ such that $PY = YQ$. Then $P = Q = D \otimes I_w$ for some $D \in \mathsf{GL}(w, \mathbb{F})$.*

Proof: Taking $Y = I_w \otimes I_w = I_{w^2}$, we get $P = Q$.

$$\text{Let } P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1w} \\ \vdots & \vdots & \vdots & \vdots \\ P_{w1} & P_{w2} & \dots & P_{ww} \end{bmatrix}_{w^2\times w^2}$$

where each P_{ij} is a $w \times w$ matrix for all $i, j \in [w]$. Since $PY = YP$, we have $P_{ij}X = XP_{ij}$ for each $i, j \in [w]$. By Observation 6.6, we infer that each P_{ij} is a scalar matrix whose diagonal entries we refer to as d_{ij}, i.e. $P_{ij} = d_{ij}I_w$ for each $i, j \in [w]$. Define $D = (d_{ij})_{w\times w}$ where d_{ij} is the diagonal entry of the matrix P_{ij}. Clearly $P = D \otimes I_w$. Further D is an invertible matrix. If it was not so, then there exists $c_1, \dots, c_w \in \mathbb{F}$ not all 0 such that $c_1D_1 + \dots + c_wD_w = 0$, where D_i refers to the i-th row of the matrix D. It is easy to verify that the matrix $P' = (c_i.P_{ij})_{w^2\times w^2}$ is not invertible implying $P = (P_{ij})_{w^2\times w^2}$ is not invertible which is a contradiction. Hence D is an invertible matrix. □

Observation 6.6 *Let $X = (x_{ij})_{w\times w}$ be a symbolic matrix whose entries are distinct formal variables and let $P = (p_{ij})_{w\times w}$ be a $w \times w$ matrix such that $PX = XP$. Then $P = dI_w$ for some $d \in \mathbb{F}$.*

Proof: Comparing the (i,i)-th entry of PX and XP, we can infer that the non-diagonal entries of P are 0. Hence P is a diagonal matrix with diagonal entries p_{ii} for $i \in [w]$. Now we show that all the diagonal entries are equal. The (i, j)-th entry of PX is $p_{ii}x_{ij}$ whereas the (i, j)-th entry of XP is $p_{jj}x_{jj}$ which implies $p_{ii} = p_{jj}$ for all $i, j \in [w]$. □

Chapter 7

Characterization by Symmetry

> In this chapter we discuss the group of symmetries of the Tr-IMM$_{w,d}$ polynomial and show that the Tr-IMM$_{w,d}$ polynomial is characterized by its group of symmetries.

Recall from Definition 2.23 that the group of symmetries of an n variate polynomial f is the set of all invertible $n \times n$ matrices A such that $f(\mathbf{x}) = f(A\mathbf{x})$. We start by defining the characterization by symmetry property below.

Definition 7.1 (Characterization by symmetry) *An n variate, degree d homogeneous polynomial f is said to be characterized by its group of symmetries $\mathcal{G}_f$ if for every n variate, degree d homogeneous polynomial g, the following holds: $\mathcal{G}_f = \mathcal{G}_g \iff f = \alpha \cdot g$,* for some non-zero $\alpha \in \mathbb{F}$.

One direction of the above equivalence is trivial as noted in Observation 7.1.

Observation 7.1 *Let $f = \alpha \cdot g$ for some non-zero $\alpha \in \mathbb{F}$. Then $\mathcal{G}_f = \mathcal{G}_g$.*

Proof: Let $A \in \mathcal{G}_f$, i.e $f(\mathbf{x}) = f(A\mathbf{x})$. Now $f(A\mathbf{x}) = \alpha \cdot g(A\mathbf{x}) = \alpha \cdot g(\mathbf{x}) = f(\mathbf{x})$. Hence, $A \in \mathcal{G}_g$. Similarly it can be shown that $A \in \mathcal{G}_g \implies A \in \mathcal{G}_f$. This implies $\mathcal{G}_f = \mathcal{G}_g$. □

7.1 Generating subgroups of $\mathcal{G}_{\text{Tr-IMM}}$

We briefly explain the generating subgroups of the group of symmetries $\mathcal{G}_{\text{Tr-IMM}}$ polynomial which has been described in [12]. In the reminder of this chapter, we only consider those Tr-IMM$_{w,d}$ polynomials with $w > 1$ and $d > 2$. Recall that $\text{Tr-IMM}_{w,d}$ can be written as

$$\text{Tr-IMM}_{w,d}(\mathbf{x}) = \text{Trace}(Q_1 \cdot Q_2 \dots Q_d)$$

where each Q_i is a $w \times w$ symbolic matrix whose entries are distinct variables. The three generating subgroups of $\mathcal{G}_{\text{Tr-IMM}}$ are:

1. **The Transposition Subgroup** $\mathcal{T}$: $\mathcal{T}$ is the subgroup consisting of two matrices, $\mathcal{T} = \{I_n, N\}$ where N is the $n \times n$ matrix such that $\text{Tr-IMM}_{w,d}(N\mathbf{x}) = \text{Trace}(Q_d^T \cdot Q_{d-1}^T \dots Q_1^T)$ and $N^2 = I_n$.

2. **The left-right multiplication subgroup** $\mathcal{M}_{w,d}$: Consider the following linear transformation M on the $\mathbf{x}$ variables given by:

$$\begin{aligned}
&Q_1 \mapsto X_1 := C_0 \cdot Q_1 \cdot C_1 \\
&Q_2 \mapsto X_2 := C_1^{-1} \cdot Q_2 \cdot C_2 \\
&Q_3 \mapsto X_3 := C_2^{-1} \cdot Q_3 \cdot C_3 \\
&\vdots \\
&Q_d \mapsto X_d := C_{d-1}^{-1} \cdot Q_d \cdot C_0^{-1}
\end{aligned}$$

where $C_0, \dots, C_{d-1}$ are invertible matrices in $\mathbb{F}^{w \times w}$. Clearly,

$$\begin{aligned}
\text{Tr-IMM}_{w,d}(M\mathbf{x}) &= \text{Trace}(X_1 \cdot X_2 \dots X_d) \\
&= \text{Trace}((C_0 \cdot Q_1 \cdot C_1) \cdot (C_1^{-1} \cdot Q_2 \cdot C_2) \dots (C_{d-1}^{-1} \cdot Q_d \cdot C_0^{-1})) \\
&= \text{Trace}(C_0 \cdot Q_1 \cdot Q_2 \dots \cdot Q_d \cdot C_0^{-1}) = \text{Trace}(Q_1 \cdot Q_2 \dots \cdot Q_d \cdot C_0^{-1} \cdot C_0) \\
&= \text{Tr-IMM}_{w,d}(\mathbf{x}) \enspace .
\end{aligned}$$

The equality in the penultimate line of the above equations follows from the commutativity of the trace of the matrix product of $Q_1 \cdot Q_2 \dots Q_d \cdot C_0^{-1}$ and C_0. The linear map M is defined by the d matrices $C_0, C_1, \dots, C_{d-1}$. All such M which can be described as above by some d invertible matrices $C_0, C_1, \dots, C_{d-1}$ are said to be in the *left-right multiplication subgroup* $\mathcal{M}_{w,d}$

of $\mathcal{G}_{\text{Tr-IMM}_{w,d}}$.

3. **The Circular Transformations Subgroup $\mathcal{C}$:** This group consists of $n \times n$ matrices in $\mathcal{G}_{\text{Tr-IMM}_{w,d}}$ that cyclically rotates the order of the matrices in the product expression $Q_1 \cdot Q_2 \ldots Q_d$. A linear transformation $C \in \mathcal{C}$ has the following effect on the variables of Tr-IMM$_{w,d}$: for each $i \in [d]$, $Q_i \mapsto Q_{(r-i+1) \bmod \text{d}}$ for some $r \in [d]$. This does not change the trace of the product as $\text{Trace}(Q_1 \cdot Q_2 \ldots Q_r \ldots Q_d) = \text{Trace}(Q_r \cdot Q_{r+1} \ldots Q_d \cdot Q_1 \ldots Q_{r-1})$.

Theorem 7.1 expresses $\mathcal{G}_{\text{Tr-IMM}}$ in terms of its generating subgroups and has been proved in [12]. Before stating the theorem, we recall some useful definitions from group theory.

Definition 7.2 (Normal Subgroup) *A subgroup U of a group G is said to be a normal subgroup of G (denoted by $U \trianglelefteq G$) if for all $g \in g$, $gUg^{-1} \subseteq U$.*

Definition 7.3 (Semidirect Product) *Let G be a group and N, H be its subgroups. G is said to be a semidirect product of U and H (denoted by $G = U \rtimes H$) if $G = UH$, $U \trianglelefteq G$ and $U \cap H = \{e\}$.*

For brevity, we denote $\mathcal{M}_{w,d}$ by $\mathcal{M}$ in the following theorem.

Theorem 7.1 (Symmetries of Tr-IMM) $\mathcal{G}_{\text{Tr-IMM}} = \mathcal{M} \rtimes (\mathcal{C} \rtimes \mathcal{T})$. [1]

Let Q_{1i} denote the i-th row of Q_1 and Q_{dj} denote the j-th column of Q_d for $i, j \in [w]$. Recall that IMM$_{w,d}$ is defined as the $(1,1)$-th entry of the product $Q_1 \cdot Q_2 \ldots Q_d$. So $\text{IMM}_{w,d} = Q_{11} \cdot Q_2 \ldots Q_{d-1} \cdot Q_{d1}$. Analogous to the subgroup $\mathcal{M}_{w,d}$ of $\mathcal{G}_{\text{Tr-IMM}_{w,d}}$,the left right multiplication subgroup for the IMM$_{w,d}$ polynomial is defined below.

The left-right multiplication subgroup $\mathcal{M}'_{w,d}$ of IMM$_{w,d}$ polynomial The subgroup $\mathcal{M}'_{w,d}$ con-

[1] Observe that $\mathcal{C}$ is a normal subgroup of $\mathcal{C} \rtimes \mathcal{T}$ and $\mathcal{M}$ is a normal subgroup of $\mathcal{G}_{\text{Tr-IMM}}$.

sists of linear maps M' that have the following effect on the variables of $\text{IMM}_{w,d}$.

$$\begin{aligned}
Q_{11} &\mapsto X_{11} := Q_{11} \cdot C_1 \\
Q_2 &\mapsto X_2 := C_1^{-1} \cdot Q_2 \cdot C_2 \\
&\vdots \\
Q_{d-1} &\mapsto X_{d-1} := C_{d-2}^{-1} \cdot Q_{d-1} \cdot C_{d-1} \\
Q_{d1} &\mapsto X_{d1} := C_{d-1}^{-1} \cdot Q_{d1}
\end{aligned}$$

where $C_1, \ldots, C_{d-1}$ are invertible matrices in $\mathbb{F}^{w \times w}$. Define $\text{IMM}_{w,d}^{(i)} := Q_{1i} \cdots Q_2 \ldots Q_{d-1} \cdot Q_{di}$. Hence, $\text{Tr-IMM}_{w,d} = \text{IMM}_{w,d}^{(1)} + \ldots + \text{IMM}_{w,d}^{(w)}$. We now make the following observation.

Observation 7.2 *There is an injective map $\phi : \mathcal{M}^{'}_{w,d} \mapsto \mathcal{M}_{w,d}$ defined as follows: If $M' \in \mathcal{M}^{'}_{w,d}$ is defined by the matrices $C_1, \ldots, C_{d-1}$ then $M := \phi(M')$ is the linear map defined by the matrices $I_w, C_1, \ldots, C_{d-1}$. In particular, for all $M' \in \mathcal{M}^{'}_{w,d}$, $\text{Tr-IMM}_{w,d}(M\mathbf{x}) = \text{IMM}_{w,d}^{(1)}(M'\mathbf{x}) + \ldots + \text{IMM}_{w,d}^{(d)}(M'\mathbf{x})$.*

Proof: Consider a linear map $M' \in \mathcal{M}'_{w,d}$ defined by the matrices $C_1, \ldots, C_{d-1}$. For each $i \in [d]$, applying M' on the variables of $\text{IMM}_{w,d}^{(i)}$ has the following effect:

$$\begin{aligned}
&Q_{1i} \mapsto X_{1i} := Q_{1i} \cdot C_1 \\
&Q_2 \mapsto X_2 := C_1^{-1} \cdot Q_2 \cdot C_2 \\
&\vdots \\
&Q_{d-1} \mapsto X_{d-1} := C_{d-2}^{-1} \cdot C_{d-1} \cdot C_{d-1} \\
&Q_{di} \mapsto X_{di} := C_{d-1}^{-1} \cdot C_{di}
\end{aligned}$$

The matrix M corresponds to applying M' simultaneously to the variables of $\text{IMM}_{w,d}^{(1)}, \ldots, \text{IMM}_{w,d}^{(1)}$ is the following linear map which we denote by M.

$$\begin{aligned}
&Q_1 \mapsto Q_1 \cdot C_1 = Q_{11} \cdot C_1 | \ldots | Q_{1w} \cdot C_1 \\
&Q_2 \mapsto C_1^{-1} \cdot Q_2 \cdot C_2 \\
&\vdots
\end{aligned}$$

$$Q_{d-1} \mapsto C_{d-2}^{-1} \cdot Q_{d-1} \cdot C_{d-1}$$
$$Q_d \mapsto C_{d-1} \cdot Q_d = C_{d-1}^{-1} \cdot Q_{d1}|\ldots|C_{d-1}^{-1} \cdot Q_{dw}$$

Since $\mathrm{IMM}_{w,d}^{(i)}(M'\mathbf{x}) = \mathrm{IMM}_{w,d}^{(i)}(\mathbf{x})$ for all $i \in [d]$, $\text{Tr-IMM}_{w,d}(M\mathbf{x}) = \text{Tr-IMM}_{w,d}(\mathbf{x})$ implying $M \in \mathcal{G}_{\text{Tr-IMM}}$. It is also easy to see that $M \in \mathcal{M}_{w,d}$ and is defined by the matrices $I_w, C_1, \ldots, C_{d-1}$. □

We will see in the next section that $\text{Tr-IMM}_{w,d}$ is characterized by symmetry just due to the presence of the left-right multiplication subgroup.

7.2 $\text{Tr-IMM}_{w,d}$ is characterized by its group of symmetries

In [20], it has been shown that the $\mathrm{IMM}_{w,d}$ polynomial is characterized by it's group of symmetries which we state in Lemma 7.1.

Lemma 7.1 *Let f be a homogeneous n variate, degree d polynomial over $\mathbb{F}$ with $\mathcal{M'}_{w,d} \subseteq \mathcal{G}_f$ and $|F| > d+1$. Then $f = \alpha \cdot \mathrm{IMM}_{w,d}$ for some non-zero $\alpha \in \mathbb{F}$.*

We use Lemma 7.1 and the following claim to establish characterization by symmetry for the $\text{Tr-IMM}_{w,d}$ polynomial.

Claim 7.1 *Let f be a homogeneous n variate, degree d polynomial over $\mathbb{F}$ with $\mathcal{M}_{w,d} \subseteq G_f$ and $|F| > d+1$. Then f is a set-multilinear polynomial in the variable sets $\mathbf{x}_1, \ldots, \mathbf{x}_d$, i.e every monomial in f has at most one variable from each $\mathbf{x}_i, i \in [d]$.*

Proof: Since $|F| > d+1$, there exists $\rho \neq 0 \in \mathbb{F}$ which is not an e-th root of unity for all $e \leq d$. Fix a $k \in [d-1]$. Consider $M \in \mathcal{M}_{w,d}$ which scales the variables in Q_k by ρ and the variables in Q_{k+1} by ρ^{-1}. Since $f(\mathbf{x}) = f(M\mathbf{x})$ and $\rho^e \neq 1$ for any $e \leq d$,the number of variables from $\mathbf{x}_k$ must be equal to the number of variables from $\mathbf{x}_{k+1}$ in any monomial of f. This is true for any $k \in [d-1]$ and f being a homogeneous degree d polynomial implies f is a set-multilinear polynomial in the variable sets $\mathbf{x}_1, \ldots, \mathbf{x}_d$. □

Lemma 7.2 (Characterization by Symmetry) *Let f be a homogeneous n variate, degree d polynomial over $\mathbb{F}$ with $\mathcal{M}_{w,d} \subseteq G_f$ and $|F| > d+1$. Then $f = \alpha \cdot \text{Tr-IMM}_{w,d}$ for some non-zero $\alpha \in \mathbb{F}$.*

Proof: From Claim 7.1, it follows that f is a set-multilinear polynomial in the variable sets $\mathbf{x}_1,\ldots,\mathbf{x}_d$. Hence f can be written as follows:

$$f = \sum_{i,j\in[w]} x_{ij}^{(1)} g_{ij} = \sum_{j\in[w]} x_{1j}^{(1)} g_{1j} + \ldots + \sum_{j\in[w]} x_{wj}^{(1)} g_{wj}$$

where the g_{ij} for $i,j \in [w]$ are set-multilinear polynomials in the variable sets $\mathbf{x}_2,\ldots,\mathbf{x}_d$. For $i \in [w]$, let $f_i := \sum_{j\in[w]} x_{ij}^{(1)} g_{ij}$. Thus $f = f_1 + \ldots + f_w$. Let $\mathbf{x}_{1i}$ denote the variables in the i-th row of Q_1 and $\mathbf{x}_{dj}$ denote the variables in the j-th column of Q_d. Clearly f_i is s set-multilinear polynomial in the variable sets $\mathbf{x}_{1i},\mathbf{x}_2,\ldots,\mathbf{x}_d$. Infact we show that f_i is a set-multilinear polynomial in the variable sets $\mathbf{x}_{1i},\mathbf{x}_2,\ldots,\mathbf{x}_{di}$. Consider a linear map $M \in \mathcal{M}_{w,d}$ defined by the matrices $C_0,\ldots,C_{d-1}$ where C_0 is a $w\times w$ diagonal matrix whose (i,i)-th entry is ρ and the remaining entries are 1, and $C_1,\ldots,C_{d-1}$ are $w \times w$ identity matrices. It is easy to infer that M replaces the $\mathbf{x}$ variables by the following linear forms: $\mathbf{x}_{1i} \mapsto \rho\cdot\mathbf{x}_{1i}$, $\mathbf{x}_{di} \mapsto \rho^{-1}\cdot\mathbf{x}_{di}$, and all other variables are mapped to themselves. Since each monomial in f_i contains exactly one variable from $\mathbf{x}_{1i}$, to cancel the effect of scaling by ρ each monomial in f_i must contain exactly one variable from $\mathbf{x}_{di}$. Hence f_i is a set-multilinear polynomial in the variable sets $\mathbf{x}_{1i},\mathbf{x}_2,\ldots,\mathbf{x}_{di}$. In the next part of the proof, first we show that for each $i \in [w]$, $f_i = \alpha_i \cdot \text{IMM}_{w,d}^{(i)}$ for some non-zero $\alpha_i \in \mathbb{F}$ and then finally prove that $\alpha_1 = \ldots = \alpha_w = \alpha$ implying $f = \alpha\cdot(\text{IMM}_{w,d}^{(1)} + \ldots + \text{IMM}_{w,d}^{(w)}) = \alpha\cdot\text{Tr-IMM}_{w,d}$.

From Lemma 7.1, to show that $f_i = \alpha_i \cdot \text{IMM}_{w,d}^{(i)}$ for some non-zero $\alpha \in \mathbb{F}$, it is sufficient to show that $\mathcal{M'}_{w,d} \subset \mathcal{G}_{f_i}$. Recall the injective map $\phi : \mathcal{M'}_{w,d} \mapsto \mathcal{M}_{w,d}$ defined in Observation 7.2. Let $M' \in \mathcal{M'}_{w,d}$ be defined by the $w\times w$ invertible matrices $C_1,\ldots C_{d-1}$ and $M := \phi(M') \in \mathcal{M}_{w,d}$ be defined by the matrices $I_w, C_1,\ldots,C_{d-1}$. Since f_i is a set-multilinear polynomial in $\mathbf{x}_{1i},\mathbf{x}_2,\ldots,\mathbf{x}_{di}$ for all $i \in [w]$, applying M on the variables of f is equivalent to simultaneously applying M' on $f_1,\ldots,f_w$. But as $M \in \mathcal{M}_{w,d} \subseteq \mathcal{G}_f$, $f(\mathbf{x}) = f(M\mathbf{x}) = f_1(M'\mathbf{x}) + \ldots + f_w(M'\mathbf{x})$. Since $f_1,\ldots,f_w$ are pairwise monomial disjoint we infer that $f_i(M'\mathbf{x}) = f_i(\mathbf{x})$ for $i \in [w]$. Thus $M' \in \mathcal{G}_{f_i}$ implying $\mathcal{M'}_{w,d} \subset \mathcal{G}_{f_i}$ for each $i \in [w]$. We now show that $\alpha_1 = \alpha_2 = \ldots = \alpha_w$. Let $i \neq j$ and $i,j \in [w]$. Let I_{ij} be the matrix obtained by swapping i-th and j-th rows of the identity matrix I_w. Consider the linear map $M \in \mathcal{M}_{w,d}$ defined by the $w \times w$ matrices $C_0 = I_{ij}, C_1 = I_w,\ldots,C_{d-1} = I_w$. M replaces the $\mathbf{x}$ variables as follows: It exchanges the i-th and j-th rows of Q_1, the i-th and j-th columns of Q_d, i.e $Q_{1i} \leftrightarrow Q_{1j}$ and $Q_{di} \leftrightarrow Q_{dj}$. The remaining variables are mapped to themselves.

$$\begin{aligned} f(\mathbf{x}) &= \alpha_1 \mathrm{IMM}^{(1)}_{w,d} + \ldots + \alpha_i \mathrm{IMM}^{(i)}_{w,d} + \ldots + \alpha_j \mathrm{IMM}^{(j)}_{w,d} + \ldots + \alpha_w \mathrm{IMM}^{(w)}_{w,d} \\ f(M\mathbf{x}) &= \alpha_1 \mathrm{IMM}^{(1)}_{w,d} + \ldots + \alpha_i \mathrm{IMM}^{(j)}_{w,d} + \ldots + \alpha_j \mathrm{IMM}^{(i)}_{w,d} + \ldots + \alpha_w \mathrm{IMM}^{(w)}_{w,d} \ . \end{aligned}$$

As $f(\mathbf{x}) = f(M\mathbf{x})$ and $\mathrm{IMM}^{(i)}_{w,d}$ and $\mathrm{IMM}^{(j)}_{w,d}$ are monomial disjoint, we infer that $\alpha_i = \alpha_j$. Since this is true for arbitrary $i \neq j$, $\alpha_1 = \ldots = \alpha_w = \alpha$. □

Bibliography

[1] Scott Aaronson. P vs np. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:4, 2017. URL https://eccc.weizmann.ac.il/report/2017/004. 6

[2] Manindra Agrawal and Nitin Saxena. Equivalence of f-algebras and cubic forms. In *STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science, Marseille, France, February 23-25, 2006, Proceedings*, pages 115–126, 2006. doi: 10.1007/11672142_8. URL https://doi.org/10.1007/11672142_8. 5, 8

[3] Manindra Agrawal and V Vinay. Arithmetic circuits: A chasm at depth four. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 67–75. IEEE, 2008. 3, 4

[4] Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical computer science*, 22(3):317–330, 1983. 3

[5] Elwyn R Berlekamp. Factoring polynomials over finite fields. *Bell System Technical Journal*, 46 (8):1853–1859, 1967. 29

[6] David G Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, pages 587–592, 1981. 29

[7] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965. 1

[8] Laszlo Csanky. Fast parallel matrix inversion algorithms. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 11–12. IEEE, 1975. 66

[9] Stephen A Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-nc. *arXiv preprint arXiv:1601.06319*, 2016. 1

[10] Michael A Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 243–252. IEEE, 2013. 56

[11] Ankit Garg, Nikhil Gupta, Neeraj Kayal, and Chandan Saha. Determinant equivalence test over finite fields and over q. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 26, page 42, 2019. 7, 12, 13, 66

[12] Fulvio Gesmundo. Geometric aspects of iterated matrix multiplication. *Journal of Algebra*, 461: 42–64, 2016. 32, 74, 76

[13] Joshua Abraham Grochow. *Symmetry and equivalence relations in classical and geometric complexity theory, PhD Thesis.* The University of Chicago, 2012. 6, 22, 63

[14] Brian Hall. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer, 2015. 21, 22

[15] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. 4

[16] KA Kalorkoti. A lower bound for the formula size of rational functions. *SIAM Journal on Computing*, 14(3):678–687, 1985. 3

[17] Erich Kaltofen and Barry M Trager. Computing with polynomials given byblack boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *Journal of Symbolic Computation*, 9(3):301–320, 1990. 10, 66

[18] Neeraj Kayal. Affine projections of polynomials: extended abstract. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 643–662, 2012. doi: 10.1145/2213977.2214036. URL https://doi.org/10.1145/2213977.2214036. 5, 6, 7, 8, 9, 12, 23, 24, 29, 66

[19] Neeraj Kayal. Affine projections of polynomials. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 643–662. ACM, 2012. 22

[20] Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of full rank algebraic branching programs. *ACM Transactions on Computation Theory (TOCT)*, 11(1):2, 2018. iv, 6, 7, 10, 12, 20, 24, 36, 37, 42, 46, 48, 56, 78

[21] Alexander Kirillov Jr. *An introduction to Lie groups and Lie algebras*, volume 113. Cambridge University Press, 2008. 21, 22

[22] Adam R. Klivans and Amir Shpilka. Learning arithmetic circuits via partial derivatives. In *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, pages 463–476, 2003. doi: 10.1007/978-3-540-45167-9_34. URL `https://doi.org/10.1007/978-3-540-45167-9_34`. 31, 62

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

[24] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982. 29

[25] Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. 1, 3

[26] Ketan D Mulmuley and Milind Sohoni. Geometric complexity theory i: An approach to the p vs. np and related problems. *SIAM Journal on Computing*, 31(2):496–526, 2001. 6

[27] Jacques Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 33–48. Springer, 1996. 5

[28] Nitin Saxena. Morphisms of rings and applications to complexity. *Indian Institute of Technology Kanpur*, 2006. 5

[29] Volker Strassen. Die berechnungskomplexität von elementarsymmetrischen funktionen und von interpolationskoeffizienten. *Numerische Mathematik*, 20(3):238–251, 1973. 1

[30] Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 696–707. Ieee, 2017. 1

[31] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Information and Computation*, 240:2–11, 2015. 3

[32] Thomas Thierauf. The isomorphism problem for read-once branching programs and arithmetic circuits. In *Chicago Journal of Theoretical Computer Science*. Citeseer, 1998. 5

[33] Leslie G Valiant. Completeness classes in algebra. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 249–261. ACM, 1979. 2, 3, 18

[34] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8 (2):189–201, 1979. 2, 3, 19

[35] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and algebraic computation*, pages 216–226. Springer, 1979. 25