Equivalence test for the trace iterated matrix multiplication polynomial

Janaky Murthy
M.Tech Research

Advisor: Prof. Chandan Saha

Department of Computer Science and Automation
IISc Bangalore

Overview

- Introduction and Motivation
- Problem Statement
- Our Results
- Approach

What are Equivalent polynomials?

Definition (Equivalent polynomials)

$$g(x_1, x_2) = x_1 + x_2^2$$

$$f(x_1, x_2) = x_1 + x_2 + x_2^2 \ .$$

If we replace the variables of g as follows, we obtain f.

$$x_1 \to x_1 + x_2$$

$$x_2 \to x_2 \ .$$

What are Equivalent polynomials?

Definition (Equivalent polynomials)

$$g(x_1, x_2) = x_1 + x_2^2$$

$$f(x_1, x_2) = x_1 + x_2 + x_2^2 \ .$$

If we replace the variables of g as follows, we obtain f.

$$\underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{x}} = \begin{bmatrix} x_1 + x_2 \\ x_2 \end{bmatrix} ; \quad f(\mathbf{x}) = g(A\mathbf{x})$$

What are Equivalent polynomials?

Definition (Equivalent polynomials)

Two n-variate, degree d polynomials f and g (over a field $\mathbb{F}$) are said to be **equivalent** if there exists an *invertible* matrix $A \in \mathbb{F}^{n \times n}$ such that $f(\mathbf{x}) = g(A\mathbf{x})$.

The Equivalence Testing Problem: Can we *efficiently* check if two polynomials f and g are equivalent?

Complexity of Equivalence Testing

Depends on the underlying field.

- **over finite fields**: NP $\cap$ co-AM
  [Thierauf(1998), Saxena(2006)]
- **over** $\mathbb{Q}$: not even known if it is decidable or not!
- **over other fields**: reduces to solving system of polynomial equations (which could possibly be a harder problem).

Relation to other Isomorphism problems

Isomorphism problem: Check if there is a **bijection** between two *structures* that **preserves some relation on the structure**.

Examples: Graph Isomorphism, Algebra Isomorphism, Tensor Isomorphism.

Graph Isomorphism: Two graphs are isomorphic if there is a **bijection** between the vertex sets which **preserves the edge relation**. Given two graphs, check if they are isomorphic.

Algebra Isomorphism

$(\mathcal{A}, +, *)$ is a $\mathbb{F}$-**Algebra** if:

- $(\mathcal{A}, +)$ is a $\mathbb{F}$-vector space.
- $(\mathcal{A}, +, *)$ is a ring.
- the ring multiplication is compatible with the scalar multiplication of the field, i.e $k(B * C) = (kB) * C = B * (kC)$ for all $B, C \in \mathcal{A}$ and $k \in \mathbb{F}$.

Example The set of all $m \times m$ matrices $(\mathcal{M}_m, +, *)$.

Algebra Isomorphism: Given bases of two algebras (as structure table), check if there is a **bijection** that **preserves the** $+$ **and** $*$ **operations**.

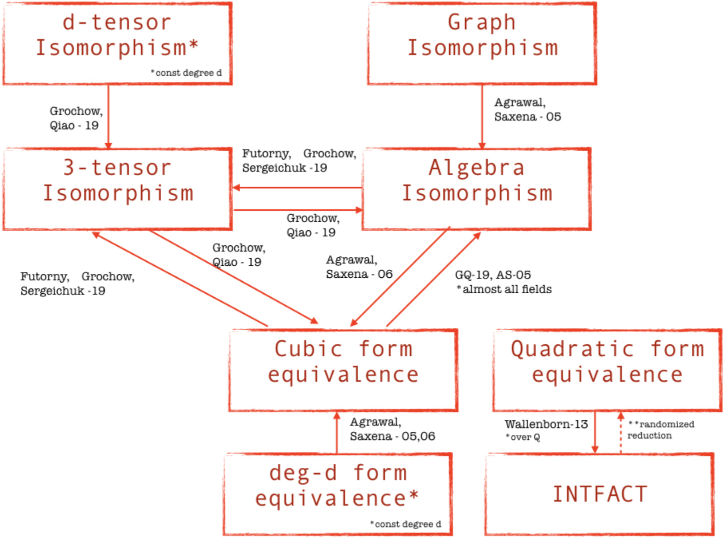
d-tensor Isomorphism

Consider a partition of n variables into d sets. A d-**tensor** is a degree d homogeneous polynomial such that each monomial contains exactly one variable from each of the d variable sets.

Example: $f = x_1 x_4 + x_2 x_4 + x_3 x_6$ is a 2-tensor.

d-**tensor Isomorphism:** Given two d-tensors f and g, check if there exists invertible matrices $B_1, \ldots, B_d$ such that $f(\mathbf{x}_1, \ldots, \mathbf{x}_d) = g(B_1\mathbf{x}_1, \ldots, B_d\mathbf{x}_d)$.

Connections between the isomorphism problems



A natural variant of Equivalence Testing

Equivalence test for special polynomial families: Check if a polynomial f is equivalent to some $g \in \mathcal{G}$ where $\mathcal{G} = \{g_1, g_2, \ldots\}$ is a polynomial family.

Some popular polynomial families: Permanent, Determinant, Power Symmetric polynomial, Sum of Products polynomial, Elementary Symmetric polynomial, Iterated Matrix Multiplication (IMM) polynomial, Trace Iterated Matrix Multiplication (Tr-IMM) polynomial, Design polynomials etc...

Motivation from Geometric Complexity Theory

An n-variate, degree d homogeneous polynomial f: **A point in the vector space** $\mathbb{C}^N$ (where $N = \binom{n+d}{d}$).

Orbit of f: $\mathcal{O}(f) = \{g : g(\mathbf{x}) = f(A\mathbf{x}), A \text{ is invertible}\}$.

Orbit Closure of f: $\widehat{\mathcal{O}(f)}$ - The Zariski closure of $\mathcal{O}$.

Motivation from Geometric Complexity Theory

Perm vs Det problem: Show that padded permanent *is not* in the **orbit closure** of (poly-sized) determinant polynomial.

This question also makes sense for permanent vs any other polynomial family $\mathcal{G}$ where $\mathcal{G}$ is a complete for some low complexity circuit class $\mathcal{C}$.

Equivalence test for some well known polynomial families

[Kayal(2012)] gave efficient randomized algorithms for equivalence testing of the **Permanent polynomial** family , **Power Symmetric polynomial** family, **Sum of Product polynomial** family, **Elementary Symmetric polynomial** family *over any field*.

From now on we assume a stronger **search version** of the equivalence testing problem.

Determinant Equivalence Testing

The Determinant polynomial family: $\{\mathrm{Det}(X_n)\}_{n\geq 1}$, where $\mathrm{Det}(X_n)$ denotes the determinant of $n \times n$ symbolic matrix X_n.

Determinant Equivalence Testing (DET)

- An efficient randomized algorithm is known over :
  - $\mathbb{C}$ [Kayal(2012)]
  - finite fields of sufficiently large characteristic - Garg,Gupta,Kayal,Saha [GGKS19].
  - For fixed n, DET can be efficiently done given oracle access to INTFACT [GGKS19].
- But it is as hard as **Integer Factoring (INTFACT)** over $\mathbb{Q}$ [GGKS19].

IMM Equivalence Testing

The Iterated Matrix Multiplication Polynomial Family

$\text{IMM}_{w,d} := (1,1)$-th entry of $(X_1 \cdot X_2 \ldots X_d)$ where each X_i is a $w \times w$ symbolic matrix.

An efficient randomized equivalence test for the **Iterated Matrix Multiplication polynomial** (IMM) over $\mathbb{Q}, \mathbb{C}$ and finite fields is known from Kayal,Nair,Saha,Tavenas [KNST17].

IMM vs Determinant Equivalence testing

Both IMM and Determinant polynomial families are complete for the circuit class VBP, yet they can not have similar algorithmic complexity for the equivalence testing problem (over $\mathbb{Q}$) unless INTFACT is easy.

The Trace Iterated Matrix Multiplication Polynomial

Definition (The Trace Iterated Matrix Multiplication Polynomial)

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

$w = 2, d = 3.$

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

The Trace Iterated Matrix Multiplication Polynomial

Definition (The Trace Iterated Matrix Multiplication Polynomial)

Let $Q_1, \ldots, Q_d$ be $w \times w$ symbolic matrices whose entries are distinct (formal) variables. Then the **Trace Iterated Matrix Multiplication Polynomial** denoted as $\text{Tr-IMM}_{w,d}$ is defined as the trace of the product of these matrices.

$$\text{Tr-IMM}_{w,d} = \text{tr}(Q_1 \cdot Q_2 \ldots Q_d) \ .$$

Equivalence test for Tr-IMM (TRACE)

It is syntatically close to the IMM polynomial, which is the $(1, 1)$-th entry of the matrix product.

Is the complexity of TRACE similar to the equivalence test for IMM polynomial?

Equivalence test for Tr-IMM (TRACE)

It is syntatically close to the IMM polynomial, which is the $(1, 1)$-th entry of the matrix product.

Is the complexity of TRACE similar to the equivalence test for IMM polynomial?

Or does it resemble that of DET?

Equivalence test for Tr-IMM (TRACE)

Problem Statement (Equivalence test for Tr-$\text{IMM}_{w,d}$ polynomial (TRACE))

Given *blackbox access* to an n-variate degree d polynomial f, check *efficiently* **if** f **is equivalent to** Tr-$\text{IMM}_{w,d}$. If yes, then compute an invertible matrix $A \in \mathbb{F}^{n \times n}$ such that $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$

Could there be some relation between special cases of the isomorphism problem and the special cases of equivalence testing?

Some special cases of the Isomorphism Problems

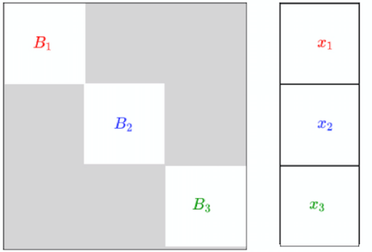
Full Matrix Algebra Isomorphism (FMAI) Given a basis of an algebra $\mathcal{A} \subseteq \mathcal{M}_m$, determine if $\mathcal{A}$ is isomorphic to $\mathcal{M}_w$ where $w^2 = \dim(\mathcal{A})$. If yes, compute an isomorphism from $\mathcal{A} \to \mathcal{M}_w$.

Some special cases of the Isomorphism Problems

Matrix Multiplication Tensor Isomorphism (MMTI) Given a 3-tensor f, check if it is isomorphic to any tensor in the Tr-IMM$_{w,3}$ family, i.e check if

$$f(\mathbf{x}) = \text{Tr-IMM}_{w,3}(B_1\mathbf{x}_1, B_2\mathbf{x}_2, B_3\mathbf{x}_3) = \text{Tr-IMM}_{w,3}(B\mathbf{x})$$

and if yes, output B_1, B_2, B_3.

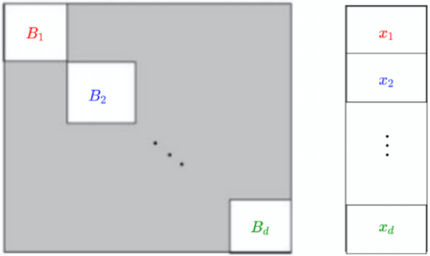


Some special cases of the Isomorphism Problems

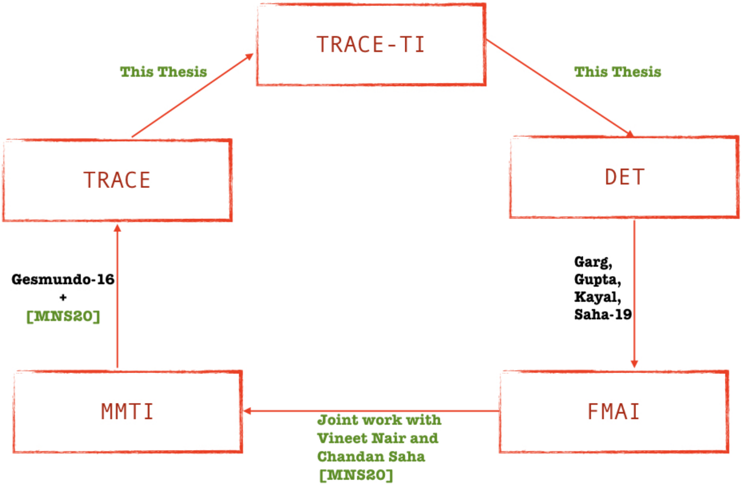
Tensor Isomorphism for Tr-IMM **(TRACE-TI)** Given a d-tensor f, check if it is isomorphic to any tensor in the $\text{Tr-IMM}_{w,d}$ family, i.e check if

$$f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(B_1\mathbf{x}_1, \ldots, B_d\mathbf{x}_d) = \text{Tr-IMM}_{w,d}(B\mathbf{x}).$$

and if yes, output $B_1, \ldots, B_d$.



Results


TRACE-TI
This Thesis
This Thesis
TRACE
DET
Gesmundo-16
+
[MNS20]
Garg,
Gupta,
Kayal,
Saha-19
MMTI
FMAI
Joint work with
Vineet Nair and
Chandan Saha
[MNS20]

Results

Theorem 1 (**TRACE is randomized polynomial time Turing reducible to DET**)

Given oracle access to DET *over* $\mathbb{F}$, TRACE *can be solved in* **randomized**, **polynomial time**

polynomial time: $poly(n, \beta)$ running time

randomized: $1 - o(1)$ success probability.

Approach

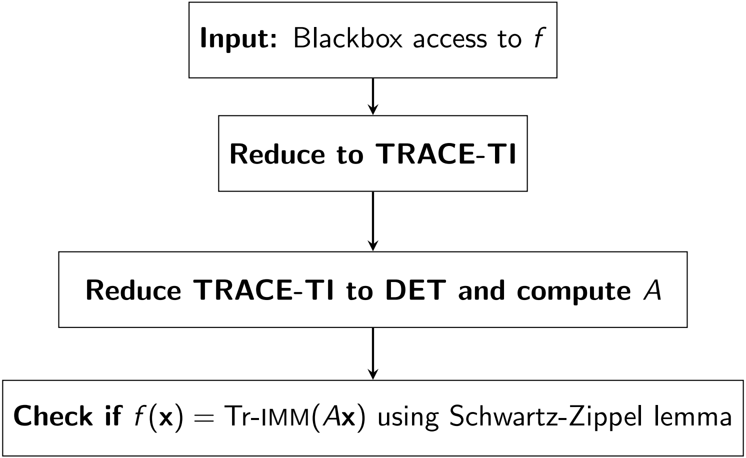


Figure: High level view of the Algorithm

Part-I: Reduction to TRACE-TI

TRACE: Is $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(A\mathbf{x})$ for some **invertible** matrix A?

TRACE-TI: Is $f(\mathbf{x}) = \text{Tr-IMM}_{w,d}(B\mathbf{x})$ for some **invertible, block-diagonal matrix** B?

Remark: An efficient randomized algorithm for TRACE-TI over $\mathbb{C}$ was given in [Grochow(2012)] which does not involve reduction to DET.

Part-I: Reduction to TRACE-TI

$$\text{Tr-IMM}(\mathbf{x}) = \text{tr}(Q_1 \cdot Q_2 \dots Q_d)$$

$$f = \text{Tr-IMM}(A\mathbf{x}) = \text{tr}(X_1 \cdot X_2 \dots X_d)$$

For example,

$$Q_i = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}, X_i = \begin{bmatrix} x_1 + x_6 & 2x_1 \\ x_1 + 2x_4 & x_4 - x_9 \end{bmatrix}$$

$\mathcal{X}_i$ - space spanned by the linear forms in X_i. The **Layer Spaces** of f are $\mathcal{X}_1, \dots, \mathcal{X}_d$.

Part-I: Reduction to TRACE-TI

1.**Compute a bases for the layer spaces** $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of f.

Part-I: Reduction to TRACE-TI

1.**Compute a bases for the layer spaces** $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of f.

2. **Compute a linear map** $\hat{A}$ **which maps each basis vector to a distinct variable**.

Part-I: Reduction to TRACE-TI

1.**Compute a bases for the layer spaces** $\mathcal{X}_1, \ldots, \mathcal{X}_d$ of f.

2. **Compute a linear map** $\hat{A}$ **which maps each basis vector to a distinct variable**.

3. **Define a new polynomial** $h(\mathbf{x}) = f(\hat{A}\mathbf{x})$. Since we mapped each basis vector to a distinct variable, h **is a** d**-tensor**.

$$h(\mathbf{x}) = f(\hat{A}\mathbf{x}) = \text{Tr-IMM}(A\hat{A}\mathbf{x})$$

We compute $\hat{A}$ such that $A\widehat{A}$ is block-diagonal. **This is the TRACE-TI problem!**

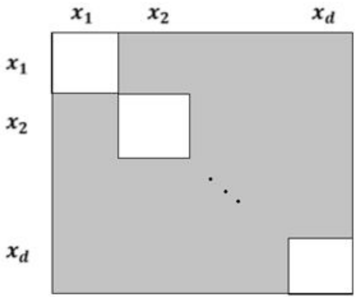
Computing a basis for the layer spaces

Associated with any n-variate polynomial f, there is a vector space called the **Lie Algebra** $\mathfrak{g}_f$ (of the group of symmetries) of f which consists of $n \times n$ matrices $E = (e_{ij})_{n\times n}$ satisfying

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

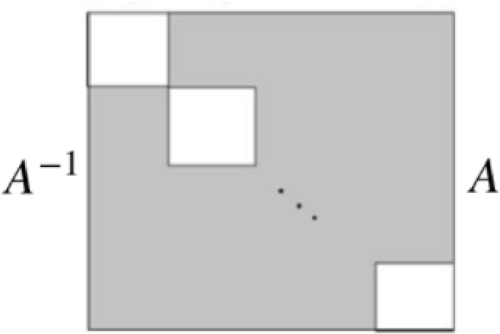
Computing a basis for the layer spaces

The basis elements of the Lie Algebra of Tr-IMM are **block-diagonal** matrices [Gesmundo(2016)].



Computing a basis for the layer spaces

The corresponding basis elements of the Lie Algebra of $f \sim$ Tr-IMM looks like:



Computing a basis for the layer spaces

We compute a bases of the Lie Algebra of f.

We exploit this relationship to compute a bases for the **irreducible invariant subspaces** $\mathcal{V}_1, \ldots, \mathcal{V}_d$ of $\mathfrak{g}_f$.

Given a bases of these irreducible invariant subspaces, we then compute a bases of the **layer spaces** of f and then **reorder them appropriately**.

Part-II: Reduction from TRACE-TI to DET

$$f = \mathrm{tr}(X_1 \cdot X_2 \ldots X_{d-1} \cdot X_d)$$

Part-II: Reduction from TRACE-TI to DET

$$f = \mathrm{tr}(X_1 \cdot X_2 \ldots X_{d-1} \cdot X_d) = Y_1 \cdot Y_2 \ldots Y_{d-1} \cdot Y_d$$

where,

$$Y_1 = [X_1(1,*), \ldots, X_1(w,*)]_{1\times w^2}$$

$$Y_d = [X_d(*,1)^T, \ldots, X_d(*,w)^T]_{w^2\times 1}$$

$$Y_i = \begin{bmatrix} X_i & & & & \\ & \ddots & & & \\ & & X_i & & \\ & & & \ddots & \\ & & & & X_i \end{bmatrix}_{w^2\times w^2} \qquad \text{for } i \in [2,d].$$

Part-II: Reduction from TRACE-TI to DET

1. Using **set-multilinear ABP reconstruction** [Klivans,Shpilka(2003)], we compute $Y'_1, \ldots, Y'_d$ such that:

$$f = Y'_1 \cdot Y'_2 \ldots Y'_{d-1} \cdot Y'_d$$

$$Y'_i = T_{i-1}^{-1} \begin{bmatrix} X_i & & & & \\ & \ddots & & & \\ & & X_i & & \\ & & & \ddots & \\ & & & & X_i \end{bmatrix} T_i \text{ for } i \in [2, d-1]$$

Idea: Block-diagonalize the matrices $Y'_2, \ldots, Y'_{d-1}$.

Part-II: Reduction from TRACE-TI to DET

2. For each intermediate matrix, compute blackbox access to circuit computing $c_i \cdot \det(X_i)$ from Y_i'.

3. Use **DET** to compute $\widehat{X_i}$ that satisfies exactly one of the following:

$$X_i = A_i \cdot \widehat{X_i} \cdot B_i$$
$$X_i = A_i \cdot \widehat{X_i}^T \cdot B_i$$

Part-II: Reduction from TRACE-TI to DET

$$Y_i' = T_{i-1}^{-1} \begin{bmatrix} X_i & & & & \\ & \ddots & & & \\ & & X_i & & \\ & & & \ddots & \\ & & & & X_i \end{bmatrix} T_i$$

Part-II: Reduction from TRACE-TI to DET

$$Y_i' = T_{i-1}^{-1} \begin{bmatrix} A_i \widehat{X}_i B_i & & & & \\ & \ddots & & & \\ & & A_i \widehat{X}_i B_i & & \\ & & & \ddots & \\ & & & & A_i \widehat{X}_i B_i \end{bmatrix} T_i$$

Part-II: Reduction from TRACE-TI to DET

$$Y_i' = P_i \begin{bmatrix} \widehat{X_i} & & & & \\ & \ddots & & & \\ & & \widehat{X_i} & & \\ & & & \ddots & \\ & & & & \widehat{X_i} \end{bmatrix} Q_i$$

4. Compute $\widehat{P_i}, \widehat{Q_i}$ for all $i \in [2, d-1]$.

(Ideally, we would want $\widehat{P_i}^{-1} Y_i' \widehat{Q_i}^{-1}$ to be block-diagonal).

Part-II: Reduction from TRACE-TI to DET

5. Using the $\widehat{P_i}, \widehat{Q_i}, Y_i'$'s , we compute $X_2', \ldots X_{d-1}'$ such that:

$$X_2' \cdot X_3' \ldots X_{d-1}' = \alpha \cdot A \cdot X_2 \cdot X_3 \ldots X_{d-1} \cdot B$$

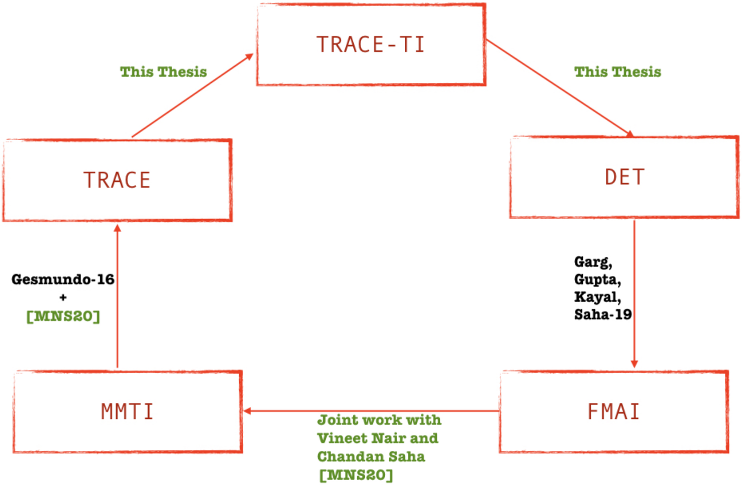
6. Compute X_1', X_d' (using ABP reconstruction techniques):

$$X_1' = \alpha^{-1} \cdot X_1 \cdot A^{-1} \text{ and } X_d' = B \cdot X_d$$

So,

$$X_1 \cdot X_2 \ldots X_{d-1} \cdot X_d = X_1' \cdot X_2' \ldots X_{d-1}' \cdot X_d'$$

Conclusion


TRACE-TI
This Thesis
This Thesis
TRACE
DET
Gesmundo-16
+
[MNS20]
Garg,
Gupta,
Kayal,
Saha-19
MMTI
FMAI
Joint work with
Vineet Nair and
Chandan Saha
[MNS20]

Acknowledgements

I thank my advisor Prof. Chandan Saha for his guidance and support during the course of my research program.

I want to thank Vineet Nair for mentoring me. I also want to thank him for his contributions to the thesis.

Thanks to my labmates Vineet, Nikhil and Bhargav for their wonderful company.

Acknowledgements

The question regarding whether the equivalence test for IMM can be extended to Tr-IMM or not was asked by Avi Wigderson to Vineet Nair at CCC'17 after the presentation of their work on IMM equivalence test.

Christian Ikenymeyer also pointed out that the Tr-IMM polynomial is more interesting to mathematicians compared to the IMM polynomial and encouraged to look at this problem.

Acknowledgements

I want to thank all my friends here at IISc and elsewhere for the wonderful memories I have had with them.

Last but not the least, I want to thank amma, appa, Sankar and Meenu for their love and encouragement :)

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

Nitin Saxena.
Morphisms of rings and applications to complexity.
Indian Institute of Technology Kanpur, 2006.

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

Joshua Abraham Grochow.
Symmetry and equivalence relations in classical and geometric complexity theory.
The University of Chicago, 2012.

Fulvio Gesmundo.

Geometric aspects of iterated matrix multiplication.
Journal of Algebra, 461:42–64, 2016.