CHENNAI MATHEMATICAL INSTITUTE

A study of the rank computation problem for linear matrices

by

Somnath Bhattacharjee

Thesis submitted for the degree *Master of Science* in Computer Science at Chennai Mathematical Institute

Supervisor: Prof. Partha Mukhopadhyay

Graduation: July, 2024

Student number: MCS202221

Dedicated to my mother, whose unwavering support has brought me here, and to my grandfather, whose memory I cherish deeply.

Acknowledgements

Earning a degree is a beautiful journey, filled with guidance and support from many along the way. First and foremost, I would like to thank my mother for her unconditional love, unwavering support, and determination, which have brought me to this point.

I also wish to express my profound gratitude to my advisor, Prof. Partha Mukhopadhyay. Words cannot fully convey my appreciation for his guidance and mentorship. Over the past three years, he has introduced me to the Complexity Theory, helped me develop a deep understanding of the subject. With patience, he has taught me everything from the art of asking the right questions to the proper approach to research. Working with him has been a beautiful learning journey, enriched by our enlightening discussions on various topics beyond academia.

I would also like to thank Prof. Markus Bläser for playing a significant role in my academic journey. He hosted me as an intern at MPI Saarbrücken in the summer of '22 and extended that internship online for an additional year. This period was a pivotal learning experience for me. Thanks to his guidance, I achieved my first publication. Beyond academics, I learned many exciting lessons from Markus, including travel tips in Germany (he explained that on the blackboard in his office!).

A special thanks to Pranjal da, Amit da, and Abhranil da, who became cherished elder brothers before they were my academic advisors. From preparing for PhD applications to providing hints for assignments, extending deadlines, responding to frequent treat requests (not all of which were fulfilled), and engaging in discussions on random topics as well as academic matters, they were my go-to contacts and my favorites throughout this journey.

I would like to thank KV for being both an academic and non-academic guide during my time at CMI. I feel incredibly fortunate to have been taught by such student-friendly and wonderful professors, including Samir, Prajakta, Srivatsan, Nithin, Sumantha da, Pramath, Upendra, Rath, and Ramadas. I am grateful to all of them. I owe a debt of gratitude to CMI for offering many excellent undergraduate and graduate courses that provided me with a strong foundation. I want to thank Prof. Nitin Saxena for supervising my reading project on PIT and introducing me to algebraic complexity. I am also thankful to Prof. Iddo Tzameret and Prof. Ilya Volkovich for guiding me on research projects. Finally, I would like to thank Shubhangi and Swastik, as well as the theory group at the

University of Toronto, for allowing me to join their group as a PhD student.

The most significant aspect of my five-year journey at CMI, especially during the challenging two years of the COVID pandemic, has been the friendships and social life I have cherished. Throughout this time, I have made many friends who have enriched my experience. I want to thank Saswata, my co-author and one of my closest friends, my two (not so) wonderful roommates, Nilava and Spandan, members of the ”Zone group” Abhishek, Arghyadip, Shourjyo, Ankit bhaiya, Anurag, Monsey, Deepro, both Aritras, Sagnik, seniors Swarnadip da, Utsab da, Shubhayan da, and many more. Additionally, I am grateful for friends from outside CMI. Yes, I have many real friends (something I am proud to flex about). I apologize for not being able to mention each of you, and like Fermat, I'd like to shield myself with the excuse that the margin is indeed too small to list the names of all you wonderful people. These five years have been a mixture of various emotions, and you were there for me every time.

I would like to thank all non-academic staffs in CMI including the administrative staffs, cleaning staffs, security and mess people. I acknowledge all the COVID workers and warriors for their endless effort to make the world better place during the COVID time. Finally I thank CMI and NDP (my school) for everything.

Last but not the least, I thank my family and all the well wishers for their constant love and support.

Abstract

One of the most important open problems in algebraic complexity theory is to design an efficient deterministic algorithm for computing the rank of linear matrices (usually stated over a set of commutative variables). At present, we know efficient deterministic algorithms when the variables do not commute, and when the variables partially commute (in a limited sense). Also, for the commutative case, a fully polynomial-time rank approximation is known. In this study, we focus on the approximation problem further. We provide a gentle outline of the known algorithms and then highlight some aspects of an approximation algorithm in the setting of partially commutative variables.

Keywords: Rank computation; Rank approximtion; Partially commutative rank.

Contents

Table of abbreviations

A table containing a list of abbreviations that will be used throughout text.

ABP	Algebrai Branching Program
NC rank	Non-commutative rank
PC rank	Partially-commutative rank
PIT	Polynomial Identity Testing
PTAS	Poly Time Approximation Scheme

Chapter 1

Introduction

1.1 Motivation and Background:

Unlocking the mysteries of the Rank problem stands as a pivotal quest within Complexity Theory. It says given a $s \times s$ polynomial matrix M where the matrix entries are linear in $\mathbb{F}[X_1, \dots, X_n]$,

$$A(\overline{\mathbf{X}}) = A_0 + A_1\overline{\mathbf{X}}_1 + \cdots + A_n\overline{\mathbf{X}}_n$$

we have to determine the rank of A in $\mathbb{F}(\overline{\mathbf{X}})$ efficiently, ie, in polynomial time in terms of s and n. This problem was introduced by Edmonds (1967) and now known as *Edmonds' problem*. Through out this thesis we will assume $\mathbb{F}$ has characteristic 0 and cost of computation of field elements is free.

The commutative rank problem is a fundamental problem, which generalizes several algorithmic problems in algebraic complexity theory and graph theory. It is well known that the Polynomial Identity Testing (PIT) is equivalent with Edmonds' problem (we have shown this in the thesis), hence this problem has a randomised efficient algorithm as PIT is well known for being in RP (Schwartz (1980); Zippel (1979). Hence one can expect derandomization is possible. Further more Kabanets and Impagliazzo (2003) proved derandomising Edmond's problem will lead to super-polynomial lower bounds which is the holy-grail of complexity theory. The graph theoretic aspects were studied by Lovász (1979) which finds relation between rank computation problem and matching and matroid related problems.

The best known deterministic algorithms for this problem are for restricted cases. For example when A_is are rank 1 matrices (Gurjar and Thierauf (2017)) or rank 2 skew-symmetric matrices (Gurjar, Oki, and Raj (2024)). But no sub-exponential time algorithm is known for general cases. Since we don't know how to find exact rank efficiently, it is natural to ask can we efficiently output some number r which approximates the actual rank. This kind of (approximation) algorithm known as deterministic Poly Time Approximation Scheme (PTAS).

Definition 1.1.1. *A PTAS or approximation algorithm for the rank problem will be given $A(\overline{\mathbf{X}})$ and an constant $0 < \varepsilon < 1$, we will output r s.t. $r \leq \mathsf{rk}(A) \leq r(1+\varepsilon)$ in poly($(ns)^{1/\varepsilon}$) time*

It is natural that the error margin ε and the time complexity will be inversely dependent. Thanks to Bläser, Jindal, and Pandey (2018) we have a deterministic PTAS for the rank computation.

The rank problem is also well-studied in the *non-commutative setting*, ie, when the variables $\overline{\mathbf{X}}$ are non-commutative. But for this first we should have some notion of field. Thankfully we have a notion of skew field which contains this non-commutative field and the (non-commutative) rank is well defined there (Cohn (1995); Amitsur (1966); Fortin and Reutenauer (2004)). Unlike the previous case, we have multiple algorithms that computes non-commutative rank (NC rank) efficiently. First using operator scaling Garg, Gurvits, Oliveira, and Wigderson (2019) gave polynomial time algorithm for NC rank. Ivanyos, Qiao, and Subrahmanyam (2015) gave another efficient algorithm based on greedy method which generalises even for higher degree entries. Recently Hamada and Hirai (2020) designed another poly time algorithm for NC rank using convex optimisation.

Clearly the current status of this problem in commutative regime differs very much from non-commutative regime. Even when the question comes to lower bounds for Algebraic Branching Programs (ABP), the knowledge gap is still the same. We have exponential ABP lower bounds for non-commutative polynomials due to Nisan's work, whereas for commutative settings, the best known ABP lower bound result is quadratic from the work of Chatterjee, Kumar, She, and Volk (2019).

This enormous gap in the understandings of non-commutative and commutative regime motivates us for this project. Can we parameterized the commutativity of the variables $\overline{\mathbf{X}}_i$s and get a generalized result in the context of hardness and efficiency, which in the extreme cases (i.e, in the case where no variables commute with each other or all of them commutes) gives the same results we know already? More precisely we partition the variables into k buckets

$$\overline{\mathbf{X}} := \overline{\mathbf{X}}_1 \sqcup \cdots \sqcup \overline{\mathbf{X}}_k$$

And define the commutativity only across the buckets, ie, $\overline{\mathbf{X}}_i$ variables can commute with $\overline{\mathbf{X}}_j$ variables $\iff i \neq j$. Note if we set $k = 1$, we get the non-commutative setting, for $k = n$ we get the commutative setting. Hence this model sets a bridge between this two extreme cases. Recently Klep, Vinnikov, and Volčič (2020) proved the existence of universal skew field $\mathfrak{U}_k$ which contains this *partially commutative* ring. Hence we have the notion of rank in this model (we call this PC rank or partially commutative rank). The PC rank computation also captures the problem of equivalence testing of multitape weighted automata, which is a long standing open problem (Harju and Karhumäki (1991); Worrell (2013). Arvind, Chatterjee, and Mukhopadhyay (2024) gave a $O(n^{k^k})$ algorithm for the PC rank computation. But here the dependency on k is doubly exponential.

1.2 Contribution and Organisation of the Thesis:

The breakthrough result by Ivanyos et al. (2015) motivates the key approaches for rank computation. Based on their idea, Bläser et al. (2018) designed the PTAS for commutative rank. Both the papers uses wong sequence for the correctness of the algorithm. Later Bhargava, Bläser, Jindal, and Pandey (2019) simplified the PTAS and gave a new analysis using simple linear algebraic techniques. Bases on their idea, Chatterjee and Mukhopadhyay (2023) simplified [Ivanyos et al. (2015)]. Based on all these ideas we design a poly(n^k) approximation scheme for PC rank computation.

In this thesis we presented the key ideas behind the greedy approach of [Ivanyos et al. (2015)]. The main idea is find any *witness* for some rank r and greedily try to find another witness for rank $r+1$; if not possible conclude the rank is r. And finally reduce this rank increment step into efficient PIT problem.

We described the preliminary tools in chapter 2. The chapter 3 is dedicated to describe the greedy ideas and the key distinguisher between commutative and non-commutative rank computation efficiency. We described the simplified PTAS for commutative rank in section 3.1 and simplified NC rank computation in section 3.2. In chapter 4 we described the rank computation ideas in partially commutative model. section 4.1 shows the approximation scheme for PC rank; section 4.2 describes the idea to solve PIT in this model given in the work by Arvind et al. (2024) which will essentially lead to their $O(n^{k^k})$ algorithm to find the exact PC rank. And finally we discussed about the open areas in chapter 5

Chapter 2

Preliminaries

2.1 Rank Problem:

Given an $s \times s$ matrix $A \in \mathbb{F}[\overline{\mathbf{X}}]^{s\times s}$ where $\overline{\mathbf{X}}$ is a set of n variables $X_1, \ldots, X_n$ and entries of the matrix A are linear polynomials from $\mathbb{F}[\overline{\mathbf{X}}]$. The commutative rank of A is the rank of A in the field of rational functions $\mathbb{F}(\overline{\mathbf{X}})$. We define this by $\mathsf{crk}(A)$. Here are an alternative definition of commutative rank which can be easily verifiable

Theorem 2.1.1. *Let $A(\overline{\mathbf{X}}) = A_0 + A_1X_1 + \cdots + A_nX_n$ where $A_i \in \mathbb{F}^{s\times s}$, then $\mathsf{crk}(A)$ is the maximum rank obtained by the matrices in the vector space spanned by A_is.*

Corollary 2.1.2. *For any $\overline{\alpha} \in \mathbb{F}^n$, $\mathsf{crk}(A(\overline{\mathbf{X}})) = \mathsf{crk}(A(\overline{\alpha} + \overline{\mathbf{X}}))$*

Similarly we can define the non-commutative rank of A (where the variables $\overline{\mathbf{X}}$ behaves non-commutatively). But first we have to define the notion of the field. We define $\mathbb{F}(\langle\overline{\mathbf{X}}\rangle)$ be the smallest *skew field* which contains $\mathbb{F}\langle\overline{\mathbf{X}}\rangle$, ie, $\mathbb{F}$ adjoining the non-commutative variables $\overline{\mathbf{X}}$. And the non-commutative rank of A, defined by $\mathsf{ncrk}(A)$ is the inner rank (ie, minimum r s.t. A can be written as product of $s \times r$ and $r \times s$ matrix from the field) of A in the field $\mathbb{F}\langle(\overline{\mathbf{X}})\rangle$. For details we refer to Cohn (1995). Note we can define notion of left action right action of $\mathbb{F}\langle(\overline{\mathbf{X}})\rangle$, hence we can define linear independences accordingly. Here is some equivalent definitions of ncrk:

Theorem 2.1.3. *The following statements are equivalent:*

1. *$\mathsf{ncrk}(A) = r$*
2. *Left row rank of $A = r$*
3. *Right row rank of $A = r$*
4. *Left column rank of $A = r$*
5. *Right column rank of $A = r$*

6. *the largest full rank minor of A has dimension $r \times r$*

For the detailed proofs we refer to Cohn (1995) again. Now we need an alternative definition of ncrk analogus to theorem 2.1.1. Now let $A = A_0 + \sum A_i X_i$. For $d \in \mathbb{N}$ let $T_d(A)$ be the $sd \times sd$ matrix space spanned by $T(Y) := A_0 \otimes Y_0 + \sum A_i \otimes Y_i$ for all $Y_i \in \mathbb{F}^{d\times d}$. Now let $r_d(A)$ be the maximum rank of the matrices in $T_d(A)$. Then the following lemma says:

Lemma 2.1.4. *(Regularity lemma, Ivanyos et al. (2015); Derksen and Makam (2015); Fortin and Reutenauer (2004)) $\mathsf{ncrk}(A) = \lim_{d\to\infty} \frac{r_d(A)}{d}$. Futhermore that sequences reaches to ncrk for $d \leq s+1$*

Ivanyos et al. (2015) gave a contructive version of the above lemma

Lemma 2.1.5. *Suppose there is an assignment in $T_d(A)$ s.t. the rank $> rd$, then we can efficiently construct a new assignment for $T_d(A)$ of rank $\geq (r+1)d$*

Now the rank problem will be given A, we have to compute rank of A (commutative and non-commutative model both) efficiently, ie, $poly(s, n)$ time. We called the above problems by *singular* and *NC-singular* problem. The above lemma was crucial for Ivanyos et al. (2015) to find $\mathsf{ncrk}(A)$ efficiently. The above lemma also called blow-up definition.

Finally due to Fortin and Reutenauer (2004) we have

Lemma 2.1.6. $\mathsf{crk}(A) \leq \mathsf{ncrk}(A) \leq 2\mathsf{crk}(A)$

2.2 Partially Commutative Model:

Now suppose the variable set $\overline{\mathbf{X}}$ partitioned into k buckets: $\overline{\mathbf{X}} := \overline{\mathbf{X}}_1 \sqcup \overline{\mathbf{X}}_2 \sqcup \cdots \sqcup \overline{\mathbf{X}}_k$ with the property that each variables in $\overline{\mathbf{X}}_i$ is non-commutative with the variables in $\overline{\mathbf{X}}_i$ but commutes with the variables outside $\overline{\mathbf{X}}_i$. We will call this partition by *commutativity partition.* Note this model generalize the commutativity nature of both the non-commutative and commutative model we saw already, when $k = n$, we have our commutative model, when $k = 1$ we are in non-commutative setting.

Now if we need to find rank in this setting, we have to define what that is, and for that we need the underlying field. Thanks to Klep et al. (2020), we can construct a field which contains $\mathbb{F}$ adjoining $\overline{\mathbf{X}}_i$s preserving their commutative nature (we need the fact that $\mathbb{F}$ has characteristic 0). Given the partition on $\overline{\mathbf{X}}$ we denote this field by $\mathfrak{U}_k$.

Now the rank of A in $\mathfrak{U}_k$, denoted by $\mathsf{rk}_k(A)$ is the inner rank of A. Easy to note $\mathsf{ncrk} = \mathsf{rk}_1$ and $\mathsf{crk} = \mathsf{rk}_n$ Now here is the following remark:

Remark 2.2.1. *theorem 2.1.3 is true even in Partially commutative setting.*

Finally some generalization of results from commutative fields

Lemma 2.2.1. *Assume $n \geq n$. If $A_{n\times n}$ is a full rank matrix and $B_{n\times n}$ r rank matrix, then rank of AB is r. The same thing happens for left multiplication by $m \times m$ matrix of rank r*

Proof. Suppose $B = P_{n\times r}Q_{r\times n}$. Then $AB = APQ = (AP)_{n\times r}Q_{r\times n}$. Hence rank of $AB \leq r$.

Now suppose $AB = M_{n\times(r-1)}N_{(r-1)\times n}$ then $B = A^{-1}MN$ hence B has rank $\leq r-1$ which is a contradiction One thing can be noted that *most of the rank properties* which are true for non-commutative settings and commutative settings, ie, two extreme cases, are also true for the generalized case, ie, partially commutative setting. And the proves follows analogously. ∎

Lemma 2.2.2. *(Rank-nullity theorem) If dim of the kernel of a matrix $A_{n\times} \geq r$, then rank of $A \leq n - r$*

Proof. So we know there is $B_{n\times r}$ rank r matrix s.t. $AB = 0$

Since rank of B is r we have invertible matrix $M_{n\times n}$ s.t. $MB = \begin{bmatrix} I_r \\ S \end{bmatrix} =: C$ (M is the row operation matrix)

So $0 = AB = AM^{-1}C$. Hence first r columns of AM^{-1} is 0. So rank of $AM^{-1} \leq n - r$. Hence from lemma 1 we are done. ∎

Now suppose $\overline{\mathbf{X}}$ has two partitions and one is the refinement of the other, $\overline{\mathbf{X}} = \bigsqcup_{i=1}^{k} \overline{\mathbf{X}}_i = \bigsqcup_{j=1}^{k'} \overline{\mathbf{X}}'_j$ and $k' \geq k$ and for all j, $\overline{\mathbf{X}}'_j \subseteq \overline{\mathbf{X}}_i$ for some i then we have generalisation of lemma 2.1.6.

Lemma 2.2.3. $\mathsf{rk}_{k'}(A) \leq \mathsf{rk}_k(A) \leq 2\mathsf{rk}_{k'}(A)$

Now we will generalise the blow-up definition for partially commutative model. We fix the commutativity partition $\overline{\mathbf{X}} = \overline{\mathbf{X}}_1 \sqcup \cdots \sqcup \overline{\mathbf{X}}_k$ and $A = A_0 + \sum_{i=1}^{k} \sum_{X\in\overline{\mathbf{X}}_i} A_X X$. For $\overline{d} := (d_1, \ldots, d_k) \in \mathbb{N}^k$ with $d' = d_1 \times \cdots \times d_k$, let $T_{\overline{d}}$ be the $sd' \times sd'$ matrix spaces spanned by the matrices of the form $T(Y) := A_0 \otimes I_{d'} + \sum_{i=1}^{k} \sum_{x\in\overline{\mathbf{X}}_i} A_X \otimes I_{d_1} \otimes \cdots \otimes I_{d_{i-1}} \otimes Y_X \otimes I_{d_{i+1}} \otimes \cdots \otimes I_{d_k}$ where $Y_X \in \mathbb{F}^{d_i\times d_i}$. Now let $r_{\overline{d}}$ is the maximum rank of the matrices in $T_{\overline{d}}$, then

Theorem 2.2.4. *Arvind et al. (2024) For each $d_i \geq s+1$, $\mathsf{rk}_k(A) = \frac{r_{\overline{d}}(A)}{d'}$*

We are interested in finding $\mathsf{rk}_k(A)$ given commutativity partition and A. Thanks to Arvind et al. (2024), we know this can be done in $(ns)^{2^{O(k \log k)}}$ time which is not good when k is $\omega(1)$. We want to improve this via help of approximation.

2.3 ABP and PIT:

In algebraic complexity, we may have discussed for various computation models, but for this thesis's purpose we will discuss about ABPs:

Definition 2.3.1. *An **Algebraic Branching Program (ABP)** over $\mathbb{F}[\overline{\mathbf{X}}]$ is a directed single source single sink multilayered graph, each edges are weighted by linear polynomials from $\mathbb{F}[\overline{\mathbf{X}}]$, and each edge is from the vertices of some layer i to the vertices of layer $i+1$. Furthermore the first and the last layer contains only one vertex, namely source and sink respectively.*

For each path p in the ABP, the weight of the path, $wt(p) = \prod_{e \in p} wt(e)$

For each vertex (also called node) v in the ABP with source s, the polynomial computed by v is $\sum_{p: s \text{ to } v \text{ path}} wt(p)$

The polynomial computed at the sink will be called the polynomial computed by the ABP.

The size of the largest layer of the ABP will be called the width of the ABP and the length of the ABP will be number of layers -1. The size of the ABP will be number of edges in the ABP which is clearly bounded by above by length $\times$ width.

Note the above computational is independent of the commutative behaviour of the variables. This model can work on any commutativity partition over the variables.

We will also talk about another computational model, namely Iterated Matrix Multiplication (IMM) which is just multiplication of matrices whose entries are linear polynomials. And clearly this model can work on any commutativity partition. But this model is equivalent to ABPs, so instead of stating another definition, we just write the following theorem which is easy to verify:

Theorem 2.3.2. *For a polynomial $p(\overline{\mathbf{X}}) \in \mathbb{F}[\mathbf{x}]$:*

there exists a sequence of matrices $M_0, \ldots, M_d$ whose entries are linear polynomials from $\mathbb{F}[\overline{\mathbf{X}}]$ and M_0 is $1 \times w$ matrix, M_d is $w \times 1$ matrix, and all others M_is are $w \times w$ matrices and $p(\overline{\mathbf{X}}) = M_0 M_1 \ldots M_d \iff p(\overline{\mathbf{X}})$ can be computed by length d, width w ABP

The idea of homogenisation of ABPs are well-known and can found in the survey by (Shpilka and Yehudayoff, 2010, chapter 2). That can be generalized to Partially commutative model as well.

Lemma 2.3.3. *Given an ABP of length l width w computing a polynomial f of degree d, we can efficiently construct another multi-sink ABP of width dw and length l which computes each homogeneous parts of f. More over one can construct ABP for the coefficients in poly size. Further more each node in the ABP computes a homogeneous polynomial and each edge in the ABP is linear homogeneous.*

The Polynomial Identity Testing (PIT) problem for ABP (or any polynomial class) is given an ABP (or any instance from that class), we have to check whether that ABP is identically zero or not efficiently. Here efficiently means polynomial in size of the input.

Randomized algorithm for PIT in known. But for commutative settings, no deterministic algorithm is known in general. But again for non-commutative settings, PIT for ABP can be done in poly time due to Raz and Shpilka (2004).

Note this advancement in non-commutative setting plays a major role to compute the ncrk efficiently.

The following result is slightly stronger than the above result and useful to find the ncrk:

Theorem 2.3.4. *Arvind, Mukhopadhyay, and Srinivasan (2008) Given a noncommutative ABP of width w and length d computing a polynomial $p(\overline{\mathbf{X}})$, there is a deterministic poly(n, d, w) time algorithm which outputs a nonzero assignment from $d+1 \times d+1$ matrix algebra if p is non-zero.*

We called the number of monomials in a polynomial $p(\overline{\mathbf{X}})$ by sparsity of that polynomial denoted by $||p(\overline{\mathbf{X}})||$. Note if the sparsity is less, then the PIT will be easy, the details of the following theorem can be found in (Saxena, 2009, theorem 2.1)

Theorem 2.3.5. *In commutative settings, PIT for any polynomial class with the promise of sparsity bounded by m, can be done in poly time on input size and m. Furthermore we can find an non-zero assignments as well if the polynomial is non-zero*

Since constant length ABP computes constant degree polynomials we have the following corollary:

Corollary 2.3.6. *PIT for constant length commutative ABP is in P. Furthermore we can find non-zero assignment if the polynomial is non-zero. Furthermore this is true in partially commutative model as well.*

2.4 Reducing non-linear matrices to linear case

Here we will show any matrix with polynomial entries can be converted to a linear matrix preserving the co-rank described in Arvind, Chatterjee, Ghosal, Mukhopadhyay, and Ramya (2023a). Let $A(\overline{\mathbf{X}})$ be a matrix polynomial with entries A_{ij}. And each A_{ij} has a width w length $l-2$ ABP. That means from theorem 2.3.2, we have a sequence of $w \times w$ linear matrices $L_{ij}^{(1)}, \dots, L_{ij}^{l}$ s.t. the $1,1$th entry of $\prod_k L_{ij}^{(k)}$ is A_{ij}. We first show A_{ij} can be computed by inverse of linear matrix.

Let $m = w(l+1)$. Define

$$M^{(ij)}(\overline{\mathbf{X}}) := \begin{bmatrix} 0 & L_{ij}^{(1)} & 0 & \dots & 0 \\ 0 & 0 & L_{ij}^{(2)} & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & & \dots & L_{ij}^{(l)} \\ 0 & & & \dots & 0 \end{bmatrix}_{m \times m}$$

Note $(M^{(ij)})^m = 0$. Further more the $1,m$th entry of $I + M^{(ij)} + (M^{(ij)})^2 + \dots + (M^{(ij)})^{m-1}$ is A_{ij}

Lemma 2.4.1. *there exists row and column vector $U^{(ij)}, V^{(ij)}$ with field entries s.t. $U^{(ij)}(I_m - M^{(ij)})^{-1} V^{(ij)} = A_{ij}$*

Proof. Note $(M^{(ij)})^m = 0$. Further more the $1,m$th entry of $I + M^{(ij)} + (M^{(ij)})^2 + \dots + (M^{(ij)})^{m-1}$ is A_{ij}

Check $(I_m - M^{(ij)})^{-1} = I + M^{(ij)} + (M^{(ij)})^2 + \dots + (M^{(ij)})^{m-1}$. Hence from the fact stated above we are done ■

Let M be the diagonal block matrix of size $s^2 m \times s^2 m$ whose all $m \times m$ diagonals are $I_m - M^{(11)}, I_m - M^{(12)}, \dots, I_m - M^{(ss)}$ Finally define the block matrix for some B, C

$$L := \begin{bmatrix} M & B_{s^2 m \times m} \\ C_{m \times s^2 m} & 0 \end{bmatrix}_{s^2 m + m \times s^2 m + m}$$

Theorem 2.4.2. *We can set B, C from field entries such a way that co-rank of L and A are same*

Proof. Note M is full rank. Hence from lemma 3.1.2 if we set $CM^{-1}B = A$ then we are done. Note M^{-1} is the block diagonal matrices with entries $(I - M^{(ij)})^{-1}$. Using the above lemma we can plugin the $U^{(ij)}, V^{(ij)}$ in B, C such a way that $CM^{-1}B$ gives A. ■

Remark 2.4.1. *This idea can be used to find rank of matrices with entries in the form of ABPs. This will do the job when we are computing the exact rank. But this will fail for rank approximation.*

Corollary 2.4.3. *Rank computation for matrices with entries in the form of ABPs reduces to Edmonds' problem*

Chapter 3

Greedy Approach for Rank computation

3.1 Rank approximation in commutative setting:

To tackle the rank problem we will go with most common idea: greedy way. We will find an assignment for $\overline{\mathbf{X}}$ so that assigning that the matrix rank becomes r, and then we will try to find another assignment from the previous one so that the new assignment makes the rank $r+1$. If we can not find such assignment efficiently we will say we are *almost* done.

Intuitively PIT should have a deeper connection with the rank problem. PIT can be thought of a decision version of the rank computation problem. It is not clear whether efficient PIT for ABP solves the rank problem efficiently. In the next subsection we will show that.

Note though it might be intuitive, the reverse direction is not clear as well, ie, how rank problem will imply PIT for ABP. Here is the following result which solves this case:

Theorem 3.1.1. *Mahajan and Vinay (1997) Each s size commutative ABP, we can construct a $poly(s) \times poly(s)$ matrix M with linear polynomial entries in $poly(s)$ time s.t.* $\det(M) =$ *the polynomial computed by the ABP*

Hence PIT for ABP reduces to check whether a matrix is full rank or not.

3.1.1 Decision to Search for commutative rank:

Note first we can assume all the entries in A are homogeneous as if there is some constant term, we can homogenise that preserving the rank. Details can be found in Bhargava et al. (2019)

Suppose we have an assignment $\overline{\alpha} := (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}^n$ s.t. $A(\overline{\alpha})$ has rank r. Then clearly $\mathsf{crk}(A) \geq r$ from theorem 2.1.1. Now we want to check whether rank increment is possible or not. Depending on that we can answer rank problem

1. Find another assignment $\overline{\beta}$ s.t. $A(\beta)$ has rank $\geq r+1$. In this case we can recurs on $r+1$

2. If we can not find such assignment declare $A(x)$ has rank r. We know from theorem 2.1.1 we are correct.

Now consider the matrix $A(\overline{\alpha}+\overline{\mathbf{X}})$. Since rank of $A(\overline{\alpha})$ has rank $1 \leq r \leq s$, we have invertible linear transformations U, V s.t. $UA(\overline{\alpha})$ is a diagonal matrix whose first r diagonals are 1, others 0. Hence we can write

$$UA(\overline{\alpha}+\overline{\mathbf{X}})V := \begin{array}{r} r \text{ rows} \left\{ \right. \\ n-r \text{ rows} \left\{ \right. \end{array} \underbrace{\underbrace{\begin{bmatrix} I_r - L(\overline{\mathbf{X}}) & B(\overline{\mathbf{X}}) \\ D(\overline{\mathbf{X}}) & C(\overline{\mathbf{X}}) \end{bmatrix}}_{}}_{r \qquad \text{n-r}}$$

Where B, C, D are matrices with linear homogeneous polynomial entries.

WLOG we can assume $U, V = I_s$, else we can rename UAV by A.

Now it is a standard fact that

Lemma 3.1.2. *for any square matrix* $B_{s\times s} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ *where* B_{11} *is a* $r \times r$ *full rank matrix, then* $\mathsf{rk}(B) = r + \mathsf{rk}(B_{22} - B_{21}B_{11}^{-1}B_{12})$. *Furthermore* $\det(B) = \det(B_{11}) \times \det(D[I_r - L]^{-1}B)$

Hence from the above lemma and corollary 2.1.2 we can say $\mathsf{crk}(A) = r + \mathsf{crk}(C - D[I_r - L]^{-1}B)$. Note here the inverse is taken over the function field $\mathbb{F}(\overline{\mathbf{X}})$.

Furthermore note the i, j^{th} entry of $C - D[I_r - L]^{-1}B$ is the determinant $\begin{vmatrix} I_r - L & B_j \\ C - D_i & C_{ij} \end{vmatrix}_{r+1\times r+1}$ where D_i is the i^{th} row of D and B_j is the j^{th} column of B. Hence each entry is a polynomial of degree atmost r.

Hence we get an if and only if condition for rank increment:

$$C - D[I_r - L]^{-1}B = 0 \implies \mathsf{crk}(A) = r, C - D[I_r - L]^{-1}B \neq 0 \implies \mathsf{crk}(A) > r \quad (3.1)$$

Now consider the power series expansion

$$C - D[I_r - L]^{-1}B = C - \sum_{t\geq 0} DL^tB$$

Note each term DL^tB in the power series is a $n-r \times n-r$ matrix with entries from $t+2$ degree homogeneous polynomial. Since the degree is increasing and the entries are homogeneous we can say

$$C - D[I_r - L]^{-1}B = 0 \iff \forall t \geq 0 \quad DL^tB = 0 \quad \&C = 0$$

Now since $D[I_r - L]^{-1}B$ has entries of degree atmost $r \leq s$, we can check the power series only upto s many entries. So we have

$$D[I_r - L]^{-1}B = 0 \iff \forall t \in [s] \quad DL^tB = 0 \quad \&C = 0 \tag{3.2}$$

Now from theorem 2.3.2, we can see the rank increment step reduces to PIT.

Now we have to see how to find an assignment for rank $r+1$ from α. Now suppose $\overline{\beta}$ is an assignment s.t. for some $t \in [s]$, DL^tB or C has i, j^{th} entry non-zero (we can find this $\overline{\beta}$ efficiently from corollary 2.3.6). Now we can say $A(\overline{\alpha} + t\overline{\beta})$ has a minor of rank $r+1$ where t is a new variable. Furthermore we know this minor, hence for any assignment of t, we can find its determinant. Also determinant of that minor is a non-zero univariate polynomial of degree atmost s. So there exists $y \in [s+1]$ s.t. if we substitute y for t, we get a non-zero value of the minor. Hence $\overline{\alpha} + y\overline{\beta}$ will be the assignment for rank $r+1$ which can be found efficiently (provided we can find $\overline{\beta}$ efficiently) by the process discussed above.

Hence the algorithm will be:

1. If A is zero, output $\mathsf{crk}(A) = 0$. Else set $r = 1$ and find a non-zero assignment s.t. the $1, 1^{\text{th}}$ entry becomes non-zero, name it $\overline{\alpha}$
2. For r: compute D, B, L, C from $A(\overline{\alpha} + \overline{\mathbf{X}})$ from the above way.
3. For each $t \in [s]$, check whether DL^tB or $C = 0$ or not.
4. Suppose we find for some t, DL^tB or $C = 0$, then we will find an assignment for rank $r+1$ and call it new $\overline{\alpha}$ and will set r to $r+1$ and will go back to step 2
5. Else we will output $\mathsf{crk}(A) = r$

The correctness will follow from eqs. (3.1) and (3.2). Now clearly if PIT for ABP is in P, then the algorithm is efficient.

3.1.2 Power series truncation implies rank approximation:

In the previous algorithm the only problem was we can do PIT for DL^tT when t is $\omega(1)$. But can we say if we check for constantly many ts, r is very closed to the rank? The answer is yes and we will see how.

Here is the crucial lemma that generalise lemma 3.1.2

Lemma 3.1.3. *(Bläser et al., 2018, lemma 18) for any square matrix* $B_{s\times s} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ *where* B_{11} *is a* $r \times r$ *full rank matrix. And first* k *terms in the following series is* 0

$$B_{22},\ B_{21}B_{12},\ B_{21}B_{11}B_{12},\ B_{21}B_{11}^2B_{12}, \dots,\ B_{21}B_{11}^iB_{12}, \dots$$

then $\mathsf{rk}(B) = r + r/k$*.*

Now assume again $\overline{\alpha}$ is an assignment that gives rank r. Let D, B, L, C be the same as defined in the above subsection.

Finally we can have the following theorem:

Theorem 3.1.4. *If first* k *terms of the series*

$$C,\ DB, \dots,\ DL^tB, \dots$$

is 0*, then* $r \leq \mathsf{crk}(A) \leq r + r/k$

Proof. Assume first k terms are 0. From the lemma 3.1.3, we know that if first k terms of the following seies is 0, then we are done

$$C, DB, D(I-L), \dots, D(I-L)^iB$$

We will prove the i^{th} term of the above series is 0 for $i \leq k$. When $i = 1, 2$, the term is C or DB, hence it is 0. Else the i^{th} term is $D(I-L)^{i-2}B = \sum_{e=0}^{i-2} c_e DL^eB$ where c_e is some non zero constant. Note all the DL^eB are 0. Hence the i^{th} is zero as well ∎

Now from the above theorem suppose for $\varepsilon > 0$ we set $k = \left\lceil \frac{1}{\varepsilon} \right\rceil$, then checking PIT for first k terms in the power series will give

$$r \leq \mathsf{crk}(A) \leq r(1 + 1/k)$$
$$\implies \mathsf{crk}(A)[1-\varepsilon] \leq r \leq \mathsf{crk}(A)$$

So if ε is constant, then k is constant, hence checking PIT will be easy. So the final algorithm will be: given A, ε

1. If A is zero, output $\mathsf{crk}(A) = 0$. Else set $r = 1$ and find a non-zero assignment s.t. the $1,1^{\text{th}}$ entry becomes non-zero, name it $\overline{\alpha}$. Set $k = \left\lceil \frac{1}{\varepsilon} \right\rceil$
2. For r: compute D, B, L, C from $A(\overline{\alpha} + \overline{\mathbf{X}})$ from the above way.
3. For each $t \in [0, \dots, k-2]$, check whether DL^tB or $C = 0$ or not.
4. Suppose we find for some t, DL^tB or $C = 0$, then we will find an assignment for rank $r+1$ and call it new $\overline{\alpha}$ and will set r to $r+1$ and will go back to step 2 (we know $\mathsf{crk}(A) > r$ in that case)
5. Else we will output r (we know we have achieved ε approximation)

3.2 Exact non-commutative rank computation:

Here the idea will be again the same. We will find a witness $\alpha \in \mathbb{F}^{d\times d}$ for rank r, then will check whether rank increment possible or not efficiently.

3.2.1 Rank-increment:

Let $d \geq s+1$. Define Z_i to be the $d \times d$ variable matrix $\mathbf{Z}_i := \begin{bmatrix} Z_{11}^{(i)} & \dots & Z_{1n}^{(i)} \\ \vdots & \ddots & \vdots \\ Z_{n1}^{(i)} & \dots & Z_{nn}^{(i)} \end{bmatrix}$. We mention that here $Z_{jk}^{(i)}$ variables are non-commutative.

$$\tilde{A}_d = A_0 \otimes I_d + \sum_{i=1}^{n} A_i \otimes Z_i$$

Lemma 3.2.1. *Chatterjee and Mukhopadhyay (2023)* $\mathsf{ncrk}(\tilde{A})_d = d \times \mathsf{ncrk}(A)$

Based on the above lemma we can design the rank increment step. Suppose $(\alpha_1, \dots, \alpha_n) \in (\mathbb{F}^{d\times d})^n$ be a witness for rank r at dth level, ie, $\tilde{A}_d(\overline{\alpha})$ has rank rd (Note since $d \geq s+1$, this definition of witness makes sense due to lemma 2.1.4).

Hence WLOG

$$\tilde{A}(\overline{\mathbf{Z}} + \overline{\alpha}) = \begin{bmatrix} I_{rd} - L(\overline{\mathbf{Z}}) & B(\overline{\mathbf{Z}}) \\ D(\overline{\mathbf{Z}}) & C(\overline{\mathbf{Z}}) \end{bmatrix}$$

Where L, B, C, D are constant free polynomial matrices with linear entries. Hence from Gaussian elimination we have the following lemma analogous to lemma 3.1.2

Lemma 3.2.2. $\mathsf{ncrk}(\tilde{A}_d) = rd + \mathsf{ncrk}(C - D[I-L]^{-1}B)$

Again we know $[I-L]^{-1} = \sum_{i\geq 0} DL^iB$. Note the $(t+2)$th degree homogeneous part in the sum is DL^tB (C if $t=-1$) as C, L, B, D are linear homogeneous. Hence by truncating the higher degree terms we can say $C - D[I-L]^{-1}B = 0 \iff \sum_{i=0}^{rs} DL^iB = 0 \iff C, DB, \dots, DL^{rs}B = 0$ (since the inverse can have degree atmost rs)

Now from theorem 2.3.4 we know we can check $C, DB, \dots, DL^{rs}B = 0$ or not efficiently. Hence from lemma 3.2.1 we can decide $\mathsf{ncrk}(A) > r$ or not efficiently.

3.2.2 Finding witness for higher rank and Blow-up control:

Now if for some t, DL^tB is non-zero then from (Chatterjee and Mukhopadhyay, 2023, corolary 20)

Theorem 3.2.3. *We can find $(\beta_1, \ldots, \beta_n)$ where β_i is a $2rd^2 \times 2rd^2$ matrix with entries from $\mathbb{F}$ in poly(n, r, d) time s.t. $\tilde{A}_{2rd^2}(\overline{\beta})$ has rank $> r$, ie, $r + 1$ rank witness at $2rd^2$ level*

Proof. Note $C - D \sum_{i=0}^{rs} B$ is an ABP of length d on nd^2 many non-commutative variables. Using theorem 2.3.4 We can find an assignment $\gamma_1^{(11)}, \ldots, \gamma_{dd}^{(1)}, \ldots, \gamma_{d^2}^{(n)}$ of dimension $2rd \times 2rd$ (we can pad 0s if necessary) s.t. assigning this gives non-zero outputs. Note we can rewrite the γs as n many matrices $\gamma^{(k)}$ with entries $\gamma_{ij}^{(k)}$ as block matrices. So over all $\gamma^{(k)}$s are $2rd^2 \times 2rd^2$ size

Hence we will lift it to $\tilde{A}_{2rd^2}$. We know assigning $\alpha_i \otimes I_d + t\gamma^{(i)}$ to Z_i for a new variable t will give a non-zero output. Now this is univariate PIT for bounded degree. We can find a value for t easily. ■

Note the blow up in the dimension of the witness can lead to an exponential size witness in the end which will not be efficient. So we have to control the blow-up to find efficient algorithm.

For that we again will take the help from Ivanyos et al. (2015)

Theorem 3.2.4. *Given a witness $\overline{\beta}$ of rank r at dth level, we can find a witness $\overline{\gamma}$ at $r + 1$th level for rank r*

Proof. Consider a sub-matrix A' in $\tilde{A}_d$ such that $\mathsf{ncrk}(A')$ is at least rd. From each matrix in the tuple $\overline{\beta}$, remove the last row and the column to get another tuple $\overline{\beta'}$. We claim that the corresponding sub-matrix A'' in $\tilde{A}_d(\overline{\beta'})$ is of rank $> (r-1)(d-1)$ as long as $d > r+1$. Otherwise

$$\begin{aligned} \mathsf{ncrk}(A') &\leq \mathsf{ncrk}(A'') + 2r \\ &\leq (r-1)(d-1) + 2r \\ &\leq rd + 1 + r - d < rd && \text{(contradiction)} \end{aligned}$$

Now we can use the lemma 2.1.5 on the tuple $\overline{\beta'}$ to obtain another witness of dimension $d - 1$ which is a witness of rank r. Applying the procedure repeatedly, we can control the blow-up in the dimension within $r + 1$ ■

Hence applying theorem 3.2.3 we can find a higher rank witness and then reduce it to atmax sth level using theorem 3.2.4

Corollary 3.2.5. *Given $\overline{\beta}$ s.t. $DL^tB \neq 0$, we can find $\overline{\gamma}$ of witness of rank $r' > r$ at $r' + 1$th level*

3.2.3 Final algorithm

So the final algorithm will be: On input $A(\overline{\mathbf{X}})_{s \times s}$

1. Set $r = 1, d = 2$. Start with $\tilde{A}_d$, find $\overline{\alpha} \in (\mathbb{F}^{2\times 2})^n$ s.t. the top left 2×2 minor in $\tilde{A}(\overline{\alpha})$ is identity, others are 0

2. For $r, d, \tilde{A}_d$ construct L, B, D, C defined in the above way. Using theorem 2.3.4 check for $C, DB, \ldots, DL^{rs}B = 0$. If 0 output rank r and break.

3. Else using corollary 3.2.5 find the witness at r'th level of rank r'. Set $r = r'$, $d = r' + 1$, call this new witness $\overline{\alpha}$, repeat step 2

We know this algorithm will terminate as in the step 2, the algorithm will break if we reach the rank, and we showed that step 3 greedily reaches to the NC rank. Hence the correctness follows as well. The time complexity is polynomial as time taken by each step is polynomial and repeatition of step 2 is atmax s times as in each time the value for r is incremented and $r \leq s$ always.

Here is a small remark: If we were interested in rank approximation, then note we can check upto k many terms in step 2 and approximate the rank of $\tilde{A}_d$ upto $\frac{1}{k}$ as we can prove lemma 3.1.3 works in non-commutative setting as well. Which will lead us to the $\frac{1}{k}$ rank approximation for A as well (by dividing the entire equation by d). But since we can do the entire PIT we don't need approximation, but this idea will be generalized to Partially commutative model.

Chapter 4

Partially Commutative Rank Computation

4.1 Decision to Search:

In this chapter we will generalize the two algorithms stated in the last chapter. Hence the idea will be very much similar with those cases.

Let $\overline{\mathbf{X}} := \overline{\mathbf{X}}_1 \sqcup \cdots \sqcup \overline{\mathbf{X}}_k$ a k-commutativity partition given. We write

$$A(\overline{\mathbf{X}}) = A_0 + \sum_{i=1}^{k} \sum_{X \in \overline{\mathbf{X}}_i} A_X X \tag{4.1}$$

Now we will give a more generalized version of lemma 3.2.1. Let for $i \in [k]$ and $X \in \overline{\mathbf{X}}_i$, let $d_i \geq s$, define the variable matrix $\mathbf{Z}_i^{(X)} = \begin{bmatrix} Z_{i,11}^{(X)} & \dots & Z_{i,1d}^{(X)} \\ \vdots & \ddots & \vdots \\ Z_{i,n1}^{(X)} & \dots & Z_{i,nn}^{(X)} \end{bmatrix}$. Here $Z_{i,jk}^{(X)}$ commutes with $Z_{i',j'k'}^{(Y)} \iff i \neq i'$. Note this define another k-commutativity partition over $\mathbf{Z}$ variables. We fix this partition. Finally define

$$\tilde{A}_{d_1,\ldots,d_k} := A_0 \otimes I_{d'} + \sum_{i=1}^{k} \sum_{x \in \overline{\mathbf{X}}_i} A_X \otimes I_{d_1} \otimes \cdots \otimes I_{d_{i-1}} \otimes Z_i^{(X)} \otimes I_{d_{i+1}} \otimes \cdots \otimes I_{d_k} \tag{4.2}$$

where $d' = d_1 d_2 \ldots d_k$. Now the following theorem is expected

Theorem 4.1.1. $d' \mathsf{rk}_k(A) = \mathsf{rk}_k(\tilde{A}_{\overline{d}})$

4.1.1 Rank increment:

Let $\overline{\alpha}$ be an assignment for $\mathbf{Z}$ in eq. (4.2) s.t. $\mathsf{rk}(\tilde{A}_{\overline{d}}) = rd'$. We call this assignment as witness for rank r at $\overline{d}$ level. Again WLOG we can write

$$\tilde{A}_{\overline{d}}(\mathbf{Z}+\alpha) = \begin{bmatrix} I_{rd'} - L(\mathbf{Z}) & B(\mathbf{Z}) \\ D(\mathbf{Z}) & C(\mathbf{Z}) \end{bmatrix}$$

Where L, B, C, D are polynomial matrices with linear constant free entries. Now the following lemma follows from Gaussian elimination

Lemma 4.1.2. $\mathsf{rk}_k(A) > r \iff \mathsf{rk}_k(\tilde{A}_{\overline{d}}) > rd' \iff C - D[I-L]^{-1}B \neq 0$

Further more from power series expansion

$$\begin{aligned} C - D[I-L]^{-1}B = 0 &\iff C - \sum_{t\geq 0} DL^tB = 0 \\ &\iff C - \sum_{t=0}^{d's} DL^tB = 0 \qquad \text{(as degree can be atmax } sd'\text{)} \\ &\iff C, DB, \ldots, DL^{sd'}B = 0 \\ &\qquad \text{(as } D, L, C, T \text{ are linear homogeneous)} \end{aligned}$$

Hence the problem reduces to PIT for ABP. But note in the above all D, L, B has size d', if all d_is are polynomial in n, then $d' = O(n^k)$

4.1.2 Finding higher rank witness and Blow-up control:

Now suppose we find some assignment $\overline{\gamma}$ s.t. $DL^tB = 0$ for some t, this will assure rank increment is possible. Now we need to find higher rank witness. This following theorem is analogous to theorem 3.2.3

Theorem 4.1.3. *We can find an witness $\overline{\beta}$ from $\overline{\gamma}$ of rank $> r$ at $(2rd_1^2, \ldots, 2rd_k^2)$*

And similar to theorem 3.2.4 we can control the blow up

Theorem 4.1.4. *We can find an witness of rank $r+1$ at $(\underbrace{r+1, \ldots, r+1}_{k \text{ many}})$ level*

Now finally we can show the lemma 3.1.3 works in any partially commutative field

Theorem 4.1.5. *Given a matrix $B_{n\times n} := \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ in a skew field $\mathfrak{U}_k$ where B_{11} is $r \times r$ full rank and $B_{22}, B_{21}B_{12}, B_{21}B_{11}B_{12}, \ldots, B_{21}B_{11}^{k-2}B_{12}$ all are 0, $k \geq 1$*

Then B has rank $\leq r + r/k$

Proof. Suppose B_{12} has rank $\leq r/k$, then $\begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix}$ has r linear independent columns (under left or right action) and $\begin{bmatrix} B_{12} \\ B_{22} \end{bmatrix}$ has atmax r/k many linear independent columns. So rank of $B \leq r + r/k$. We can say the same if B_{21} has rank $\leq r/k$.

Now suppose rank of $B_{12} := l > r/k$. We know the rank nullity theorem holds from lemma 2.2.2

Let $C_1^{(i)}, \dots, C_{n-r}^{(i)}$ are the columns of $B_{11}^{i-1}B_{12}$.

Assume $\{C_j^{(i)}\}$ has atleast $(k-1)l$ many linear independent vectors.

That means there exists $(k-1)l$ many linear independent vectors in the kernel of B_{21}. Hence B_{21} has rank $\leq r - (k-1)l \leq r - (k-1)r/k = r/k$.

Now suppose $\{C_j^i\}$ does not have this property. We will convert B to B^* s.t. the submatrices has the same property. We will obtain B^* by multiplying fullrank matrices hence it will preserve rank. Finally for B^*, we will prove $C_j^{*(i)}$ has dimension $\geq (k-1)l$

Now suppose $\{C_j^{(i)}\}$ has dim $< (k-1)l$

Since B_{12} has rank l, we can do invertible column operations s.t. only l columns in B_{12} survives. Lets call this new matrix B'_{12}. Note that on B we can perform these invertible column operations so that the right upper $n - r \times r$ block B_{12} becomes B'_{12} and all other blocks B_{11}, B_{21}, B_{22} remains the same. Further more since the column operations are invertible it will not change the rank. Lets call the new matrix B' and the new columns of $B_{11}^{i-1}B'_{12}$ $C_j'^{(i)}$.

Note there are only $(k-1)l$ many $C_j^{(i)}$s are non-zero and since the total dimension is less than $(k-1)l$ we have $y_{ij} \in \mathbb{F}$ s.t. $\sum_{i,j} C_j^{(i)} y_{ij} = 0$ where not all y_{ij} are 0.

Which simply means there are $y_1, \dots, y_{k-1} \in \mathbb{F}^n$ where not all y_is are 0, s.t.

$$\sum_{i=1}^{k-1} B_{11}^{i-1} B'_{12} y_i = 0$$

Moreover, we can assume that these vectors y_i only have non-zero entries in the places that corresponds to non-zero columns of B'_{12}

Further more we can assume $y_1 \neq 0$, else we can replace each y_i by y_{i+j} where j is the minimum index s.t. $y_j \neq 0$. That makes y_1 non zero.

So that means $B'_{12}y_1 + B_{11} \sum B_{11}^{i-2} B'_{12} = 0$ which implies columns of B'_{12} and B_{11} are linearly dependent.

That means by some invertible column operations, we can drop the rank of B'_{12} by 1.

After the transformation we call this new matrix B^* and the blocks B^*_{11}, B^*_{12}, B^*_{21}, B^*_{22}. Furthermore notice $B^*_{11} = B_{11}$ and $B^*_{21} = B_{21}$

First note rank of B^* and B is same but rank of B^*_{12} is $l-1$. Now

Claim 4.1.6. *B^*_{22}, $B^*_{21}(B^*_{11})^i B^*_{12}$ for $i = 0, \dots, k-2$ are 0*

Suppose the tth co-ordinate of y_1 is non zero. Then the column operation on B' will be: the $r+t$ column of B' is anyway 0 now, so this column operation will not change B_{22}. So $B^*_{22} = 0$

Now for $B_{21}B^i_{11}B^*_{12}$, suppose $B_{21}B^i_{11} = A$ and $B'_{12} = \begin{bmatrix} c_1 & \dots & c_{n-r} \end{bmatrix}$ and $B^*_{12} = \begin{bmatrix} c^*_1 & \dots & c^*_{n-r} \end{bmatrix}$.

Then $c_i = c^*_i$ if $i \neq t$ and $c^*_t = c_t + x_1 + \cdots + x_s$ where $Ax_i = 0$ we know. So $A \begin{bmatrix} c^*_1 & \dots & c^*_{n-r} \end{bmatrix} = \begin{bmatrix} Ac^*_1 & \dots & Ac_t + Ax_1 + \cdots + Ax_s & + \dots & Ac^*_{n-r} \end{bmatrix} = 0$ ■

Hence repeating this way we can make the dimension of $C^{(i)}_j$ to be $(k-1)\mathsf{rk}_k(B_{12})$ ■

Hence applying this to the series $C, DB, \dots, DL^tB$ we have the corollary

Corollary 4.1.7. *if first k terms of the series is 0, $\mathsf{rk}_k(\tilde{A}_{\overline{d}}) \leq rd'(1 + 1/k)$. Hence $\mathsf{rk}_k(A) \leq r(1 + 1/k)$*

Hence the final algorithm will be simple: Set $k = \frac{1}{\varepsilon}$

We will start with 1×1 non-zero entry. And suppose for an rank r witness $\overline{\alpha}$, we will construct C, B, D, L. And check for $C, DB, \dots, DL^{k-2}B = 0$. If one of them is non-zero then we can find a $r+1$ rank witness, else we can argue the PC rank is bounded by $r + r/k$. The correctness follows from the above discussion.

Note since every time after substituting the witness the dimension of $\tilde{A}_{\overline{d}}$ becomes n^k, our running time is poly(n^k).

4.2 Solving the PC-PIT

In this section we will describe the PIT algorithm for ABP in Partially Commutative model (we call it $\mathsf{PC\text{-}PIT}_k$ problem) given by Arvind et al. (2024). The key idea will be reducing the PIT to rank computation in $k-1$ commutativity partition.

The key idea will be based on [Raz and Shpilka (2004)]. But directly applying their idea will have some difficulties in dealimng with the coefficients. We will deal with that.

Theorem 4.2.1. *Given a k commutativity partition, $\mathsf{PC\text{-}PIT}_k$ of width w, length l ABP can be solved in poly(w, l, n) time using oracle access to PC Rank computation over $k-1$ commutativity partition on $\overline{\mathbf{X}} \backslash \overline{\mathbf{X}}_1 := \overline{\mathbf{X}}_2 \sqcup \cdots \sqcup \overline{\mathbf{X}}_k$*

Proof. Note if an ABP is 0 then all the homogeneous parts are 0. Hence we will first homogenise the ABP using lemma 2.3.3. So WLOG assume the ABP is homogeneous

Firstly, we explain how $\mathsf{PC\text{-}PIT}_k$ finds the substitution matrices for the variables in $\overline{\mathbf{X}}_1$. We view the edge labels as affine linear forms over the variables in $\overline{\mathbf{X}}_1$ and the coefficients are over the ring $\mathbb{F}\langle\overline{\mathbf{X}}\backslash\overline{\mathbf{X}}_1\rangle$ inside $\mathfrak{U}_{k-1}$

Inductively, at the jth level, suppose the monomials computed are $m_1, m_2, \ldots, m_{w'}$ in $\overline{\mathbf{X}}_1^j$, where $w \leq w'$. Let the corresponding coefficient vectors be $v_1, v_2, \ldots, v_{w'}$ over the ring $\mathbb{F}\langle\overline{\mathbf{X}}\backslash\overline{\mathbf{X}}_1\rangle^w$ (here the kth entry of v_i is the coefficient of m_i in the kth node). Again by lemma 2.3.3, entries v_i are given by ABPs over $\mathbb{F}\langle\overline{\mathbf{X}}\backslash\overline{\mathbf{X}}_1\rangle$ of size poly(lw). Moreover, the vectors $v_1, \ldots, v_{w'}$ are $\mathfrak{U}_{k-1}$ -spanning set for the coefficient vectors of monomials at layer j .

Now, for the $(j+1)$th level, we need to compute at most w many $\mathfrak{U}_{k-1}$-linearly independent vectors from the at most nw many coefficient vectors of the $\{m_i x_j | i \in [w'] . j \in [n]\}$. . Clearly, this is the problem of computing the rank of these at most nw coefficient vectors whose entries are ABPs over the variables in $\overline{\mathbf{X}}\backslash\overline{\mathbf{X}}_1$. This is because, given a set of w-dimensional $\mathfrak{U}_{k-1}$-linearly independent vectors $v_1 \ldots, v_{l'}$ and another vector v, the rank of this matrix with $l'+1$ columns is l' precisely if v is in the span of $v_1, \ldots, v_{l'}$. The columns of the matrix are $v_1, \ldots, v_l$ and we can make it a square matrix by padding with zero columns. Now from corollary 2.4.3 we can reduce this rank computation problem to rank computation for linear matrix problem preserving the commutativity partition. ■

Note here $\mathsf{PC\text{-}PIT}_k$ reduces to PC rank computation in $k-1$ partition, which again reduces to $\mathsf{PC\text{-}PIT}_{k-1}$ So recursing on k we can reduce it to $\mathsf{PC\text{-}PIT}_1$ which is just NC PIT which has polytime algorithm. But as we seen in the previous section reducing rank problem to the PIT problem there is a blow-up of k in the power. There will be k-step recursion and each time k will come in the power. So final time for the $\mathsf{PC\text{-}PIT}_k$ $O(n^{k^k})$. And since the rank increment step needs this times, there can be atmax s many rank increent possible, we know the time required for exact rank computation is $O(n^{k^k})$

Chapter 5

Conclusion

We are concluding this thesis by highlighting the benefits of greedy approaches for rank computation. Finally we can discuss about some open areas

1. Note the algorithm by Ivanyos et al. (2015) is highly whitebox, ie, we need the information of the matrix entries. Can we design non-trivial black box algorithm for rank computation, ie, for s, n we will output a set of assignments for n variables, for any n-variate input of size s, there will be one assignment which will be the witness for the actual rank. Note the famous Rational Identity Testing (RIT) problem reduces to NC rank even in black box sense. Recently Arvind, Chatterjee, and Mukhopadhyay (2023b) gave quasi-poly black-box algorithm for RIT.

2. Can we design an FPT algorithm for PC Rank (even in approximation case) where the parameter will be k? Note this will surely generalize the best known algorithms in both the extreme cases.

3. Can we answer something about lower bound question in Partially commutative model since this completely unexplored. Further more can the lower bound in this model generalise the extreme cases?

Bibliography

S.A Amitsur. Rational identities and applications to algebra and geometry. *Journal of Algebra*, 3(3):304–359, 1966. ISSN 0021-8693. doi: https://doi.org/10.1016/0021-8693(66)90004-4. URL `https://www.sciencedirect.com/science/article/pii/0021869366900044`.

V. Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan. New results on noncommutative and commutative polynomial identity testing. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 268–279, 2008. doi: 10.1109/CCC.2008.22.

V. Arvind, Abhranil Chatterjee, and Partha Mukhopadhyay. Trading determinism for noncommutativity in edmonds' problem, 2024.

Vikraman Arvind, Abhranil Chatterjee, Utsab Ghosal, Partha Mukhopadhyay, and C. Ramya. On identity testing and noncommutative rank computation over the free skew field. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 6:1–6:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023a. doi: 10.4230/LIPICS.ITCS.2023.6. URL `https://doi.org/10.4230/LIPIcs.ITCS.2023.6`.

Vikraman Arvind, Abhranil Chatterjee, and Partha Mukhopadhyay. Black-box identity testing of noncommutative rational formulas in deterministic quasipolynomial time. *CoRR*, abs/2309.15647, 2023b. doi: 10.48550/ARXIV.2309.15647. URL `https://doi.org/10.48550/arXiv.2309.15647`.

Vishwas Bhargava, Markus Bläser, Gorav Jindal, and Anurag Pandey. A deterministic PTAS for the algebraic rank of bounded degree polynomials. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 647–661. SIAM, 2019. doi: 10.1137/1.9781611975482.41. URL `https://doi.org/10.1137/1.9781611975482.41`.

Markus Bläser, Gorav Jindal, and Anurag Pandey. A deterministic PTAS for the commutative rank of matrix spaces. *Theory Comput.*, 14(1):1–21, 2018. doi: 10.4086/TOC.2018.V014A003. URL `https://doi.org/10.4086/toc.2018.v014a003`.

Abhranil Chatterjee and Partha Mukhopadhyay. The noncommutative edmonds' problem re-visited. *CoRR*, abs/2305.09984, 2023. doi: 10.48550/ARXIV.2305.09984. URL `https://doi.org/10.48550/arXiv.2305.09984`.

Prerona Chatterjee, Mrinal Kumar, Adrian She, and Ben Lee Volk. A quadratic lower bound for algebraic branching programs. *CoRR*, abs/1911.11793, 2019. URL `http://arxiv.org/abs/1911.11793`.

P.M. Cohn. *Skew Fields: Theory of General Division Rings*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995. ISBN 9780521432177. URL `https://books.google.co.in/books?id=u-4ADgUgpSMC`.

Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants, 2015.

Jack Edmonds. Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, page 241, 1967. URL `https://api.semanticscholar.org/CorpusID:12065561`.

MARC Fortin and Christophe Reutenauer. Commutative/noncommutative rank of linear matrices and subspaces of matrices of low rank. *Séminaire Lotharingien de Combinatoire [electronic only]*, 52, 01 2004.

Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Operator scaling: theory and applications. *Foundations of Computational Mathematics*, pages 1–68, 2019.

Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-nc. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 821–830. ACM, 2017. doi: 10.1145/3055399.3055440. URL `https://doi.org/10.1145/3055399.3055440`.

Rohit Gurjar, Taihei Oki, and Roshan Raj. Fractional linear matroid matching is in quasi-nc, 2024.

Masaki Hamada and Hiroshi Hirai. Computing the nc-rank via discrete convex optimization on cat(0) spaces, 2020.

T. Harju and J. Karhumäki. The equivalence problem of multitape finite automata. *Theoretical Computer Science*, 78(2):347–355, 1991. ISSN 0304-3975. doi: https://doi.org/10.1016/0304-3975(91)90356-7. URL `https://www.sciencedirect.com/science/article/pii/0304397591903567`.

Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive noncommutative rank computation in deterministic polynomial time over fields of arbitrary characteristics. *CoRR*, abs/1512.03531, 2015. URL `http://arxiv.org/abs/1512.03531`.

Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the Thirty-Fifth Annual ACM*

Symposium on Theory of Computing, STOC '03, page 355–364, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136749. doi: 10.1145/780542.780595. URL `https://doi.org/10.1145/780542.780595`.

Igor Klep, Victor Vinnikov, and Jurij Volčič. Multipartite rational functions. *Documenta Mathematica*, 25:1285–1313, 2020. ISSN 1431-0643. doi: 10.4171/dm/777. URL `http://dx.doi.org/10.4171/dm/777`.

László Lovász. On determinants, matchings and random algorithms. volume 79, pages 565–574, 01 1979.

Meena Mahajan and V. Vinay. A combinatorial algorithm for the determinant. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '97, page 730–738, USA, 1997. Society for Industrial and Applied Mathematics. ISBN 0898713900.

R. Raz and A. Shpilka. Deterministic polynomial identity testing in non commutative models. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 215–222, 2004. doi: 10.1109/CCC.2004.1313845.

Nitin Saxena. Progress on polynomial identity testing. *Electron. Colloquium Comput. Complex.*, TR09, 2009. URL `https://api.semanticscholar.org/CorpusID:2126874`.

J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, oct 1980. ISSN 0004-5411. doi: 10.1145/322217.322225. URL `https://doi.org/10.1145/322217.322225`.

Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, 01 2010. doi: 10.1561/0400000039.

James Worrell. Revisiting the equivalence problem for finite multitape automata. 03 2013. ISBN 978-3-642-39211-5. doi: 10.1007/978-3-642-39212-2_38.

Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, EUROSAM '79, page 216–226, Berlin, Heidelberg, 1979. Springer-Verlag. ISBN 3540095195.