2016 IEEE 57th Annual Symposium on Foundations of Computer Science

A deterministic polynomial time algorithm for non-commutative rational identity testing

Ankit Garg
Microsoft Research New England
Cambridge, USA
Email: garga@microsoft.com

Leonid Gurvits
Department of Computer Science
The City College of New York
New York, USA
Email: l.n.gurvits@gmail.com

Rafael Oliveira
Department of Computer Science
Princeton University
Princeton, USA
Email: rmo@cs.princeton.edu

Avi Wigderson
Institute for Advanced Study
Princeton, USA
Email: avi@math.ias.edu

***Abstract*—Symbolic matrices in *non-commuting* variables, and the related structural and algorithmic questions, have a remarkable number of diverse origins and motivations. They arise independently in (commutative) invariant theory and representation theory, linear algebra, optimization, linear system theory, quantum information theory, and naturally in non-commutative algebra.**

In this paper we present a deterministic polynomial time algorithm for testing if a symbolic matrix in *non-commuting* variables over $\mathbb{Q}$ is invertible or not. The analogous question for *commuting* variables is the celebrated polynomial identity testing (PIT) for symbolic determinants. In contrast to the commutative case, which has an efficient probabilistic algorithm, the best previous algorithm for the non-commutative setting required exponential time [1] (whether or not randomization is allowed).

The main (simple!) technical contribution of this paper is an analysis of an existing "operator scaling" algorithm due to Gurvits [2], which solved some special cases of the same problem we do (these already include optimization problems like matroid intersection). This analysis of the running time of Gurvits' algorithm combines results from some of these different fields. It lower bounds a parameter of quantum maps called *capacity*, via degree bounds from algebraic geometry on the *Left-Right* group action, which in turn is relevant due to certain characterization of the free skew (non-commutative) field.

Via the known connections above, our algorithm efficiently solves several problems in different areas which had only exponential-time algorithms prior to this work. These include the "word problem" for the free skew field (namely identity testing for rational expressions over non-commuting variables), testing if a quantum operator is "rank decreasing", and the membership problem in the null-cone of a natural group action arising in Geometric Complexity Theory (GCT). Moreover, extending our algorithm to actually compute the *non-commutative rank* of a symbolic matrix, yields an efficient factor-2 approximation to the standard *commutative rank*. This naturally suggests the challenge to improve this approximation factor, noting that a fully-polynomial approximation scheme may lead to a deterministic PIT algorithm. Finally, our algorithm may also be viewed as efficiently solving a family of structured systems of quadratic equations, which seem general enough to encode interesting decision and optimization problems [1].

***Keywords*-Non-commutative computation; Rational identity testing; Derandomization; Optimization;**

I. Introduction

The main object of study in this paper are *symbolic* matrices whose entries are linear functions in variables $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ over a field[2] $\mathbb{F}$. Any such matrix can be expressed as a linear combination of the variables with matrix coefficients

$$L = x_1 A_1 + x_2 A_2 + \cdots + x_m A_m$$

where $A_1, A_2 \ldots, A_m$ are $n \times n$ matrices[3] over $\mathbb{F}$.

The main computational problem we will be concerned with in this paper is determining whether such a symbolic matrix is invertible or not. This problem has a dual life, depending on whether the variables commute or do not commute. In both cases inversion is over the appropriate field of rational functions in the variables: the familiar one $\mathbb{F}(\mathbf{x})$ in the commutative case, and the *free skew field* $\mathbb{F}\langle\langle\mathbf{x}\rangle\rangle$ (which we will define and discuss) in the non-commutative case[4].

In the *commutative* case this problem has an illustrious history and significance. It was first explicitly stated by Edmonds [3], and shown to have a randomized polynomial time

[1]This is an extended abstract. For the full version, we refer the reader to https://arxiv.org/abs/1511.03730

[2]Our main results are for the rationals $\mathbb{Q}$ (and hold for the Reals and Complex numbers as well). However many of the questions are interesting for any field, and some results we mention hold for finite fields.

[3]For all purposes we may assume that the matrices A_i are linearly independent, namely span a space of matrices of dimension exactly m.

[4]For intuition the reader can view this field as the home of all expressions in the variables and constants from $\mathbb{F}$ formed by all arithmetic operations: addition, multiplication and inversion. We only caution that while the same definition works in the commutative case, here the resulting objects cannot in general be simplified to ratios of polynomials. Luckily, as we will see, the problem SINGULAR can also be defined without reference to this field; indeed, it has purely commutative formulations.

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.95

algorithm by Lovasz [4]. The completeness of determinant for arithmetic formulas by Valiant [5] means that singularity captures the celebrated Polynomial[5] Identity Testing (PIT) problem, and so in this commutative setting we will refer to it as PIT. Derandomizing Lovasz' probabilistic algorithm for PIT (namely, proving PIT $\in \mathcal{P}$) became all-important overnight when Kabanets and Impagliazzo [6] showed it would imply nontrivial arithmetic or Boolean lower bounds well beyond current reach. Thus, finding a deterministic polynomial time algorithm for PIT has been the object of intensive study in the past decade, and most results provide such algorithms for polynomials defined by restricted classes of formulae. While this paper makes no direct progress on PIT, it suggests a different type of attack on it through the non-commutative angle.

The *non-commutative* case turns out to be significant to an even larger and more diverse set of areas. Any algorithm for the singularity problem in non-commuting variables (a problem we will call SINGULAR) also solves the following problems, via known reductions and equivalences (which are of central importance to our results). We will define them and discuss their history below.

1) (Non-Commutative Algebra) Solving the *Word Problem* for the free skew field[6].
2) (Invariant Theory) Testing membership in the null-cone of the *Left-Right group action* [7]
3) (Quantum Information Theory) Deciding if a *completely positive quantum map* is rank-decreasing.
4) (Algebraic Complexity) Computing a factor-2 approximation of the *commutative* rank of a symbolic matrix[8].
5) (Combinatorial Optimization) Solving certain structured families of quadratic equations[9].

With so many implications, what is the complexity of the best algorithm for SINGULAR? Unlike the commutative PIT, it is not even clear from its definition that SINGULAR is decidable. Indeed, its decidability proof requires some very nontrivial characterizations, and other important results in both commutative and non-commutative algebra, starting with Cohn [7]. The best results yield a deterministic *exponential time* upper bound on its complexity (see two very different proofs in [1], [8]), the best known before this work. No sub-exponential algorithm was known even allowing randomness.

The main result of this paper is a *deterministic* polynomial time algorithm for this problem, for $\mathbb{F} = \mathbb{Q}$! As obvious corollaries, the five problems listed above are in $\mathcal{P}$ as well. We note that exciting subsequent work extends our result below to all fields of sufficiently high characteristic, via a very different algorithm [9]. We now formally state our result, followed by an elementary, commutative definition of the problem it solves.

Theorem I.1. *For non-commutative variables over $\mathbb{Q}$,* SINGULAR $\in \mathcal{P}$. *More specifically, there is a deterministic algorithm which, given m $n \times n$ integer matrices $A_1, \ldots A_m$ with entries of bit-size b, decides in time* poly(n, m, b) *if the matrix $L = \sum_{i=1}^{m} x_i A_i$ is invertible over the free skew field.*

One of the many equivalent formulations alluded to above of the problem SINGULAR is the following. A symbolic matrix $L = \sum_{i=1}^{m} x_i A_i$ over a field F is singular if and only if *all* the following (infinitely many) polynomials defined by the A_i vanish identically for all integers d:

$$\text{Det}(\sum X_i \otimes A_i) = 0 \tag{1}$$

where $X_1, X_2, \ldots X_m$ are $d \times d$ matrices whose entries are distinct *commuting* variables, and Det is the commutative determinant polynomial (on $dn \times dn$ matrices).

The alert reader will have noticed that in the commutative PIT problem, singularity is captured by a *single* polynomial identity, namely the case $d = 1$ above! Somehow, testing if a given tuple of matrices satisfies the infinite system of identities above seems now easier than testing the single one. The same reader will have asked if an infinite system is needed, or does a finite subset of them suffices. The answer (spoiler alert), which we will reach at the end of this introduction, is that a finite, exponential upper bound $\exp(\text{poly}(n))$ on the largest d to consider was known, and that this bound was key to our polynomial upper bound on the running time of the algorithm. Subsequent to our work (Section I-F) this upper bound on d was dramatically improved to linear $O(n)$, which does not affect our analysis much, but has lead to the different algorithm mentioned above.

We now begin our journey explaining the many origins and incarnations of the problem SINGULAR, and the algorithm we use to solve it, all of which were *known*! The main (and simple) contribution of this work is showing that the combination of the connections of equivalences, and known results in the respective areas can be combined to (1) explain what that algorithm actually computes, and (2) to analyze its running time. We will also generalize it to compute the *non-commutative rank* of a symbolic matrix. A peculiar point to make already at this point is that while the problem at hand is purely algebraic, the algorithm solving it is purely analytic.

Due to the multiple cross connections of the different areas involved, there are probably many ways to weave this long and meandering story, describing past work, evolution of ideas, and the many consequences of the algorithm (some accounts of subsets of these connections appear also e.g. in [1], [2], [10]). The algorithm itself, its (interesting) origins and our analysis of it are described on subsection I-D.

A. Word problems, identity tests and the free skew field

Word problems and their complexity are central throughout mathematics, arising whenever mathematical objects in a certain class have several representations. In such cases, a basic

[5]And rational functions, which in the commutative case are simply ratios of polynomials.

[6]Namely, the identity testing problem for rational functions over non-commutative variables.

[7]The most basic special case of the *orbit-closure intersection problem*: are two sets of linear operators equivalent if we allow a change of basis in both their domain and range.

[8]To solve PIT we must compute this rank exactly.

[9]A special case is solving geometric matroid intersection *in the dark*, a notion invented by Gurvits [2] in which the two matroids are given implicitly.

problem is whether two such representations describe the same object. Often, indeed, some problems of this form have served as early examples of decidable and undecidable problems[10].

For us, the objects in question are polynomials and rational functions in commuting and non-commuting variables. Their standard representations will be arithmetic formulas (which we focus on here) and circuits, which take the variables (and constants in $\mathbb{F}$) as inputs, and use plus, times and *inverse* gates. An excellent exposition of arithmetic complexity models and the state-of-art in the subject is [13], which discusses at length the polynomial identity testing problem in both settings (without inversion). The study of this problem in the non-commutative setting when inversion is allowed[11] was initiated in [10]. This study requires the algebraic framework of the *free skew field*, in which the resulting objects live.

The first to construct the (universal) free skew field $\mathbb{F}\langle\langle \mathbf{x} \rangle\rangle$ was Amitsur [14], a structure which contains all non-commutative polynomials (namely the algebra $\mathbb{F}\langle \mathbf{x} \rangle$), is closed under all arithmetic operations, and is universal in a sense we will not define here. This field contains as formal objects all formulas $\phi(\mathbf{x})$ as above (namely using plus, times and inverse over the variables $\mathbf{x}$ and constants from $\mathbb{F}$), like $x^{-1}+y^{-1}+1$, $(x+xy^{-1}x)^{-1}+(x+y)^{-1}-x^{-1}$, etc., under the following equivalence relation. Two such rational expressions on the same set of variables are *equivalent* if for every d, and every way of substituting $d \times d$ matrices for the variables, if both are defined[12] then they compute the same matrix[13]. The reader may observe the similarity (which will recur) of this dimension "blow-up" of variables to matrices as in equation (1). The elements of the free skew field are thus the equivalence classes of formulae defined above.

A very different construction of $\mathbb{F}\langle\langle \mathbf{x} \rangle\rangle$ was given by Cohn [15]. Its essence is that formulas can be computed by the inverse of an appropriate symbolic matrix. This is the analog of Valiant's completeness theorem for the determinant [16] in the commutative case. A simple consequence is that in the non-commutative setting SINGULAR captures polynomial and rational function identity testing for formulae over the free skew field.

Theorem I.2 ([15]). *There is an efficient algorithm which converts every arithmetic formula $\phi(\mathbf{x})$ in* non-commuting *variables $\mathbf{x}$ of size s to a symbolic matrix L_ϕ of size* $\mathrm{poly}(s)$*, such that the rational expression computed by ϕ is identically zero if and only if $L_\phi \in$* SINGULAR.

The structure of the free skew field is so complex that unlike the commutative case, even decidability of SINGULAR (and thus rational identity testing) is far from obvious. The first to prove decidability was Cohn in [7], [17]. The first explicit bound on time was given by Cohn and Reutenauer in [8], reducing it to a system of commutative polynomial equations, which puts it in $\mathcal{PSPACE}$, and thus in exponential time. Two different exponential time algorithms follow from [1], [8]. From our Theorem I.1 we conclude (using Cohn's completeness theorem above) a *deterministic, polynomial time* rational identity test for non-commuting rational expressions.

Corollary I.3. *The non-commutative rational identity testing problem is in $\mathcal{P}$. Namely, there is an algorithm which for any non-commutative formula over $\mathbb{Q}$ of size s and bit complexity b determines in* $\mathrm{poly}(s,b)$ *steps if it is identically zero.*

When division is *not* allowed, efficient deterministic identity tests of non-commutative *polynomials* were known [18], not only for formula but also for the stronger model of arithmetic branching programs (ABPs). However, as this model is efficiently simulated by matrix inversion (see Theorem 6.5 in [10]), our algorithm provides an alternative (and very different) proof of the following theorem, at least over $\mathbb{Q}$.

Theorem I.4 ([18]). *There is a deterministic polynomial time rational identity testing algorithm for non-commutative ABPs.*

B. Commutative and non-commutative rank of symbolic matrices

There are many mathematical sources, motivations and results regarding the rank of *commutative* symbolic matrices, which are much older than the complexity theory interest in the PIT problem (which of course is the special case of determining if the commutative rank is full). Some of the many references to this body of work can be found in the papers [19], [20]. In some of these works, the *non-commutative* rank (often implicitly via different characterizations) is used to give upper bound on the commutative rank, and their relationship becomes of interest.

We focus on this connection here, and explain how our main result implies a deterministic *approximation algorithm* to the commutative rank. We will use the same notation L for both a symbolic matrix, as well as the subspace of matrices spanned by it (when fixing the variables to constants in the field). We also define and elaborate on what is known regarding the commutative and non-commutative ranks, from now on denoted by $\mathrm{rank}(L)$ and $\text{nc-rank}(L)$.

Fact I.5. *Given a* commutative *symbolic matrix $L(\mathbf{x})$, its rank over $\mathbb{F}(\mathbf{x})$ is r, denoted by* $\mathrm{rank}(L(\mathbf{x})) = r$*, iff r is the* maximal *rank of any matrix in the subspace L over $\mathbb{F}$, spanned by the A_i. Equivalently,* $\mathrm{rank}(L(\mathbf{x}))$ *is the smallest r such that there exists a factorization $L = KM$ such that K has r columns, M has r rows, and the entries of both K, M are rational functions in $\mathbb{F}(\mathbf{x})$.*

While the characterization above is simple, the two given in the theorem below are very substantial, mostly developed by Cohn for his construction of the free skew field. The first characterization we present here is due to Fortin and Reutenauer [19] who heavily use Cohn's techniques. The

[10]E.g. deciding if two knots diagrams describe the same knot was proved decidable by Haken [11], and deciding if two presentations with generators and relations describe the same group was proved undecidable by Rabin [12]

[11]Inversion is not an issue in the commutative setting as every commutative rational function is a ratio of two polynomials.

[12]Namely, neither attempts to invert a singular matrix.

[13]As it happens, the 2nd expression above is equivalent to zero; this is the famous Hua's identity.

second was used by Cohn for proving decidability of the word problem over the free skew field.

Theorem I.6. *Given a* non-commutative *symbolic matrix* $L(\mathbf{x})$*, its rank over* $\mathbb{F}\langle\langle\mathbf{x}\rangle\rangle$ *is* r*, denoted by* nc-rank$(L(\mathbf{x})) = r$*, iff the space* L *is* r-decomposable*. Namely,* r *is the* minimal *number such that there exist invertible matrices* B, C *over* $\mathbb{F}$ *such that* BLC *has a minor of zeros of size* $i \times j$ *with* $i + j = 2n - r$ *(note that this may be viewed as an algebraic analog of the Hall condition for maximum bipartite matching). Equivalently,* nc-rank$(L(\mathbf{x}))$ *is the smallest* r *such that there exists a factorization* $L = KM$ *such that* K *has* r *columns,* M *has* r *rows, and the entries of both* K, M *having entries which are* polynomials *in* $\mathbb{F}\langle\mathbf{x}\rangle$*, of degree at most one.*[14]

We can extend our main Theorem I.1 from testing singularity to efficiently computing the non-commutative rank over $\mathbb{Q}$.

Theorem I.7. *There is a deterministic algorithm which, given* m $n \times n$ *integer matrices* $A_1, \ldots A_m$ *with entries of bit-size* b*, computes* nc-rank(L) *in time* poly(n, m, b) *(where* $L = \sum_{i=1}^{m} x_i A_i$*).*

It is not hard to see from the definitions that for every L we have rank$(L) \leq$ nc-rank(L). These two ranks can be different, as it is the case for the 3×3 skew symmetric matrix, whose rank is 2 but non-commutative rank is 3. Taking many copies of this matrix we see that there can be a factor $3/2$ gap between the two: for any r there are matrices L with rank$(L) = 2r$ and nc-rank$(L) = 3r$. However, Fortin and Reutenauer [19] proved that this gap is never more than a factor of 2, so our main result implies an efficient factor-2 approximation of commutative rank.

Theorem I.8 ([19])**.** *For every* L *we have* nc-rank$(L) \leq$ 2rank(L).

Corollary I.9. *There is a polynomial time algorithm which for every symbolic matrix in* commuting *variables over* $\mathbb{Q}$ *approximates* rank(L) *to within a factor of 2.*

We find the question of efficiently obtaining a better approximation ratio (indeed, even an approximation scheme) a very interesting problem. It is a different relaxation of the commutative PIT problem that as far as we are aware of has not been studied until now.

Another interesting corollary of our main theorem (as well as Valiant's determinant completeness proof[15]) is that it reduces the commutative PIT problem to a seemingly much simpler restriction of it. Call an $n \times n$ symbolic matrix L *extreme* if rank$(L) \geq n - 1$ and nc-rank$(L) = n$.

Corollary I.10. *There is a deterministic polynomial time reduction from the general PIT problem to the PIT problem for extreme matrices.*

C. Compression spaces, optimization and Gurvits' algorithm G

This subsection describes the origin and motivation of the algorithm underlying our main theorem.

An important class of spaces of matrices, studied e.g. in [20]–[25] for different motivations in algebra and geometry, is the class of *compression spaces*. These are simply all linear spaces L for which rank$(L) =$ nc-rank(L).

A deterministic polynomial time algorithm solving the *commutative* PIT for compression spaces L (over $\mathbb{Q}$) was discovered by Gurvits [2], a paper which serves as the starting point for this work. Indeed, it solves the following slightly more general problem.

Theorem I.11 ([2])**.** *There is a deterministic polynomial time algorithm, algorithm* G*, which for every* n *and every* $n \times n$ *matrix* L *given by a set of integer matrices* $(A_1, A_2, \ldots, A_m)$*, outputs "singular" or "invertible", and its output is guaranteed to be correct*[16] *when either* rank$(L) = n$ *or* nc-rank$(L) < n$ *over* $\mathbb{Q}$*.*

In particular, note that the algorithm always gives the correct answer for compression spaces. Gurvits motivates his algorithm primarily by suggesting a completely different approach to the commutative PIT problem (on which we elaborated above), which solves it efficiently for this interesting family of symbolic matrices. Indeed, he notes that a few general families of matrix spaces are compression spaces, including those spanned by *PSD* matrices, or by *upper triangular* matrices, or by *rank-1* matrices. Gurvits notes that, as his algorithm does not depend on the actual given generators A_i, but only on the space L they span, it can solve some optimization problems (including *geometric matroid intersection*) "in the dark" (see details in [2]).

Gurvits' paper deviates from most recent progress on deterministic PIT algorithms (e.g. [26]–[30] among many others) which focus on polynomials computed by a variety of restricted classes of arithmetic circuits. Gurvits' algorithm G solves PIT in cases which we do not know how to classify in arithmetic complexity terms, but rather via structural properties of the symbolic matrix L as above (e.g. compression spaces). Very few examples of approaching deterministic PIT from this direction are known, and include the algebraic algorithms for perfect matching and rank completion of [31]–[33], nearly all being special cases of compression spaces. Making progress on PIT by relaxing these structural constraints and solving it outside compression spaces would be extremely interesting. We note that Corollary I.10 above shows that the

[14]Note the striking difference to the commutative case, which may also hint to why the non-commutative case may be easier in a sense. The "rigidity" of non-commutative polynomials affords such a simple factorization if and only if it exists with rational function entries from $\mathbb{F}\langle\langle\mathbf{x}\rangle\rangle$. It is not hard to see using this definition that computing nc-rank(L) thus reduces to a system of quadratic equations in the (commutative) coefficients (in $\mathbb{F}$) of the entries of the factors, and so can be solved e.g. using the Gröbner basis algorithm. Our algorithm determines solvability of such systems in polynomial time!

[15]Observing that the reduction generates matrices which are nearly upper triangular, except having 1's rather than 0's just below the diagonal.

[16]For both the commutative and non-commutative definitions.

commutative PIT problem *reduces* to the question of determining whether a given symbolic matrix L *is* a compression space (over $\mathbb{Q}$).

Yet another interesting direction (taken here) is to better understand what algorithm G actually does. Our main contribution in this paper is (1) observing that despite its focus on the commutative PIT problem, Gurvits' algorithm G actually solves the non-commutative SINGULAR problem, and (2) analyzing the algorithm to show that it does so in polynomial time. To understand both, we now turn to describe algorithm G.

D. Permanents, quantum operators, origins and analysis of algorithm G

Here we give only an intuitive informal description (with imprecise parameters), which makes it easy to explain its origins and nature, as well as its analysis. We already mentioned one peculiar property of algorithm G, namely that while the problem SINGULAR it solves in Theorem I.1 is purely *algebraic*, the algorithm itself is purely *analytic*; it generates from the input a sequence of matrices and tests if it is convergent. Another interesting property is that algorithm G arises as a "quantum analog" of another analytic algorithm with very different motivation that we now discuss. This is algorithm S of Sinkhorn [34], designed to solve the *matrix scaling* problem defined below. We note that this problem (besides being the "classical version" of the problem we solve) is interesting in its own right; it was initially developed for applications in numerical analysis, and has since found many other applications (see survey and references in [35], who used it as a basis for their deterministic algorithm for approximating the permanent of non-negative matrices).

The *matrix scaling* problem (over $\mathbb{R}$) gets as input a single non-negative matrix A, and attempts to find if it can be "scaled" to a doubly-stochastic one (namely, having the entries in every row and column sum to 1). Here "scaling" refers to multiplying rows and columns by positive constants. Thus we seek positive diagonal matrices B, C (called scaling factors) such that BAC is (nearly) doubly stochastic, or determine that no such scaling is possible.

Two different analyses of Sinkhorn's algorithm S, one of [35] and the other in the unpublished [36] inspire algorithm G and its analysis in [2].

We now describe algorithm S. We set up notation which will be later easy to generalize for describing algorithm G. For a non-negative matrix A, let $R(A)$ denote the diagonal matrix whose (i,i)-entry is the inverse of the L_1 norm of row i (which here is simply the sum of its entries as A is non-negative). Similarly $C(A)$ is defined for the columns[17].

Algorithm S gets as input a non-negative integer matrix A. For a fixed polynomial (in the input size) number of iterations it repeats the following two steps

- Normalize rows: $A \leftarrow R(A) \cdot A$
- Normalize columns: $A \leftarrow A \cdot C(A)$

[17]A "non-triviality" assumption is that no row or column in A is all zero.

We describe the [35] analysis for algorithm S.

What does this algorithm do? It is clear that in alternate steps either $R(A) = I$ or $C(A) = I$, where I is the identity matrix. Thus A itself alternates[18] being row-stochastic and column-stochastic. The question is whether both converge to I together, namely, if this process converts A to a *doubly stochastic* matrix.

In [35] it is proved that this happens if and only if $\text{Per}(A) > 0$, where Per is the permanent polynomial. Moreover, convergence (in the limit) is easy to detect after very few iterations! If we define $\text{ds}(A) = ||R(A)-I||^2+||C(A)-I||^2$ as a notion of distance between A and the doubly stochastic matrices, then the convergence test is simply whether $\text{ds}(A) < 1/n^2$. If it is that small at the end of the algorithm, then $\text{Per}(A) > 0$, and otherwise $\text{Per}(A) = 0$ (in particular, this 2-line algorithm solves in particular the bipartite perfect matching problem!)

The analysis of convergence of algorithm S in [35] is extremely simple, using the permanent itself as a "progress measure" on the sequence of matrices produced S. It has three parts:

1) Initially, $\text{Per}(A)$ is inverse exponentially large, (specifically, if A is row-stochastic with rational entries of bit-length b, then $\text{Per}(A) > 1/(bn)^{O(n)}$),
2) The arithmetic-geometric mean inequality guarantees that iterations never decrease the permanent, and increase it by a factor of $1 + 1/n^2$ when $ds(A) > 1/n^2$ for the current A,
3) The permanent of any row-stochastic or column stochastic matrix is upper bounded by 1.

This 3-step analysis clearly implies that after polynomially many iterations we can tell if the sequence ever converges or not.

Now lets move to describing Gurvits' algorithm G, which introduces into our story quantum information theory! As it happens, algorithm G is best viewed as a *quantum* analog of algorithm S! Informally, in quantum analogs of classical situations two things typically happen: diagonal matrices (which commute) become general matrices (which do not), and the L_1 norm is replaced by L_2. This happens here as well, and we do so almost syntactically, referring the reader to [2] for the quantum information theoretic intuition and meaning of all notions we mention.

The input to algorithm G is a symbolic matrix $L = \sum_i x_i A_i$, given by the $n \times n$ integer matrices $(A_1, A_2, \ldots, A_m)$. Briefly, L is viewed as a *completely positive (quantum) operator*, or map, on PSD matrices, mapping such a (complex valued) matrix P to $L(P) = \sum_i A_i P A_i^\dagger$ (P is typically a "density matrix" describing a quantum state, namely a PSD matrix with unit trace, and the operator L will typically preserve trace or at least not increase it). The dual operator L^* acts by $L^*(P) = \sum_i A_i^\dagger P A_i$. These maps provide us with quantum analog notions of what it means for

[18]This algorithm may indeed be viewed as a special case of a heuristic called "alternate minimization" in convex optimization.

an operator L to be row-stochastic (if $L(I) = I$) and column-stochastic (if $L^*(I) = I$)[19].

We now turn to generalized scaling. Instead of positive diagonal matrices, we now allow any complex non-singular matrices B, C. Thus, given L, we ask if if there are such "scaling factors" B, C such that BLC (interpreted as mapping the tuple $(A_1, A_2, \ldots, A_m)$ to the tuple $(BA_1C, BA_2C, \ldots, BA_mC)$) is (nearly) doubly stochastic in the sense above.

As in the classical case, it is easy to satisfy one of these conditions by appropriate left or right basis change of all A_i. Given L let $R(L)$ and $C(L)$ be defined[20] by $R(L) = (\sum_i A_i A_i^\dagger)^{-\frac{1}{2}}$, and $C(L) = (\sum_i A_i^\dagger A_i)^{-\frac{1}{2}}$. Note that $R(L) = L(I)^{-\frac{1}{2}}$ and $C(L) = L^*(I)^{-\frac{1}{2}}$.

With this notation, Gurvits' algorithm G essentially mimics its classical sibling algorithm S above. On input $(A_1, A_2, \ldots, A_m)$ algorithm G repeats, for a fixed polynomial (in the input size) number of iterations, the following analogous two steps

- Normalize rows: $L \leftarrow R(L) \cdot L$
- Normalize columns: $L \leftarrow L \cdot C(L)$

So again, analogs of row and column scalings are performed alternately, simultaneously on all matrices A_i. It is clear, as above, that after each step either $R(L) = I$ or $C(L) = I$. It is natural to wonder under what conditions does this sequence converges to a doubly stochastic operator, namely both $R(L)$ and $C(L)$ simultaneously approach I.

One can similarly define in an analogous way to the classical setting a "distance from double stochastic" by $\text{ds}(L) = ||R(L) - I||^2 + ||C(L) - I||^2$, and and test (after polynomially many iterations) if $\text{ds} < 1/n^2$. This is precisely what algorithm G does. It is not hard to see that if L (with non-commuting variables) is singular (or equivalently, if L is rank-decreasing), then there is no convergence, and the test above fails.

To study convergence Gurvits [2] introduces an alternative progress measure, a function on completely positive maps L he calls *capacity* which we now define. Let $\text{cap}(L)$ be the infimum of $\text{Det}(L(P))$ over all PSD matrices P of determinant 1. This measure is chosen, like the Permanent above, to easily reflect the changes after each step. It turns out the steps (2), (3) of the 3-part analysis above (replacing permanent by capacity and A by L) follow by the same simple proofs of the classical case. The problem is proving an exponential lower bound for the initial input L. Unable to prove a lower bound in general, Gurvits proved the following conditional lower bound.

Lemma I.12. *Let L be a row-stochastic operator of dimension n and rational entries with bit-length b. If Det$(L) \not\equiv 0$ (L is non-singular over* commutative *variables), then cap$(L) \geq 1/(bn)^{O(n)}$.*

This clearly suffices for proving Theorem I.11 which handles compression spaces. *The next two paragraphs contain the main technical contribution of the paper.*

We now explain how we handle the general case, in which $\text{Det}(L) \equiv 0$ (so L is commutatively singular), but with non-commuting variables L is not singular. Recall that we need an inverse exponentially large lower bound on the capacity of L. The idea is to use Lemma I.12 above, but for a different operator. The source of this new operator is the equation (1), arising from Amitsur's and Cohn's characterizations of the free skew field, which expresses SINGULAR in terms of the vanishing of all blow ups of L by $d \times d$ variable matrices X_i. We can deduce from it, that as our initial L is not singular in $\mathbb{Q}\langle\langle \mathbf{x} \rangle\rangle$, there must be some finite d such that $\text{Det}(\sum_i X_i \otimes A_i) \not\equiv 0$. By Schwartz-Zippel [37], [38], there must be m $d \times d$ constant matrices D_i (with integers entries of bit-length at most bdn) such that $\text{Det}(\sum_i D_i \otimes A_i) \neq 0$. Associating with the matrices D_i a completely positive map $M = (D_1, D_2, \ldots, D_m)$ acting on $d \times d$ matrices, and using the natural tensor product of quantum maps, we obtain a new operator $M \otimes L$ defined by the set $(D_i \otimes A_j)_{i,j=1}^m$ that acts on $dn \times dn$ matrices. For this new composite operator the condition of Lemma I.12 holds by the choice of D_i, and we immediately get from it

$$\text{cap}(M \otimes L) \geq 1/(bdn)^{O(nd)}.$$

This is nice, but we need a bound on the capacity of the original operator L. This follows from a very easy lemma, which essentially says that the *normalized* capacity is sub-multiplicative under tensor products.

Lemma I.13. *Let M and L be completely positive maps on dimensions d, n respectively. Then*

$$\textit{cap}(M \otimes L)^{1/dn} \leq \textit{cap}(M)^{1/d} \textit{cap}(L)^{1/n}.$$

Using the bound above on $\text{cap}(M \otimes L)$, and the fact that capacity never exceeds 1, we obtain

$$\text{cap}(L) \geq \text{cap}(M \otimes L)^{1/d} \geq 1/(bdn)^{O(n)}.$$

Thus, we now see that to have an inverse exponential lower bound on $\text{cap}(L)$ it suffices to have an exponential upper bound on the blow-up dimension d as a function of n. We turn now to explore the source of this upper bound.

E. Invariant theory, the left-right group action, and blow-up dimension bounds

Invariant theory[21] deals with understanding the symmetries of mathematical objects, namely transformations of the underlying space which leave an object unchanged or *invariant*. Such a set of transformations always form a group (and every group arises this way). One major question in this field is, given a group acting on a space, characterize the objects left invariant under all elements of the group. Here we will only discuss very specific space and actions: polynomials (with commuting variables!) that are left invariant under certain linear transformations of the variables. The

[19] This condition is equivalent to having the original map L to be "trace preserving", namely satisfy $\text{tr}L(P) = \text{tr}(P)$.

[20] Again using a "non-triviality" assumption these matrices are invertible.

[21] The books [39]–[41] provide expositions on the general theory. More focused discussions towards our applications appear in the appendix of [10] and Section 1.2 of [42].

invariant polynomials under such action are clearly closed under addition and multiplication, and thus form a ring (called the *invariant ring*). The *null-cone* of the action is simply the set of all assignments to variables which make *all* non-constant homogenous invariant polynomials vanish.

A general, important theorem of Hilbert [43] assures us that the invariant rings under such linear actions there is always a finite generating set of polynomials (and hence we have a finite upper bound on their maximum degree). Obtaining upper bounds on the degree of generating sets, and finding descriptions of minimal generating sets for natural actions are the classical goals of this area. More modern one[22] is obtaining succinct descriptions and efficient computation of these invariants (e.g. see [42], [44]).

What we consider here is the *left-right action* on m $n \times n$ matrices, where a pair $(B, C) \in SL_n(\mathbb{F})^2$ takes $(Y_1, Y_2, \ldots, Y_m)$ to $(BY_1C, BY_2C, \ldots, BY_mC)$. The study of the invariant ring of polynomials (in the mn^2 variables sitting in the entries of these matrices) for this action was done[23] by [48]–[51]. Magically, it will look very familiar to equation (1) used above.[24]

Theorem I.14 ([48]–[51]). *Over algebraically closed fields of characteristic* 0*, the invariant ring of polynomials of the left-right action above is generated by all polynomials of the form Det* $(\sum_i D_i \otimes Y_i)$*, for all integers* d *and all* $d \times d$ *matrices* D_i*.*

It is probably worthwhile to stress the connection forged between the commutative and non-commutative worlds by this theorem when combined with Amitsur's and Cohn's constructions of the skew field. A set of matrices $(A_1, A_2, \ldots, A_m)$ is in the null-cone of the left-right action if and only if the symbolic matrix $L = \sum_i x_i A_i$ is singular in the free skew field! In other words, the non-commutative SINGULAR problem (and thus rational identity testing, and the word problem in the skew field) arises completely naturally in commutative algebra and invariant theory. Of course, SINGULARITY itself is invariant under the left-right action (indeed, even by any invertible matrices B, C, not necessarily of determinant 1), so one expects a connection, and hope now that algebraic geometric tools will aid in solving these non-commutative problems. And indeed they do!

The required bound on the minimal blow-up dimension $d = d(n)$ of testing membership in the null cone of the left-right action (needed in the previous section) follows directly from upper bounds on the degree of generators for the invariant ring of this action. The first explicit (doubly exponential) bound was proved by Popov [52], followed by a singly exponential bound by Derksen [53], that was sharpened in [1]:

Theorem I.15 ([1]). $d(n) \leq (n+1)!$

[22]Arising in particular in the GCT program of Mulmuley and Sohoni

[23]We note that this is part of the larger project of understanding *quiver representations*, started by the works of Procesi, Razmysolov, and Formanek [45]–[47].

[24]Note though that the roles of which matrices in the tensor product are variable, and which are constant, has switched!

Plugging in this exponential bound, the best known prior to our work, in the bound for cap(L) of the previous section completes the proof of Theorem I.1!

The question of obtaining polynomial degree upper bounds (known for other group actions, e.g. the simultaneous conjugation of a tuple of matrices) remained a challenging open problem (both from the original algebraic geometric motivations, as well as more recent ones, including the complexity theoretic problems raised in [10] on the question of division elimination in non-commutative formulae computing polynomials, and lower bounds on such formulae. Luckily for us, the existing exponential bound was sufficient for a polynomial running time proof.

F. Subsequent work

Only weeks after we published our work, major developments happened which further improved our understanding of the issues discussed in this paper.

First, Derksen and Makam [54], using a concavity argument and the regularity lemma of [1], proved that the minimal blow-up dimension $d(n)$ for matrix invariants is actually upper bounded by $d(n) \leq n + 1$, which is an exponential improvement over Theorem I.15 (which was open for over a decade). This answers open problems in [10] division elimination and lower bounds for non-commutative formulae with division.

Then, a few weeks later, [9] simplified the proof of [54] and proved that SINGULAR $\in \mathcal{P}$ for large enough fields, generalizing our result. Moreover, they solve not only the decision problem (of whether a matrix L is singular), but also the search problem of finding a factorization $L = KM$ where K, M are linear matrices over $\mathbb{F}\langle \mathbf{x} \rangle$. Their algorithm is essentially combinatorial and is completely different then ours.

G. Discussion and open problems

The main result of this paper is that a natural, simple iterative algorithm converges exponentially fast, and solves in polynomial time an array of problems in very different computational and mathematical areas (due to the many equivalent formulations of non-commutative rank of a symbolic matrix). We thus feel that, as an algorithmic technique its power is far from fully revealed as yet. Three particular exciting directions we believe are worth exploring are the following. First, understand the families of systems of quadratic equations solved efficiently by this algorithm, and find applications of optimization problems which can be reduced to such a system[25]. Second, determine to what extent it can be useful to resolve or approach the formidable commutative PIT problem[26]. Third, our application to computing membership in the null-come of the Left-Right group action is a special case of the more interesting problem of orbit-closure intersection,

[25]We have noted that geometric matroid intersection is one such problem.

[26]We have noted that a seemingly simple form of PIT is already as general as the whole problem in Corollary I.10 and it can approximate commutative rank in Corollary I.9.

which is more basic in invariant theory and more relevant to GCT - can one solve it with these techniques.

More abstractly, this work further strengthens the connections of the different areas (and the people working in them) that bear upon this special problem. The recent rapid progress mentioned above will hopefully lead to better understanding and more applications, in both math and CS.

ACKNOWLEDGEMENTS

We would like to thank Harm Derksen, Pavel Hrubes, Louis Rowen and K. V. Subrahmanyam for helpful discussions.

Ankit Garg's research partially supported by NSF grant CCF-1149888, Simons Collaboration on Algorithms and Geometry, Simons Fellowship in Theoretical Computer Science and Siebel Scholarship. Rafael Oliveira's research was partially supported by NSF Career award (1451191) and by CCF-1523816 award. Avi Wigderson's research was partially supported by NSF grant CCF-1412958.

REFERENCES

[1] G. Ivanyos, Y. Qiao, and K. V. Subrahmanyam, "Non-commutative edmonds' problem and matrix semi-invariants," *http://arxiv.org/abs/1508.00690*, August 2015.

[2] L. Gurvits, "Classical complexity and quantum entanglement," *Journal of Computer and System Sciences*, vol. 69, no. 3, pp. 448–484, 2004.

[3] J. Edmonds, "Systems of distinct representatives and linear algebra," *Journal of research of the National Bureau of Standards*, vol. 71, no. 241-245, 1967.

[4] L. Lovasz, "On determinants, matchings, and random algorithms," *Fundamentals of Computation Theory*, pp. 565–574, 1979.

[5] L. Valiant, "The complexity of computing the permanent," *Theoretical Computer Science*, vol. 8, pp. 189–201, 1979.

[6] V. Kabanets and R. Impagliazzo, "Derandomizing polynomial identity tests means proving circuit lower bounds," *Computational Complexity*, vol. 13, pp. 1–46, 2004.

[7] P. M. Cohn, "The word problem for free fields." *The Journal of Symbolic Logic*, vol. 38, no. 2, pp. 309–314, 1973.

[8] P. M. Cohn and C. Reutenauer, "On the construction of the free field," *International journal of Algebra and Computation*, vol. 9, no. 3, pp. 307–323, 1999.

[9] G. Ivanyos, Y. Qiao, and K. V. Subrahmanyam, "Constructive noncommutative rank computation in deterministic polynomial time over fields of arbitrary characteristics," *arXiv preprint arXiv:1512.03531*, December 2015.

[10] P. Hrubes and A. Wigderson, "Non-commutative arithmetic circuits with division," *ITCS*, 2014.

[11] W. Haken, "Theorie der normalflachen," *Acta Math*, vol. 105, pp. 245–375, 1961.

[12] M. O. Rabin, "Recursive unsolvability of group theoretic problems," *Annals of Mathematics*, vol. 67, no. 172-194, 1958.

[13] A. Shpilka and A. Yehudayoff, *Arithmetic Circuits: A Survey of Recent Results and Open Questions*. NOW, Foundations and Trends in Theoretical Computer Science, 2010, vol. 5, no. 3-4.

[14] S. Amitsur, "Rational identities and applications to algebra and geometry," *Journal of Algebra*, vol. 3, pp. 304–359, 1966.

[15] P. M. Cohn, "The embedding of firs in skew fields," *Proceedings of the London Mathematical Society*, vol. 23, pp. 193–213, 1971.

[16] L. G. Valiant, "Completeness classes in algebra," in *Proceedings of the eleventh annual ACM symposium on Theory of computing*. ACM, 1979, pp. 249–261.

[17] P. M. Cohn, "The word problem for free fields: A correction and an addendum," *Journal of Symbolic Logic*, vol. 40, no. 1, pp. 69–74, 1975.

[18] R. Raz and A. Shpilka, "Deterministic polynomial identity testing in non commutative models," *Computational Complexity*, vol. 14, pp. 1–19, 2005.

[19] M. Fortin and C. Reutenauer, "Commutative/noncommutative rank of linear matrices and subspaces of matrices of low rank," 2004.

[20] B. Gelbord and R. Meshulam, "Spaces of singular matrices and matroid parity," *European Journal of Combinatorics*, vol. 23, no. 4, pp. 389–397, 2002.

[21] J. Dieudonné, "Sur une généralisation du groupe orthogonal à quatre variables," *Arch. Math.*, vol. 1, pp. 282–287, 1949.

[22] M. D. Atkinson, "Spaces of matrices with several zero eigenvalues," *Bulletin of the London Mathematical Society*, vol. 12, no. 89-95, 1980.

[23] M. D. Atkinson and S. Lloyd, "Large spaces of matrices of bounded rank," *Quarterly Journal of Math. Oxford*, vol. 31, pp. 253–262, 1980.

[24] L. B. Beasley, "Nullspaces of spaces of matrices of bounded rank," *Current trends in matrix theory*, 1987.

[25] D. Eisenbud and J. Harris, "Vector spaces of matrices of low rank," *Advances in Math*, vol. 70, pp. 135–155, 1988.

[26] A. Klivans and D. Spielman, "Randomness efficient identity testing of multivariate polynomials," in *Proceedings of the 33rd Annual STOC*, 2001.

[27] Z. Dvir and A. Shpilka, "Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits," *SIAM J. Comput*, 2006.

[28] N. Kayal and N. Saxena, "Polynomial identity testing for depth 3 circuits," *Computational Complexity*, 2007.

[29] S. Saraf and I. Volkovich, "Black-box identity testing of depth 4 multilinear circuits," in *Proceedings of the 43rd annual STOC*, 2011.

[30] M. Forbes and A. Shpilka, "Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs," *FOCS*, pp. 243–252, 2013.

[31] J. F. Geelen, "An algebraic matching algorithm," *Combinatorica*, vol. 20, no. 1, pp. 61–70, 2000.

[32] G. Ivanyos, M. Karpinski, and N. Saxena, "Deterministic polynomial time algorithms for matrix completion problems," *SIAM journal on computing*, vol. 39, no. 8, pp. 3736–3751, 2010.

[33] S. A. Fenner, R. Gurjar, and T. Thierauf, "Bipartite perfect matching is in quasi-nc," 2015.

[34] R. Sinkhorn, "A relationship between arbitrary positive matrices and doubly stochastic matrices," *The Annals of Mathematical Statistics*, vol. 35, pp. 876–879, 1964.

[35] N. Linial, A. Samorodnitsky, and A. Wigderson, "A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents," *STOC*, pp. 644–652, 1998.

[36] L. Gurvits and P. N. Yianilos, "The deflation-inflation method for certain semidefinite programming and maximum determinant completion problems," *Technical Report, NECI*, 1998.

[37] J. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM*, vol. 27, pp. 701–717, 1980.

[38] R. Zippel, "Probabilistic algorithms for sparse polynomials," *EUROSAM*, pp. 216–226, 1979.

[39] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms*, third edition ed. Undergraduate Texts in Mathematics. Springer, New York, 2007.

[40] H. Derksen and G. Kemper, *Computational Invariant Theory*. Springer-Verlag, Berlin, 2002, vol. 130.

[41] H. Kraft and C. Procesi, "Classical invariant theory, a primer," *https://math.unibas.ch/uploads/x4epersdb/files/primernew.pdf*, 1996.

[42] M. Forbes and A. Shpilka, "Explicit noether normalization for simultaneous conjugation via polynomial identity testing," *RANDOM*, 2013.

[43] D. Hilbert, "Uber die vollen invariantensysteme," *Math. Ann.*, vol. 42, pp. 313–370, 1893.

[44] K. Mulmuley, "Geometric complexity theory v: Equivalence between blackbox derandomization of polynomial identity testing and derandomization of noether's normalization lemma," *FOCS*, pp. 629–638, 2012.

[45] C. Procesi, "The invariant theory of nn matrices," *Advances in Mathematics*, vol. 19, pp. 306–381, 1976.

[46] J. P. Razmyslov, "Trace identities of full matrix algebras over a field of characteristic zero," *Mathematics of the USSR-Izvestiya*, vol. 8, no. 4, p. 727, 1974.

[47] E. Formanek, "Generating the ring of matrix invariants," *Ring Theory*, pp. 73–82, 1986.

[48] H. Derksen and J. Weyman, "Semi-invariants of quivers and saturation for littlewood-richardson coefficients." *Journal of the American Mathematical Society*, vol. 13, no. 3, pp. 467–479, 2000.

[49] M. Domokos and A. N. Zubkov, "Semi-invariants of quivers as determinants," *Transformation Groups*, vol. 6, no. 1, pp. 9–24, 2001.

[50] A. Schofield and M. V. den Bergh, "Semi-invariants of quivers for arbitrary dimension vectors," *Indagationes Mathematicae*, vol. 12, no. 1, pp. 125–138, 2001.

[51] B. Adsul, S. Nayak, and K. V. Subrahmanyam, "A geometric approach to the kronecker problem ii : rectangular shapes, invariants of n n matrices, and a generalization of the artin-procesi theorem," *Manuscript, available at http://www.cmi.ac.in/ kv/ANS10.pdf*, 2010.
[52] V. L. Popov, "The constructive theory of invariants," *Izvestiya: Mathematics*, vol. 19, no. 2, pp. 359–376, 1982.
[53] H. Derksen, "Polynomial bounds for rings of invariants," *Proceedings of the American Mathematical Society*, vol. 129, no. 4, pp. 955–964, 2001.
[54] H. Derksen and V. Makam, "Polynomial degree bounds for matrix semi-invariants," *arXiv preprint arXiv:1512.03393*, 2015.