Descriptive Complexity of Linear Equation Systems and Applications to Propositional Proof Complexity

Martin Grohe
RWTH Aachen University
grohe@informatik.rwth-aachen.de

Wied Pakusa
University of Oxford
wied.pakusa@cs.ox.ac.uk

***Abstract*—We prove that the solvability of systems of linear equations and related linear algebraic properties are definable in a fragment of fixed-point logic with counting that only allows polylogarithmically many iterations of the fixed-point operators. This enables us to separate the descriptive complexity of solving linear equations from full fixed-point logic with counting by logical means. As an application of these results, we separate an extension of first-order logic with a rank operator from fixed-point logic with counting, solving an open problem due to Holm [21].**

We then draw a connection from this work in descriptive complexity theory to graph isomorphism testing and propositional proof complexity. Answering an open question from [7], we separate the strength of certain algebraic graph-isomorphism tests. This result can also be phrased as a separation of the algebraic propositional proof systems "Nullstellensatz" and "monomial PC".

I. Introduction

There are two quite different perspectives on this work. The first is from descriptive complexity and finite model theory: we prove that the solvability of systems of linear equations and related linear algebraic properties are definable in a fragment of fixed-point logic with counting (FPC) that only allows polylogarithmically many iterations of the fixed-point operators (POLYLOG-FPC). To place this result, it may be helpful to view FPC as an approximation of the complexity class PTIME and the fragment POLYLOG-FPC as the corresponding approximation of the complexity class NC consisting of all problems solvable in parallel polylogarithmic time. Our result allows us to separate the descriptive complexity of solving linear equations from full FPC by logical means. As an application of these results, we separate an extension of first-order logic with a rank operator from fixed-point logic with counting, solving an open problem due to Holm [21].

The second perspective on our work is from graph isomorphism testing and propositional proof complexity. Answering an open question from [7], we separate the strength of certain algebraic graph-isomorphism tests. This result can also be phrased as a separation of the algebraic propositional proof systems "Nullstellensatz" and "monomial PC".

Maybe the key insight of our paper is that there is a connection between these two topics and that descriptive complexity can be used to answer the open problem on the graph isomorphism algorithms. In fact, this open problem was our starting point, and the paper is the result of solving it.

In the following, we present both the descriptive complexity side and the graph isomorphism/proof complexity side in more detail, giving context and references.

A. Descriptive Complexity of Solving Linear Equations

Fixed-point logic with counting has played a central role in descriptive complexity theory ever since Immerman [23] proposed it as a candidate for a logic capturing polynomial time. Over the last ten years we have learned that the logic is significantly more expressive than previously thought, both in graph theoretic contexts [18], [20], [27] and, quite unexpectedly, also in algebraic contexts [3], [12], [21]. A culmination of the work on definability of algebraic properties is Anderson, Dawar, and Holm's [3] result that the solvability of linear programs is expressible in FPC. Of course this implies that the solvability of systems of linear equations is also expressible in FPC. This is not obvious, because typical algorithms like Gaussian elimination, choosing a pivot element in each step, cannot be described in a "choiceless" logic like FPC. And it should be noted that this result only holds over the fields of characteristic 0 and not over fields of characteristic $p \neq 0$, see [4], [13], [17].

We take a closer look at the logical resources required to express the solvability of linear equations. Our intuition was that since the computational complexity of this problem is in NC, it might be possible to express it in the fragment POLYLOG-FPC. This turned out to be true. The key step of the proof is to transform, by a logical interpretation, a given system of linear equations into an equivalent system where we can define a linear order on the variables and equations.

We also prove a related, quite powerful, technical result stating that simultaneous similarity of two sequences of definable matrices can be expressed in counting logic with polylogarithmic quantifier depth. Very briefly, two matrix sequences are simultaneously similar if there is a single similarity transformation which pairwise relates corresponding matrices in the two sequences.

We can now use an old result due to Immerman [22], [24] that separates the logics POLYLOG-FPC and FPC (actually, Immerman works with different logics, but this result follows) to prove that certain logical systems based on solving systems of linear equations must be strictly weaker than FPC. As an

The second author was partially supported by a DFG grant (PA 2962/1-1).

978-1-5090-3018-7/17/$31.00 ©2017 European Union

immediate application, this shows that the extension of first-order logic by a rank operator [13], which expresses the rank of matrices over the field of rationals, is strictly weaker than FPC, answering an open question from [21].

B. Algebraic Approaches to Graph Isomorphism

In recent years, graph isomorphism algorithms based on generic mathematical programming and algebraic techniques have received considerable attention (e.g. [5], [7], [8], [19], [30], [31]). The idea is to encode a graph-isomorphism instance into a system of linear or polynomial equations or inequalities and then to solve this system by standard (linear) algebraic means. The most basic of these approaches is to write, for two given graphs, an integer linear program (ILP) whose solutions correspond to the isomorphisms between the graphs and then study the LP-relaxation of this ILP. The solutions to this linear program are known as *fractional isomorphisms*. Note that solving the linear program gives us a sound, but not complete isomorphism test: if there is no solution, we know that the graphs are not isomorphic; if there is a solution we know nothing. It follows from a beautiful result due to Tinhofer [36] via a correspondence between logic and the so-called Weisfeiler-Leman algorithm due to Immerman and Lander [25] that the linear program has a solution (that is, there is a fractional isomorphism between the two graphs) if, and only if, the graphs are indistinguishable in the 2-variable fragment of first-order logic with counting. Atserias and Maneva [5] extended Tinhofer's result to a correspondence between the Sherali-Adams hierarchy of increasingly "tighter" LP-relaxations of the original ILP and the hierarchy of finite-variable fragments of first-order logic with counting.

In [7], Berkholz jointly with the first author of this paper, studied an algebraic approach where an instance of the graph isomorphism problem is encoded by a system of linear and quadratic equations, which then can be solved by Gröbner basis techniques. The strength of this approach can best be studied in the framework of propositional proof complexity, specifically for the algebraic proof systems *polynomial calculus (PC)* [11] and *Hilbert's Nullstellensatz* [6]. In [7], an intermediate proof system between Nullstellensatz and PC, the *monomial PC* was introduced, and it was shown that this system precisely characterises the hierarchy of finite-variable fragments of first-order logic with counting. Here the number of variables corresponds to the degree of the polynomials used in a refutation in the proof system (proving that the original system of equations is not satisfiable). The question whether the monomial PC is stronger than Nullstellensatz was left open. Remarkably, this question is equivalent to the question whether the nonnegativity constraints on all variables in the linear programs of the Sherali-Adams hierarchy mentioned above can be omitted. We answer this question by proving that monomial PC is strictly stronger.

The proof is by reductions to our descriptive-complexity theoretic results. Finding Nullstellensatz refutations amounts to solving systems of linear equations and hence can be expressed in finite-variable first-order logic with counting and polylogarithmic quantifier depth. Monomial PC has the strength of full FPC when it comes to distinguishing graphs. Now we can use Immerman's result (POLYLOG-FPC < FPC) to separate the two.

C. Outline

We introduce the polylogarithmic restriction of fixed-point logic with counting (POLYLOG-FPC) in Section III. Formulas of POLYLOG-FPC can be translated into equivalent formulas of counting logic with polylogarithmic quantifier rank (Theorem III.1). An old result due to Immerman (Theorem III.1) yields the separation of POLYLOG-FPC from full FPC. After we discussed the encoding of linear-algebraic objects by means of finite relational structures in Section IV, we explore the power of POLYLOG-FPC with respect to linear-algebraic queries over the complex field in Section V. One of our main results is that POLYLOG-FPC can define the solvability of linear equation systems (Theorem V.3). More generally, we show in Section VI that POLYLOG-FPC can distinguish between all sequences of matrices which are not simultaneously unitarily similar (Theorem VI.4). This implies, for instance, that a polylogarithmic number of iterations of the Weisfeiler-Lehman method suffices in order to distinguish graphs which are not cospectral. Finally, we come to the main application of our POLYLOG-FPC-definability results in Section VII, where we separate the power of the two propositional proof systems "Nullstellensatz" and "polynomial calculus" with respect to the graph (non)-isomorphism problem (Theorem VII.2).

II. Preliminaries

We assume that the reader has a solid background in logic. To fully understand and appreciate our results, familiarity with the ideas and techniques of finite model theory will be necessary (see [15], [24], [29], [16]). We briefly review the logics most important here.

In this paper, we only consider finite relational structures; a *τ-structure* $\mathfrak{A}$ has a finite *universe* A and a relation $R^A \subseteq A^k$ for each k-ary relation symbol R in the *vocabulary* τ. The class of all (finite) τ-structures is denoted by $\mathrm{Str}(\tau)$. We fix an encoding of pairs $(\mathfrak{A}, \mathfrak{B})$ of τ-structures as structures $\langle \mathfrak{A}, \mathfrak{B} \rangle$ of some vocabulary τ_{pair}, and we let $\mathrm{Pair}(\tau)$ denote the class of all pairs of τ-structures encoded this way. We view graphs $G = (V(G), E(G))$ as $\{E\}$-structures with universe $V(G)$ and binary edge relation $E(G)$. We denote the class of all pairs of graphs, encoded as $\{E\}_{\text{pair}}$-structures, by GraphPair.

Let us briefly review first-order logic FO. First-order formulas of vocabulary τ, or FO$[\tau]$-formulas, are built from atomic formulas $Rx_1, \ldots, x_k$ for k-ary $R \in \tau$ and $x = y$ using the usual Boolean connectives and existential and universal quantifiers ranging over the universe of a structure.

We next introduce *counting logic* C which is the (syntactic) extension of FO that allows counting quantifiers $\exists^{\geq m} x$ ("there exist at least m values for x") for each m. Note that C and FO are equally expressive, because $\exists^{\geq m} x$ can be expressed in FO using m quantifiers and m distinct variables for x. However, the quantifier rank of formulas in C may be significantly lower

than that of their equivalent counterparts in FO, because a counting quantifier $\exists^{\geq m} x$ only increases the quantifier rank by one. The same holds for the number of variables in a formula. By C^k we denote the fragment of C consisting of all formulas with at most k (free or bound) variables.

We write $\mathfrak{A} \equiv^k \mathfrak{B}$ if the structures $\mathfrak{A}$ and $\mathfrak{B}$ cannot be distinguished by any sentence in C^k, we write $\mathfrak{A} \equiv_r \mathfrak{B}$ if both structures cannot be distinguished by any sentence of C of quantifier rank at most r, and we write $\mathfrak{A} \equiv_r^k \mathfrak{B}$ if both structures cannot be distinguished by any sentence of C^k of quantifier rank at most r.

Inflationary fixed-point logic IFP is the extension of FO by a fixed-point operator with an inflationary semantics. To simplify the presentation, we only consider a special case and let $\varphi(X, \vec{x})$ be a formula that has a k-tuple $\bar{x} = (x_1, \ldots, x_k)$ of free *individual variables* ranging over the universe of a structure and a free k-ary *relation variable* X ranging over k-ary relations on the universe. (The general case is that φ has additional free variables.) For every structure $\mathfrak{A}$, we define a sequence of relations $X^{(i)} \subseteq A^k$, for $i \in \mathbb{N}$, by $X^{(0)} := \varnothing$ and

$$X^{(i+1)} := X^{(i)} \cup [\![\varphi(X^{(i)}, \bar{x})]\!]^{\mathfrak{A}} \qquad \text{for all } i \in \mathbb{N}, \tag{II.1}$$

where $[\![\varphi(X^{(i)}, \vec{x})]\!]^{\mathfrak{A}}$ denotes the set of all $\bar{a} \in A^k$ such that $\mathfrak{A}$ satisfies φ if the relation variable X is interpreted by $X^{(i)}$ and the individual variables in $\bar{x}$ are interpreted by $\bar{a}$. Since we have $X^{(0)} \subseteq X^{(1)} \subseteq X^{(2)} \subseteq \cdots \subseteq A^k$ and A is finite, the sequence reaches a fixed-point $X^{(n)} = X^{(n+1)}$, which we denote by $X^{(\infty)}$. We use the following syntax for the *ifp-operator*:

$$[\textsc{ifp}\, X\bar{x} \,.\, \varphi(\bar{x})]\,(\bar{x}), \tag{II.2}$$

In the structure $\mathfrak{A}$, this formula defines the relation $X^{(\infty)}$.

Example II.1. The following IFP-formula defines the transitive closure of a binary relation R:

$$[\textsc{ifp}\; Txy \,.\, (Rxy \vee \exists z(Txz \wedge Tzy))]\,(x, y). \tag{II.3}$$

Fixed-point logic with counting (FPC) is the extension of inflationary fixed-point logic by counting terms. Formulas of FPC are evaluated over the *two-sorted extension* of an input structure $\mathfrak{A}$ by a linear order of the size of the input structure. More precisely, we denote by $\mathfrak{A}^{\#}$ the two-sorted extension of a τ-structure $\mathfrak{A} = (A, R_1, \ldots, R_k)$ by a linear order $(\{0, \ldots, n\}, <)$ of length $n + 1$, where $n = |A|$, i.e. the two-sorted structure $\mathfrak{A}^{\#} = (A, R_1, \ldots, R_k, \{0, \ldots, n\}, <)$ where the universe of the first sort (also referred to as *vertex sort*) is A and the universe of the second sort (also referred to as *number sort* or *counting sort*) is $\{0, \ldots, n\}$.

For both the vertex and the number sort we have a collection of typed first-order variables. We also allow relation variables X of mixed types, i.e. a relation variable X of type $(k, \ell) \in \mathbb{N} \times \mathbb{N}$ ranges over relations $R \subseteq A^k \times \{0, \ldots, n\}^\ell$. In particular, we allow fixed-point operators over relations of mixed type. For example if X is of type $(k, \ell) \in \mathbb{N} \times \mathbb{N}$, then $[\textsc{ifp}\, X\bar{x} \,.\, \varphi(\bar{x})]\,(\bar{x})$ is a formula of FPC where the tuple of variables $\bar{x} = x_1, \ldots, x_{k+\ell}$ has to be compatible with the type of the relation symbol X, that is $x_1, \ldots, x_k$ are vertex variables and $x_{k+1}, \ldots, x_{k+\ell}$ are number variables. To relate the vertex and the number sort, fixed-point logic with counting allows the formation of *counting terms* to define sizes of sets. More precisely, for each formula φ we can form a *counting term* $s = [\#x \,.\, \varphi]$ whose value $s^{\mathfrak{A}} \in \{0, \ldots, n\}$ in a structure $\mathfrak{A}$ of size n is the number of elements $a \in A$ such that $\mathfrak{A} \models \varphi(a)$.

Logical interpretations: The logical counterpart of the notion of (algorithmic) reductions is the notion of *logical interpretations*. Basically, an interpretation $\mathcal{I}$ transforms each structure $\mathfrak{A}$ into a new structure $\mathfrak{B} = \mathcal{I}(\mathfrak{A})$ and this transformation is defined by formulas in some logic L. For the applications in this paper it suffices to give a formal definition for the case L = FO.

Let σ, τ be vocabularies with $\tau = \{S_1, \ldots, S_\ell\}$ where S_i is an s_i-ary relation symbol. An FO-*interpretation of* $\mathrm{Str}(\tau)$ *in* $\mathrm{Str}(\sigma)$ is a tuple

$$\mathcal{I} = \big(\varphi_\delta(\bar{x}), \varphi_1(\bar{x}_1, \ldots, \bar{x}_{s_1}), \ldots, \varphi_\ell(\bar{x}_1, \ldots, \bar{x}_{s_\ell})\big),$$

where $\varphi_\delta, \varphi_1, \ldots, \varphi_\ell \in \mathrm{FO}(\sigma)$, and where $\bar{x}, \bar{x}_1, \ldots$ are tuples of pairwise distinct variables of the same length $d \geq 1$ ($\mathcal{I}$ is then called a *d-dimensional* interpretation). Let $\mathrm{FO}[\sigma \to \tau]$ denote the set of FO-interpretations of $\mathrm{Str}(\tau)$ in $\mathrm{Str}(\sigma)$. The quantifier rank of $\mathcal{I}$ is the maximal quantifier rank of the first-order formulas $\varphi_\delta, \varphi_1, \ldots, \varphi_\ell$.

With every d-dimensional interpretation $\mathcal{I} \in \mathrm{FO}[\sigma \to \tau]$ we associate a mapping $\mathcal{I} : \mathrm{Str}(\sigma) \to \mathrm{Str}(\tau)$ as follows. For $\mathfrak{A} \in \mathrm{Str}(\sigma)$ we define the τ-structure $\mathcal{I}(\mathfrak{A}) = \mathfrak{B} \in \mathrm{Str}(\tau)$ over the universe $B = \{\bar{b} \in A^d : \mathfrak{A} \models \varphi_\delta(\bar{b})\}$ by setting

$$S_i^{\mathfrak{B}} = \{(\bar{b}_1, \ldots, \bar{b}_{s_i}) \in B^{s_i} : \mathfrak{A} \models \varphi_i(\bar{b}_1, \ldots, \bar{b}_{s_i})\}$$

for each $S_i \in \tau$.

The notion of an algorithmic reduction is the central tool in complexity theory to determine the relative algorithmic complexity between problems. Analogously, one can use logical interpretations to analyse the relative *descriptive* complexity between problems. In particular, just as in the case of complexity classes, many logics are closed under certain notions of interpretations in a way similar as stated in the following lemma for counting logic and FO-interpretations.

Lemma II.2. *Let $\mathcal{I} \in \mathrm{FO}[\sigma \to \tau]$ be a first-order interpretation from $\mathrm{Str}(\tau)$ in $\mathrm{Str}(\sigma)$ of dimension $d \geq 1$ and quantifier rank $r \geq 0$. For every sentence $\psi \in \mathrm{C}$ of counting logic of vocabulary τ with quantifier rank at most $\ell \geq 1$ we can find a sentence $\psi^{\mathcal{I}} \in \mathrm{C}$ of vocabulary σ with quantifier rank at most $r + d \cdot \ell$ such that for every σ-structure $\mathfrak{A}$ we have $\mathfrak{A} \models \psi^I$ if, and only if, $\mathcal{I}(\mathfrak{A}) \models \psi$.*

III. The Polylogarithmic Fragment of Fixed-point Logic with Counting

We introduce a restriction of fixed-point logic with counting where we bound the iterations of fixed-point operators by a polylogarithmic function. While fixed-point logic with counting (FPC) is tailored to expressing polynomial-time properties of finite structures, this polylogarithmic variant

(POLYLOG-FPC) is defined as a counterpart of the complexity class NC to capture problems which are efficiently parallelisable. To get more intuition, recall that Anderson and Dawar proved that FPC can express precisely those properties of finite structures that can be decided by uniform families of *symmetric* circuits of polynomial size [2]. The idea of POLYLOG-FPC is to capture those properties of finite structures which can be decided by uniform families of symmetric circuits of polynomial size *and* of polylogarithmic depth.

To obtain *polylogarithmic fixed-point logic with counting*, denoted by POLYLOG-FPC, we specify polylogarithmic bounds for the fixed-point operators in the formulas of FPC. More specifically, we use fixed-point operators only in the form $\left[\text{IFP}^{\leq \log^r(n)} X\bar{x}\,.\,\varphi(\bar{x})\right](\bar{x})$, for a constant $r \geq 1$. The semantics of this formula is determined as above except for that we stop the inflationary evaluation of the fixed point after at most $\log^r(n)$ steps (no matter of whether the inflationary fixed point was reached). That is, the formula defines the relation $X^{(\lceil \log^r(n) \rceil)}$, as defined in (II.1).

Many natural fixed-point processes converge after a polylogarithmic number of steps, so we have $X^{(\lceil \log^r(n) \rceil)} = X^\infty$. An example is the formula of Example II.1 defining the transitive closure, which converges after a logarithmic number of steps.

It is easy to see that every formula of POLYLOG-FPC can be evaluated in polylogarithmic parallel time and thus the data complexity of POLYLOG-FPC is in NC. Conversely, it is not hard to show that over *ordered* structures, POLYLOG-FPC can express all problems that can be solved in NC (we always refer to the uniform version of NC). In other words, POLYLOG-FPC captures NC on ordered structures. This even holds in the absence of counting, since over ordered structures one can use the fixed-point operators to simulate counting terms.

A. Embedding into Counting Logic

It is known that every formula $\varphi \in \text{FPC}$ can be translated into a family of formulas $(\varphi_n)_{n\geq 1} \in \text{C}^k$, for some $k \geq 1$, such that for all $n \geq 1$, the formula φ_n is equivalent to φ on structures of size at most n, see e.g. [32]. If we apply this embedding to POLYLOG-FPC, then we obtain formulas φ_n with polylogarithmic quantifier rank.

Theorem III.1. *For every sentence $\varphi \in$ POLYLOG-FPC there are constants $c, k, \ell \geq 1$ such that for all $n \geq 2$ there is a sentence $\varphi_n^* \in \text{C}^k$ of quantifier rank at most $c \cdot \log^\ell(n)$ that is equivalent to φ on structures of size at most n.*

B. Immerman's Lower Bound on the Quantifier Rank of Counting Logic

Polylogarithmic fixed-point logic with counting is a strict fragment of full fixed-point logic with counting. This follows from an old construction by Immerman [22].

Theorem III.2 ([22], see also [9], [24])**.** *For sufficiently large $m \geq 1$ there are graphs G_m, H_m with the following properties.*
(1) $|G_m| = |H_m| = n$, and $n < m^{1+\log(m)}$.
(2) $G_m \equiv_m H_m$.
(3) $G_m \not\equiv^3 H_m$.

In words, the graphs G_m and H_m cannot be distinguished by any sentence of counting logic C of quantifier rank at most m, although these graphs can be distinguished by a sentence of the same logic when we allow unbounded quantifier rank and only three variables. Moreover, the size of the graphs G_m and H_m is quasipolynomial in m.

It follows from Immerman's result that FPC is more powerful than POLYLOG-FPC. In fact, a property that witnesses this separation is the graph isomorphism problem over the class of graphs G_m and H_m from Theorem III.2. More precisely, the class $\mathcal{K} = \{\langle G, H\rangle \in \text{GraphPair} : G, H \in \{G_m, H_m\} \text{ for some } m \geq 1\}$ is POLYLOG-FPC-definable, but not $\mathcal{K}_{\text{iso}} = \{\langle G_m, G_m\rangle : m \geq 1\} \cup \{\langle H_m, H_m\rangle : m \geq 1\}$. On the other hand, $\mathcal{K}_{\text{iso}}$ is definable in FPC.

To see this, first recall from Theorem III.1 that over structures of size n every formula of POLYLOG-FPC can be translated into an equivalent formula in counting logic C with a polylogarithmic bound on the quantifier rank (the exponent of the polylogarithmic function is fixed by the POLYLOG-FPC-formula). The size of the graphs G_m and H_m is quasipolynomial in m. Note that if we apply a polylogarithmic function to a quasipolynomial function in m, then we obtain a polylogarithmic function in m which clearly grows slower than the linear function m. Hence the resulting sentences are not able to distinguish between Immerman's graphs G_m and H_m for large enough $m \geq 1$. On the other hand, it is known that FPC can express equivalence in three-variable counting logic, see e.g. [32]. This suffices to distinguish between G_m and H_m.

Corollary III.3. POLYLOG-FPC < FPC.

IV. Linear-algebraic Objects as Relational Structures

A central aim of this paper is to study the POLYLOG-FPC-definability of important linear-algebraic problems, such as the solvability of linear equation systems, over the field $\mathbb{Q}$ of rationals and the field $\mathbb{C}$ of complex numbers. In this section we discuss how one can encode linear-algebraic objects such as matrices, vectors, and so on, by finite relational structures. It will be convenient for us to work over the field $\mathbb{C}$ throughout. However, the input vectors and matrices to our linear-algebraic problems will always be rationals or Gaussian rationals, that is, complex numbers with rational real and imaginary parts. Note that a system of linear equations with rational coefficients always has a rational solution, and a linear system whose coefficients are Gaussian rationals has a solution in the Gaussian rationals.

Complex numbers: We start to explain how to encode complex numbers $c = \text{Re}(c) + i \cdot \text{Im}(c)$ with $\text{Re}(c), \text{Im}(c) \in \mathbb{Q}$. Let $\tau_{\text{lin}} = \{<\}$ and let $\tau_{\mathbb{C}} = \tau_{\text{lin}} \uplus \{N_{\text{Re}}, D_{\text{Re}}, N_{\text{Im}}, D_{\text{Im}}, S_{\text{Re}}, S_{\text{Im}}\}$ where $<$ is a binary relation symbol and where $N_{\text{Re}}, D_{\text{Re}}, N_{\text{Im}}, D_{\text{Im}}$ are unary relation symbols, and where $S_{\text{Re}}, S_{\text{Im}}$ are nullary predicates. We consider $\tau_{\mathbb{C}}$-structures $(A, <, N_{\text{Re}}, D_{\text{Re}}, N_{\text{Im}}, D_{\text{Im}}, S_{\text{Re}}, S_{\text{Im}})$ where $<$ is a linear order on A. Clearly, we can uniquely identify such structures with structures over the universe

$\{0,\dots,n\}$ with $<$ being the natural order. The predicates $N_{\mathrm{Re}}, D_{\mathrm{Re}}, N_{\mathrm{Im}}, D_{\mathrm{Im}}$ represent the binary encodings of the absolute values of the numerators and denominators of the real and imaginary part of c. For example, the numerator of the absolute value of the real part is encoded by N_{Re} and is determined as $\sum_{i \in N_{\mathrm{Re}}} 2^i$. The nullary predicates $S_{\mathrm{Re}}, S_{\mathrm{Im}}$ are used to encode the signs of $\mathrm{Re}(c)$ and $\mathrm{Im}(c)$, respectively. Of course, we require some simple consistency conditions, for example $D_{\mathrm{Re}} \neq \emptyset \neq D_{\mathrm{Im}}$ (non-zero denominators). We denote by $\mathcal{K}_{\mathbb{C}}$ the (first-order definable) class consisting of all $\tau_{\mathbb{C}}$-structures which encode complex numbers.

Matrices: Formally, matrices M (with complex coefficients) are mappings $M : I \times J \to \mathbb{C}$ for two non-empty index sets I, J. In general we do not require that the index sets I and J are ordered. To stress this fact we sometimes speak of *unordered* matrices. To represent unordered matrices by structures we use the vocabulary $\tau_{\mathrm{Mat}} = \{I, J, L, <, N_{\mathrm{Re}}, D_{\mathrm{Re}}, N_{\mathrm{Im}}, D_{\mathrm{Im}}, S_{\mathrm{Re}}, S_{\mathrm{Im}}\}$ with unary relation symbols I, J, L, binary relation symbols $<, S_{\mathrm{Re}}, S_{\mathrm{Im}}$ and 4-ary relation symbols $N_{\mathrm{Re}}, D_{\mathrm{Re}}, N_{\mathrm{Im}}, D_{\mathrm{Im}}$. We consider τ_{Mat}-structures $(A, I, J, L, <, N_{\mathrm{Re}}, D_{\mathrm{Re}}, N_{\mathrm{Im}}, D_{\mathrm{Im}}, S_{\mathrm{Re}}, S_{\mathrm{Im}})$ where $<$ is a linear order on L, where $L, I, J \neq \emptyset$, $I \cup J \cup L = A$, and where for every $i \in I$ and $j \in J$, the structure $(L, <, N_{\mathrm{Re}}(i, j, -, -), D_{\mathrm{Re}}(i, j, -, -), N_{\mathrm{Im}}(i, j, -, -), D_{\mathrm{Im}}(i, j, -, -), S_{\mathrm{Re}}(i, j), S_{\mathrm{Im}}(i, j))$ encodes the (i, j)-th entry $M(i, j) \in \mathbb{C}$ of the matrix $M : I \times J \to \mathbb{C}$ as a complex number as explained in the preceding paragraph.

We denote by $\mathcal{K}_{\mathrm{Mat}}$ the (first-order definable) class consisting of all τ_{Mat}-structures which encode matrices in this sense. We usually identify matrices with their structural encodings. A matrix $M \in \mathcal{K}_{\mathrm{Mat}}$, $M : I \times J \to \mathbb{C}$, is a *square matrix* if $I = J$. We identify (column) vectors with mappings $b : I \times \{0\} \to \mathbb{C}$ and row vectors with mappings $b : \{0\} \times J \to \mathbb{C}$. In particular vectors are structures in $\mathcal{K}_{\mathrm{Mat}}$. We speak of an *ordered matrix* $M \in \mathcal{K}_{\mathrm{Mat}}$ if $I, J \subseteq L$, that is if the encoding of M provides a linear order on the index sets for the rows and columns.

Families of matrices: For the sake of conciseness, we do not explicitly specify the following encodings. We consider *pairs of matrices* compatible with *matrix addition*, that is pairs of matrices $M : I \times J \to \mathbb{C}$ and $N : I \times J \to \mathbb{C}$ over the same index sets I and J. Let $\tau_{\mathrm{pairMat}(+)}$ denote an appropriate vocabulary to encode such pairs and let $\mathcal{K}_{\mathrm{pairMat}(+)}$ denote the class of $\tau_{\mathrm{pairMat}(+)}$-structures representing such pairs. We further consider structures which encode pairs of matrices $M : I \times J \to \mathbb{C}$ and $N : J \times K \to \mathbb{C}$ which are compatible with *matrix multiplication*. Let $\tau_{\mathrm{pairMat}(*)}$ denote an appropriate vocabulary to encode such pairs and let $\mathcal{K}_{\text{pair-Mat}(*)}$ be the class of $\tau_{\mathrm{pairMat}(*)}$-structures representing such matrix pairs.

Moreover, we consider *(ordered) sequences of square matrices over a common index set*, that is sequences of the form $\mathcal{M} = (M_0, \dots, M_{s-1})$ where all M_i are square matrices $M_i : I \times I \to \mathbb{C}$. All matrices in this sequence are compatible with matrix addition and multiplication. In particular, since the sequence is ordered, the product of the sequence is also well-defined. Let τ_{seqMat} denote an appropriate vocabulary to encode ordered sequences of square matrices and let $\mathcal{K}_{\mathrm{seqMat}}$ be the class of τ_{seqMat}-structures representing such sequences.

Linear equation systems: We also need an encoding for linear equation systems with complex coefficients by finite relational structures. We can represent every such system in the form $M \cdot x = b$ for an $I \times J$-coefficient matrix $M : I \times J \to \mathbb{C}$ and an I-vector of constants $b : I \to \mathbb{C}$. We already saw above that we can encode such pairs (M, b) by relational structures. Hence, we let τ_{les} denote an appropriate vocabulary to encode linear equation systems (M, b) and let $\mathcal{K}_{\mathrm{les}}$ be the class of τ_{les}-structures which encode linear equation systems.

V. Linear Algebra in Polylogarithmic Fixed-point Logic with Counting

We explore the expressive power of POLYLOG-FPC with respect to queries from linear algebra over the complex field $\mathbb{C}$. We will see that many non-trivial properties, such as the solvability of linear equation systems and the coefficients of the characteristic polynomial, can be expressed in POLYLOG-FPC. Central ideas that underly our approaches have been used, for example, also by Holm and Laubner [21], [28] in order to obtain definability results for the stronger logic FPC.

As POLYLOG-FPC captures NC on ordered structures, we can express every NC-decidable property over the numeric sort in POLYLOG-FPC. For instance, we can define in POLYLOG-FPC the addition and multiplication of (sets of) complex numbers (in binary representation). Also, over ordered inputs, we can define a host of important problems from linear algebra in POLYLOG-FPC over $\mathbb{C}$ and over finite fields, such as (iterated) matrix multiplication, computing the coefficients of the characteristic polynomial, computing the matrix rank, and also deciding the solvability of linear equation systems, see e.g. [26].

However, this does *not* mean that the corresponding problems remain POLYLOG-FPC-definable over *unordered* inputs. In fact, Atserias, Bulatov, and Dawar showed that the solvability of linear equation systems over *finite* fields can *not* be defined even in full FPC [4]. Surprisingly, the picture changes if we consider queries from linear algebra over the complex field. Here it turns out that the above mentioned queries are all definable in POLYLOG-FPC also over unordered inputs.

A. Matrix Arithmetic

We first observe that matrix addition can be defined in POLYLOG-FPC. Consider a pair of matrices $(M, N) \in \mathcal{K}_{\mathrm{pairMat}(+)}$, $M : I \times J \to \mathbb{C}$ and $N : I \times J \to \mathbb{C}$. Then we have that $(M + N)(i, j) = M(i, j) + N(i, j)$. Hence, we only have to express the addition of complex numbers. As complex numbers are ordered objects this is possible in POLYLOG-FPC, since POLYLOG-FPC can express every NC-property of ordered inputs.

For us the most important observation is that the multiplication of two *unordered* complex matrices is POLYLOG-FPC-definable. Assume we have a representation of two matrices $(M, N) \in \mathcal{K}_{\text{pair-Mat}(*)}$, $M : I \times J \to \mathbb{C}$ and $N : J \times K \to \mathbb{C}$, compatible for matrix multiplication. Recall that the (i, k)-th entry of the product $(M \cdot N)$ is given as $M(i, k) =$

$\sum_{j\in J} M(i,j)\cdot N(j,k)$. Again, the multiplication of the complex numbers $M(i,j)\cdot N(j,k)$ is definable in POLYLOG-FPC, because this is an NC-computable function. The interesting question, however, is how to evaluate the *unordered* sum of complex numbers in the above equation. Here we crucially rely on the counting mechanism of POLYLOG-FPC in order to reduce this task to the evaluation of an *ordered* sum of complex numbers. First we consider the POLYLOG-FPC-definable linear preorder $\preceq$ on J that is defined as $j \preceq j'$ if $(M(i,j)\cdot N(j,k)) \leq (M(i,j')\cdot N(j',k))$, where $\leq$ is the order on $\mathbb{C}$ defined as $x \leq y$ if $\mathrm{Re}(x) < \mathrm{Re}(y)$ or $\mathrm{Re}(x) = \mathrm{Re}(y)$ and $\mathrm{Im}(x) \leq \mathrm{Im}(y)$. The resulting equivalence classes $j/_\sim \in J/_\sim$ consist of elements which contribute the same complex number to the above sum. If we denote the sizes of these classes by $|j/_\sim|$, then we get

$$\sum_{j\in J} M(i,j)\cdot N(j,k) = \sum_{j/_\sim \in J/_\sim} |j/_\sim|\cdot M(i,j)\cdot N(j,k).$$

Since $\preceq$ induces a linear order on $J/_\sim$ and since we can determine the sizes of the equivalence classes $|j/_\sim|$ with counting terms we have indeed reduced the evaluation of an unordered sum of complex numbers to the evaluation of an *ordered* sum. In this way we obtain the POLYLOG-FPC-definability of matrix multiplication.

The above generalises for products over ordered sequences of square matrices. Indeed, since POLYLOG-FPC can express products of pairs of matrices, we can evaluate in POLYLOG-FPC products (of polynomial length) using a standard divide-and-conquer approach.

Let us also mention that the above technique to evaluate unordered sums of complex numbers in POLYLOG-FPC has many further applications. For example, it can be used to show that the *trace* $\mathrm{tr}(M) = \sum_{i\in I} M(i,i)$ of a square matrix $M \in \mathbb{C}^{I\times I}$ is definable in POLYLOG-FPC.

B. Linear Equation Systems over the Complex Field

We describe a new, and surprisingly simple, POLYLOG-FPC-definable reduction that transforms a given linear equation system with *unordered* sets of variables and equations into an *equivalent* linear equation system with *ordered* sets of variables and equations (over the complex field $\mathbb{C}$). Since the solvability of linear equation systems (over the field $\mathbb{C}$) can be decided in NC, it follows from this reduction that the solvability of an (unordered) linear equation system over $\mathbb{C}$ can be defined in POLYLOG-FPC.

Let $M\cdot x = b$ be a linear equation system for an $I\times J$-matrix M over $\mathbb{C}$ and an I-vector b over $\mathbb{C}$. The first step is to transform this system into a (solvability-)equivalent linear equation system whose coefficient matrix is a Hermitian matrix, that is a square complex matrix that is equal to its own conjugate transpose. Recall that the *conjugate transpose* of a matrix $S : I\times I \to \mathbb{C}$ is the matrix S^* which results from the transpose S^T of S by taking the complex conjugates of all entries, for instance:

$$\begin{pmatrix} i & 1+i \\ 2-i & 2 \end{pmatrix}^* = \begin{pmatrix} -i & 2+i \\ 1-i & 2 \end{pmatrix}.$$

Let $K := I \uplus J$. Then the linear equation system $N\cdot y = c$ over $\mathbb{C}$ with the Hermitian $K\times K$-coefficient matrix N and the K-vector c given as

$$N = \begin{pmatrix} 0 & M \\ M^* & 0 \end{pmatrix}, \quad c = \begin{pmatrix} b \\ 0 \end{pmatrix},$$

is (solvability-)equivalent to the original system and can easily be obtained from $M\cdot x = b$.

From now on assume that $M\cdot x = b$ is a linear equation system over $\mathbb{C}$ given by an Hermitian matrix $M \in \mathbb{C}^{I\times I}$ and an I-vector $b \in \mathbb{C}^I$. The important consequence of M being Hermitian is the following well known fact from linear algebra. Recall that the *kernel* $\ker(M)$ of a matrix $M \in \mathbb{C}^{I\times J}$ is the set of all $c \in \mathbb{C}^J$ such that $Mc = 0$.

Lemma V.1. *If $M \in \mathbb{C}^{I\times I}$ is Hermitian, then $\ker(M) = \ker(M^i)$ for all $i \geq 1$.*

Proof. This easily follows by induction on $i \geq 1$. The case $i = 1$ is trivial so let $i = 2$. Clearly, we have $\ker(M) \subseteq \ker(M^2)$ and in general $\ker(M) \subseteq \ker(M^i)$ for all $i \geq 1$. On the other hand, if $M^2\cdot a = 0$, then also $M^*Ma = 0$ since $M = M^*$. Hence, also $a^*M^*Ma = 0$. This implies that $(Ma)^*(Ma) = 0$ which in turn implies that $Ma = 0$ and hence $a \in \ker(M)$. For $i > 2$ we have that $M^i a = 0$ implies $M^{i-1}(Ma) = 0$. By induction hypothesis, we have $Ma \in \ker(M)$, hence $a \in \ker(M^2) = \ker(M)$. □

We remark that an analogous result for symmetric matrices over finite fields does not hold. The following lemma is the key step in our transformation of an unordered system of linear equations into an ordered system.

Lemma V.2. *Let $M \in \mathbb{C}^{I\times I}$ be Hermitian, $b \in \mathbb{C}^I$, $m = |I|$. The linear equation system $M\cdot x = b$ is solvable if, and only if, $b \in$ Span$(Mb,\ldots,M^m b)$.*

Proof. For the backward direction, note that if $b = \sum_{i=1}^m z_i M^i b \in \mathrm{Span}(Mb,\cdots,M^m b)$, then $c = \sum_{i=1}^m z_i M^{i-1} b$ is a solution for the system $Mx = b$.

For the forward direction, assume that $Mc = b$ for a solution $c \in \mathbb{C}^I$. The set $\{b, Mb,\ldots,M^m b\}$ has size $m+1$ and so it is linearly dependent. Hence, let $z_0 b + z_1 Mb + \cdots + z_m M^m b = 0$ for a non-zero $z = (z_0,\ldots,z_m) \in \mathbb{C}^{m+1}$. Let $i \geq 0$ be minimal such that $z_i \neq 0$. If $i = 0$, then we are done. Otherwise $z_i M^i b + \cdots + z_m M^m b = 0$ for some $i > 0$. Now we make use of the fact that $Mc = b$. Then the former equation can be rewritten as $z_i M^{i+1} c + \cdots + z_m M^{m+1} c = 0$. We obtain $M^{i+1}(z_i c + \cdots + z_m M^{m-i} c) = 0$. By Lemma V.1 we conclude that $M(z_i c + \cdots + z_m M^{m-i} c) = 0$. Again by making use of the fact that $Mc = b$ we get $z_i b + z_{i+1} Mb + \cdots + z_m M^{m-i} b = 0$. Since $z_i \neq 0$ the claim follows. □

Let N denote the $I\times\{1,\ldots,m\}$-matrix over $\mathbb{C}$ whose i-th column is the I-vector $M^i b$. By the above, we conclude that the linear equation system $M\cdot x = b$ is (solvability-)equivalent to the system $N\cdot y = b$ whose columns are indexed by the *ordered* set $\{1,\ldots,m\}$. The lexicographical order induced by this order on the rows of N is a linear order up to duplicates of

equations. Hence, we obtain an equivalent system with ordered sets of variables and equations. It follows from our earlier discussion about the POLYLOG-FPC-definability of matrix multiplication that our outlined reduction can be expressed in POLYLOG-FPC. The solvability of the ordered system can be defined in POLYLOG-FPC since this is an NC-property.

Theorem V.3. *There exists a* POLYLOG-FPC*-sentence φ of vocabulary τ_{les} such that for all linear equation systems $(M,b) \in \mathcal{K}_{les}$ we have*

$$(M,b) \text{ is solvable} \iff (M,b) \models \varphi.$$

For later reference, we also state a version in terms of counting logic with polylogarithmic quantifier rank.

Corollary V.4. *There exists $t \geq 1$ such that for all sufficiently large $n \geq 1$ we can find a sentence $\varphi_n \in \mathrm{C}$ of vocabulary τ_{les} such that $\mathrm{qr}(\varphi_n) \leq \log^t(n)$, and such that for all linear equation systems $(M,b) \in \mathcal{K}_{les}$ of size $|(M,b)| \leq n$ we have*

$$(M,b) \text{ is solvable} \iff (M,b) \models \varphi_n.$$

C. Coefficients of the Characteristic Polynomial

We next show that also the coefficients of the characteristic polynomial of complex matrices can be defined in POLYLOG-FPC. This result has many interesting consequences. For instance, it follows that the determinant and the rank of complex matrices are POLYLOG-FPC-definable.

The definability of the characteristic polynomial for complex matrices has been established in [21] with respect to full FPC. The idea there is to show that *Csanky's algorithm* can be expressed in FPC. Our observation is that for the simulation of this algorithm a POLYLOG-FPC-formula suffices.

Let us recall Csanky's algorithm. For details we refer to [26]. Let $M \in \mathbb{C}^{I \times I}$ be an $I \times I$-matrix over $\mathbb{C}$. Let $n = |I|$. Then the characteristic polynomial χ_M of M is

$$\chi_M = \det(x \cdot \mathrm{id}_I - M) = s_0 \cdot x^n - s_1 x^{n-1} + s_2 x^{n-2} - \cdots \pm s_n,$$

where the coefficients s_i can be obtained via the following linear recurrences: $s_0 = 1$, $s_1 = \mathrm{tr}(M)$, and, more generally,

$$s_k = \frac{1}{k}(s_{k-1}\,\mathrm{tr}(M) - s_{k-2}\,\mathrm{tr}(M^2) + \cdots \pm \mathrm{tr}(M^k)) \quad \cdots$$
$$s_n = \frac{1}{n}(s_{n-1}\,\mathrm{tr}(M) - s_{n-2} tr(M^2) + \cdots \pm \mathrm{tr}(M^n)) = \det(M).$$

These relations can be obtained from Newton's identities for symmetric polynomials. Now, if we want to define this system of linear recurrences in POLYLOG-FPC, then we only have to determine the powers M^i of the matrix M for all $i \leq n = |I|$ and then compute the corresponding traces $\mathrm{tr}(M^i)$, that is the sums over the diagonal entries of the matrices M^i. It follows from our earlier observations that this is possible in POLYLOG-FPC. The question remains whether the *unique solution* of this system, which is the characteristic polynomial of the matrix M, can be defined in POLYLOG-FPC. Again, this follows from the fact that this unique solution can be computed in NC and the observation that the above system of linear recurrences is *ordered*. To see this, we express the above system as a matrix equation $L \cdot s = c$ for a *lower triangular matrix* L. Then the unique solution of the system is given as $s = L^{-1}c$. Hence, in order to compute the solution it suffices to compute the inverse L^{-1} of the non-singular lower triangular matrix L. This can be done in NC by using a divide-and-conquer strategy based on the following identity which holds for all non-singular lower triangular square matrices $L \in \mathrm{Mat}_n(\mathbb{C})$ and square matrices $A, B, C \in \mathrm{Mat}_{n/2}(\mathbb{C})$:

$$\text{for } L = \begin{pmatrix} A & 0 \\ C & B \end{pmatrix}, \text{ we have } L^{-1} = \begin{pmatrix} A^{-1} & 0 \\ -B^{-1}CA^{-1} & B^{-1} \end{pmatrix}.$$

As mentioned before, from this definability result we can extract POLYLOG-FPC-definability results for the determinant, the matrix rank, matrix inverses, and so on, via the standard reductions, see for example [26]. For completeness, we recall the steps which reduce the computation of the rank of a matrix $M \in \mathbb{C}^{I \times J}$ to the computation of a characteristic polynomial. The first step is to obtain a Hermitian matrix. Using similar arguments as in the proof of Lemma V.1 it is easy to check that the rank of the matrix M is the same as the rank of the matrix $M^* \cdot M$. Since $M^* \cdot M$ is Hermitian, we get:

$$\mathrm{rk}(M) = \mathrm{rk}(M^*M) = \mathrm{rk}\left((M^*M)^2\right).$$

Finally, for matrices $N \in \mathbb{C}^{I \times I}$ with the property $\mathrm{rk}(N) = \mathrm{rk}(N^2)$, the matrix rank can be determined using the following well-known result from linear algebra.

Theorem V.5. *Let $N \in \mathbb{C}^{I \times I}$ with $|I| = n$ and such that $\mathrm{rk}(N) = \mathrm{rk}(N^2)$. Then $\mathrm{rk}(N) = n - k$ where $k \geq 0$ is maximal such that x^k divides the characteristic polynomial $\chi_N(x)$.*

D. Application to First-order Rank Logic

In his dissertation [21], Holm introduced an extension of first-order logic by operators which can compute the rank of matrices over the rationals $\mathbb{Q}$. He proved that this logic, denoted by $\mathrm{FOR}_{\mathbb{Q}}$, is contained in FPC and that over ordered structures it captures the exact logspace counting hierarchy, a complexity class introduced by Allender and Ogihara [1] to capture the (algorithmic) complexity of deciding singularity for integer matrices. Holm leaves open whether $\mathrm{FOR}_{\mathbb{Q}}$ is a strict fragment of FPC. From our above definability results it follows that Holm's logic $\mathrm{FOR}_{\mathbb{Q}}$ is contained in POLYLOG-FPC. Hence we get the following.

Theorem V.6. $\mathrm{FOR}_{\mathbb{Q}} \leq$ POLYLOG-FPC $<$ FPC.

VI. Simultaneous Unitary Similarity

In this section we review a classical theorem of Specht from 1940 which provides a characterisation for the *simultaneous unitary similarity problem* for pairs of ordered sequences of square matrices (of the same dimension) over the field of complex numbers. Our motivation is to use this criterion to show that counting logic with polylogarithmic quantifier rank can distinguish between all pairs of (non-isomorphic) graphs which have different linear-algebraic properties over the field of complex numbers.

Recall that the set $\mathbb{C}^{I\times I}$ of all $(I\times I)$-square matrices over $\mathbb{C}$ forms a $\mathbb{C}$-algebra. By $\mathrm{GL}_I(\mathbb{C})$ we denote the multiplication group of all non-singular matrices $S \in \mathbb{C}^{I\times I}$. A non-singular matrix $S \in \mathbb{C}^{I\times I}$ is called *unitary* if its inverse is the conjugate transpose, that is, $S^{-1} = S^*$. Two matrices $A, B \in \mathbb{C}^{I\times I}$ are *similar* if there exists an invertible matrix $S \in \mathrm{GL}_I(\mathbb{C})$ such that $SA = BS$. We consider a refinement of this equivalence relation and say that two matrices $A, B \in \mathbb{C}^{I\times I}$ are *unitarily similar* if we can find a unitary matrix $S \in \mathrm{GL}_I(\mathbb{C})$ such that $SA = BS$.

We will show that if two graphs cannot be distinguished by fixed-point logic with counting with a polylogarithmic number of iterations, or equivalently, by a formula of counting logic C of polylogarithmic quantifier rank, then all finite sequences of matrices definable by a fixed first-order interpretation over the graphs are (in fact simultaneously) related via a unitary similarity transformation. This means, in particular, that all of these matrices are similar.

Example VI.1. As a simple application, where we just need to look at the adjacency matrices of the graphs, we see that POLYLOG-FPC can distinguish between graphs which are not cospectral, that is which have different (multi-)sets of eigenvalues.

Let $s \geq 1$ and let $\mathcal{M} = (M_1, \ldots, M_s)$ and $\mathcal{N} = (N_1, \ldots, N_s)$ denote two sequences of square matrices $M_i, N_i \in \mathbb{C}^{I\times I}$ with complex coefficients and with the same index sets I. We say that the sequences $\mathcal{M}$ and $\mathcal{N}$ are *simultaneously unitarily similar* for short *s.u.s.*, if there exists a single unitary matrix $S \in \mathrm{GL}_I(\mathbb{C})$ such that $SM_i = N_iS$ for all $1 \leq i \leq s$.

To see whether two ordered families of complex matrices are s.u.s. we make use of the following characterisation by Specht. Let $\Sigma_s = \{x_1, x_1^*, \ldots, x_s, x_s^*\}$ denote the alphabet consisting of the letters x_i and x_i^* for all $1 \leq i \leq s$. For a finite word $x \in \Sigma_s^{<\omega}$ over Σ_s let $x_{\mathcal{M}} \in \mathbb{C}^{I\times I}$ denote the matrix which results from x by replacing all letters x_i and x_i^* by the matrices M_i and M_i^*, respectively, and by evaluating the resulting product. In particular, for the empty word $x = \varepsilon \in \Sigma_s^{<\omega}$ we agree to set $x_{\mathcal{M}} = \mathrm{id}_I$. The matrices $x_{\mathcal{N}} \in \mathbb{C}^{I\times I}$ for words $x \in \Sigma_s^{<\omega}$ are defined in the same way.

Two sequences $\mathcal{M} = (M_1, \ldots, M_s)$ and $\mathcal{N} = (N_1, \ldots, N_s)$ of complex matrices $M_i, N_i \in \mathbb{C}^{I\times I}$ are *trace equivalent* if for all words $x \in \Sigma_s^{<\omega}$ we have $\mathrm{tr}(x_{\mathcal{M}}) = \mathrm{tr}(x_{\mathcal{N}})$. Moreover, for $k \geq 1$, the sequences $\mathcal{M}$ and $\mathcal{N}$ are *k-trace equivalent* if for all words $x \in \Sigma_s^{\leq k}$ we have $\mathrm{tr}(x_{\mathcal{M}}) = \mathrm{tr}(x_{\mathcal{N}})$.

Theorem VI.2 (Specht's Theorem [35], [37], see also [34])**.** *Two sequences $\mathcal{M} = (M_1, \ldots, M_s)$ and $\mathcal{N} = (N_1, \ldots, N_s)$ of complex square matrices $M_i, N_i \in \mathbb{C}^{I\times I}$ are simultaneously unitarily similar if, and only if, they are trace equivalent.*

Specht's Theorem provides us with a criterion to test whether a pair of matrix sequences is s.u.s. in POLYLOG-FPC. Indeed, we only have to verify for every word $x \in \Sigma_s^{<\omega}$ whether $\mathrm{tr}(x_{\mathcal{M}}) = \mathrm{tr}(x_{\mathcal{N}})$ holds or not. This comes down to defining products and traces of matrices which, as we saw earlier, we can do in POLYLOG-FPC. Specifically, we have to express the following steps in POLYLOG-FPC for every word $x \in \Sigma_s^{<\omega}$:

- first of all, we define the conjugate transposes M_i^* and N_i^* of all input matrices M_i and N_i, and
- secondly, we determine the products $x_{\mathcal{M}}$ and $x_{\mathcal{N}}$ specified by the word x, and
- finally, we compare the traces of $x_{\mathcal{M}}$ and $x_{\mathcal{N}}$.

Basically all these steps are expressible in POLYLOG-FPC. However, there are two obvious problems. The most striking one is that there infinitely many words $x \in \Sigma_s^{<\omega}$ for which we have to test whether $\mathrm{tr}(x_{\mathcal{M}}) = \mathrm{tr}(x_{\mathcal{N}})$ holds or not. In particular, the length of these words $x \in \Sigma_s^{<\omega}$ is not bounded which means that also the sizes of (the representations of) the entries of the matrices $x_{\mathcal{M}}$ and $x_{\mathcal{N}}$ can become arbitrarily large. Fortunately it is not necessary to check all words in $\Sigma_s^{<\omega}$. In fact, Pearcy [33] proved that the two sequences $\mathcal{M}$ and $\mathcal{N}$ of square matrices are trace equivalent if, and only if, they are $2n^2$-trace equivalent where $n \geq 1$ denotes the dimension of the square matrices M_i, N_i, that is size of the index set I.

Theorem VI.3 ([33], see also [34])**.** *Two sequences $\mathcal{M} = (M_1, \ldots, M_s)$ and $\mathcal{N} = (N_1, \ldots, N_s)$ of complex square matrices $M_i, N_i \in \mathbb{C}^{I\times I}$ of dimension $n = |I|$ are trace equivalent if, and only if, $\mathcal{M}$ and $\mathcal{N}$ are $2n^2$-trace equivalent.*

This means that we only have to consider products $x_{\mathcal{M}}$ and $x_{\mathcal{N}}$ for all words $x \in \Sigma_s^{\leq 2n^2}$. In particular, the length of such products is polynomially bounded in the dimension of the input matrices which also gives us a polynomial bound on the (representations of the) entries of the matrices $x_{\mathcal{M}}$ and $x_{\mathcal{N}}$. From Section V we know that matrix products of polynomial length can be evaluated in POLYLOG-FPC. Also defining traces of matrices is possible in POLYLOG-FPC as this corresponds to the summation over an (unordered) set of complex numbers.

Unfortunately, there is a second problem: there still are exponentially many words $x \in \Sigma_s^{\leq 2n^2}$ which we have to check. Clearly, we cannot go through all words explicitly in POLYLOG-FPC. However, since we do not aim for an POLYLOG-FPC-definability result, this does not cause any trouble. In fact what we can do is to make use of the embedding of POLYLOG-FPC into counting logic C (Theorem III.1) which gives us for sequences of matrices $\mathcal{M}$ and $\mathcal{N}$ of a fixed size and for every *fixed* word $x \in \Sigma_s^{\leq 2n^2}$ a sentence in C with a polylogarithmic quantifier rank that expresses the trace condition $\mathrm{tr}(x_{\mathcal{M}}) = \mathrm{tr}(x_{\mathcal{N}})$ for $x_{\mathcal{M}}$ and $x_{\mathcal{N}}$. We can then take the conjunction over all these sentences for $x \in \Sigma_s^{\leq 2n^2}$, which does not increase the quantifier rank, to obtain a definition for the simultaneous similarity of the sequences $\mathcal{M}$ and $\mathcal{N}$ in C with polylogarithmic quantifier rank.

Theorem VI.4. *There are $k, r \geq 1$ such that the following holds. Let $\mathcal{M} = (M_1, \ldots, M_s) \in \mathcal{K}_{seqMat}$ and $\mathcal{N} = (N_1, \ldots, N_s) \in \mathcal{K}_{seqMat}$ be sequences of complex square matrices over index sets I and J of the same size, that is $M_i \in \mathbb{C}^{I\times I}$ and $N_i \in \mathbb{C}^{J\times J}$ and $|I| = |J|$, such that the size of $\mathcal{M}$ and $\mathcal{N}$ is n (where $n \geq 1$ is large enough and measures*

the size of the structures $\mathcal{M}$ and $\mathcal{N}$). If $\mathcal{M} \equiv^k_{\log^r(n)} \mathcal{N}$, then $\mathcal{M}$ and $\mathcal{N}$ are simultaneously unitarily similar.

Theorem VI.4 basically says that in counting logic with polylogarithmic quantifier rank we can express *all* linear-algebraic properties of finite structures which are invariant under unitary similarity (over the complex field). Let us give a more concrete application. We fix some constant $\ell \geq 1$. Now consider a sequence of formulas $\vec{\varphi} = (\varphi_1(x_1, \ldots, x_\ell, y_1, \ldots, y_\ell), \ldots, \varphi_s(x_1, \ldots, x_\ell, y_1, \ldots, y_\ell))$ of counting logic over graphs. Then this sequence defines in every graph $G = (V, E)$ a sequence of square matrices $G^{\vec{\varphi}} = (M_1^{\varphi}, \ldots, M_s^{\varphi})$ with entries $\{0, 1\} \subseteq \mathbb{C}$ over the index set V^ℓ in the obvious way:

$$M_i^{\varphi}(\bar{v}, \bar{w}) = 1, \text{ if, and only if, } G \models \varphi_i(\bar{v}, \bar{w}).$$

Now what Theorem VI.4 says is that we can find constants k (the dimension) and r (the polylogarithmic exponent) such that whenever two graphs G and H of size n are equivalent with respect to all formulas of k-variable counting logic with quantifier rank $\log^r(n)$, then the two sequences of matrices $G^{\vec{\varphi}}$ and $H^{\vec{\varphi}}$ are simultaneously unitarily similar.

Finally, let us remark that our POLYLOG-FPC-definability result in this section does not hold over finite fields. In fact, over finite fields, similarity of matrices cannot be defined even in full FPC [4]. Furthermore, quite interestingly, the criterion of simultaneous similarity of matrices over *finite fields* has been used by Dawar and Holm in [14] to define an isomorphism test which is provably strictly stronger than the Weisfeiler-Lehman method. Our results in this section show that this isomorphism test collapses to the usual Weisfeiler-Lehman method if considered over the complex field.

VII. Application to Graph Isomorphism Testing and Proof Complexity

We now turn to the main application of our definability results, which on a high level we already described in Section I-B. We start by introducing the relevant algebraic proof systems and the encoding of the graph isomorphism problem. (For details, we refer to [7] and the references cited there.)

A. Algebraic Proof Systems

Let $\mathbb{F}$ be a field (in this paper, $\mathbb{F}$ will always be the field $\mathbb{C}$ of complex numbers). We consider *polynomial equations* over a set of variables X_j, $j \in J$, ranging over $\mathbb{F}$. We do not assume to have an ordering on the index set J of variables, that is we consider polynomial equations over *unordered* sets of variables. We denote by $\mathbb{F}[\vec{X}]$ the ring of (multivariate) polynomials in variables X_j, $j \in J$, and with coefficients from the field $\mathbb{F}$. For a multi-index $\alpha : J \to \mathbb{N}$ we let the *monomial* X^α be defined as $X^\alpha = \Pi_{j \in J} X_j^{\alpha(j)}$. Then polynomials $f \in \mathbb{F}[\vec{X}]$ can be written as $f = \sum_\alpha f_\alpha \cdot X^\alpha$ where the $f_\alpha \in \mathbb{F}$ are coefficients from the field $\mathbb{F}$ and such that $f_\alpha \neq 0$ for finitely many α only. The *degree* $\deg(X^\alpha)$ of a monomial X^α is defined as $|\alpha| = \sum_{j \in J} \alpha(j)$, and the degree $\deg(f)$ of a polynomial $f = \sum_\alpha f_\alpha \cdot X^\alpha$ is defined as the maximal degree of a monomial X^α occurring in f with non-zero coefficient $f_\alpha \neq 0$. Note that $\deg(f \cdot g) = \deg(f) + \deg(g)$ for all non-zero polynomials $f, g \in \mathbb{F}[\vec{X}] \setminus \{0\}$.

A *polynomial equation* is an equation of the form $f = 0$ for a polynomial $f \in \mathbb{F}[\vec{X}]$. For better readability, we usually omit the equality "$= 0$" when we specify polynomial equations, that is we identify polynomials $f \in \mathbb{F}[\vec{X}]$ with the corresponding polynomial equations $f = 0$. A *system of polynomial equations* is a set $\mathcal{P} = \{f_i : i \in I\}$ consisting of polynomials $f_i \in \mathbb{F}[\vec{X}]$ for all $i \in I$ where I is an (unordered) index set. A *solution* of $\mathcal{P}$ is a common zero $\bar{a} \in \mathbb{F}^J$ of all polynomials in $\mathcal{P}$.

In what follows, we only consider systems $\mathcal{P} = \{f_i : i \in I\}$ which contain for every variable $X = X_j$, $j \in J$, the polynomial equation $(X^2 - X) = 0$. Note that these axioms $(X^2 - X) = 0$ enforce that each variable $X = X_j$, $j \in J$, can only take values 0 or 1.

The *algebraic proof systems* we shall introduce next can be used to *refute* a system $\mathcal{P}$ of equations, that is, to prove that it has no solution.

Nullstellensatz Proof System: This proof system is based on *Hilbert's Nullstellensatz*, an important result from algebra saying that the non-solvability of a system $\mathcal{P} = \{f_i : i \in I\}$ is equivalent to the existence of polynomials $g_i \in \mathbb{F}[\vec{X}], i \in I$, such that $\sum_{i \in I} g_i \cdot f_i = 1$. The polynomials g_i are called a *Nullstellensatz refutation* for the system $\mathcal{P}$. It can be shown that by the axioms $X^2 - X$ one can restrict to polynomials g_i whose degree is linear in the number of variables. Hence, Hilbert's Nullstellensatz can be used to search effectively, but of course not efficiently, for proofs for the non-solvability of the polynomial system $\mathcal{P}$. If we want to systematically search for Nullstellensatz refutations *efficiently*, that means in polynomial time, then we have to restrict the search space for the polynomials g_i. The most important approach is to restrict the degree of the polynomials g_i by a constant. Indeed, if we fix $d \geq 1$, then we can check efficiently, namely in time $n^{O(d)}$, whether there are polynomials $g_i \in \mathbb{F}[\vec{X}]$ satisfying that

- the degree of all products $g_i \cdot f_i$ is bounded by d, and
- $\sum_{i \in I} g_i \cdot f_i = 1$.

Nullstellensatz proofs via linear equation systems: We can reduce the question of whether a system $\mathcal{P}$ of polynomial equations in n variables has a Nullstellensatz refutation of degree at most d to the question of solving a system of linear equations in $n^{O(d)}$. This not only shows that we can find a refutation of fixed degree d in polynomial time, but also will enable us to connect Nullstellensatz refutations to definability in POLYLOG-FPC. The reduction goes back to [10].

As a first step, let us simplify the situation a little bit. In fact, the axioms $(X^2 - X) = 0$ allow us to restrict to *linearised* polynomials. Formally, we define the linearisation operator $\mathrm{Lin} : \mathbb{F}[\vec{X}] \to \mathbb{F}[\vec{X}]$ as the $\mathbb{F}$-linear operator uniquely determined by $\mathrm{Lin}(X^\alpha) = X^\beta$ where $\beta(j) = 1$ if $\alpha(j) > 0$ and $\beta(j) = 0$ if $\alpha(j) = 0$. Then Lin respects addition and $\mathbb{F}$-scalar multiplication, but not multiplication. For example $\mathrm{Lin}(X \cdot (X - 1)) \neq \mathrm{Lin}(X) \cdot \mathrm{Lin}(X - 1)$. However, we obtain the following weak preservation identity for multiplication. If $f, g \in \mathbb{F}[\vec{X}]$, then $\mathrm{Lin}(f \cdot g) = \mathrm{Lin}(\mathrm{Lin}(f) \cdot \mathrm{Lin}(g))$.

This implies that it always suffices to look for Nullstellensatz refutations consisting of multilinear polynomials g_i.

Let $d \geq 1$ be the degree bound and let D denote the set of all linearised monomials of degree $\leq d$. We associate with each monomial $X^\alpha \in D$ the set $m \subseteq J$ of variable (indices) which occur in X^α, that is $m = \{j \in J : \alpha(j) = 1\}$. Then $|m| \leq d$ and $X^\alpha = \Pi_{j \in m} X_j = X^m$. Let us now consider the $\mathbb{F}$-vector space $V = \mathbb{F}^D$ with basis D. Then we obtain the obvious $\mathbb{F}$-vector space isomorphism between the vector space V and the vector space of all multilinear polynomials $f \in \mathbb{F}[\vec{X}]$ with degree $\deg(g) \leq d$.

Let $g_i = \sum_{\alpha, |\alpha| \leq d} g_i^\alpha X^\alpha$ be a polynomial with $g_i^\alpha \in \mathbb{F}$ and such that $\deg(g_i \cdot f_i) \leq d$. Then $\deg(g) \leq d_i := d - \deg(f_i)$. We observe that $g_i \cdot f_i = \sum_{\alpha, |\alpha| \leq d} g_i^\alpha \cdot X^\alpha \cdot f_i$ and that $\mathrm{Lin}(g_i \cdot f_i) = \sum_{\alpha, |\alpha| \leq d} g_i^\alpha \cdot \mathrm{Lin}(\mathrm{Lin}(X^\alpha) \cdot \mathrm{Lin}(f_i))$. Altogether, this shows that there exist polynomials $g_i \in \mathbb{F}[\vec{X}]$, $i \in I$, with $\deg(g_i \cdot f_i) \leq d$ and $\sum_{i \in I} \mathrm{Lin}(g_i \cdot f_i) = 1$ if, and only if, the polynomial 1 is an $\mathbb{F}$-linear combination of the set of all polynomials $G_\mathcal{P} = \{h_i^m := \mathrm{Lin}(X^m \cdot f_i) : i \in I, m \subseteq J, |m| \leq d_i\}$. Note that this set $G_\mathcal{P}$ of lifted and linearised polynomials h_i^m has polynomial size (for fixed $d \geq 1$) and it can easily be constructed from $\mathcal{P}$. Since we can identify the polynomials h_i^m with vectors in V we can search for Nullstellensatz proofs of degree d by solving a linear equation system. More concretely, if we let M denote a matrix whose columns are the vectors $h_i^m \in V$ and if we further let $b \in V$ denote the vector corresponding to the polynomial $1 \in \mathbb{F}[\vec{X}]$, then we have that a Nullstellensatz refutation from $\mathcal{P}$ of degree d exists if, and only if, we find a linear combination x such that $M \cdot x = b$.

The polynomial calculus: The Nullstellensatz proof system is a "static" system in which we have to find the whole refutation in a single step. The polynomial calculus (PC) is a dynamic system which allows us to use polynomials that we have derived in subsequent derivation steps.

Starting with the axioms in $\mathcal{P}$, one can derive new equations, according to the following rules:

- *Linear combinations.* Every $\mathbb{F}$-linear combination of the derived polynomials can be derived as well.
- *Lifting (up to degree d).* If we can derive $f \in \mathbb{F}[\vec{X}]$, then we can also derive $(X^\alpha \cdot f)$ for every monomial X^α such that $\deg(X^\alpha \cdot f) \leq d$.

If we can obtain the constant polynomial $1 \in \mathbb{F}[\vec{X}]$ by iteratively applying these two rules, then a corresponding derivation is called a *PC-refutation* (of degree d) for the non-solvability of our system $\mathcal{P}$. As in the case of the Nullstellensatz proof system the axioms $X^2 - X = 0$ guarantee that we can always restrict to linearised polynomials when we consider PC-proofs. If we restrict the lifting rule of the polynomial calculus to such polynomials f which are either axioms or monomials, then we obtain the *monomial PC*. This propositional proof system lies between Nullstellensatz and polynomial calculus and was introduced in [7].

Finally, let us remark that also for the PC and the monomial PC one can efficiently check, namely in time $n^{\mathcal{O}(d)}$, whether a refutation of degree d exists using Gröbner basis computations, see [11].

Encoding of the graph isomorphism problem: We recall the encoding of the graph isomorphism problem as a system of polynomial equations from [7]. Let $G = (V, E)$ and $H = (W, F)$ be two graphs. We define a system $\mathcal{P}_{\text{iso}}(G, H)$ of polynomial equations which is solvable if, and only if, the two graph G and H are isomorphic. For every $(v, w) \in V \times W$ we introduce a variable $X[v \mapsto w]$. The idea is that a solution of $\mathcal{P}_{\text{iso}}(G, H)$ encodes an isomorphism between the graphs G and H. This isomorphism is determined in the obvious way: if the variable $X[v \mapsto w]$ is set to one, then v is mapped to w under this isomorphism. To this end, we include the following polynomial equations in our system $\mathcal{P}$:

$$\text{(B1)} \qquad \sum_{w \in W} X[v \mapsto w] - 1 = 0 \quad \text{for every } v \in V$$

$$\text{(B2)} \qquad \sum_{v \in V} X[v \mapsto w] - 1 = 0 \quad \text{for every } w \in W$$

$$\text{(ISO)} \quad X[v_1 \mapsto w_1] \cdot X[v_2 \mapsto w_2] = 0$$

for every mapping $v_1 v_2 \mapsto w_1 w_2$ that is *not* a local isomorphism.

We include the axioms $X[v \mapsto w]^2 - X[v \mapsto w] = 0$ for all variables $X[v \mapsto w]$ as usual. Then, over fields of characteristic zero, the equations (B1) and (B2) guarantee that solutions of $\mathcal{P}_{\text{iso}}(G, H)$ encode a bijection between the vertex sets V and W of the graphs G and H, and the equations (ISO) guarantee that this bijection respects the edge relations. Thus, the system $\mathcal{P}_{\text{iso}}(G, H)$ is solvable if, and only if, the graphs G and H are isomorphic.

The obvious question is: can we find a fixed degree $d \geq 1$ such that for all pairs of non-isomorphic graphs G and H we can prove the non-solvability of $\mathcal{P}_{\text{iso}}(G, H)$ in the polynomial calculus with degree d (or maybe even in the Nullstellensatz proof system or in the monomial PC)? If this would be true, then we had found an efficient algorithm for the graph isomorphism problem. Unfortunately, Berkholz and Grohe obtain linear lower bounds on the degree d which is required for the polynomial calculus to decide the graph isomorphism problem [7], [8]. Their lower bounds hold for the PC over all finite fields and over the rationals (or complex numbers).

However, what Grohe and Berkholz left open in [7] is the question about the *relative* expressive power of the three proof systems (Nullstellensatz, polynomial calculus, and monomial PC) with respect to the graph isomorphism problem. On the other hand, they established the following precise correspondence between the monomial PC and counting logic C. Recall that $\equiv^d$ denotes equivalence of two structures in the d-variable fragment C^d of counting logic C.

Theorem VII.1 ([7]). *Let G, H be graphs and $d \geq 1$. Then $\mathcal{P}_{\text{iso}}(G, H)$ has a monomial PC-refutation of degree d if, and only if, $G \not\equiv^d H$.*

B. Lower Bounds on Nullstellensatz Refutations

We now present the main application of our definability results for polylogarithmic fixed-point logic with counting which is the following lower bound on the complexity of Nullstellensatz proofs for the graph isomorphism problem.

Theorem VII.2. *For every $r \geq 1$ and for infinitely many $n \geq 1$ there exists graphs G, H of size n such that*

- $\mathcal{P}_{\text{iso}}(G,H)$ *has no Nullstellensatz refutation of degree at most* $\log^r(n)$.
- $\mathcal{P}_{\text{iso}}(G,H)$ *has a monomial PC refutation of degree* 3.

For $d \geq 1$ and for graphs G and H let us write $G \equiv_d^{\text{NST}} H$ if the graphs G and H cannot be distinguished in the Nullstellensatz proof system with degree d, that is if one cannot find a Nullstellensatz refutation of degree d for $\mathcal{P}_{\text{iso}}(G,H)$.

Lemma VII.3. *There is $c \geq 1$ such that for all $d \geq 1$ we can find a quantifier-free first-order interpretation $\mathcal{I}_d \in \text{FO}[\{E\}_{\text{pair}}, \tau_{les}]$ of dimension $c \cdot d$ such that for all $(G,H) \in$ GraphPair with $|G| = |H|$ we have*

$$G \equiv_d^{NST} H \Leftrightarrow \textit{the linear eq. system } \mathcal{I}_d(G,H) \textit{ is solvable}.$$

Proof sketch. Let $d \geq 1$ be the dimension for the Nullstellensatz proof system. By definition, $G \equiv_d^{\text{NST}} H$ if, and only if, $\mathcal{P}_{\text{iso}}(G,H)$ has no Nullstellensatz refutation of degree d. As explained in Section VII-A, we can express this as the solvability problem for a linear equation system over $\mathbb{Q}$ (note that here we have to take the dual system as we want to capture that $\mathcal{P}_{\text{iso}}(G,H)$ has *no* refutation). To obtain this linear equation system we basically have to lift all axioms in $\mathcal{P}_{\text{iso}}(G,H)$ by monomials of degree at most d and linearise the resulting polynomials. Suppose that $G = (V,E)$, $H = (W,F)$.

We obtain the following system of linear equations. The variables are $X[\bar{v} \mapsto \bar{w}]$ for tuples $\bar{v} = (v_1, \ldots, v_k) \in V(G)^k$, $\bar{w} = (w_1, \ldots, w_k) \in V(H)^k$, where $1 \leq k \leq d$ and $v_i \neq v_j, w_i \neq w_j$ for $1 \leq i < j \leq k$. The variable $X[\bar{v} \mapsto \bar{w}]$ corresponds to the multilinear monomial $\prod_{i=1}^k X[v_i \mapsto w_i]$.

$$\text{(LB1)} \qquad \sum_{w \in W} X[\bar{v}v \mapsto \bar{w}w] - X[\bar{v} \mapsto \bar{w}] = 0$$

for every v and $\bar{v}, \bar{w}$ of length $k, 0 \leq k < d$

$$\text{(LB2)} \qquad \sum_{v \in V} X[\bar{v}v \mapsto \bar{w}w] - X[\bar{v} \mapsto \bar{w}] = 0$$

for every w and $\bar{v}, \bar{w}$ of length $k, 0 \leq k < d$

$$\text{(LISO)} \qquad X[\bar{v} \mapsto \bar{w}] = 0$$

for all $\bar{v}, \bar{w}$ of length $\leq d$ such that $\bar{v} \mapsto \bar{w}$ is not a local isomorphism.

In the equations (LB1) and (LB2) we also included the case $k = 0$, that is, $\bar{v}, \bar{w}$ are empty tuples. One can read $X[() \mapsto ()]$ as a convenient notation for the constant 1. However, it is better to include $X[() \mapsto ()]$ as a new variable and to add the equation $X[() \mapsto ()] = 1$. The above system of linear equations is solvable if, and only if, there is no Nullstellensatz refutation of $\mathcal{P}_{\text{iso}}(G,H)$ of degree $d \geq 1$.

We can easily define this system using a quantifier-free first-order interpretation of dimension $c \cdot d$ (with $c \geq 1$ large enough) in the structure $\langle G, H \rangle$. Essentially, what we need to define is the matrix of this system of equations. The columns of this matrix are indexed by pairs of tuples $(\bar{v}, \bar{w}) \in \bigcup_{k=0}^d V^k \times W^k$. The rows are indexed by the equations, which we can describe by sufficiently long tuples of vertices of the two graphs. Fortunately, the matrix only has entries in $\{-1, 0, 1\}$, which makes the encoding a bit easier. Details of the interpretation depend on the exact encoding of the pair (G,H) of graphs as an $\{E\}_{\text{pair}}$-structure $\langle G, H \rangle$ as well as the exact encoding of matrices, and we leave them to the reader. □

In the following it is important to recall that if we interpret a structure $\mathfrak{B} = \mathcal{I}(\mathfrak{A})$ in a structure $\mathfrak{A}$ using a d-dimensional interpretation $\mathcal{I}$, then the size of $\mathfrak{B}$ is bounded by n^d where $n = |\mathfrak{A}|$. The main ingredient of the proof of Theorem VII.2 is our definability result in Theorem V.3 showing that POLYLOG-FPC can define the solvability of linear equation systems over the complex field. From this we also obtained a definability result for counting logic with polylogarithmic quantifier rank (Corollary V.4). Also we make use of Immerman's construction to separate the polylogarithmic fragment of counting logic from the full logic (Theorem III.2).

Proof of Theorem VII.2. Let $r \geq 1$ be fixed. We start with Immerman's construction and choose according to Theorem III.2 for large enough $m \geq 1$ graphs G_m and H_m of size $|G_m| = |H_m| = n$, with $n < m^{1+\log(m)}$, and such that $G_m \equiv_m H_m$ and $G_m \not\equiv^3 H_m$. Then by Theorem VII.1 we already know that $\mathcal{P}_{\text{iso}}(G_m, H_m)$ has a monomial PC-refutation of degree 3. For better readability, let $d = \log^r(n)$.

We now use Lemma VII.3 to find a constant $c \geq 1$ and a quantifier-free first-order interpretation $\mathcal{I}_d$ of dimension $c \cdot d$ which translates each pair of graphs (G^*, H^*), $G^*, H^* \in \{G_m, H_m\}$, into a linear equation system $\mathcal{I}_d(G^*, H^*)$ which is solvable if, and only if, $G^* \equiv_d^{\text{NST}} H^*$. Note that the size of this linear equation system is bounded by $n^{c \cdot d}$. Since d is bounded polylogarithmically in n, and since n is bounded quasipolynomially in m, we conclude that d is bounded polylogarithmically in m. Hence the size of the system $\mathcal{I}_d(G^*, H^*)$ is bounded quasipolynomially in m.

We now apply Corollary V.4 to obtain sentences φ_m of counting logic which define the solvability of these linear equation systems $\mathcal{I}_d(G^*, H^*)$ with a quantifier-rank which is polylogarithmically bounded in the size of the linear equation systems $\mathcal{I}_d(G^*, H^*)$. Note that by the above observation it follows that the quantifier rank of the sentences φ_m is polylogarithmically bounded in m. Using the interpretation lemma (Lemma II.2) we now translate the formulas φ_m back into equivalent formulas working over the original pairs of graphs (G^*, H^*). Since the dimension of the interpretation $\mathcal{I}_d$ is polylogarithmically bounded in m this just gives a polylogarithmic blow-up for the quantifier rank of φ_m. Hence, altogether we can obtain formulas ψ_m of counting logic whose quantifier rank is polylogarithmically bounded in m and such that $(G^*, H^*) \models \psi_m$ if, and only if, $G^* \equiv_d^{\text{NST}} H^*$.

Finally, since the graphs G_m and H_m are equivalent with respect to formulas of counting logic with quantifier rank m we conclude that for large enough m the formula ψ_m cannot distinguish between the pairs (G_m, G_m) and (G_m, H_m). Since $G_m \equiv_d^{\text{NST}} G_m$ we conclude that $G_m \equiv_d^{\text{NST}} H_m$. □

VIII. Conclusion

We studied the expressive power of POLYLOG-FPC which is the fragment of fixed-point logic with counting (FPC) with polylogarithmic bounds on the number of fixed-point iterations. An old construction by Immerman shows that POLYLOG-FPC is weaker than FPC. Many non-trivial linear-algebraic properties over $\mathbb{C}$ or $\mathbb{Q}$ can be expressed in POLYLOG-FPC, for example the solvability of linear equation systems. This separates the expressive power of linear equation systems over $\mathbb{Q}$ from full FPC and answers an open question due to Holm about the power of first-order rank logic over $\mathbb{Q}$. In fact, our results show that, from the logical perspective, solving linear equation systems (over $\mathbb{Q}$) is provably simpler than solving linear programs. We further applied our new POLYLOG-FPC-definability results to obtain lower bounds on the complexity of proofs in the Nullstellensatz proof system. This new, and quite surprising, connection between finite model theory and proof complexity is fascinating and we aim to study it in more detail in the future. Let us elaborate on some open problems.

First of all, many questions remain about the relative power of the three propositional proof systems Nullstellensatz, monomial PC, and polynomial calculus, with respect to the graph isomorphism problem. For instance, we still do not know whether monomial PC can be separated from the full polynomial calculus. Can finite model theory also help to answer this question? Also, it would be interesting to see in how far our separation results between Nullstellensatz and polynomial calculus can be transferred to finite fields.

A second line of research concerns the structure of the logic POLYLOG-FPC. The power of FPC can be characterised in terms of families of symmetric circuits of polynomial size [2]. We expect that a similar characterisation for POLYLOG-FPC can be obtained with respect to families of symmetric circuits of polynomial size and polylogarithmic depth.

References

[1] E. Allender and M. Ogihara. Relationships Among PL, #L, and the Determinant. *ITA*, 30(1):1–21, 1996.

[2] M. Anderson and A. Dawar. On Symmetric Circuits and Fixed-Point Logics. In *31st International Symposium on Theoretical Aspects of Computer Science*, page 41, 2014.

[3] M. Anderson, A. Dawar, and B. Holm. Solving Linear Programs without Breaking Abstractions. *Journal of the ACM*, 62(6):48:1–48:26, 2015.

[4] A. Atserias, A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410(18):1666–1683, 2009.

[5] A. Atserias and E. Maneva. Sherali–Adams Relaxations and Indistinguishability in Counting Logics. *SIAM Journal on Computing*, 42(1):112–137, 2013.

[6] P. Beame, R. Impagliazzo, J. Krajicek, T. Pitassi, and P. Pudlak. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 794–806, 1994.

[7] C. Berkholz and M. Grohe. Limitations of Algebraic Approaches to Graph Isomorphism Testing. In *ICALP*, Lecture Notes in Computer Science, pages 155–166. Springer, 2015.

[8] C. Berkholz and M. Grohe. Linear Diophantine Equations, Group CSPs, and Graph Isomorphism. In *Proceedings of the of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (to appear)*, 2017.

[9] C. Berkholz and J. Nordström. Near-Optimal Lower Bounds on Quantifier Depth and Weisfeiler-Leman Refinement Steps. In *LICS*, pages 267–276. ACM, 2016.

[10] S. Buss. Lower Bounds on Nullstellensatz Proofs via Designs. In *Proof Complexity and Feasible Arithmetics*, pages 59–71. American Mathematical Society, 1998.

[11] M. Clegg, J. Edmonds, and R. Impagliazzo. Using the Groebner Basis Algorithm to Find Proofs of Unsatisfiability. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 174–183, New York, NY, USA, 1996. ACM.

[12] A. Dawar. On the Descriptive Complexity of Linear Algebra. In *WoLLIC*, volume 5110 of *Lecture Notes in Computer Science*, pages 17–25. Springer, 2008.

[13] A. Dawar, M. Grohe, B. Holm, and B. Laubner. Logics with Rank Operators. In *Proceedings of the 24th IEEE Symposium on Logic in Computer Science*, pages 113–122, 2009.

[14] A. Dawar and B. Holm. Pebble games with algebraic rules. In *Automata, Languages, and Programming*, pages 251–262. Springer, 2012.

[15] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 2nd edition, 1999.

[16] E. Grädel, P.G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M.Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications*. Springer-Verlag, 2007.

[17] E. Grädel and W. Pakusa. Rank logic is dead, long live rank logic! In *Computer Science Logic (CSL'15)*, Leibniz International Proceedings in Informatics (LIPIcs), 2015.

[18] M. Grohe. Fixed-Point Definability and Polynomial Time on Graphs with Excluded Minors. *Journal of the ACM*, 59(5), 2012.

[19] M. Grohe and M. Otto. Pebble Games and Linear Equations. *Journal of Symbolic Logic*, 80(3):797–844, 2015.

[20] B. Grußien. *Capturing Polynomial Time and Logarithmic Space using Modular Decompositions and Limited Recursion*. PhD thesis, Humboldt-Universität zu Berlin, 2016.

[21] B. Holm. *Descriptive complexity of linear algebra*. PhD thesis, University of Cambridge, 2010.

[22] N. Immerman. Number of Quantifiers is Better Than Number of Tape Cells. *J. Comput. Syst. Sci.*, 22(3):384–406, 1981.

[23] N. Immerman. Expressibility as a complexity measure: results and directions. In *Proceedings of the 2nd IEEE Symposium on Structure in Complexity Theory*, pages 194–202, 1987.

[24] N. Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999.

[25] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In A. Selman, editor, *Complexity theory retrospective*, pages 59–81. Springer-Verlag, 1990.

[26] D. Kozen. *Design and Analysis of Algorithms*. Texts and Monographs in Computer Science. Springer, 1992.

[27] B. Laubner. Capturing Polynomial Time on Interval Graphs. In *Proceedings of the 25th IEEE Symposium on Logic in Computer Science*, pages 199–208, 2010.

[28] B. Laubner. *The structure of graphs and new logics for the characterization of Polynomial Time*. PhD thesis, Humboldt-Universität Berlin, 2011.

[29] L. Libkin. *Elements of Finite Model Theory*. Springer-Verlag, 2004.

[30] P. Malkin. Sherali–Adams relaxations of graph isomorphism polytopes. *Discrete Optimization*, 12:73–97, 2014.

[31] R. O'Donnell, J. Wright, C. Wu, and Y. Zhou. Hardness of Robust Graph Isomorphism, Lasserre Gaps, and Asymmetry of Random Graphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1659–1677, 2014.

[32] M. Otto. *Bounded Variable Logics and Counting*. Springer, 1997.

[33] C. Pearcy. A complete set of unitary invariants for operators generating finite w^*-algebras of type I. *Pacific J. Math.*, 12(4):1405–1416, 1962.

[34] H. Shapiro. A survey of canonical forms and invariants for unitary similarity. *Linear Algebra and its Applications*, 147:101 – 167, 1991.

[35] W. Specht. Zur Theorie der Matrizen. II. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 50:19–23, 1940.

[36] G. Tinhofer. A note on compact graphs. *Discrete Applied Mathematics*, 30:253–264, 1991.

[37] N. A. Wiegmann. Necessary and sufficient conditions for unitary similarity. *Journal of the Australian Mathematical Society*, 2(1):122–126, 1961.