On the Expressive Power of Linear Algebra on Graphs

Floris Geerts
University of Antwerp, Antwerp, Belgium
http://adrem.uantwerpen.be/floris.geerts
floris.geerts@uantwerpen.be

Abstract

Most graph query languages are rooted in logic. By contrast, in this paper we consider graph query languages rooted in linear algebra. More specifically, we consider MATLANG, a matrix query language recently introduced, in which some basic linear algebra functionality is supported. We investigate the problem of characterising equivalence of graphs, represented by their adjacency matrices, for various fragments of MATLANG. A complete picture is painted of the impact of the linear algebra operations in MATLANG on their ability to distinguish graphs.

2012 ACM Subject Classification Information systems → Query languages; Mathematics of computing → Graph theory; Theory of computation → Logic

Keywords and phrases matrix query languages, graph queries, graph theory, logics

Digital Object Identifier 10.4230/LIPIcs.ICDT.2019.7

Related Version Full version: https://arxiv.org/abs/1812.04379.

1 Introduction

Motivated by the importance of linear algebra for machine learning on big data [7, 8, 12, 46, 51] there is a current interest in languages that combine matrix operations with relational query languages in database systems [22, 35, 40, 41, 43]. Such hybrid languages raise many interesting questions from a database theoretical point of view. It seems natural, however, to first consider query languages for matrices alone. These are the focus of this paper.

More precisely, we continue the investigation of the expressive power of the matrix query language MATLANG, recently introduced as an analog for matrices of the relational algebra on relations [9]. Intuitively, queries in MATLANG are built-up by composing several linear algebra operations. The language MATLANG was shown to be subsumed by aggregate logic with only three non-numerical variables. Conversely, MATLANG can express all queries from graph databases to binary relations that can be expressed in first-order logic with three variables. The four-variable query asking if a graph contains a four-clique, however, is not expressible [9].

In this paper, we further zoom in on the expressive power of MATLANG on graphs. In particular, we investigate when two graphs are *equivalent* relative to some fragment of MATLANG. These fragments are defined by allowing only certain linear algebra operations in the queries and are denoted by $\mathsf{ML}(\mathcal{L})$, with $\mathcal{L}$ the list of allowed operations. A total of six (sensible) fragments are considered and $\mathsf{ML}(\mathcal{L})$-equivalence of graphs, i.e., their agreement on all sentences in $\mathsf{ML}(\mathcal{L})$, is characterised. Our results are as follows.

- For starters, we have the fragment $\mathsf{ML}(\cdot, \mathsf{tr})$ that allows for matrix multiplication $(\cdot)$ and trace (tr) computation (i.e., taking the sum of diagonal elements of a matrix). Equivalence of graphs relative to $\mathsf{ML}(\cdot, \mathsf{tr})$ coincides with being co-spectral, or equivalently, to having the same number of closed walks of any length (Section 5).

22nd International Conference on Database Theory (ICDT 2019).
Editors: Pablo Barcelo and Marco Calautti; Article No. 7; pp. 7:1–7:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- Another fragment, $\mathsf{ML}(\cdot, ^*, \mathbb{1})$, allows for matrix multiplication, conjugate transposition (*) and the introduction of the vector $\mathbb{1}$, consisting of all ones. Here, equivalence coincides with having the same number of (not necessarily closed) walks of any length (Section 6).
- When allowing both tr and $\mathbb{1}$, equivalence relative to $\mathsf{ML}(\cdot, \mathsf{tr}, \mathbb{1})$ coincides, not surprisingly, to having the same number of closed and non-closed walks of any length (Section 6).
- More interesting is the fragment $\mathsf{ML}(\cdot, ^*, \mathbb{1}, \mathsf{diag})$, which also allows for the operation $\mathsf{diag}(\cdot)$ that turns a vector into a diagonal matrix with that vector on its diagonal. In this case, equivalence coincides with having a so-called common equitable partition, or equivalently, to C^2-equivalence. Here, C^2 denotes the two-variable fragment of C, the extension of first-order logic with counting (Section 7).
- The combination of tr with diag results in a stronger notion of equivalence: Graphs are equivalent relative to $\mathsf{ML}(\cdot, \mathsf{tr}, \mathbb{1}, \mathsf{diag})$ when they are C^2-equivalent *and* co-spectral (Section 7).
- Finally, equivalence relative to $\mathsf{MATLANG}$ is shown to correspond to C^3-equivalence, the three-variable fragment of C (Section 8). This is in agreement with the results from Brijder et al. [9] mentioned earlier.

We remark that each of these fragments can be extended with addition and scalar multiplication at no increase in distinguishing power. We exhibit examples *separating* all fragments.

The characterisations are shown in a purely algebraic way, without relying on simulations in logic. Underlying are reductions of $\mathsf{ML}(\mathcal{L})$-equivalence of graphs to *similarity notions* of their adjacency matrices. For example, it is known that two graphs G and H are C^2-equivalent if and only if they are fractionally isomorphic [48, 53, 54]. This means that the adjacency matrices A_G of G and A_H of H satisfy $A_G \cdot S = S \cdot A_H$ for some doubly stochastic matrix S. As another example, C^3-equivalence of graphs corresponds to $A_G \cdot O = O \cdot A_H$ for some orthogonal matrix O that is also an isomorphism between the cellular algebras of G and H [20]. We provide similar characterisations for all our matrix query language fragments. It is worth pointing out that beyond $\mathsf{MATLANG}$, C^k-equivalence, for $k \geq 4$, can also be characterised in terms of solutions to linear problems [3, 29, 44].

Moreover, whenever possible, we also provide characterisations in terms of *spectral properties* of graphs. A wealth of results exists in spectral graph theory on what information can be obtained from the adjacency matrix, or from other matrices like the Laplacian, of a graph [10, 16, 26]. We rely quite a bit on known results in that area. Nevertheless, we believe that the connections made in this paper are of interest in their own right. They relate combinatorial and spectral graph invariants by means of query languages. We refer to work by Fürer [24, 25] for more examples of the power of graph invariants and to Dawar et al. [20] for connections between logic, combinatorial and spectral invariants.

Although links to logics such as C^2 and C^3 are made, the connection between $\mathsf{MATLANG}$, rank logics and fixed-point logics with counting, as studied in the context of the descriptive complexity of linear algebra [18, 17, 19, 27, 30, 34], is yet to be explored. Similarly for connections to logic-based graph query languages [2, 5].

2 Background

We denote the set of real numbers by $\mathbb{R}$ and the set of complex numbers by $\mathbb{C}$. The set of $m \times n$-matrices over the real (resp., complex) numbers is denoted by $\mathbb{R}^{m \times n}$ (resp., $\mathbb{C}^{m \times n}$). Vectors are elements of $\mathbb{R}^{m \times 1}$ (or $\mathbb{C}^{m \times 1}$). The entries of an $m \times n$-matrix A are denoted by A_{ij}, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. The entries of a vector v are denoted by v_i, for $i = 1, \ldots, m$. We often identify $\mathbb{R}^{1 \times 1}$ with $\mathbb{R}$, and $\mathbb{C}^{1 \times 1}$ with $\mathbb{C}$. The following classes of

matrices are of interest in this paper: square matrices (elements in $\mathbb{R}^{n\times n}$ or $\mathbb{C}^{n\times n}$), symmetric matrices (such that $A_{ij} = A_{ji}$ for all i and j), *doubly stochastic* matrices ($A_{ij} \in \mathbb{R}$, $A_{ij} \geq 0$, $\sum_{j=1}^{n} A_{ij} = 1$ and $\sum_{i=1}^{m} A_{ij} = 1$ for all i and j), *doubly quasi-stochastic* matrices ($A_{ij} \in \mathbb{R}$, $\sum_{j=1}^{n} A_{ij} = 1$ and $\sum_{i=1}^{m} A_{ij} = 1$ for all i and j), and *orthogonal* matrices ($O \in \mathbb{R}^{n\times n}$, $O^{\mathsf{t}} \cdot O = I = O \cdot O^{\mathsf{t}}$, where O^{t} denotes the transpose of O obtained by switching rows and columns, $\cdot$ denotes matrix multiplication, and I is the identity matrix in $\mathbb{R}^{n\times n}$).

We only need a couple of notions of linear algebra. We refer to the textbook by Axler [4] for more background. An *eigenvalue* of a matrix A is a scalar λ in $\mathbb{C}$ for which there is a non-zero vector v satisfying $A \cdot v = \lambda v$. Such a vector is called an *eigenvector* of A for eigenvalue λ. The *eigenspace* of an eigenvalue is the vector space obtained as the span of a maximal set of linear independent eigenvectors for this eigenvalue. Here, the *span* of a set of vectors just refers to the set of all linear combinations of vectors in that set. A set of vectors is linear independent if no vector in that set can be written as a linear combination of other vectors. The *dimension* of an eigenspace is the minimal number of eigenvectors that span the eigenspace.

We will only consider undirected graphs without self-loops. Let $G = (V, E)$ be such a graph with vertices $V = \{1, \ldots, n\}$ and unordered edges $E \subseteq \{\{i, j\} \mid i, j \in V\}$. The *order* of G is simply the number of vertices. Then, the *adjacency matrix* of a graph G of order n, denoted by A_G, is an $n \times n$-matrix whose entries $(A_G)_{ij}$ are set to 1 if and only if $\{i, j\} \in E$, all other entries are set to 0. The matrix A_G is a symmetric real matrix with zeroes on its diagonal. The *spectrum* of an undirected graph can be represented as $\mathsf{spec}(G) = \begin{pmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_p \\ m_1 & m_2 & \cdots & m_p \end{pmatrix}$, where $\lambda_1 < \lambda_2 < \cdots < \lambda_p$ are the distinct real eigenvalues of the adjacency matrix A_G of G, and where $m_1, m_2, \ldots, m_p$ denote the dimensions of the corresponding eigenspaces. Two graphs are said to be *co-spectral* if they have the same spectrum. We introduce other relevant notions throughout the paper.

3 Matrix Query Languages

As described in Brijder et al. [9], matrix query languages can be formalised as compositions of linear algebra operations. Intuitively, a linear algebra operation takes a number of matrices as input and returns another matrix. Examples of operations are matrix multiplication, conjugate transposition, computing the trace, just to name a few. By closing such operations under composition "matrix query languages" are formed. More specifically, for linear algebra operations $\mathsf{op}_1, \ldots, \mathsf{op}_k$ the corresponding matrix query language is denoted by $\mathsf{ML}(\mathsf{op}_1, \ldots, \mathsf{op}_k)$ and consists of expressions formed by the following grammar:

$$e := X \mid \mathsf{op}_1\big(e_1, \ldots, e_{p_1}\big) \mid \cdots \mid \mathsf{op}_k\big(e_1, \ldots, e_{p_k}\big),$$

where X denotes a *matrix variable* which serves to indicate the input to expressions and p_i denotes the number of inputs required by operation op_i. We focus on the case when only a single matrix variable X is present. The treatment of multiple variables is left for future work.

The semantics of an expression $e(X)$ in $\mathsf{ML}(\mathsf{op}_1, \ldots, \mathsf{op}_k)$ is defined inductively, relative to an *assignment* ν of X to a matrix $\nu(X) \in \mathbb{C}^{m\times n}$, for some dimensions m and n. We denote by $e\big(\nu(X)\big)$ the result of evaluating $e(X)$ on $\nu(X)$. As expected, we define $\mathsf{op}_i(e_1(X), \ldots, e_{p_i}(X))(\nu(X)) := \mathsf{op}_i\big(e_1(\nu(X)), \ldots, e_{p_i}(\nu(X))\big)$ for linear algebra operation op_i. In Table 1 we list the operations constituting the basis matrix query language $\mathsf{MATLANG}$, introduced in Brijder et al. [9]. In the table we also show their semantics. We note that

Table 1 Linear algebra operations (supported in MATLANG [9]) and their semantics. In the first operation, for $A_{ji} \in \mathbb{C}$, A^*_{ji} denotes complex conjugation. In the last operation, $\Omega = \bigcup_{k>0} \Omega_k$, where Ω_k consists of functions $f : \mathbb{C}^k \to \mathbb{C}$.

conjugate transposition ($\mathsf{op}(e) = e^*$)		
$e(\nu(X)) = A \in \mathbb{C}^{m\times n}$	$e(\nu(X))^* = A^* \in \mathbb{C}^{n\times m}$	$(A^*)_{ij} = A^*_{ji}$
one-vector ($\mathsf{op}(e) = \mathbb{1}(e)$)		
$e(\nu(X)) = A \in \mathbb{C}^{m\times n}$	$\mathbb{1}(e(\nu(X)) = \mathbb{1} \in \mathbb{C}^{m\times 1}$	$\mathbb{1}_i = 1$
diagonalization of a vector ($\mathsf{op}(e) = \mathsf{diag}(e)$)		
$e(\nu(X)) = A \in \mathbb{C}^{m\times 1}$	$\mathsf{diag}(e(\nu(X)) = \mathsf{diag}(A) \in \mathbb{C}^{m\times m}$	$\mathsf{diag}(A)_{ii} = A_i$, $\mathsf{diag}(A)_{ij} = 0,\ i \neq j$
matrix multiplication ($\mathsf{op}(e_1, e_2) = e_1 \cdot e_2$)		
$e_1(\nu(X)) = A \in \mathbb{C}^{m\times n}$, $e_2(\nu(X)) = B \in \mathbb{C}^{m\times o}$	$e_1(\nu(X)) \cdot e_2(\nu(X)) = C \in \mathbb{C}^{m\times o}$	$C_{ij} = \sum_{k=1}^{n} A_{ik} \times B_{kj}$
matrix addition ($\mathsf{op}(e_1, e_2) = e_1 + e_2$)		
$e_i(\nu(X)) = A^{(i)} \in \mathbb{C}^{m\times n}$	$e_1(\nu)(X) + e_2(\nu(X)) = B \in \mathbb{C}^{m\times n}$	$B_{ij} = A^{(1)}_{ij} + A^{(2)}_{ij}$
scalar multiplication ($\mathsf{op}(e) = c \times e,\ c \in \mathbb{C}$)		
$e(\nu(X)) = A \in \mathbb{C}^{m\times n}$	$c \times e(\nu(X)) = B \in \mathbb{C}^{m\times n}$	$B_{ij} = c \times A_{ij}$
trace ($\mathsf{op}(e) = \mathsf{tr}(e)$)		
$e(\nu(X)) = A \in \mathbb{C}^{m\times m}$	$\mathsf{tr}(e(\nu(X)) = c \in \mathbb{C}$	$c = \sum_{i=1}^{m} A_{ii}$
$e(\nu(X)) = A \in \mathbb{C}^{m\times 1}$	$\mathsf{tr}(e(\nu(X)) = c \in \mathbb{C}$	$c = \sum_{i=1}^{m} A_i$
pointwise function application ($\mathsf{op}(e_1, \ldots, e_p) = \mathsf{apply}[f](e_1, \ldots, e_p)$), $f : \mathbb{C}^p \to \mathbb{C} \in \Omega$		
$e_i(\nu(X)) = A^{(i)} \in \mathbb{C}^{m\times n}$	$\mathsf{apply}[f]\big(e_1(\nu(X)), \ldots, e_p(\nu(X))\big) = B \in \mathbb{C}^{m\times n}$	$B_{ij} = f(A^{(1)}_{ij}, \ldots, A^{(p)}_{ij})$

restrictions on the dimensions are in place to ensure that operations are well-defined. Using a simple type system one can formalise a notion of well-formed expressions which guarantees that the semantics of such expressions is well-defined [9]. We only consider well-formed expressions from here on.

▶ Remark 3.1. The list of operations in Table 1 differs slightly from the list presented in Brijder et al. [9]: We explicitly mention scalar multiplication ($\times$) and addition ($+$), and the trace operation (tr), all of which can be expressed in MATLANG. Hence, MATLANG and $\mathsf{ML}(\cdot, ^*, \mathsf{tr}, \mathbb{1}, \mathsf{diag}, +, \times, \mathsf{apply}[f], f \in \Omega)$ are equivalent.

4 Expressive Power

As mentioned in the introduction, we are interested in the expressive power of matrix query languages. In this paper, we consider sentences in these languages. We define an expression $e(X)$ in $\mathsf{ML}(\mathsf{op}_1, \ldots, \mathsf{op}_k)$ to be a *sentence* if $e(\nu(X))$ returns a 1×1-matrix for any assignment ν of X. We note that the type system of MATLANG allows to check whether an expression in $\mathsf{ML}(\mathcal{L})$ is a sentence (see Brijder et al. [9] for more details). Having defined sentences, a notion of equivalence naturally follows.

▶ **Definition 4.1.** *Two matrices A and B in $\mathbb{C}^{m\times n}$ are said to be* $\mathsf{ML}(\mathsf{op}_1, \ldots, \mathsf{op}_k)$-equivalent, *denoted by $A \equiv_{\mathsf{ML}(\mathsf{op}_1,\ldots,\mathsf{op}_k)} B$, if and only if $e(A) = e(B)$ for all sentences $e(X)$ in* $\mathsf{ML}(\mathsf{op}_1, \ldots, \mathsf{op}_k)$.

In other words, equivalent matrices cannot be distinguished by sentences in the matrix query language under consideration. Equivalence with regards to sentences resembles the standard notion of equivalence used in logic.

We aim to *characterise* equivalence for various matrix query languages. We will, however, not treat this problem in full generality and instead, to gain intuition, start by considering *adjacency matrices of undirected graphs*. The corresponding notion of equivalence on graphs is defined, as expected:

▶ **Definition 4.2.** *Two graphs G and H of the same order are said to be* $\mathsf{ML}(\mathsf{op}_1, \ldots, \mathsf{op}_k)$-equivalent*, denoted by $G \equiv_{\mathsf{ML}(\mathsf{op}_1,\ldots,\mathsf{op}_k)} H$, if and only if their adjacency matrices are* $\mathsf{ML}(\mathsf{op}_1, \ldots, \mathsf{op}_k)$*-equivalent.*

▶ Remark 4.3. One could imagine defining equivalence with regards to arbitrary expressions, i.e., expressions in MATLANG that are not necessarily sentences. Such a notion would be too strong, however. Indeed, requiring that $e(A_G) = e(A_H)$ for arbitrary expressions $e(X)$ would imply that $A_G = A_H$ and hence $G = H$.

In the following sections we consider graph equivalence for various fragments, starting from simple fragments only supporting a couple of operations, up to the full MATLANG matrix query language.

5 Expressive Power of the Matrix Query Language $\mathsf{ML}(\cdot, \mathsf{tr})$

The smallest fragment we consider is $\mathsf{ML}(\cdot, \mathsf{tr})$. This is a very restrictive fragment since the only sentences that one can express are of the form (i) $\#\mathsf{cwalk}_k(X) := \mathsf{tr}(X^k)$, where X^k stands for the kth power of X, i.e., X multiplied k times with itself, and (ii) products of such sentences. We note that, when evaluated on an adjacency matrix A_G, $\#\mathsf{cwalk}_k(A_G)$ counts the number of *closed walks of length k* in G.

Indeed, the entries of the powers A_G^k of adjacency matrix A_G are known to correspond to the number of walks of length k in G. Recall that a *walk of length k* in a graph $G = (V, E)$ is a sequence $(v_0, v_1, \ldots, v_k)$ of vertices of G such that consecutive vertices are adjacent in G, i.e., $\{v_{i-1}, v_i\} \in E$ for all $i = 1, \ldots, k$. Furthermore, a *closed walk* is a walk that starts in and ends at the same vertex. Hence, $\#\mathsf{cwalk}_k(A_G) = \sum_i (A_G^k)_{ii}$ indeed counts closed walks of length k in G. Closed walks of length 0 correspond, as usual, to vertices in G.

The following characterisations are known to hold.

▶ **Proposition 5.1** ([10, 16])**.** *Let G and H be two graphs of the same order. The following are equivalent:*

- *G and H have the same total number of closed walks of length k, for all $k \geq 0$,*
- $\mathsf{tr}(A_G^k) = \mathsf{tr}(A_H^k)$ *for all $k \geq 0$,*
- *G and H are co-spectral, and*
- *there exists a real orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$.*

▶ **Example 5.2.** The graphs G_1 ([illegible]) and H_1 ([illegible]) are the smallest pair (in terms of number of vertices) of non-isomorphic co-spectral graphs of the same order [13]. Note that the isolated vertex in G_1 ensures that G_1 and H_1 have the same number of vertices (and thus the same number of closed walks of length 0).
A characterisation of $\mathsf{ML}(\cdot, \mathsf{tr})$-equivalence now easily follows.

▶ **Proposition 5.3.** *For two graphs G and H of the same order, $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr})} H$ if and only if there exists a real orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$ if and only if G and H have the same number of closed walks of any length.*

Proof. By definition, if $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr})} H$, then $e(A_G) = e(A_H)$ for any sentence $e(X)$ in $\mathsf{ML}(\cdot, \mathsf{tr})$. This holds in particular for the sentences $\#\mathsf{cwalk}_k(X) := \mathsf{tr}(X^k)$ in $\mathsf{ML}(\cdot, \mathsf{tr})$, for

$k \geq 1$. That is, $G \equiv_{\mathsf{ML}(\,\cdot\,,\,\mathsf{tr})} H$ implies that $\mathsf{tr}(A_G^k) = \mathsf{tr}(A_H^k)$ for all $k \geq 1$. Since G and H are of the same order and $A_G^0 = A_H^0 = I$ (by convention), $\mathsf{tr}(A_G^0) = \mathsf{tr}(A_H^0) = \mathsf{tr}(I) = n$. From the previous proposition it then follows that there exists an orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$.

For the converse, assume that $A_G \cdot O = O \cdot A_H$ for some orthogonal matrix O. We already remarked that sentences in $\mathsf{ML}(\,\cdot\,,\mathsf{tr})$ are products of sentences of the form $\#\mathsf{cwalk}_k(X) := \mathsf{tr}(X^k)$. It now suffices to observe that $\mathsf{tr}(P \cdot A \cdot P^{-1}) = \mathsf{tr}(A)$ for any matrix A and any invertible matrix P. In particular, $\mathsf{tr}(A_G^k) = \mathsf{tr}(O^\mathsf{t} \cdot A_G^k \cdot O) = \mathsf{tr}(O^\mathsf{t} \cdot O \cdot A_H^k) = \mathsf{tr}(A_H^k)$. ◀

From an expressiveness point of view, it tells that $\mathsf{ML}(\,\cdot\,,\mathsf{tr})$-equivalence of two graphs implies that their adjacency matrices share the same rank, characteristic polynomial, determinant, eigenvalues, and their algebraic multiplicities, geometric multiplicities of eigenvalues, just to name a few.

Given that the trace operation is a linear mapping, i.e., $\mathsf{tr}(cA + dB) = c\mathsf{tr}(A) + d\mathsf{tr}(B)$ for matrices A and B and complex numbers c and d, one would expect that matrix addition $(+)$ and scalar multiplication $(\times)$ can be added to $\mathsf{ML}(\,\cdot\,,\mathsf{tr})$ without an increase in expressiveness. Indeed, one can rewrite sentences in $\mathsf{ML}(\,\cdot\,,\mathsf{tr},+,\times)$ as a linear combination of sentences in $\mathsf{ML}(\,\cdot\,,\mathsf{tr})$. Combined with the linearity of $\mathsf{tr}(\,\cdot\,)$, Proposition 5.3 can be extended as follows.

▶ **Corollary 5.4.** *For two graphs G and H of the same order, we have that $G \equiv_{\mathsf{ML}(\,\cdot\,,\,\mathsf{tr})} H$ if and only if $G \equiv_{\mathsf{ML}(\,\cdot\,,\,\mathsf{tr},+,\times)} H$.*

We can further strengthen Corollary 5.4 by allowing the application of *any* function $f : \mathbb{C}^p \to \mathbb{C}$ in Ω, *provided* that $\mathsf{apply}[f](e_1, \ldots, e_p)$ is only allowed when each e_i is a sentence. That is, we only allow pointwise function applications on "scalars". The restriction of such function applications is denoted by $\mathsf{apply_s}[f]$, for $f \in \Omega$. Indeed, $G \equiv_{\mathsf{ML}(\,\cdot\,,\,\mathsf{tr},+,\times)} H$ implies that $e(A_G) = e(A_H)$ for any sentence $e(X)$ in $\mathsf{ML}(\,\cdot\,,\mathsf{tr},+,\times)$. Clearly, when $e_i(A_G) = e_i(A_H)$ for all $i = 1, \ldots, p$, $\mathsf{apply_s}[f](e_1(A_G), \ldots, e_p(A_G)) = \mathsf{apply_s}[f](e_1(A_H), \ldots, e_p(A_H))$.

▶ **Corollary 5.5.** *For two graphs G and H of the same order, $G \equiv_{\mathsf{ML}(\,\cdot\,,\,\mathsf{tr},+,\times)} H$ if and only if $G \equiv_{\mathsf{ML}(\,\cdot\,,\,\mathsf{tr},+,\times,\mathsf{apply_s}[f],f\in\Omega)} H$.*

Finally, we can also add conjugate transposition $(^*)$ without increasing the expressive power, *provided* that we mildly restrict the class Ω of pointwise functions. More precisely, we assume that Ω is *closed under complex conjugation* in the sense that for every $f \in \Omega$ also the composition * and f is in Ω. This assumption, together with standard properties of complex conjugation and conjugate transposition (in particular, $(A \cdot B)^* = B^* \cdot A^*$, $(A^*)^* = A$ and linearity) and using the fact that adjacency matrices of undirected graphs are symmetric, allows one to rewrite expressions in $\mathsf{ML}(\,\cdot\,,{}^*,\mathsf{tr},+,\times,\mathsf{apply_s}[f], f \in \Omega)$ such that * is only applied on scalars. As a consequence, any expression in $\mathsf{ML}(\,\cdot\,,{}^*,\mathsf{tr},+,\times,\mathsf{apply_s}[f], f \in \Omega)$ is equivalent to an expression in $\mathsf{ML}(\,\cdot\,,\mathsf{tr},+,\times,\mathsf{apply_s}[f], f \in \Omega)$.

▶ **Corollary 5.6.** *Let Ω be a class of pointwise functions that is closed under complex conjugation. Then, for two graphs G and H of the same order, $G \equiv_{\mathsf{ML}(\,\cdot\,,\,\mathsf{tr},+,\times,\mathsf{apply_s}[f],f\in\Omega)} H$ if and only if $G \equiv_{\mathsf{ML}(\,\cdot\,,\,{}^*,\,\mathsf{tr},+,\times,\mathsf{apply_s}[f],f\in\Omega)} H$.*

As a consequence, the graphs G_1 (⊡) and H_1 (⋇) from Example 5.2 cannot be distinguished by sentences in $\mathsf{ML}(\,\cdot\,,{}^*,\mathsf{tr},+,\times,\mathsf{apply_s}[f], f \in \Omega)$.

As we will see later, including any other operation from Table 1, such as $\mathbb{1}(\,\cdot\,)$, $\mathsf{diag}(\,\cdot\,)$ or pointwise function applications on vector or matrices, requires additional constraints on the orthogonal matrix O linking A_G with A_H.

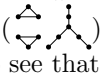
▶ Remark 5.7. Corollaries 5.4, 5.5 and 5.6 hold for *any* fragment that we will consider.

6 The Impact of the $\mathbb{1}(\cdot)$ Operation

The $\mathbb{1}(\cdot)$ operation, which returns the all-ones vector $\mathbb{1}$[1], allows to extract other information from graphs than just the number of closed walks. Indeed, consider the sentences

$$\#\mathsf{walk}_k(X) := (\mathbb{1}(X))^* \cdot X^k \cdot \mathbb{1}(X) \text{ and } \#\mathsf{walk}'_k(X) := \mathsf{tr}(X^k \cdot \mathbb{1}(X)),$$

in $\mathsf{ML}(\cdot, {}^*, \mathbb{1})$ and $\mathsf{ML}(\cdot, \mathsf{tr}, \mathbb{1})$, respectively. When applied on adjacency matrix A_G of a graph G, $\#\mathsf{walk}_k(A_G)$ (and also $\#\mathsf{walk}'_k(A_G)$) returns the *number of (not necessarily closed) walks* in G of length k. In relation to the previous section, co-spectral graphs do not necessarily have the same number of walks of any length. Similarly, graphs with the same number of walks of any length are not necessarily co-spectral.

▶ **Example 6.1.** It can be verified that the co-spectral graphs G_1 () and H_1 () of Example 5.2 have 16 versus 20 walks of length 2, respectively. As a consequence, $\mathsf{ML}(\cdot, {}^*, \mathbb{1})$ and $\mathsf{ML}(\cdot, \mathsf{tr}, \mathbb{1})$ can distinguish G_1 from H_1 by means of the sentences $\#\mathsf{walk}_2(X)$ and $\#\mathsf{walk}'_2(X)$, respectively. By contrast, the graphs G_2 () and H_2 () are not co-spectral, yet have the same number of walks of any length. It is easy to see that G_2 and H_2 are not co-spectral (apart from verifying that their spectra are different): H_2 has 12 closed walks of length 3 (because of the triangles), whereas G_2 has none. We argue below why they have the same number of walks. As a consequence, $\mathsf{ML}(\cdot, \mathsf{tr})$ (and thus also $\mathsf{ML}(\cdot, \mathsf{tr}, \mathbb{1})$) can distinguish G_2 and H_2. It follows from Proposition 6.6 below that these graphs cannot be distinguished by $\mathsf{ML}(\cdot, {}^*, \mathbb{1})$.

Graphs sharing the same number of walks of any length have been investigated before in spectral graph theory [14, 15, 32, 49]. To state a spectral characterisation, the so-called *main spectrum* of a graph needs to be considered. The main spectrum of a graph is the set of eigenvalues whose eigenspace is not orthogonal to the $\mathbb{1}$ vector. More formally, for an eigenvalue λ and corresponding eigenspace, represented by a matrix V whose columns are eigenvectors of λ that span its eigenspace, the *main angle* β_λ of λ's eigenspace is $\frac{1}{\sqrt{n}}\|V^\mathsf{t} \cdot \mathbb{1}\|_2$, where $\|\cdot\|_2$ is the Euclidean norm. Hence, main eigenvalues are those with a non-zero main angle. Two graphs are said to be *co-main* if they have the same set of main eigenvalues and corresponding main angles. Intuitively, the importance of the orthogonal projection on $\mathbb{1}$ stems from the observation that $\#\mathsf{walk}_k(A_G)$ can be expressed as $\sum_i \lambda_i^k \beta_{\lambda_i}^2$ where the λ_i's are eigenvalues of A_G. Clearly, only those eigenvalues λ_i for which $\beta_{\lambda_i} > 0$ matter when computing $\#\mathsf{walk}_k(A_G)$. This results in the following characterisation.

▶ **Proposition 6.2** (Theorem 1.3.5 in Cvetković et al. [16]). *Two graphs G and H of the same order are co-main if and only if they have the same total number of walks of length k, for every $k \geq 0$.*

Furthermore, the following proposition follows implicitly from the proof of Theorem 3 in van Dam et al. [56] (and is also shown in Theorem 1.2 in Dell et al. [21] in the context of distinguishing graphs by means of *homomorphism vectors*).

▶ **Proposition 6.3.** *Two graphs G and H of the same order have the same total number of walks of length k, for every $k \geq 0$, if and only if there is a doubly quasi-stochastic matrix Q such that $A_G \cdot Q = Q \cdot A_H$, i.e., such that $A_G \cdot Q = Q \cdot A_H$, $Q \cdot \mathbb{1} = \mathbb{1}$ and $Q^\mathsf{t} \cdot \mathbb{1} = \mathbb{1}$ hold.*

[1] We use $\mathbb{1}$ to denote the all ones *vector* (of appropriate dimension) and use $\mathbb{1}(\cdot)$ (with brackets) for the corresponding all ones *operator*.

▶ **Example 6.4** (Continuation of Example 6.1). Consider the subgraph G_3 ( ) of G_2 and the subgraph H_3 ( ) of H_2. We have that $A_{G_3} \cdot Q = Q \cdot A_{H_3}$ for

$$A_{G_3} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, A_{H_3} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, Q = \begin{bmatrix} 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \end{bmatrix},$$

and hence by Proposition 6.3, G_3 and H_3 have the same number of walks on any length.

As it turns out, the values of the sentences $\#\mathsf{walk}_k(A_G)$ mentioned earlier, that count the number of walks of length k in G, fully determine the value of any sentence in $\mathsf{ML}(\cdot, {}^*, \mathbb{1})$.

▶ **Lemma 6.5.** *Let G and H be two graphs of the same order. Then, $G \equiv_{\mathsf{ML}(\cdot,{}^*,\mathbb{1})} H$ if and only if $\#\mathsf{walk}_k(A_G) = \#\mathsf{walk}_k(A_H)$ for all $k \geq 1$.*

The proof involves an analysis of expressions in $\mathsf{ML}(\cdot, {}^*, \mathbb{1})$. We may thus conclude from Proposition 6.3 and Lemma 6.5 that:

▶ **Proposition 6.6.** *For two graphs G and H of the same order, $G \equiv_{\mathsf{ML}(\cdot,{}^*,\mathbb{1})} H$ if and only if there exists a doubly quasi-stochastic matrix Q such that $A_G \cdot Q = Q \cdot A_H$ if and only if G and H have the same number of walks of any length.*

When it comes to $\mathsf{ML}(\cdot, \mathsf{tr}, \mathbb{1})$, we know from Propositions 5.1 and 5.3 that $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1})} H$ implies that G and H are co-spectral. Combined with Proposition 6.2 and the fact that the sentence $\#\mathsf{walk}'_k(X)$ counts the number of walks of length k, we have that $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1})} H$ implies that G and H are co-spectral and co-main. The following is known about such graphs.

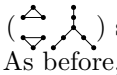
▶ **Proposition 6.7** ([38, 56]). *Two co-spectral graphs G and H of the same order are co-main if and only if there exists an orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$ and $O \cdot \mathbb{1} = \mathbb{1}$.*

In other words, $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1})} H$ implies the existence of an orthogonal matrix O such that $O \cdot \mathbb{1} = \mathbb{1}$ (i.e., O is also doubly quasi-stochastic) and $A_G \cdot O = O \cdot A_H$. An analysis of expressions in $\mathsf{ML}(\cdot, \mathsf{tr}, \mathbb{1})$ shows that the converse also holds.

▶ **Proposition 6.8.** *For two graphs G and H of the same order, $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1})} H$ if and only if there exists an orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$ and $O \cdot \mathbb{1} = \mathbb{1}$ if and only if G and H have the same number of closed walks and the same number of walks of any length.*

An alternative characterisation (also in van Dam et al. [56]) is that G and H are co-spectral and co-main if and only if both G and H *and* their complement graphs $\bar{G}$ and $\bar{H}$ are co-spectral. Here, the complement graph $\bar{G}$ of G is the graph with adjacency matrix given by $J - A_G - I$, where J is the all ones matrix, and similarly for $\bar{H}$.

▶ **Example 6.9** (Continuation of Example 6.1). Consider the subgraph G_4 ( ) of G_2 and the subgraph H_4 ( ) of H_2. These are known to be the smallest non-isomorphic co-spectral graphs with co-spectral complements [31]. From Proposition 6.8 it then follows that G_4 and H_4 have the same number of walks of any length. Combined with our earlier observation in Example 6.4 that also G_3 and H_3 have this property, we may conclude that $G_2 = G_3 \cup G_4$ ( ) and $H_2 = H_3 \cup H_4$ ( ) have the same number of walks of any length, as anticipated in Example 6.1.

We remark that as a consequence of Propositions 6.6 and 6.8, $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1})} H$ implies that $G \equiv_{\mathsf{ML}(\cdot,{}^*,\mathbb{1})} H$. We already mentioned in Example 6.1 that the graphs G_2 () and H_2 () show that the converse does not hold.

As before, we observe that addition, scalar multiplication, conjugate transposition and pointwise function application on scalars can be included at no increase in expressiveness.

▶ **Corollary 6.10.** *Let G and H be two graphs of the same order. Then,*

- $G \equiv_{\mathsf{ML}(\cdot,{}^*,\mathbb{1},+,\times,\mathsf{apply_s}[f],f\in\Omega)} H$ *if and only if* $G \equiv_{\mathsf{ML}(\cdot,{}^*,\mathbb{1})} H$*, and*
- $G \equiv_{\mathsf{ML}(\cdot,{}^*,\mathsf{tr},\mathbb{1},+,\times,\mathsf{apply_s}[f],f\in\Omega)} H$ *if and only if* $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1})} H$*,*

where Ω is assumed to be closed under complex conjugation.

7 The Impact of the diag(·) Operation

We next consider the operation $\mathsf{diag}(\cdot)$ which takes a vector as input and returns a diagonal matrix with the input vector on its diagonal. The smallest fragments in which vectors (and sentences) can be defined are $\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1})$ and $\mathsf{ML}(\cdot,{}^*,\mathbb{1})$. Therefore, in this section we consider equivalence with regards to $\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1},\mathsf{diag})$ and $\mathsf{ML}(\cdot,{}^*,\mathbb{1},\mathsf{diag})$.

Using $\mathsf{diag}(\cdot)$ we can again extract new information from graphs.

▶ **Example 7.1.** Consider graphs G_4 () and H_4 (). In G_4 we have vertices of degrees 0 and 2, and in H_4 vertices of degrees 1, 2 and 3. We will count the number of vertices of degree 3. To this aim consider the sentence $\#\mathsf{3degr}(X)$ given by

$$\left(\frac{1}{6}\right) \times \mathbb{1}(X)^* \cdot \big(\mathsf{diag}(X \cdot \mathbb{1}(X)) \cdot \mathsf{diag}(X \cdot \mathbb{1}(X) - \mathbb{1}(X)) \cdot \mathsf{diag}(X \cdot \mathbb{1}(X) - 2 \times \mathbb{1}(X))\big) \cdot \mathbb{1}(X),$$

in which we, for convenience, allow addition and scalar multiplications. Each of the subexpressions $\mathsf{diag}(X \cdot \mathbb{1}(X) - d \times \mathbb{1}(X))$, for $d = 0, 1$ and 2, sets the diagonal entry corresponding to vertex v to 0 when v has degree d. By taking the product of these diagonal matrices, entries that are set to 0 will remain zero in the resulting diagonal matrix. This implies that the only non-zero diagonal entries are those corresponding to vertices of degree different from 0, 1 and 2. In other words, only for vertices of degree 3 the diagonal entries carry a non-zero value, i.e., value $3(3-1)(3-2)$. By appropriately rescaling by the factor $\frac{1}{6} = \frac{1}{3(3-1)(3-2)}$, the diagonal entries for the degree three vertices are set to 1, and then summed up. Hence, $\#\mathsf{3degr}(X)$ indeed counts the number vertices of degree three in G_4 and H_4. Since $\#\mathsf{3degr}(A_{G_4}) = [0] \neq [1] = \#\mathsf{3degr}(A_{H_4})$ we can distinguish these graphs.

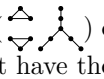
The use of the diagonal matrices and their products as in our example sentence $\#\mathsf{3degr}(X)$ can be generalised to obtain information about so-called *iterated degrees* of vertices in graphs, e.g., to identify and/or count vertices that have a number of neighbours each of which have neighbours of specific degrees. Such iterated degree information is closely related to *equitable partitions* of graphs (see e.g., Scheinerman et al. [50]). We phrase our results in terms of such partitions instead of iterated degree sequences.

7.1 Equitable Partitions

Formally, an *equitable partition* $\mathcal{V} = \{V_1, \ldots, V_\ell\}$ of G is partition of the vertex set of G such that for all $i, j = 1, \ldots, \ell$ and $v, v' \in V_i$, $\mathsf{deg}(v, V_j) = \mathsf{deg}(v', V_j)$. Here, $\mathsf{deg}(v, V_j)$ is the number of vertices in V_j that are adjacent to v. In other words, an equitable partition is such that the graph is regular within each part, and is bi-regular between any two different parts.

A graph always has a *trivial* equitable partition: simply treat each vertex as a part by its own. Most interesting is the *coarsest* equitable partition of a graph, i.e., the *unique* equitable partition for which any other equitable partition of the graph is a refinement thereof [50]. Two graphs G and H are said to have a *common* equitable partition if there exists an equitable partition $\mathcal{V} = \{V_1, \ldots, V_\ell\}$ of G and an equitable partition $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ of H such that (a) the sizes of the parts agree, i.e., $|V_i| = |W_i|$ for each $i = 1, \ldots, \ell$, and (b) $\mathsf{deg}(v, V_j) = \mathsf{deg}(w, W_j)$ for any $v \in V_i$ and $w \in W_i$ and any $i, j = 1, \ldots, \ell$. We note that, due to condition (b) the trivial partitions of graphs do not always result in a common equitable partition. In other words, not every two graphs (of the same order) have a common equitable partition. Proposition 7.2 below characterises when they do have a common partition. Equitable partitions naturally arise as the result of the *colour refinement procedure* [6, 28, 57], also known as the 1-dimensional Weisfeiler-Lehman algorithm, used as a subroutine in graph isomorphism solvers. Furthermore, there is a close connection to the study of *fractional isomorphisms* of graphs [50, 53], already mentioned in the introduction. We recall: two graphs G and H are said to be fractional isomorphic if there exists a doubly stochastic matrix S such that $A_G \cdot S = S \cdot A_H$. Furthermore, a logical characterisation of graphs with a common equitable partition exists.

▶ **Proposition 7.2** ([53], [36]). *Let G and H be two graphs of the same order. Then, G and H are fractional isomorphic if and only if G and H have a common equitable partition if and only if $G \equiv_{\mathsf{C}^2} H$.*

▶ **Example 7.3.** The matrix linking the adjacency matrices of G_3 () and H_3 () in Example 6.4 is in fact a doubly stochastic matrix (all its entries are either 0 or $\frac{1}{2}$). Hence, G_3 and H_3 have a common equitable partition, which in this case consists of a single part consisting of all vertices. This generally holds for graphs that are k-regular (meaning, each vertex is adjacent to k vertices) for the same k [48, 50]. By contrast, graphs G_2 () and H_2 () do not have a common equitable partition. Indeed, fractional isomorphic graphs must have the same degree sequence [50], which does not hold for G_2 and H_2. For the same reason, G_1 () and H_1 (), and G_4 () and H_4 () are not fractional isomorphic.

To related equitable partitions to $\mathsf{ML}(\cdot, \mathsf{tr}, \mathbb{1}, \mathsf{diag})$- and $\mathsf{ML}(\cdot, {}^*, \mathbb{1}, \mathsf{diag})$-equivalence, we show that the presence of $\mathsf{diag}(\cdot)$ allows to formulate a number of expressions, denoted by $\mathsf{eqpart}_i(X)$, for $i = 1, \ldots, \ell$, that together extract the *coarsest equitable partition* from a given graph.

In the following, $\mathcal{L}$ can be either $\{\cdot, \mathsf{tr}, \mathbb{1}, \mathsf{diag}\}$ or $\{\cdot, {}^*, \mathbb{1}, \mathsf{diag}\}$. Furthermore, we denote by $\mathcal{L}^+$ the extension of $\mathcal{L}$ with linear combinations (i.e., $+$ and $\times$), pointwise function applications on scalars (i.e., $\mathsf{apply}_\mathsf{s}[f]$, $f \in \Omega$) and conjugate transposition (*). The corresponding matrix query languages are denoted by $\mathsf{ML}(\mathcal{L})$ and $\mathsf{ML}(\mathcal{L}^+)$, respectively.

We start by reducing the problem of $\mathsf{ML}(\mathcal{L}^+)$-equivalence to $\mathsf{ML}(\mathcal{L})$-equivalence.

▶ **Lemma 7.4.** *Let G and H be two graphs of the same order. Then, $G \equiv_{\mathsf{ML}(\mathcal{L})} H$ if and only if $G \equiv_{\mathsf{ML}(\mathcal{L}^+)} H$.*

This lemma is verified by showing that expressions in $\mathsf{ML}(\mathcal{L}^+)$ can be seen as linear combinations of expressions in $\mathsf{ML}(\mathcal{L})$, in an analogous way as in the proof of Corollary 6.10. For example, it is clear that $\#\mathsf{3degr}(X)$ can be written as such a linear combination.

We next relate $G \equiv_{\mathsf{ML}(\mathcal{L}^+)} H$ and common equitable partitions of G and H.

▶ **Proposition 7.5.** *Let G and H be two graphs of the same order. Then, $G \equiv_{\mathsf{ML}(\mathcal{L}^+)} H$ implies that G and H have a common equitable partition.*

Proof. We show that the algorithm $\textsc{cgcr}(A_G)$, described in Kersting et al. [39], which computes the coarsest equitable partition of a graph can be simulated by expressions in $\mathsf{ML}(\mathcal{L}^+)$. To describe a partition $\mathcal{V} = \{V_1, \ldots, V_\ell\}$ of the vertex set of G we use *indicator vectors*. More precisely, we define $\mathbb{1}_{V_i}$ as the $n \times 1$-vector which has a "1" for those entries corresponding to vertices in V_i and has all its other entries set to "0". It is clear that we can also recover partitions from indicator vectors. The simulation of $\textsc{cgcr}(A_G)$ results in a number of expressions, denoted by $\mathsf{eqpart}_i(X)$ for $i = 1, \ldots, \ell$, in $\mathsf{ML}(\mathcal{L}^+)$ that *depend on* G and such that the set $\{\mathsf{eqpart}_i(A_G)\}$ consists of indicator vectors of the coarsest equitable partition of G. Since the algorithm $\textsc{cgcr}(A_G)$ is phrased in linear algebra terms [39], its simulation follows easily. Underlying this simulation is the use of products of diagonal matrices as a means of taking conjunctions of indicator vectors, similar to the propagation of zeroes used in $\#\mathsf{3degr}(X)$.

The expressions $\mathsf{eqpart}_i(X)$ are constructed based on G. Next, using our assumption $G \equiv_{\mathsf{ML}(\mathcal{L}^+)} H$, we show that the vectors $\mathsf{eqpart}_i(A_H)$, for $i = 1, \ldots, \ell$, also correspond to the coarsest equitable partition of H. This is done in a number of steps:

1. We verify that each $\mathsf{eqpart}_i(A_H)$ is also an indicator vector containing the same number of 1's as $\mathsf{eqpart}_i(A_G)$.
2. We verify that any distinct pair of indicator vectors in $\{\mathsf{eqpart}_i(A_H)\}$ have no common entry holding value "1". This implies that the set $\{\mathsf{eqpart}_i(A_H)\}$ also represents a *partition*.
3. Finally, we verify that the set $\{\mathsf{eqpart}_i(A_H)\}$ corresponds to an *equitable* partition of H which, together with the partition corresponding to $\{\mathsf{eqpart}_i(A_G)\}$, witnesses that G and H have a *common* equitable partition. Since $\{\mathsf{eqpart}_i(A_G)\}$ is an equitable partition,

$$\mathsf{diag}(\mathsf{eqpart}_i(A_G)) \cdot A_G \cdot \mathsf{diag}(\mathsf{eqpart}_j(A_G)) = \mathsf{deg}(v, V_j) \times \mathsf{diag}(\mathsf{eqpart}_i(A_G)),$$

for some $v \in V_i$. Here, $\mathcal{V} = \{V_1, \ldots, V_\ell\}$ denotes the equitable partition corresponding to indicator vectors $\{\mathsf{eqpart}_i(A_G)\}$. Then, $G \equiv_{\mathsf{ML}(\mathcal{L}^+)} H$ implies that $\mathsf{diag}(\mathsf{eqpart}_i(A_H)) \cdot A_H \cdot \mathsf{diag}(\mathsf{eqpart}_j(A_H))$ is the diagonal matrix $\mathsf{deg}(v, V_j) \times \mathsf{diag}(\mathsf{eqpart}_i(A_H))$. Hence, we have that $\mathsf{deg}(w, W_j) = \mathsf{deg}(w', W_j)$, for any $w, w' \in W_i$, and furthermore, $\mathsf{deg}(v, V_j) = \mathsf{deg}(w, W_j)$. Here, we denote by $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ the partition corresponding to $\{\mathsf{eqpart}_i(A_H)\}$.

All combined, we may conclude that G and H have indeed a common equitable partition. ◀

7.2 Characterisations

For $\mathsf{ML}(\cdot, {}^*, \mathbb{1}, \mathsf{diag}, +, \times, \mathsf{apply}_\mathsf{s}[f], f \in \Omega)$ we also have the converse.

▶ **Proposition 7.6.** *Let G and H be two graphs of the same order. If G and H have a common equitable partition, then $e(A_G) = e(A_H)$ for any sentence $e(X)$ in $\mathsf{ML}(\cdot, {}^*, \mathbb{1}, \mathsf{diag}, +, \times, \mathsf{apply}_\mathsf{s}[f], f \in \Omega)$.*

Proof. Let $\mathcal{V} = \{V_1, \ldots, V_\ell\}$ and $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ be the common coarsest equitable partitions of G and H, respectively. Denote by $\{\mathbb{1}_{V_i}\}$ and $\{\mathbb{1}_{W_i}\}$, for $i = 1, \ldots, \ell$, the corresponding indicator vectors. We know from Proposition 7.2 that there exists a doubly stochastic matrix S such that $A_G \cdot S = S \cdot A_H$. In fact, S can be assumed to have a *block structure* in which the only non-zero blocks are those relating $\mathbb{1}_{V_i}$ and $\mathbb{1}_{W_i}$ [50]. As a consequence, $\mathbb{1}_{V_i} = S \cdot \mathbb{1}_{W_i}$ and $\mathbb{1}_{V_i}^\mathsf{t} \cdot S = \mathbb{1}_{W_i}^\mathsf{t}$ for $i = 1, \ldots, \ell$. The key insight in the proof is that when $e(A_G)$ is an $n \times 1$-vector, it can be written as a linear combination of $\mathbb{1}_{V_i}$'s,

say $\sum a_i \times \mathbb{1}_{V_i}$. Moreover, also $e(A_H) = \sum a_i \times \mathbb{1}_{W_i}$. As a consequence, $e(A_G) = S \cdot e(A_H)$ meaning that $e(A_G)$ is just a permutation of $e(A_H)$. For this to hold, it is essential that we work with equitable partitions common to G and H. For example, if $e(X) := X \cdot \mathbb{1}(X)$ then

$$\begin{aligned} e(A_G) = A_G \cdot \mathbb{1} &= \sum_{i=1}^{\ell} A_G \cdot \mathbb{1}_{V_i} = \sum_{i,j=1}^{\ell} \mathsf{deg}(v_i, V_j) \times \mathbb{1}_{V_i} \\ &= \sum_{i,j=1}^{\ell} \mathsf{deg}(w_i, W_j) \times (S \cdot \mathbb{1}_{W_i}) = S \cdot e(A_H), \end{aligned}$$

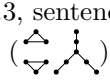
for some $v_i \in V_i$ and $w_i \in W_i$. The challenging case in the proof is when $e(X) := \mathsf{diag}(e'(X))$. Based on the decomposition of $n \times 1$-vectors and the block structure of S, we have

$$\mathsf{diag}(e'(A_G)) \cdot S = \sum_{i=1}^{\ell} a_i \times \mathsf{diag}(\mathbb{1}_{V_i}) \cdot S = \sum_{i=1}^{\ell} a_i \times (S \cdot \mathsf{diag}(\mathbb{1}_{W_i})) = S \cdot \mathsf{diag}(e'(A_H)),$$

which allows to prove that $A_G \cdot S = S \cdot A_H$ implies that $e(A_G) = e(A_H)$ for all sentences in our fragment. ◀

All combined, we obtain the following characterisation.

▶ **Theorem 7.7.** *Let G and H be two graphs of the same order. Then, $G \equiv_{\mathsf{ML}(\cdot,^*,\mathbb{1},\mathsf{diag})} H$ if and only if $G \equiv_{\mathsf{ML}(\cdot,^*,\mathbb{1},\mathsf{diag},+,\times,\mathsf{apply_s}[f],f\in\Omega)} H$ if and only if there is a doubly stochastic matrix S such that $A_G \cdot S = S \cdot A_H$ if and only if $G \equiv_{\mathsf{C}^2} H$.*

As a consequence, following Example 7.3, sentences in $\mathsf{ML}(\cdot,^*,\mathbb{1},\mathsf{diag})$ can distinguish G_1 () and H_1 (), G_2 () and H_2 (), G_4 () and H_4 (), but cannot distinguish G_3 () and H_3 ().

We next turn our attention to $\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1},\mathsf{diag})$- and $\mathsf{ML}(\cdot,^*,\mathsf{tr},\mathbb{1},\mathsf{diag},+,\times,\mathsf{apply_s}[f],f \in \Omega)$-equivalence. Theorem 5.3 implies that G and H are co-spectral and we thus need to combine the existence of a common equitable partition with the existence of an orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$. We remark that we cannot simply require O to be doubly stochastic as this would imply that O is a permutation matrix[2], which in turn would imply that G and H are isomorphic, contradicting that our fragments cannot go beyond C^3-equivalence, as we see later.

A characterisation is obtained inspired by a characterisation of simultaneous equivalence of the so-called 1-dimensional Weisfeiler-Lehman closure of adjacency matrices [52]. Let $\mathcal{V} = \{V_1, \ldots, V_\ell\}$ and $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ be common equitable partitions of G and H. Following Thüne [52], we say that an orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$ is *compatible with $\mathcal{V}$ and $\mathcal{W}$* if O can be block partitioned into ℓ orthogonal matrices O_i of size $|V_i|$ such that $\mathbb{1}_{V_i} = O \cdot \mathbb{1}_{W_i}$, for all $i = 1, \ldots, \ell$. Given this notion, we have the following characterisation.

▶ **Theorem 7.8.** *Let G and H be graphs of the same order. Then the following holds: $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1},\mathsf{diag})} H$ if and only if $G \equiv_{\mathsf{ML}(\cdot,^*,\mathsf{tr},\mathbb{1},\mathsf{diag},+,\times,\mathsf{apply_s}[f],f\in\Omega)} H$ if and only if G and H have a common equitable partition, say $\mathcal{V}$ and $\mathcal{W}$, and furthermore $A_G \cdot O = O \cdot A_H$ for some orthogonal matrix O that is compatible with $\mathcal{V}$ and $\mathcal{W}$.*

[2] This is an immediate consequence of the Birkhoff-von Neumann Theorem which states that any doubly stochastic matrix lies in the convex hull of permutation matrices [45].

Proof. If $G \equiv_{\mathsf{ML}(\cdot,^*,\mathsf{tr},\mathbb{1},\mathsf{diag},+,\times,\mathsf{apply_s}[f],f\in\Omega)} H$, then for any k, $\mathsf{tr}(e(A_G)^k) = \mathsf{tr}(e(A_H)^k)$ for any expression $e(X)$ such that $e(A_G)$ (and thus also $e(A_H)$) is an $n \times n$-matrix. As argued in Thüne [52] this implies the existence of a single orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$ and $e(A_G) \cdot O = O \cdot e(A_H)$. (The proof relies on Specht's Theorem which relates the existence of an orthogonal matrix *simultaneously* linking sets of matrices to trace equality conditions [37].) In particular, $\mathsf{diag}(\mathsf{eqpart}_i(A_G)) \cdot O = O \cdot \mathsf{diag}(\mathsf{eqpart}_i(A_H))$, for $i = 1, \ldots, \ell$, where $\mathsf{eqpart}_i(X)$ are the expressions computing the equitable partition given in the proof of Proposition 7.5. Lemma 6 in Thüne [52] shows that O must be compatible with the common equitable partitions represented by $\mathsf{eqpart}_i(A_G)$ and $\mathsf{eqpart}_i(A_H)$.

For the converse, we argue as in Proposition 7.6, using orthogonal matrices (which preserve the trace operation) instead of doubly stochastic matrices. ◀

Note that $G \equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1},\mathsf{diag})} H$ implies $G \equiv_{\mathsf{ML}(\cdot,^*,\mathbb{1},\mathsf{diag})} H$. The converse does not hold.

▶ **Example 7.9.** Consider G_3 () and H_3 (). These graphs are fractional isomorphic but are not co-spectral. Hence, $G_3 \not\equiv_{\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1},\mathsf{diag})} H_3$ since $\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1},\mathsf{diag})$-equivalence implies co-spectrality. On the other hand, G_5 () and H_5 () are co-spectral regular graphs [55], with co-spectral complements, which cannot be distinguished by $\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1},\mathsf{diag})$.

A close inspection of the proofs of Proposition 7.6 and Theorem 7.8, shows that $G \equiv_{\mathsf{ML}(\mathcal{L}^+)} H$ implies that for any expression $e(X)$ in $\mathsf{ML}(\mathcal{L}^+)$ such that $e(A_G)$ (and thus also $e(A_H)$) is an $n \times 1$-vector, $e(A_G)$ is a permutation of $e(A_H)$. Indeed, both can be written as linear combinations of indicator vectors, $e(A_G)$ in terms of $\mathbb{1}_{V_i}$'s and $e(A_H)$ in terms of $\mathbb{1}_{W_i}$'s, using the *same* coefficients. This implies that we can allow pointwise function applications *on vectors* and scalars, denoted by $\mathsf{apply_v}[f]$, $f \in \Omega$, at no increase in expressiveness.

▶ **Corollary 7.10.** *Let G and H be two graphs of the same order. We have that $G \equiv_{\mathsf{ML}(\mathcal{L})} H$ if and only if $G \equiv_{\mathsf{ML}(\mathcal{L}^+\cup\{\mathsf{apply_v}[f],f\in\Omega\})} H$.*

▶ Remark 7.11. An equitable partition can be defined *without* the $\mathsf{diag}(\cdot)$ operation, provided that function applications on *vectors* are allowed. Hence, the same story holds when first adding pointwise function applications on vectors to $\mathsf{ML}(\cdot,^*,\mathbb{1})$ and $\mathsf{ML}(\cdot,\mathsf{tr},\mathbb{1})$, rather than first adding $\mathsf{diag}(\cdot)$ like we did in this section.

8 The Impact of Pointwise Functions on Matrices

We conclude by considering pointwise function applications on *matrices*, the only operation from Table 1 that we did not consider yet. As we will see shortly, *pointwise multiplication of matrices*, also known as the Schur-Hadamard product, is what results in an increase in expressive power. We denote the Schur-Hadamard product by the binary operator $\circ$, i.e., $(A \circ B)_{ij} = A_{ij}B_{ij}$ for matrices A and B.

▶ **Example 8.1.** We recall that in expression $\#\mathsf{3degr}(X)$ in Example 7.1, products of diagonal matrices resulted in the ability to zoom in on *vertices* that carry specific degree information. When diagonal matrices are concerned, the product of matrices coincides with pointwise multiplication of the *vectors* on the diagonals. Allowing pointwise multiplication on matrices has the same effect, but now on *edges* in graphs. As an example, suppose that we want to count the number of "triangle paths" in G, i.e., paths $(v_0, \ldots, v_k)$ of length k in G such that each edge (v_{i-1}, v_i) on the path is part of a triangle. This can be done by expression

$$\#\Delta\mathsf{paths}_k(X) := \mathbb{1}(X)^* \cdot ((\mathsf{apply}[f_{>0}](X^2 \circ X))^k \cdot \mathbb{1}(X),$$

where $f_{>0}(x) = 1$ if $x \neq 0$ and $f_{>0}(x) = 0$ otherwise [3]. Indeed, when evaluated on adjacency matrix A_G, $A_G^2 \circ A_G$ extracts from A_G^2 only those entries corresponding to paths (u, v, w) of length 2 such that (u, w) is an edge as well, i.e., it identifies edges involved in triangles. Then, $\mathsf{apply}[f_{>0}](A_G^2 \circ A_G)$ sets all non-zero entries to 1. By considering the kth power of this matrix and summing up all its entries, the number of triangle paths is obtained. It can be verified that for graphs G_5 ( ) and H_5 ( ), $\#\Delta\mathsf{paths}_2(A_{G_5}) = [160] \neq [132] = \#\Delta\mathsf{paths}_2(A_{H_5})$ and hence, they can be distinguished when the Schur-Hadamard product is available. Recall that all previous fragments could not distinguish between these two graphs.

In fact, in $\mathsf{ML}(\cdot, ^*, \mathsf{tr}, \mathbb{1}, \mathsf{diag}, +, \times, \circ)$ we can compute the *coarsest stable edge colouring* of a graph $G = (V, E)$ which arises as the result of applying the edge colouring algorithm by Weisfeiler-Lehman [6, 11, 47, 57]. Initially, an edge colouring $\chi_0 : V \times V \to \{0, 1, 2\}$ is defined such that $\chi_0(v, v) = 2$, $\chi_0(v, w) = 1$ if $\{v, w\} \in E$, and $\chi_0(v, w) = 0$ for $v \neq w$ and $\{v, w\} \notin E$. Such a colouring naturally induces a partitioning Π_{χ_0} of $V \times V$. A colouring $\chi : V \times V \to C$ for some set of colours C is called *stable* if and only if for any two pairs (v_1, v_2) and (v_1', v_2') in $V \times V$,

$$\chi(v_1, v_2) = \chi(v_1', v_2') \Leftrightarrow \mathsf{L}^2(v_1, v_2) = \mathsf{L}^2(v_1', v_2'),$$

where for a pair $(v, v') \in V \times V$ and pairs (c, d) of colours in C,

$$\mathsf{L}^2(v, v') := \{(c, d, p_{v,v'}^{c,d}) \mid p_{v,v'}^{c,d} \neq 0\} \text{ and } p_{v,v'}^{c,d} := |\{v'' \in V \mid \chi(v, v'') = c, \chi(v'', v') = d\}|.$$

In other words, $\mathsf{L}^2(v, v')$ lists the number of triangles (v, v', v'') in which (v, v'') has colour c and (v'', v) has colour d, for each pair of colours. Such a stable edge colouring χ is called *coarsest* when the corresponding edge partition Π_χ is the coarsest stable edge partition. That is, Π_χ refines Π_{χ_0}, χ is stable and any other colouring satisfying these conditions results in a finer partition than Π_χ.

Two graphs $G = (V, E)$ and $H = (W, F)$ are said to be *indistinguishable by edge colouring*, denoted by $G \equiv_{\mathsf{2WL}} H$, if the following holds. Let $\Pi_{\chi_G} = \{E_1, \ldots, E_\ell\}$ and $\Pi_{\chi_H} = \{F_1, \ldots, F_\ell\}$ be the edge partitions corresponding to stable edge colourings χ_G and χ_H of H. Then, $G \equiv_{\mathsf{2WL}} H$ if there is a bijection $\imath : \Pi_{\chi_G} \to \Pi_{\chi_H}$ such that E_i and $F_{\imath(i)}$ have the same colour and the same number of entries carrying value 1.

In the seminal paper by Cai, Fürer and Immerman [11], the following was shown.

▶ **Theorem 8.2.** *Let G and H be two graphs of the same order. Then, $G \equiv_{\mathsf{2WL}} H$ if and only if $G \equiv_{\mathsf{C}^3} H$.*

We have the following characterisation of $\mathsf{ML}(\cdot, ^*, \mathsf{tr}, \mathbb{1}, \mathsf{diag}, +, \times, \circ)$-equivalence.

▶ **Theorem 8.3.** *Let G and H be two graphs of the same order, then $G \equiv_{\mathsf{ML}(\cdot, ^*, \mathsf{tr}, \mathbb{1}, \mathsf{diag}, +, \times, \circ)} H$ if and only if $G \equiv_{\mathsf{C}^3} H$.*

Proof. We only have space here to sketch the proof. The proof is not that different from the one used in the context of equitable partitions. Let $G = (V, E)$ and $H = (W, F)$ be two graphs. First, we simulate algorithm 2-STAB(A_G) [6], that computes the coarsest stable edge colouring, by expressions $\mathsf{stabcol}_i(X)$, for $i = 1, \ldots, \ell$, in $\mathsf{ML}(\cdot, ^*, \mathsf{tr}, \mathbb{1}, \mathsf{diag}, +, \times, \circ)$. Each $\mathsf{stabcol}_i(A_G)$ is an *indicator matrix* representing the part of the partition Π of $V \times V$

[3] The use of $\mathsf{apply}[f_{>0}](\cdot)$ is just for convenience and can be simulated when evaluated on given instances using $\cdot$, $+$, $\times$ and $\circ$.

corresponding to a specific colour. Based on well-known properties of these indicator matrices (they form standard basis of the *cellular* or *coherent algebra* associated with G [33]), we show that $G \equiv_{\mathsf{ML}(\cdot,^*,\mathsf{tr},\mathbb{1},\mathsf{diag},+,\times,\circ)} H$ implies that $\{\mathsf{stabcol}_i(A_H)\}$ also represent a partition of $W \times W$ corresponding to the coarsest stable colouring of H. Finally, G and H are shown to be indistinguishable by edge colouring, based on the partitions $\{\mathsf{stabcol}_i(A_G)\}$ and $\{\mathsf{stabcol}_i(A_H)\}$. Hence, by Theorem 8.2, $G \equiv_{\mathsf{ML}(\cdot,^*,\mathsf{tr},\mathbb{1},\mathsf{diag},+,\times,\circ)} H$ implies $G \equiv_{\mathsf{C}^3} H$.

For the converse, we use that $G \equiv_{\mathsf{2WL}} H$ implies that there exists an orthogonal matrix O such that $A_G \cdot O = O \cdot A_H$ and furthermore, the mapping $Y \mapsto O \cdot Y \cdot O^{\mathsf{t}}$ is an isomorphism between the cellular algebras of G and H. In particular, it commutes with the Schur-Hadamard product [23]. This is crucial to show that $e(A_G) = e(A_H)$ for all sentences $e(X) \in \mathsf{ML}(\cdot,^*,\mathsf{tr},\mathbb{1},\mathsf{diag},+,\times,\circ)$. ◀

▶ Remark 8.4. We can do some simplification in $\mathsf{ML}(\cdot,^*,\mathsf{tr},\mathbb{1},\mathsf{diag},+,\times,\circ)$. Indeed, the trace operator can be simulated by $\mathsf{tr}(e(X)) = \mathbb{1}(X)^* \cdot (e(X) \circ \mathsf{diag}(\mathbb{1}(X))) \cdot \mathbb{1}(X)$ and can hence be omitted. Moreover, $\mathsf{diag}(\cdot)$ can be replaced by a simpler operator, denoted by Id, which returns the identity matrix of the same dimensions as the input. Indeed, $\mathsf{diag}(e(X)) = (e(X) \cdot \mathbb{1}(X)^*) \circ \mathsf{Id}(X)$. We can thus work with $\mathsf{ML}(\cdot,^*,\mathbb{1},\mathsf{Id},+,\times,\circ)$ instead.

▶ Remark 8.5. Similar to Corollary 7.10, we can allow *any* pointwise function application on matrices. This follows from the proof of Theorem 8.3 in which it is shown that for expressions $e_i(X)$, for $i = 1,\ldots,p$, such that each $e_i(A_G)$ (and thus also each $e_i(A_H)$) is an $n \times n$-matrix, $e_i(A_G) = \sum a_j^{(i)} \times \mathsf{stabcol}_j(A_G)$ and $e(A_H) = \sum a_j^{(i)} \times \mathsf{stabcol}_i(A_H)$, for scalars $a_j^{(i)} \in \mathbb{C}$. This implies that $\mathsf{apply}[f]\big(e_1(A_G),\ldots,e_p(A_G)\big) = \sum f(a_j^{(1)},\ldots,a_j^{(p)}) \times \mathsf{stabcol}_j(A_G)$, and similarly, $\mathsf{apply}[f]\big(e_1(A_H),\ldots,e_p(A_H)\big) = \sum f(a_j^{(1)},\ldots,a_j^{(p)}) \times \mathsf{stabcol}_j(A_H)$ As a consequence, $\mathsf{apply}[f]\big(e_1(A_G),\ldots,e_p(A_G)\big) \cdot O = O \cdot \mathsf{apply}[f]\big(e_1(A_H),\ldots,e_p(A_H)\big)$ for the orthogonal matrix O in the proof of Theorem 8.3. This suffices to show that $e(A_G) = e(A_H)$ for any sentence $e(X)$ in $\mathsf{ML}(\cdot,^*,\mathsf{tr},\mathbb{1},\mathsf{diag},+,\times,\mathsf{apply}[f], f \in \Omega)$, or in other words, for any sentence in MATLANG.

▶ Remark 8.6. The orthogonal matrix O in the proof of Theorem 8.3 can be taken to be compatible with the common equitable partitions of G and H, just as in Theorem 7.8. This follows from the fact that the diagonal indicator matrices $\mathsf{diag}(\mathsf{eqpart}_i(A_G))$ are part of the indicator matrices that constitute the basis of the cellular algebra of G [6].

▶ Remark 8.7. An almost direct consequence of Theorem 8.3 is that $G \equiv_{\mathsf{2WL}} H$ implies that G and H have the same number of (simple) cycles of length 7. This was known to hold for cycles of length $\ell = 1,2,\ldots,6$, and known not to hold for cycles of length greater than 7 [25]. The case $\ell = 7$ was left open in Fürer [25]. In view of Theorem 8.3 it suffices to show that we can count cycles of length $\ell = 1,2,\ldots,7$ in MATLANG. This, however, is a direct consequence of the formulas for counting cycles given in Noga et al. [1]. Indeed, a close inspection of these formulas reveals that only limited linear algebra functionality is required and hence they can be formulated in MATLANG. Although formulas exist for counting cycles of length greater than 7, they require to count the number of 4-cliques, which is not possible in MATLANG.

9 Concluding Remarks

We have characterised $\mathsf{ML}(\mathcal{L})$-equivalence for undirected graphs and clearly identified what additional distinguishing power each of the operations has. That natural characterisations can be obtained once more attests that MATLANG is an adequate matrix language. Although motivated by the increased interest in integrating linear algebra functionality in database

management systems, the presented results are primarily of theoretical interest. As such, they do not directly translate into effective procedures for evaluating or optimising linear algebra inside database systems.

We conclude with some avenues for further investigation. Although some of the results generalise to directed graphs (with asymmetric adjacency matrices), an extension to the case when queries can have multiple inputs seems challenging. The generalisation beyond graphs, i.e., for arbitrary matrices, is wide open. Of interest may also be to connect $\mathsf{ML}(\mathcal{L})$-equivalence to fragments of first-order logic (without counting). A possible line of attack could be to work over the boolean semiring instead of over the complex numbers (see Grohe and Otto [29] for a similar approach). More general semirings could open the way for modelling and querying labeled graphs using matrix query languages.

We also note that MATLANG was extended in Brijder et al. [9] with an operator inv that computes the inverse of a matrix, if it exists, and returns the zero matrix otherwise. The extension, MATLANG + inv, was shown to be more expressive than MATLANG. For example, connectedness of graphs can be checked by a single sentence in MATLANG + inv. Of course, we here consider *equivalence* of graphs. Even when considering a "classical" logic like FO^3, the three-variable fragment of first-order logic, $G \equiv_{\mathsf{FO}^3} H$ implies that G is connected if and only if H is connected. Translated to our setting, for any fragment $\mathsf{ML}(\mathcal{L})$ in which $G \equiv_{\mathsf{ML}(\mathcal{L})} H$ implies that the Laplacian $\mathsf{diag}(A_G \cdot \mathbb{1}) - A_G$ of G is co-spectral with the Laplacian of $\mathsf{diag}(A_H \cdot \mathbb{1}) - A_H$ of H, $G \equiv_{\mathsf{ML}(\mathcal{L})} H$ implies that G is connected if and only if H is connected. It even implies that G and H must have the same number of connected components, as this is determined by the multiplicity of the eigenvalue 0 of the Laplacian [10]. Nevertheless, we can also consider equivalence of graphs relative to MATLANG + inv. We observe, however, that the inverse of a matrix can be computed using $+$ and $\times$, by the Cayley-Hamilton Theorem [4], given the coefficients of the characteristic polynomial of the adjacency matrix. These coefficients can be computed using $+$, $\times$ and tr. For fragments supporting $\cdot$, $+$, $\times$ and tr, the operator inv thus does not add distinguishing power. It is unclear what the impact is of inv for smaller fragments such as $\mathsf{ML}(\cdot, \cdot\mathbb{1})$ and $\mathsf{ML}(\cdot, {}^*, \mathbb{1}, \mathsf{diag})$.

To relate our notion of equivalence more closely to the expressiveness questions studied in Brijder et al. [9], it may be interesting to investigate notions of *locality* of $\mathsf{ML}(\mathcal{L})$ expressions, as this underlies the inexpressibility of connectivity of MATLANG [42]. It would be nice if this can be achieved in purely algebraic terms, without relying on locality notions in logic.

To conclude, MATLANG was also extended with an eigen operator which returns a matrix whose columns consist of eigenvectors spanning the eigenspaces [9]. Since the choice of eigenvectors is not unique, this results in a non-deterministic semantics. We leave it for future work to study the equivalence of graphs relative to *deterministic* fragments supporting the eigen operator, i.e., such that the result of expressions does not depend on the eigenvectors returned. As a starting point one could, for example, force determinism by considering a certain answer semantics. That is, if $e(X)$ is an expression using $\mathsf{eigen}(X)$, one can define $\mathsf{cert}(e(A_G)) := \bigcap_V e(A_G, V)$, where V ranges over all bases of the eigenspaces. Distinguishability with regards to such a certain answer semantics demands further investigation.

References

1 Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. doi:10.1007/BF02523189.

2 Renzo Angles, Marcelo Arenas, Pablo Barceló, Aidan Hogan, Juan Reutter, and Domagoj Vrgoč. Foundations of Modern Query Languages for Graph Databases. *ACM Comput. Surv.*, 50(5):68:1–68:40, 2017. doi:10.1145/3104031.

3 Albert Atserias and Elitza N. Maneva. Sherali-Adams Relaxations and Indistinguishability in Counting Logics. *SIAM J. Comput.*, 42(1):112–137, 2013. doi:10.1137/120867834.

4 Sheldon Axler. *Linear Algebra Done Right*. Springer, third edition, 2015. doi:10.1007/978-3-319-11080-6.

5 Pablo Barceló. Querying Graph Databases. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS, pages 175–188, 2013. doi:10.1145/2463664.2465216.

6 Oliver Bastert. *Stabilization procedures and applications*. PhD thesis, Technical University Munich, Germany, 2001. URL: http://nbn-resolving.de/urn:nbn:de:bvb:91-diss2002070500045.

7 Matthias Boehm, Michael W. Dusenberry, Deron Eriksson, Alexandre V. Evfimievski, Faraz Makari Manshadi, Niketan Pansare, Berthold Reinwald, Frederick R. Reiss, Prithviraj Sen, Arvind C. Surve, and Shirsih Tatikonda. SystemML: Declarative machine learning on Spark. *Proceedings of the VLDB Endowment*, 9(13):1425–1436, 2016. doi:10.14778/3007263.3007279.

8 Matthias Boehm, Berthold Reinwald, Dylan Hutchison, Prithviraj Sen, Alexandre V. Evfimievski, and Niketan Pansare. On Optimizing Operator Fusion Plans for Large-scale Machine Learning in SystemML. *Proceedings of the VLDB Endowment*, 11(12):1755–1768, 2018. doi:10.14778/3229863.3229865.

9 Robert Brijder, Floris Geerts, Jan Van den Bussche, and Timmy Weerwag. On the Expressive Power of Query Languages for Matrices. In *21st International Conference on Database Theory*, ICDT, pages 10:1–10:17, 2018. doi:10.4230/LIPIcs.ICDT.2018.10.

10 Andries E. Brouwer and Willem H. Haemers. *Spectra of Graphs*. Universitext. Springer, 2012. doi:10.1007/978-1-4614-1939-6.

11 Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992. doi:10.1007/BF01305232.

12 Lingjiao Chen, Arun Kumar, Jeffrey Naughton, and Jignesh M. Patel. Towards Linear Algebra over Normalized Data. *Proceedings of the VLDB Endowment*, 10(11):1214–1225, 2017. doi:10.14778/3137628.3137633.

13 Dragoš M. Cvetković. GRAPHS AND THEIR SPECTRA. *Publikacije Elektrotehničkog fakulteta. Serija Matematika i fizika*, 354/356:1–50, 1971. URL: http://www.jstor.org/stable/43667526.

14 Dragoš M. Cvetković. The main part of the spectrum, divisors and switching of graphs. *Publ. Inst. Math. (Beograd) (N.S.)*, 23(37):31–38, 1978. URL: http://elib.mi.sanu.ac.rs/files/journals/publ/43/6.pdf.

15 Dragoš M. Cvetković, Peter Rowlinson, and Slobodan Simić. *Eigenspaces of Graphs*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1997. doi:10.1017/CBO9781139086547.

16 Dragoš M. Cvetković, Peter Rowlinson, and Slobodan Simić. *An Introduction to the Theory of Graph Spectra*. London Mathematical Society Student Texts. Cambridge University Press, 2009. doi:10.1017/CBO9780511801518.

17 Anuj Dawar. On the Descriptive Complexity of Linear Algebra. In *Proceedings of the 15th International Workshop on Logic, Language, Information and Computation*, WoLLIC, pages 17–25, 2008. doi:10.1007/978-3-540-69937-8_2.

18 Anuj Dawar, Martin Grohe, Bjarki Holm, and Bastian Laubner. Logics with Rank Operators. In *Proceedings of the 24th Annual IEEE Symposium on Logic In Computer Science*, LICS, pages 113–122, 2009. doi:10.1109/LICS.2009.24.

19 Anuj Dawar and Bjarki Holm. Pebble games with algebraic rules. *Fund. Inform.*, 150(3-4):281–316, 2017. doi:10.3233/FI-2017-1471.

20 Anuj Dawar, Simone Severini, and Octavio Zapata. Descriptive Complexity of Graph Spectra. In *Proceedings of the 23rd International Workshop on Logic, Language, Information and Computation*, WoLLIC, pages 183–199, 2016. doi:10.1007/978-3-662-52921-8_12.

21 Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász Meets Weisfeiler and Lehman. In *45th International Colloquium on Automata, Languages, and Programming*, ICALP, pages 40:1–40:14, 2018. doi:10.4230/LIPIcs.ICALP.2018.40.

22 Ahmed Elgohary, Matthias Boehm, Peter J. Haas, Frederick R. Reiss, and Berthold Reinwald. Compressed linear algebra for large-scale machine learning. *The VLDB Journal*, pages 1–26, 2017. doi:10.1007/s00778-017-0478-1.

23 Shmuel Friedland. Coherent algebras and the graph isomorphism problem. *Discrete Applied Mathematics*, 25(1):73–98, 1989. doi:10.1016/0166-218X(89)90047-4.

24 Martin Fürer. On the power of combinatorial and spectral invariants. *Linear Algebra and its Applications*, 432(9):2373–2380, 2010. doi:10.1016/j.laa.2009.07.019.

25 Martin Fürer. On the Combinatorial Power of the Weisfeiler-Lehman Algorithm. In Dimitris Fotakis, Aris Pagourtzis, and Vangelis Th. Paschos, editors, *Algorithms and Complexity*, pages 260–271. Springer, 2017. doi:10.1007/978-3-319-57586-5_22.

26 Chris Godsil and Gordon F. Royle. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer, 2001. doi:10.1007/978-1-4613-0163-9.

27 Erich Grädel and Wied Pakusa. Rank Logic is Dead, Long Live Rank Logic! In *24th EACSL Annual Conference on Computer Science Logic*, CSL, pages 390–404, 2015. doi:10.4230/LIPIcs.CSL.2015.390.

28 Martin Grohe, Kristian Kersting, Martin Mladenov, and Erkal Selman. Dimension Reduction via Colour Refinement. In *22th Annual European Symposium on Algorithms*, ESA, pages 505–516, 2014. doi:10.1007/978-3-662-44777-2_42.

29 Martin Grohe and Martin Otto. Pebble games and linear equations. *The Journal of Symbolic Logic*, 80(3):797–844, 2015. doi:10.1017/jsl.2015.28.

30 Martin Grohe and Wied Pakusa. Descriptive complexity of linear equation systems and applications to propositional proof complexity. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS, pages 1–12, 2017. doi:10.1109/LICS.2017.8005081.

31 Willem H. Haemers and Edward Spence. Enumeration of cospectral graphs. *European J. Combin.*, 25(2):199–211, 2004. doi:10.1016/S0195-6698(03)00100-8.

32 Frank Harary and Allen J. Schwenk. The spectral approach to determining the number of walks in a graph. *Pacific J. Math.*, 80(2):443–449, 1979. URL: https://projecteuclid.org:443/euclid.pjm/1102785717.

33 Donald G. Higman. Coherent algebras. *Linear Algebra and its Applications*, 93:209–239, 1987. doi:10.1016/S0024-3795(87)90326-0.

34 Bjarki Holm. *Descriptive Complexity of Linear Algebra*. PhD thesis, University of Cambridge, 2010.

35 Dylan Hutchison, Bill Howe, and Dan Suciu. LaraDB: A Minimalist Kernel for Linear and Relational Algebra Computation. In *Proceedings of the 4th ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond*, BeyondMR, pages 2:1–2:10, 2017. doi:10.1145/3070607.3070608.

36 Neil Immerman and Eric Lander. Describing Graphs: A First-Order Approach to Graph Canonization. In Alan L. Selman, editor, *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday*, pages 59–81. Springer, 1990. doi:10.1007/978-1-4612-4478-3_5.

37 Naihuan Jing. Unitary and orthogonal equivalence of sets of matrices. *Linear Algebra and its Applications*, 481:235–242, 2015. doi:10.1016/j.laa.2015.04.036.

38 Charles R Johnson and Morris Newman. A note on cospectral graphs. *Journal of Combinatorial Theory, Series B*, 28(1):96–103, 1980. doi:10.1016/0095-8956(80)90058-1.

39 Kristian Kersting, Martin Mladenov, Roman Garnett, and Martin Grohe. Power Iterated Color Refinement. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1904–1910. AAAI Press, 2014. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8377.

40 Andreas Kunft, Alexander Alexandrov, Asterios Katsifodimos, and Volker Markl. Bridging the Gap: Towards Optimization Across Linear and Relational Algebra. In *Proceedings of the 3rd ACM SIGMOD Workshop on Algorithms and Systems for MapReduce and Beyond*, BeyondMR, pages 1:1–1:4, 2016. doi:10.1145/2926534.2926540.

41 Andreas Kunft, Asterios Katsifodimos, Sebastian Schelter, Tilmann Rabl, and Volker Markl. BlockJoin: Efficient Matrix Partitioning Through Joins. *Proceedings of the VLDB Endowment*, 10(13):2061–2072, 2017. doi:10.14778/3151106.3151110.

42 Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004. doi:10.1007/978-3-662-07003-1.

43 Shangyu Luo, Zekai J. Gao, Michael Gubanov, Luis L. Perez, and Christopher Jermaine. Scalable Linear Algebra on a Relational Database System. *SIGMOD Rec.*, 47(1):24–31, 2018. doi:10.1145/3277006.3277013.

44 Peter N. Malkin. Sherali–Adams relaxations of graph isomorphism polytopes. *Discrete Optimization*, 12:73–97, 2014. doi:10.1016/j.disopt.2014.01.004.

45 Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. *Inequalities: Theory of Majorization and its Applications*. Springer, 2011. doi:10.1007/978-0-387-68276-1.

46 Hung Q. Ngo, XuanLong Nguyen, Dan Olteanu, and Maximilian Schleich. In-Database Factorized Learning. In *Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web*, AMW, 2017. URL: http://ceur-ws.org/Vol-1912/paper21.pdf.

47 Martin Otto. *Bounded Variable Logics and Counting: A Study in Finite Models*, volume 9 of *Lecture Notes in Logic*. Cambridge University Press, 2017. doi:10.1017/9781316716878.

48 Motakuri V. Ramana, Edward R. Scheinerman, and Daniel Ullman. Fractional isomorphism of graphs. *Discrete Mathematics*, 132(1-3):247–265, 1994. doi:10.1016/0012-365X(94)90241-0.

49 Peter Rowlinson. The main eigenvalues of a graph: A survey. *Applicable Analysis and Discrete Mathematics*, 1(2):455–471, 2007. URL: http://www.jstor.org/stable/43666075.

50 Edward R. Scheinerman and Daniel H. Ullman. *Fractional Graph Theory: a Rational Approach to the Theory of Graphs*. John Wiley & Sons, 1997. URL: https://www.ams.jhu.edu/ers/wp-content/uploads/sites/2/2015/12/fgt.pdf.

51 Maximilian Schleich, Dan Olteanu, and Radu Ciucanu. Learning Linear Regression Models over Factorized Joins. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD, pages 3–18, 2016. doi:10.1145/2882903.2882939.

52 Mario Thüne. *Eigenvalues of matrices and graphs*. PhD thesis, University of Leipzig, 2012.

53 Gottfried Tinhofer. Graph isomorphism and theorems of Birkhoff type. *Computing*, 36(4):285–300, 1986. doi:10.1007/BF02240204.

54 Gottfried Tinhofer. A note on compact graphs. *Discrete Applied Mathematics*, 30(2):253–264, 1991. doi:10.1016/0166-218X(91)90049-3.

55 Edwin R. van Dam and Willem H. Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its Applications*, 373:241–272, 2003. doi:10.1016/S0024-3795(03)00483-X.

56 Erwin R. van Dam, Willem H. Haemers, and Jack H. Koolen. Cospectral graphs and the generalized adjacency matrix. *Linear Algebra and its Applications*, 423(1):33–41, 2007. doi:10.1016/j.laa.2006.07.017.

57 Boris Weisfeiler. *On Construction and Identification of Graphs*. Number 558 in Lecture Notes in Mathematics. Springer-Verlag Berlin Heidelberg, 1976. doi:10.1007/BFb0089374.