Pebble Games with Algebraic Rules*

Online preprint: March 18, 2012

Anuj Dawar and Bjarki Holm

University of Cambridge Computer Laboratory
firstName.lastName@cl.cam.ac.uk

Abstract. We define a general framework for formulating model-comparison games. This framework, which we call partition games, includes as special cases the pebble games for finite-variable logics and finite-variable logics with counting. We introduce two new pebble games that are obtained by imposing linear-algebraic conditions on the moves in a partition game. The first of these, called the *matrix-equivalence game*, characterises equivalence in the finite-variable fragments of matrix-rank logic. The second, called the *invertible-map game*, yields a refinement of the equivalence defined by the matrix-equivalence game and we show that this game equivalence is polynomial-time decidable. This yields a family of polynomial-time approximations of graph isomorphism that is strictly stronger than the Weisfeiler-Lehman method.

1 Introduction

An important open question that has motivated much work in finite model theory is that of finding a logical characterisation of polynomial-time computability. That is to say, to find a logic in which a class of finite structures is expressible if, and only if, membership in the class is decidable in deterministic polynomial time (PTIME). The exact formulation of the question (see [8]) requires additional effectivity conditions which need not concern us here. By a result proved independently by Immerman [11] and Vardi [16], it is known that fixed-point logic (IFP) expresses exactly the polynomial-time properties of *ordered* finite structures, but falls short of expressing the polynomial-time properties of *all* finite structures. It was at one time conjectured by Immerman that extending IFP with a mechanism for counting would yield a logic, IFPC, sufficient for expressing all of PTIME. However, this turns out not to be the case and a counterexample was constructed by Cai, Fürer and Immerman [2]. Noting that the Cai-Fürer-Immerman construction and various other examples of properties in PTIME that are not definable in IFPC can be reduced to testing the solvability of systems of linear equations, in [4] we introduced the extension of fixed-point logic with matrix-rank operators (IFPR). This logic strictly extends the expressive power of IFPC while still being contained in PTIME. It remains an open question whether there are PTIME properties that are not definable in IFPR.

The study of ever more expressive logics has gone hand in hand with the development of tools for proving limitations on the expressive power of such logics. An

* Research supported by EPSRC grant EP/H026835/1.

important class of such tools are the pebble games, which are variations and extensions of the Ehrenfeucht-Fraïssé game for first-order logic. In particular, the *k-pebble game* characterises the relation $\equiv^L_k$ of equivalence in L^k—first-order logic with k variables. Since it can be shown that any formula of IFP is invariant under $\equiv^L_k$ for some k, this is useful in proving inexpressibility results for IFP. Similarly, inexpressibility results for IFPC are established by showing that a property is not invariant under $\equiv^C_k$ for any k. This is the relation of equivalence in C^k—first-order logic with counting quantifiers and at most k variables. The relation $\equiv^C_k$ is characterised by two different pebble games: the Immerman-Lander game [12] and the *bijection game* of Hella [9].

In addition to providing a tool for the analysis of logics, these games also provide interesting approximations of the graph isomorphism relation. In particular, the equivalence relation $\equiv^C_{k+1}$ is exactly the relation decided by the k-dimensional Weisfeiler-Lehman method (see [2] for a description of the method and the relationship with $\equiv^C_k$). This is a family of polynomial-time algorithms (the kth algorithm in the family has running time $n^{O(k)}$) which provide ever finer approximations of graph isomorphism and which approach isomorphism in the limit. A key contribution of the Cai-Fürer-Immerman construction of a property in PTIME that is not definable in IFPC is to show that there is no k such that the k-dimensional Weisfeiler-Lehman algorithm decides graph isomorphism.

The logic of matrix-rank operators also yields a family of equivalence relations $\equiv^R_{k,m,p}$ which can be used to analyse definability in IFPR and which also provide another stratification of graph isomorphism. Here, $\equiv^R_{k,m,p}$ refers to equivalence in the logic that extends first-order logic with at most k variables by means of matrix-rank operators of arity at most m for matrices over GF_p. In this paper we present a game that characterises $\equiv^R_{k,m,p}$[1]. The game, which we call the *matrix-equivalence game*, is difficult to use and it remains a challenge to deploy it to establish that there is a PTIME property not closed under $\equiv^R_{k,m,p}$ for any k, p and m.

The matrix-equivalence game and, indeed, the relations $\equiv^R_{k,m,p}$ that it characterises suffer from another limitation as approximations of the graph isomorphism relation. Namely, it is not clear whether the relations $\equiv^R_{k,m,p}$ are decidable in polynomial time. Indeed, the natural algorithm that is obtained from the definition of the matrix-equivalence game is exponential. This leads us to consider an alternative game that we call the *invertible-map game*. This game is obtained by replacing the algebraic matrix-equivalence condition with a condition of simultaneous similarity of tuples of matrices. As a result we obtain a family of equivalence relations $\approx^k_{m,p}$ which refine $\equiv^R_{k,m,p}$. Moreover, using a result of Chistov et al. [3] we are able to show that each of the relations $\approx^k_{m,p}$ is decidable in polynomial time. Therefore, this gives us a family of polynomial-time algorithms which, like the Weisfeiler-Lehman method, approximates isomorphism in the limit. This family is strictly stronger than the Weisfeiler-Lehman method in the sense that it can also distinguish the Cai-Fürer-Immerman graphs at some fixed level.

The games we introduce in this paper are formulated as *partition games*. They are so called because the Duplicator is required at each move to give a suitable partition of the game board. This partition has to satisfy certain algebraic conditions which vary ac-

[1] We have previously presented this in [5, 10] but it has not appeared in a refereed publication.

cording to the game we are considering. It turns out that the games for $\equiv_k^L$ and $\equiv_k^C$ can also be formulated as partition games, by replacing the algebraic rules of the matrix-equivalence game with weaker conditions that the partitions need to satisfy. This provides a general framework for exploring other games and, indeed, other equivalence relations on structures. So far, model-comparison games have been formulated for specific logics. Perhaps we can reverse this and extract suitable logics from well-behaved games? One such challenge is to formulate a logic that corresponds to the invertible map game that we define here.

2 Preliminaries

We assume that all structures are finite and that all vocabularies are finite and relational. Unless otherwise noted, we commonly write τ to denote a vocabulary. We write $U(\mathbf{A})$ for the universe of a structure $\mathbf{A}$ and write $\|\mathbf{A}\|$ for the cardinality of $U(\mathbf{A})$. We denote the class of all finite τ-structures with fixed tuples of $r \in \mathbb{N}$ parameters by $\mathrm{fin}[\tau; r] := \{(\mathbf{A}, \boldsymbol{a}) \mid \mathbf{A} \in \mathrm{fin}[\tau],\ \boldsymbol{a} \in U(\mathbf{A})^r\}$. We denote tuples $(v_1, \ldots, v_k)$ by $\boldsymbol{v}$ and their length by $\|\boldsymbol{v}\|$. If $\boldsymbol{v}$ is a k-tuple of elements from a set X, $i \in [k]$ and $w \in X$, then we write $\boldsymbol{v}\frac{w}{i}$ for the tuple obtained from $\boldsymbol{v}$ by replacing the i-th component with w; that is, $\boldsymbol{v}\frac{w}{i} = (v_1, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_k)$. If $m \leq k$, $\boldsymbol{i} = (i_1, \ldots, i_m) \in [k]^m$ is a tuple of distinct integers (an 'index pattern') and $\boldsymbol{w}$ is an m-tuple of elements from X, then we write $\boldsymbol{v}\frac{\boldsymbol{w}}{\boldsymbol{i}} := \boldsymbol{v}\frac{w_1}{i_1} \cdots \frac{w_m}{i_m}$.

Matrices Indexed by Unordered Sets. If F is a field and I, J are finite and non-empty sets then an $I \times J$ *matrix* over F is a function $A : I \times J \to F$. If $\|I\| = \|J\|$ then we say that A is invertible if there is a $J \times I$ matrix B such that AB is the graph of the identity function on I; that is, if $AB(x, y) = 1$ if $x = y$ and $AB(x, y) = 0$ otherwise[2]. We say that A is a *square matrix* if $I = J$.

In this paper we will focus on square $\{0, 1\}$-matrices whose rows and columns are indexed by tuples of elements from some finite and non-empty base set A. More specifically, if $B \subseteq A^{2m}$ for some $m \geq 1$, then we write χ_B for the characteristic function of B, seen as a $\{0, 1\}$-matrix indexed by $A^m \times A^m$. That is, χ_B is defined by $(\boldsymbol{a}, \boldsymbol{b}) \mapsto 1$ if $(\boldsymbol{a}, \boldsymbol{b}) \in B$ and $(\boldsymbol{a}, \boldsymbol{b}) \mapsto 0$ otherwise. We refer to $\chi_B(\boldsymbol{a}, \boldsymbol{b})$ as the *characteristic matrix of* B; the underlying field and the exponent m are usually clear from the context.

We also consider matrices expressed as a linear combination of characteristic matrices. Let $\mathbf{P} \subseteq \wp(A^{2m})$ be a non-empty collection of subsets of A^{2m} and let $\gamma : \mathbf{P} \to F$ be a function. Then we write $M_\gamma^{\mathbf{P}}$ to denote the $A^m \times A^m$ matrix over F defined by $M_\gamma^{\mathbf{P}} := \sum_{P \in \mathbf{P}} \gamma(P) \cdot \chi_P$. Typically, $\mathbf{P}$ will be a *partition* of A^{2m}; that is, a collection of non-empty and mutually disjoint subsets of A^{2m} (called *blocks*) whose union is all of A^{2m}.

Finite-variable Logics. We write L^k to denote the fragment of first-order logic using only the variables $x_1, \ldots, x_k$ and we write C^k for the extension of L^k with rules for

[2] Equivalently, it can be seen that A is invertible if there is an $J \times I$ matrix C such that CA is the graph of the identity function on J.

defining counting formulas of the kind $\exists^{\geq i} x \,.\, \varphi(x)$, for each $i \in \mathbb{N}$ (for further details, see [7, 14]). For $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ in $\mathrm{fin}[\tau; r]$, we write $(\mathbf{A}, \boldsymbol{a}) \equiv_k^L (\mathbf{B}, \boldsymbol{b})$ to indicate that for any L^k-formula φ with free variables amongst $\{x_1, \ldots, x_r\}$, it holds that $(\mathbf{A}, \boldsymbol{a}) \models \varphi$ if and only if $(\mathbf{B}, \boldsymbol{b}) \models \varphi$; $\equiv_k^C$ is defined similarity for C^k.

For each integer $i \geq 0$ and prime p, we define a quantifier $\mathrm{rk}_p^{\geq i}$ which binds exactly $2m$ variables and $(p-1)$-formulae. If $\varphi_1, \ldots, \varphi_{p-1}$ are formulae, $\boldsymbol{x}$ and $\boldsymbol{y}$ are m-tuples of pairwise distinct variables, and $\mathbf{A}$ a structure, then we let

$$\mathbf{A} \models \mathrm{rk}_p^{\geq i}(\boldsymbol{x}, \boldsymbol{y}) \,.\, (\varphi_1, \ldots, \varphi_{p-1})$$

if and only if the rank of the square $U(\mathbf{A})^m \times U(\mathbf{A})^m$ matrix $\sum_{j=1}^{p-1} j \cdot \chi_{\varphi_j^{\mathbf{A}}} \pmod p$ is at least i over GF_p, the finite field with p elements. We write $R^k_{m;p}$ to denote the logic built up in the same way as k-variable first-order logic, except that we have rules for constructing formulas with $2m$-ary rank quantifiers over GF_p instead of the rules for first-order existential and universal quantifiers. Every formula in L^k or C^k is equivalent to one of $R^{k+1}_{p;2}$ (where p is any prime), for we can simulate existential, universal and unary counting quantifiers by expressing the rank of diagonal matrices (see [4, 10, 13] for details). We write $(\mathbf{A}, \boldsymbol{a}) \equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b})$ to indicate that $\mathbf{A}$ and $\mathbf{B}$ agree on all $R^k_{m;p}$-formulae under the assignments $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. It can be shown that any formula of IFPR_p is invariant under $\equiv^R_{k,m,p}$ for some k and m [4, 10, 13], where we write IFPR_p for the fragment of fixed-point logic rank operators restricted to matrices over GF_p.

Pebble Games. A pebble game is a two-player model-comparison game where each of the two players (Spoiler and Duplicator) has a finite number of tokens ('pebbles') for placing on the game board. It can be shown that equivalence in L^k is completely characterised by a pebble game where each player has k pebbles [1, 15, 11]. This correspondence gives a purely combinatorial game method for proving inexpressibility results for k-variable logic in general and IFP in particular, since it can be shown that any formula of IFP is invariant under $\equiv_k^L$ for some k. Immerman and Lander [12] and Hella [9] later introduced separate versions of the k-pebble game for analysing the expressiveness of C^k over finite models, which can be used to establish lower bounds for IFPC over finite structures.

Class Extensions and Extension Matrices. We frequently consider relations that arise by extending a fixed tuple of elements in a structure according to some criteria. For example, consider a formula φ and let $\boldsymbol{a}$ be an assignment of values to the free variables of φ over a structure $\mathbf{A}$. Then the set of all pairs (c, d) from $\mathbf{A}$ which, when used to replace the first two elements of $\boldsymbol{a}$ to give a satisfying assignment to φ, can be seen as a binary "extension" of $\boldsymbol{a}$ in $\mathbf{A}$, defined by the formula φ. Moreover, this relation can be viewed as a $\{0, 1\}$-matrix over $\mathbf{A}$ in the usual way, which gives us a way to associate a pair $(\mathbf{A}, \boldsymbol{a})$ with a family of matrices over $\mathbf{A}$.

More formally, consider a class $\alpha \subseteq \mathrm{fin}[\tau; k]$ and let $\boldsymbol{i} = (i_1, \ldots, i_n) \in [k]^n$ be a tuple of distinct integers, $n \leq k$. Then we write $\mathrm{ext}^\alpha_{\boldsymbol{i}}$ to denote the functor on $\mathrm{fin}[\tau; k]$ defined by $\mathrm{ext}^\alpha_{\boldsymbol{i}}(\mathbf{A}, \boldsymbol{a}) := \{(b_1, \ldots, b_n) \in U(\mathbf{A})^n \mid (\mathbf{A}, \boldsymbol{a}\frac{b_1}{i_1} \cdots \frac{b_n}{i_n}) \in \alpha\}$. We refer to $\mathrm{ext}^\alpha_{\boldsymbol{i}}(\mathbf{A}, \boldsymbol{a})$ as the *$\boldsymbol{i}$-extension of* $(\mathbf{A}, \boldsymbol{a})$ *into* α. Abusing notation, if φ is a formula whose free variables are all amongst $\boldsymbol{x} = (x_1, \ldots, x_k)$, then we let $\mathrm{ext}_{\boldsymbol{i}}^{\varphi} := \mathrm{ext}_{\boldsymbol{i}}^{\alpha_\varphi}$, where $\alpha_\varphi := \{(\mathbf{A}, \boldsymbol{a}) \in \mathrm{fin}[\tau; k] \mid (\mathbf{A}, \boldsymbol{a}) \models \varphi\}$. That is,

$$\mathrm{ext}_{\boldsymbol{i}}^{\varphi} = \{(b_1, \ldots, b_n) \in U(\mathbf{A})^n \mid \mathbf{A} \models \varphi[\boldsymbol{a}\tfrac{b_1}{i_1} \cdots \tfrac{b_n}{i_n}]\} \subseteq U(\mathbf{A})^n.$$

If $n = 2m$, then we write $\mathrm{extmat}_{\boldsymbol{i}}^{\alpha}(\mathbf{A}, \boldsymbol{a})$ and $\mathrm{extmat}_{\boldsymbol{i}}^{\varphi}(\mathbf{A}, \boldsymbol{a})$ to denote the $U(\mathbf{A})^m \times U(\mathbf{A})^m$ characteristic matrices of $\mathrm{ext}_{\boldsymbol{i}}^{\alpha}(\mathbf{A}, \boldsymbol{a})$ and $\mathrm{ext}_{\boldsymbol{i}}^{\varphi}(\mathbf{A}, \boldsymbol{a})$, respectively. We refer to such matrices as *extension matrices*.

3 A Game Characterisation of Rank Logics

In this section we give a game characterisation of finite-variable logic with quantifiers for matrix rank. This gives us a combinatorial method for proving lower bounds (inexpressibility results) for fixed-point logic with rank operators.

To give the intuition behind this game, we first describe a simple "partition game" that is a based on the same game protocol. The partition game is played by two players, Spoiler and Duplicator, on a pair of relational structures $\mathbf{A}$ and $\mathbf{B}$, each with k pebbles labelled $1, \ldots, k$. At each round of the game, Spoiler removes a pebble from $\mathbf{A}$ and the corresponding pebble from $\mathbf{B}$. Unlike the classical pebble game, Duplicator is not allowed to move any pebbles herself[3]. However, in response to the challenge of the Spoiler, she is allowed to divide the game board into disjoint regions in order to restrict the possible moves that Spoiler is subsequently allowed to make. More specifically, in response to Spoiler's challenge, Duplicator partitions each of $U(\mathbf{A})$ and $U(\mathbf{B})$ into the same number of disjoint regions and matches each region in $U(\mathbf{A})$ with a unique region in $U(\mathbf{B})$ (and vice versa). Intuitively, Duplicator's strategy will be to gather in each region all those elements that lead to game positions that are sufficiently alike. In turn, Spoiler is allowed to place each of the chosen pebbles on some element of the corresponding structure, with the *restriction* that the two newly pebbled elements have to be within matching regions. That completes a round of the game. Compared with the standard pebble game, it may seem that the partition game is biased against the Duplicator, since she is not allowed to place her own pebbles after seeing where Spoiler places his. However, it can be shown that the two games are actually equivalent over finite structures.

The idea of dividing the game board into disjoint regions leads to a very generic template for designing new pebble games. For instance, if we adapt the rules so that any two matching regions need to have the same cardinality, then we get a game equivalent to the bijection game. The "matrix-equivalence game" we describe next is obtained by putting additional linear-algebraic constraints on the matching game regions.

3.1 Matrix-equivalence Game

Let k, m and p be positive integers with $2m \leq k$ and p prime. The game board of the k-pebble m-ary *matrix-equivalence game* over GF_p (or (k, m, p)-matrix-equivalence game for short) consists of two structures $\mathbf{A}$ and $\mathbf{B}$ of the same vocabulary, each with k

[3] By convention, Spoiler is male and Duplicator female.

pebbles labelled $1, \ldots, k$. The first r pebbles of $\mathbf{A}$ are initially placed on the elements of an r-tuple $\boldsymbol{a}$ of elements in $\mathbf{A}$ and the corresponding r pebbles in $\mathbf{B}$ on an r-tuple $\boldsymbol{b}$ of elements in $\mathbf{B}$. The game is played by two players, Spoiler and Duplicator. If $\|\mathbf{A}\| \neq \|\mathbf{B}\|$ or the mapping defined by the initial pebble positions is not a partial isomorphism then Spoiler wins the game immediately. Otherwise, each round of the game proceeds as follows.

1. Spoiler picks up $2m$ pebbles in some order from $\mathbf{A}$ and the $2m$ corresponding pebbles in the same order from $\mathbf{B}$.
2. Duplicator has to respond by choosing
   - a partition $\mathbf{P}$ of $U(\mathbf{A})^m \times U(\mathbf{A})^m$,
   - a partition $\mathbf{Q}$ of $U(\mathbf{B})^m \times U(\mathbf{B})^m$, with $\|\mathbf{P}\| = \|\mathbf{Q}\|$, and
   - a bijection $f : \mathbf{P} \to \mathbf{Q}$,

   for which it holds that for all labellings $\gamma : \mathbf{P} \to \mathsf{GF}_p$,
   $$\operatorname{rk}(M^{\mathbf{P}}_{\gamma}) = \operatorname{rk}(M^{\mathbf{Q}}_{\gamma\circ f^{-1}}). \tag{$\star$}$$
   Here the composite map $\gamma \circ f^{-1} : \mathbf{Q} \to \mathsf{GF}_p$ is seen as a labelling of $\mathbf{Q}$.
3. Spoiler next picks a block $P \in \mathbf{P}$ and places the $2m$ chosen pebbles from $\mathbf{A}$ on the elements of some tuple in P (in the order they were chosen earlier) and the corresponding $2m$ pebbles from $\mathbf{B}$ on the elements of some tuple in $f(P)$ (in the same order).

This completes one round in the game. If, after this exchange, the partial map from $\mathbf{A}$ to $\mathbf{B}$ defined by the pebbled positions (in addition to constants) is not a partial isomorphism, or if Duplicator is unable to produce the required partitions, then Spoiler wins the game; otherwise it can continue for another round. Observe that the condition "$\operatorname{rk}(M^{\mathbf{P}}_{\gamma}) = \operatorname{rk}(M^{\mathbf{Q}}_{\gamma\circ f^{-1}})$" is the same as saying that the two matrices should be *equivalent*, since rank is a complete invariant for matrix equivalence. This explains the name of the game.

Theorem 1. *Duplicator has a winning strategy in the (k, m, p)-matrix-equivalence game starting with positions $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ if and only if $(\mathbf{A}, \boldsymbol{a}) \equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b})$.*

We give the proof of Theorem 1 by two separate lemmas, one for each implication. To simplify the proof, we consider only positions $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ with $\|\boldsymbol{a}\| = \|\boldsymbol{b}\| = k$; that is, positions where all the pebbles are initially placed on the board.

Lemma 2. *If $(\mathbf{A}, \boldsymbol{a}) \not\equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b})$ then Spoiler has a winning strategy in the (k, m, p)-matrix-equivalence game starting with positions $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$.*

Sketch proof. We show that if $(\mathbf{A}, \boldsymbol{a}) \not\equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b})$ then Spoiler has a strategy to force the game, in a finite number of rounds, into positions that are not partially isomorphic. Spoiler's strategy is obtained by structural induction on some formula $\varphi \in R^k_{m;p}$ on which the two game positions disagree; this argument broadly resembles similar proofs for the standard pebble games (see e.g. [14]). The main difficulty of the proof is to show that if Duplicator produces partitions $\mathbf{P}$ and $\mathbf{Q}$, then Spoiler can always find a block in one of the partitions that contains both tuples that satisfy φ and tuples that satisfy $\neg\varphi$. Once he has identified such a block, Spoiler can place his pebbles in a way that ensures that the resulting game positions disagree on a formula of quantifier rank less than φ. This gives him a strategy to win the game in a finite number of moves. $\square$

In the proof of the next lemma, we show that if $(\mathbf{A}, \boldsymbol{a}) \equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b})$, then Duplicator can play one round of the (k, m, p)-matrix-equivalence game in a way that ensures that the resulting positions will also be $\equiv^R_{k,m,p}$-equivalent. This gives her a strategy to play the game indefinitely. The idea here is to let Duplicator respond to a challenge of the Spoiler with partitions $\mathbf{P}$ and $\mathbf{Q}$ that are obtained by grouping together in each partition block all the elements realising the same $R^k_{m;p}$-type (with respect to the current game positions). The bijection $f : \mathbf{P} \to \mathbf{Q}$ is similarly defined by pairing together blocks whose elements all realise the same $R^k_{m;p}$-type. We show that if Duplicator plays in this manner, then she can ensure both that condition ($\star$) is met and that Spoiler is restricted to placing his pebbles in blocks which do not distinguish the two structures. We omit the details of the proof due to page limitations; full details are given in Appendix A.

Lemma 3. *If $(\mathbf{A}, \boldsymbol{a}) \equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b})$ then Duplicator has a winning strategy in the (k, m, p)-matrix-equivalence game starting with positions $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$.*

The next lemma shows that for all m and primes p, the equivalence $\equiv^R_{k+2m-1,m,p}$ refines $\equiv^C_k$ on the class of finite structures.

Lemma 4. *Duplicator has a winning strategy in the k-pebble bijection game on $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ if she has a winning strategy in the $(k + 2m - 1, m, p)$-matrix-equivalence game starting on $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$, for any prime p and $m \in \mathbb{N}$.*

It follows from the game characterisation of finite-variable counting logics that the equivalence $\equiv^C_k$ on finite structures is decidable in polynomial time. Essentially, this is because the number of possible moves (of each kind) for Duplicator at any particular stage of the game can be inductively combined into a structural invariant that completely characterises the game equivalence, and this invariant can be constructed in polynomial time (see Otto [14] for details). In light of Theorem 1, we may therefore ask whether the game model for $R^k_{m;p}$ can be used to give a similar result for equivalence in finite-variable rank logic; that is, whether we can decide $\equiv^R_{k,m,p}$ on finite structures in polynomial time. Unfortunately, unlike the case with the counting game, there does not seem to be an effective way to encode complete game information into a polynomial-size invariant. The main problem is the game condition ($\star$); this requires Duplicator to show that each pair of matrices $M^{\mathbf{P}}_{\gamma}$ and $M^{\mathbf{Q}}_{\gamma\circ f^{-1}}$ are equivalent (that is, have the same rank) but the number of these matrices is *exponential* in the size of the partition.

4 Playing with Invertible Linear Maps

Building on the game protocol described in the previous section, we define a new pebble game based on invertible linear maps. In this game, Duplicator is required to specify a bijection between the partitions of the two game structures as the conjugacy action of a single invertible matrix. In that sense, the invertible-map game can be seen as the natural extension of the bijection game for counting logics, where we replace bijections with invertible maps. In [5] it had been asked whether such a game might characterise definability in finite-variable rank logic. We show that equivalence in the invertible-map game does refine the relations $\equiv^R_{k,m,p}$ while it is not known whether the converse holds. We also establish that equivalence in the invertible-map game can be decided in polynomial time, which is not known to be true for the $\equiv^R_{k,m,p}$. We will see one application of the game equivalence in the next section, where we define algorithms for testing graph isomorphism by playing the invertible-map game on finite graphs.

4.1 Invertible-map Game

Let k, m and p be positive integers with $2m \leq k$ and p prime. The game board of the k-pebble m-ary *invertible-map game* over GF_p (or (k, m, p)-invertible-map game for short) consists of two structures $\mathbf{A}$ and $\mathbf{B}$ of the same vocabulary, each with k pebbles labelled $1, \ldots, k$ (and initial placement of pebbles $\boldsymbol{a}$ over $\mathbf{A}$ and $\boldsymbol{b}$ over $\mathbf{B}$, as before). If $\|\mathbf{A}\| \neq \|\mathbf{B}\|$ or the mapping defined by the initial pebble positions is not a partial isomorphism, then Spoiler wins the game immediately. Otherwise, each round of the game is played as follows.

1. Spoiler picks up $2m$ pebbles in some order from $\mathbf{A}$ and the $2m$ corresponding pebbles in the same order from $\mathbf{B}$.
2. Duplicator has to respond by choosing
   - a partition $\mathbf{P}$ of $U(\mathbf{A})^m \times U(\mathbf{A})^m$,
   - a partition $\mathbf{Q}$ of $U(\mathbf{B})^m \times U(\mathbf{B})^m$, with $\|\mathbf{P}\| = \|\mathbf{Q}\|$, and
   - a non-singular $U(\mathbf{B})^m \times U(\mathbf{A})^m$ matrix S over GF_p,

   for which it holds that the map $f : \mathbf{P} \to \mathbf{Q}$ defined by
   $$P \mapsto Q \quad \text{iff} \quad S \cdot \chi_P \cdot S^{-1} = \chi_Q \qquad (\circledast)$$
   is *total* and *bijective*, where we view χ_P and χ_Q as matrices over GF_p.
3. Spoiler next picks a block $P \in \mathbf{P}$ and places the $2m$ chosen pebbles from $\mathbf{A}$ on the elements of some tuple in P (in the order they were chosen earlier) and the corresponding $2m$ pebbles from $\mathbf{B}$ on the elements of some tuple in $f(P)$ (in the same order).

This completes one round in the game. If, after this exchange, the partial map from $\mathbf{A}$ to $\mathbf{B}$ defined by the pebbled positions is not a partial isomorphism, or if Duplicator is unable to produce the necessary triple $(\mathbf{P}, \mathbf{Q}, S)$, then Spoiler has won the game; otherwise it can continue for another round.

We write $(\mathbf{A}, \boldsymbol{b}) \approx^k_{m,p} (\mathbf{B}, \boldsymbol{b})$ to denote that Duplicator has a strategy to play forever in the (k, m, p)-invertible-map game with starting positions $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$. The following lemma shows that with increasing value of k or m, we get a decreasing chain of equivalence relations on $\mathrm{fin}[\tau; k]$. The first inclusion of the lemma is trivial but the second inclusion requires results on the block structure of those invertible maps that can be played as a part of a winning strategy.

Lemma 5. *For all $k, m \in \mathbb{N}$ where $2m \leq k$, it holds that $\approx^{k+1}_{m,p} \subseteq \approx^k_{m,p}$ and $\approx^k_{m+1,p} \subseteq \approx^k_{m,p}$ for all primes p.*

In the matrix-equivalence game, it is clearly sufficient for Duplicator to demonstrate the existence of a single similarity transformation that relates all linear combinations of partition matrices, since similar matrices have the same rank. Hence, we establish that $\approx^k_{m,p}$ refines $\equiv^R_{k,m,p}$ for all values of k, m and p.

Lemma 6. *Duplicator has a winning strategy in the (k, m, p)-matrix-equivalence game starting on $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ if she has a winning strategy in the (k, m, p)-invertible-map game starting on $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$.*

4.2 Complexity of the Game Equivalence

In this section we show that for each k and vocabulary τ, there is an algorithm that decides whether $(\mathbf{A}, \boldsymbol{a}) \approx^k_{m,p} (\mathbf{B}, \boldsymbol{b})$ in time polynomial in np, where n is the size of both $\mathbf{A}$ and $\mathbf{B}$.

To simplify our notation, fix k, m, p and a vocabulary τ. In order to analyse the structure of the game equivalence, we consider a stratification of $\approx^k_{m,p}$ by the number of rounds in the game. More specifically, we let $\sim_i$ be the binary relation on $\mathrm{fin}[\tau; k]$ defined by $(\mathbf{A}, \boldsymbol{a}) \sim_i (\mathbf{B}, \boldsymbol{b})$ if Duplicator has a strategy to play for up to i rounds in the (k, m, p)-invertible-map game on $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$. This relation can be characterised inductively as follows.

Lemma 7. *For all $(\mathbf{A}, \boldsymbol{a}), (\mathbf{B}, \boldsymbol{b}) \in \mathrm{fin}[\tau; k]$ we have*

$$(\mathbf{A}, \boldsymbol{a}) \sim_0 (\mathbf{B}, \boldsymbol{b}) \quad \textit{iff}\ \mathrm{atp}(\mathbf{A}, \boldsymbol{a}) = \mathrm{atp}(\mathbf{B}, \boldsymbol{b})$$

$$(\mathbf{A}, \boldsymbol{a}) \sim_{i+1} (\mathbf{B}, \boldsymbol{b})\ \textit{iff}\ (\mathbf{A}, \boldsymbol{a}) \sim_i (\mathbf{B}, \boldsymbol{b})\ \textit{and for all}\ \boldsymbol{j} \in [k]^{2m}\ \textit{with distinct}\ \underbrace{\begin{array}{l}\textit{values there is an invertible } U(\mathbf{B})^m \times U(\mathbf{A})^m \textit{ matrix}\\ S \textit{ over } \mathsf{GF}_p \textit{ such that for all } \alpha \in \mathrm{fin}[\tau; k]/\sim_i\textit{:}\\ S \cdot \mathrm{extmat}^{\alpha}_{\boldsymbol{j}}(\mathbf{A}, \boldsymbol{a}) \cdot S^{-1} = \mathrm{extmat}^{\alpha}_{\boldsymbol{j}}(\mathbf{B}, \boldsymbol{b}).\end{array}}_{(\star\star)}$$

For the proof of the inductive step of Lemma 7, the "if" direction is fairly straightforward (it essentially specifies a sufficient response for Duplicator in one round of the game). For the converse, it needs to be shown that any partition played by Duplicator as a part of an $(i+1)$-round winning strategy has to be a refinement of the partition of the corresponding structure into $\sim_i$-equivalence classes. Combining Lemma 7 with the fact that the matrix-similarity relation is transitive, we get the following result.

Corollary 8. *$\sim_i$ is an equivalence relation on $\mathrm{fin}[\tau; k]$ for each $i \in \mathbb{N}_0$.*

In particular, since $\approx^k_{m,p}$ coincides with the intersection of the $\sim_i$ over all i (with respect to k, m and p), it follows that $\approx^k_{m,p}$ is an equivalence relation on $\mathrm{fin}[\tau; k]$.

Now consider some $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ in $\mathrm{fin}[\tau; k]$ and assume that $\|A\| = \|B\| = n$. Since the number of distinct positions in the game starting with $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ is bounded by a polynomial in n, it follows that there is some polynomial $q(X)$ (depending only on k and τ) such that if Duplicator can play the game for at least $q(n)$ rounds, then she has a strategy to play forever. In other words, $(\mathbf{A}, \boldsymbol{a}) \approx^k_{m,p} (\mathbf{B}, \boldsymbol{b})$ if and only if $(\mathbf{A}, \boldsymbol{a}) \sim_{q(n)} (\mathbf{B}, \boldsymbol{b})$. To decide $(\mathbf{A}, \boldsymbol{a}) \approx^k_{m,p} (\mathbf{B}, \boldsymbol{b})$, we inductively construct the graph of $\approx^k_{m,p}$, restricted to $\mathbf{A}$ and $\mathbf{B}$, as follows. Initially, we partition the elements of $U(\mathbf{A})^k \,\dot\cup\, U(\mathbf{B})^k$ by their atomic equivalance, which is just $\sim_0$. For the induction step, suppose we have constructed $\sim_i$. Then to compute the refinement $\sim_{i+1}$, we consider each $\sim_i$-equivalent pair $(\boldsymbol{c}, \boldsymbol{d})$ and check whether condition $(\star\star)$ of Lemma 7 is satisfied. That is, for each tuple $\boldsymbol{j} \in [k]^{2m}$, we let $\mathcal{C} = (C_\alpha)$ and $\mathcal{D} = (D_\alpha)$ be the families of extension matrices defined by $\boldsymbol{j}$ over $\boldsymbol{c}$ and $\boldsymbol{d}$, respectively, indexed by all equivalence classes of $\sim_i$ (where $C_\alpha = \mathrm{extmat}^\alpha_{\boldsymbol{j}}(\mathbf{A}, \boldsymbol{c})$ if $\boldsymbol{c}$ is defined over $\mathbf{A}$, and similarly for D_α). Here it is important to note that it suffices to consider only equivalence classes of $\sim_i$ *restricted to* $\mathbf{A}$ *and* $\mathbf{B}$. Therefore, the number of extension matrices that we need to consider is bounded by a polynomial in n.

At this stage it remains to determine whether the pair of matrix tuples $\mathcal{C}$ and $\mathcal{D}$ are *simultaneously similar*: that is, whether there is a non-singular matrix S such that $S \cdot C_\alpha \cdot S^{-1} = D_\alpha$ for all types α realised over $\mathbf{A}$ and $\mathbf{B}$. By a result of Chistov et al. [3], this problem is in polynomial time over all finite fields.

Proposition 9 (Chistov, Karpinsky and Ivanyov). *There is a deterministic algorithm that, given two families of $n \times n$ matrices $\mathcal{C} = (C_1, \dots, C_l)$ and $\mathcal{D} = (D_1, \dots, D_l)$ over a finite field GF_q, determines in time $\mathrm{poly}(n, l, q)$ whether $\mathcal{C}$ and $\mathcal{D}$ are simultaneously similar.*

By our discussion above, it follows that we can construct the graph of $\approx^k_{m,p}$ restricted to $\mathbf{A}$ and $\mathbf{B}$ in a polynomial number of steps. At each step, we need to check a polynomial number of matrix tuples for simultaneous similarity, where each tuple has polynomial length. Combined with Proposition 9, this gives us a proof of the following theorem.

Theorem 10. *For each vocabulary τ there is a deterministic algorithm that, given $(\mathbf{A}, \boldsymbol{a}), (\mathbf{B}, \boldsymbol{b}) \in \mathrm{fin}[\tau; k]$ (with $\|\mathbf{A}\| = \|\mathbf{B}\| = n$) and $m, p \in \mathbb{N}$ (with $2m \leq k$ and p prime), decides whether $(\mathbf{A}, \boldsymbol{a}) \approx^k_{m,p} (\mathbf{B}, \boldsymbol{b})$ in time $(np)^{\mathcal{O}(k)}$.*

Observe that this implies that for each fixed k, we can decide $\approx^k_{m,p}$ in polynomial time, where the parameters m and p can be a part of the input.

5 Application to the Graph Isomorphism Problem

By considering the invertible-map game equivalence $\approx^k_{m,p}$ on the class of all finite graphs, we get a family of polynomial-time algorithms for stratifying the graph isomorphism relation. More specifically, for each k, m, p with p prime and $2m \leq k$, we write $\mathrm{IM}^k_{m,p}$ to denote the following algorithm on a pair of finite graphs $\mathbf{G}$ and $\mathbf{H}$:

> If $\|\mathbf{G}\| \neq \|\mathbf{H}\|$ then output "not isomorphic". Otherwise, compute the equivalence relation $\approx^k_{m,p}$ (restricted to $\mathbf{G}$ and $\mathbf{H}$) on the set $U(\mathbf{G})^k \,\dot\cup\, U(\mathbf{H})^k$ by applying the algorithm of Theorem 10 for all tuples in $U(\mathbf{G})^k \,\dot\cup\, U(\mathbf{H})^k$. If the result is that there is some equivalence class α of $\approx^k_{m,p}$ such that $\|\alpha \cap U(\mathbf{G})^k\| \neq \|\alpha \cap U(\mathbf{H})^k\|$ then output "not isomorphic"; otherwise, output "isomorphic".

It follows from Theorem 10 that $\mathrm{IM}^k_{m,p}$ runs in polynomial time for a fixed k. While the algorithm will always correctly identify isomorphic graphs, it may fail to distinguish between non-isomorphic instances (broadly speaking, this happens if the associated game equivalence groups together different orbits of the underlying graphs into a single equivalence class). Furthermore, it can be seen that for each pair of graphs, there is always some value of k for which $\mathrm{IM}^k_{m,p}$ correctly determines isomorphism.

The procedure for $\mathrm{IM}^k_{m,p}$ that we outlined above bears strong resemblance to the well-known Weisfeiler-Lehman method for graph isomorphism (see [2] for a description of the method). It was shown by Cai, Fürer and Immerman [2] that Duplicator has a winning strategy in the $(k+1)$-pebble bijection game on $\mathbf{G}$ and $\mathbf{H}$ if and only if $\mathbf{G}$ and $\mathbf{H}$ are not distinguished by the k-dimensional Weisfeiler-Lehman algorithm (WL^k). Combining this characterisation of the Weisfeiler-Lehman algorithm with lemmas 4 and 6, we have that

$$\begin{aligned} &\mathbf{G} \text{ and } \mathbf{H} \text{ are distinguished by } \mathrm{WL}^k \\ \Rightarrow\ &\mathbf{G} \text{ and } \mathbf{H} \text{ are distinguished by } \mathrm{IM}^{k+2m}_{m,p} \text{ for } \textit{all } m \text{ and prime } p. \end{aligned}$$

In [2], Cai et al. showed how to construct for each $k \in \mathbb{N}$ a pair of non-isomorphic graphs (named "CFI graphs") that are indistinguishable in WL^k. Later, it was shown by Dawar et al. [4] that there is a fixed sentence of first-order logic with rank operators over GF_2 that can distinguish between any pair of these CFI graphs. This construction was extended further by Holm [10], who showed that for any prime p, there are families of non-isomorphic graphs that can be distinguished by first-order logic with rank operators over GF_p but not by any fixed dimension of Weisfeiler-Lehman. Hence, it follows that for each prime p, the family of $\mathrm{IM}^k_{m,p}$ algorithms provide a way of stratifying the graph isomorphism relation which goes beyond that given by the Weisfeiler-Lehman algorithms.

Proposition 11. *For each prime p and $k \geq 1$, there is a pair of non-isomorphic graphs $\mathbf{G}$ and $\mathbf{H}$ that can be distinguished by $\mathrm{IM}^3_{1,p}$ but not by WL^k.*

Finally, we remark that Derksen [6] has recently described a different family of polynomial-time algorithms that also give an approximation to graph isomorphism that goes beyond that of the Weisfeiler-Lehman method. While Derksen's method partly builds on the simultaneous-similarity algorithm of Chistov et al. [3] (Proposition 9), it also draws heavily on techniques from algebraic geometry and category theory and seems very different from the game-based approach that we describe. Nevertheless, it is an open problem whether these two approaches are compatible.

6 Discussion

A natural question that is raised by the definitions of the games we have presented in this paper, is how to use them to establish inexpressibility results. A step in this direction is presented in [10] where it is shown that for any prime p, there is a property definable in FOR_p which is not closed under $\equiv^R_{k,1,q}$ for any k and any primes $q \neq p$. It would be interesting to lift this up to arities higher than 1, but playing the game poses combinatorial difficulties.

Another interesting direction would be to establish the precise relationship between the two games we consider. While we showed that the invertible-map game gives a refinement of the matrix-equivalence game (that is, a winning strategy for Duplicator in the former gives a winning strategy in the latter), it is not known whether this refinement is strict. Might it be the case that for any k and m one can find a k' and m' so that $\equiv^{R}_{k',m',p}$ is a refinement of $\approx^{k}_{m,p}$? One way this might be established is by showing that the relations $\approx^{k}_{m,p}$ are themselves definable in IFPR. If it turns out that this is not the case, then we would have established that there is a PTIME property not in in IFPR. A natural line of investigation would then be to extract from the invertible-map game a suitable logical operator, stronger than the matrix-rank operator, that is characterised by this game.

A more general direction of research that is suggested by this work is to explore other partition games which can be defined by suitable equivalence conditions on the partition matrices. There is space here for defining new logics and also new isomorphism tests.

Acknowledgements. This research was partially done whilst the authors were participants in the programme 'Semantics and Syntax' at the Isaac Newton Institute for the Mathematical Sciences in Cambridge, UK.

References

[1] J. Barwise. On Moschovakis closure ordinals. *Journal of Symbolic Logic*, 42:292–296, 1977.

[2] J-Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

[3] A. Chistov, G. Ivanyos, and M. Karpinski. Polynomial time algorithms for modules over finite dimensional algebras. In *Proc. 1997 international symposium on symbolic and algebraic computation*, pages 68–74. ACM, 1997.

[4] A. Dawar, M. Grohe, B. Holm, and B. Laubner. Logics with rank operators. In *Proc. 24th IEEE Symp. on Logic in Computer Science*, pages 113–122, 2009.

[5] A. Dawar and B. Holm. Pebble games for logics with counting and rank. In P. Cégielski, editor, *Studies in Weak Arithmetics*, pages 99–120. CSLI Publications, 2010.

[6] H. Derksen. The graph isomorphism problem and approximate categories. 2010. arXiv:1012.2081v1.

[7] H. D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1999.

[8] Y. Gurevich. Logic and the challenge of computer science. In E. Börger, editor, *Current Trends in Theoretical Computer Science*, pages 1–57. Computer Science Press, 1988.

[9] L. Hella. Logical hierarchies in PTIME. *Information and Computation*, 129:1—19, 1996.

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

[11] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[12] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. *Complexity Theory Retrospective: In Honor of Juris Hartmanis on the Occasion of His Sixtieth Birthday, July 5, 1988*, pages 59—81, 1990.

[13] B. Laubner. *The Structure of Graphs and New Logics for the Characterization of Polynomial Time*. PhD thesis, Humboldt-Universität zu Berlin, 2011.

[14] M. Otto. *Bounded Variable Logics and Counting — A Study in Finite Models*, volume 9 of *Lecture Notes in Logic*. Springer-Verlag, 1997.

[15] B. Poizat. Deux ou trois choses que je sais de $\mathrm{L}_{\mathtt{n}}$. *Journal of Symbolic Logic*, 47(3):641–658, 1982.

[16] M. Y. Vardi. The complexity of relational query languages. In *STOC '82: Proc. 14th ACM Symp. on Theory of Computing*, pages 137–146. ACM Press, 1982.

A Detailed proof of Theorem 1

We present a proof of Theorem 1 that includes many of the details omitted from Section 3. For convenience, we restate the theorem below.

Theorem 1. *Duplicator has a winning strategy in the (k, m, p)-matrix-equivalence game starting with positions $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ if and only if $(\mathbf{A}, \boldsymbol{a}) \equiv^{R}_{k,m,p} (\mathbf{B}, \boldsymbol{b})$.*

Before we can give the proof, we need to introduce some new notation. To simplify our notation (and the proof), we consider only positions $(\mathbf{A}, \boldsymbol{a})$ and $(\mathbf{B}, \boldsymbol{b})$ with $\|\boldsymbol{a}\| = \|\boldsymbol{b}\| = k$; that is, positions where all the pebbles are initially placed on the board. The argument for the case when the tuples $\boldsymbol{a}$ and $\boldsymbol{b}$ have length $r < k$ is exactly the same, except that one has to distinguish at every turn between game moves made during the first k rounds and game moves in the subsequent rounds[4]. This has the effect of making the proof non-uniform, without actually providing any new insight.

Now consider a tuple $\Phi = (\varphi_1, \ldots, \varphi_{p-1})$ of formulae in vocabulary τ and suppose that all the formulae occurring in Φ have free variables amongst the elements of the k-tuple $\boldsymbol{x}$. Let $\boldsymbol{i} = (i_1, \ldots, i_{2m})$ be a tuple of distinct integers, with $2m \leq k$. Then we write $\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p$ to denote the "formula matrix" over GF_p defined by

$$\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p := \sum_{i=1}^{p-1} i \cdot \mathrm{extmat}^{\varphi_i}_{\boldsymbol{i}}(\mathbf{A}, \boldsymbol{a}) \qquad (\mathrm{mod}\ p).$$

That is, $\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p$ is a linear combination of the $U(\mathbf{A})^m \times U(\mathbf{A})^m$ extension matrices of the formulae φ_i, with scalar coefficients defined by the position of each formula in the tuple Φ.

Lemma 12. *Suppose $(\mathbf{A}, \boldsymbol{a}) \equiv^{R}_{k,m,p} (\mathbf{B}, \boldsymbol{b})$, with $\boldsymbol{a}$ and $\boldsymbol{b}$ k-tuples of elements. Let $\boldsymbol{x}$ be a k-tuple of variables and suppose $2m \leq k$. Then for all $\varphi_1, \ldots, \varphi_{p-1} \in R^k_{m;p}$, with $\mathrm{free}(\varphi_i) \subseteq \boldsymbol{x}$, and all tuples $\boldsymbol{i} \in [k]^{2m}$ of distinct integers, it holds that*

$$\mathrm{rk}(\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p) = \mathrm{rk}(\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{B}, \boldsymbol{b})_p),$$

where the matrix rank is taken over GF_p and $\Phi := (\varphi_1, \ldots, \varphi_{p-1})$.

[4] Note that it is possible to obtain a proof for $\|\boldsymbol{a}\| = \|\boldsymbol{b}\| = r \in [k-1]$ as a direct corollary of the situation when $\|\boldsymbol{a}\| = \|\boldsymbol{b}\| = k$. In this case, given r-tuples $\boldsymbol{a}$ and $\boldsymbol{b}$, one would consider the game with pebble positions $\boldsymbol{a}'$ and $\boldsymbol{b}'$, where the k-tuple $\boldsymbol{a}'$ is obtained from $\boldsymbol{a}$ by adding $k-r$ copies of a_1 at the end of the tuple (simulating the case when $k-r+1$ pebbles are placed on element a_1) and similarly for $\boldsymbol{b}'$. Alternatively, one could consider a game board where the structures $\mathbf{A}$ and $\mathbf{B}$ are augmented with new vertices $\star_A$ and $\star_B$, respectively, totally disjoint from the rest of the structure. Here the idea is that a pebble placed on these special elements is to be treated as being off-the-board. This latter approach has the benefit of working for all $r \leq k$, including $r = 0$, without changing the proof in any other way. See for example Ebbinghaus and Flum [7] for an application of this idea.

Proof. Let $(\mathbf{A}, \boldsymbol{a}) \in \operatorname{fin}[\tau; k]$. Then for all tuples $\Phi = (\varphi_1, \dots, \varphi_{p-1})$ of $R^k_{m;p}$-formulae, with $\operatorname{free}(\varphi_i) \subseteq \boldsymbol{x}$, and all $\boldsymbol{i} \in [k]^{2m}$, the formula

$$\operatorname{rk}_p^{=l}((x_{i_1}, \dots, x_{i_m}), (x_{i_{m+1}}, \dots, x_{i_{2m}})) \,.\, (\varphi_1, \dots, \varphi_{p-1})$$

is in $\operatorname{tp}(R^k_{m;p}; \mathbf{A}, \boldsymbol{a})$ exactly for the number $l := \operatorname{rk}(\operatorname{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p)$. The statement of the lemma now follows by considering that

$$\operatorname{tp}(R^k_{m;p}; \mathbf{A}, \boldsymbol{a}) = \operatorname{tp}(R^k_{m;p}; \mathbf{B}, \boldsymbol{b}).$$

$\square$

We now give a detailed proof of lemmas 2 and 3, which together give the proof of Theorem 1. For convenience, we restate both lemmas below.

Lemma 2. *If* $(\mathbf{A}, \boldsymbol{a}) \not\equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b})$ *then Spoiler has a winning strategy in the* (k, m, p)*-matrix-equivalence game starting with positions* $(\mathbf{A}, \boldsymbol{a})$ *and* $(\mathbf{B}, \boldsymbol{b})$.

Proof. If $(\mathbf{A}, \boldsymbol{a}) \not\equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b})$ then there is a formula $\varphi(\boldsymbol{x}) \in R^k_{m;p}$ of quantifier rank $q \in \mathbb{N}$ such that $\mathbf{A} \models \varphi[\boldsymbol{a}]$ but $\mathbf{B} \models \neg\varphi[\boldsymbol{b}]$. If $q = 0$ then the mapping $\mathbf{A} \to \mathbf{B}$ defined by the pebbled elements $\boldsymbol{a} \mapsto \boldsymbol{b}$ is not a partial isomorphism and Spoiler has won the game. For the inductive step, suppose that $q > 0$. We show that Spoiler can in one round force the game into positions $(\mathbf{A}, \boldsymbol{a}')$ and $(\mathbf{B}, \boldsymbol{b}')$ where $(\mathbf{A}, \boldsymbol{a}')$ and $(\mathbf{B}, \boldsymbol{b}')$ can be distinguished by a formula of quantifier rank $q' < q$. This gives Spoiler a strategy to win the game in a finite number of steps, as claimed. To establish the claim, we can assume without loss of generality that φ is of the form

$$\operatorname{rk}_p^{=l}((x_{i_1}, \dots, x_{i_m}), (x_{i_{m+1}}, \dots, x_{i_{2m}})) \,.\, (\varphi_1, \dots, \varphi_{p-1})$$

for some $l \geq 0$. Other cases reduce to this one through the symmetry of the claim (we have an equivalence relation) or, if φ is a Boolean combination of formulas, by replacing φ by one of its components. Set $\boldsymbol{i} = (i_1, \dots, i_{2m})$ and $\Phi = (\varphi_1, \dots, \varphi_{p-1})$. Then by assumption on φ,

$$\operatorname{rk}(\operatorname{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p) \neq \operatorname{rk}(\operatorname{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{B}, \boldsymbol{b})_p). \tag{1}$$

Spoiler now starts the round by picking up pebbles with labels $i_1, \dots, i_{2m}$. Duplicator has to respond by choosing partitions $\mathbf{P}$, $\mathbf{Q}$ and a bijection $f : \mathbf{P} \to \mathbf{Q}$, which satisfy the requirements of the game. If Duplicator fails to properly respond to the challenge of Spoiler, then Spoiler wins the game immediately, so assume that $\mathbf{P}$, $\mathbf{Q}$ and f satisfy the rank condition ($\star$). Then the following claim shows that the partitions proposed by Duplicator must contain a block with tuples that disagree on at least one of the φ_i.

Claim. There is a block $P \in \mathbf{P}$ and tuples $\boldsymbol{c} \in P$ and $\boldsymbol{d} \in f(P)$ for which there is some formula φ_i in Φ such that

$$\mathbf{A} \models \varphi_i[\boldsymbol{a}\tfrac{c_1}{i_1} \cdots \tfrac{c_{2m}}{i_{2m}}] \text{ and } \mathbf{B} \models \neg\varphi_i[\boldsymbol{b}\tfrac{d_1}{i_1} \cdots \tfrac{d_{2m}}{i_{2m}}],$$

or vice versa.

Proof of claim. Suppose, towards a contradiction, that each block $P \in \mathbf{P}$ contains only tuples that all realise one or the other, φ_i or $\neg\varphi_i$, and all tuples in $f(P)$ realise the same (corresponding) formula, for each $i \in [p-1]$. Hence, the map $\iota : \mathbf{P} \to \wp([p-1])$ that associates with each $P \in \mathbf{P}$ the set of formulae in Φ that are realised by some (and hence all) tuples in P is well-defined. Note that for each $P \in \mathbf{P}$, the formulae

$$\bigwedge_{i \in \iota(P)} \varphi_i \text{ and } \bigwedge_{i \in [1,p-1] \setminus \iota(P)} \neg\varphi_i$$

are realised by all tuples in P. Now consider the matrix $\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p$ defined over $\mathbf{A}$. By assumption, we can find a labelling $\gamma : \mathbf{P} \to \mathsf{GF}_p$ such that

$$\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p = M^{\mathbf{P}}_{\gamma} \quad \text{and} \quad \mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{B}, \boldsymbol{b})_p = M^{\mathbf{Q}}_{\gamma \circ f^{-1}}.$$

For instance, γ can be defined by $\gamma(P) := \sum_{i \in \iota(P)} i \pmod p$ for each $P \in \mathbf{P}$ (as an element of GF_p). But

$$\mathrm{rk}(\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{A}, \boldsymbol{a})_p) \neq \mathrm{rk}(\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\Phi, \mathbf{B}, \boldsymbol{b})_p)$$

by (1), while $\mathrm{rk}(M^{\mathbf{P}}_{\gamma}) = \mathrm{rk}(M^{\mathbf{Q}}_{\gamma \circ f^{-1}})$ since we assumed that Duplicator's response satisfies condition ($\star$). Therefore, we have a contradiction. $\lhd$

Now Spoiler picks some block P that satisfies the statement of the claim. This allows him to place the chosen pebbles on elements $(c_1, \ldots, c_{2m}) \in P$ and $(d_1, \ldots, d_{2m}) \in f(P)$ such that the two structures, with the corresponding pebble placements, can be distinguished by a formula of quantifier rank $q' < q$. $\square$

Lemma 3. *If* $(\mathbf{A}, \boldsymbol{a}) \equiv^{R}_{k,m,p} (\mathbf{B}, \boldsymbol{b})$ *then Duplicator has a winning strategy in the* (k, m, p)*-matrix-equivalence game starting with positions* $(\mathbf{A}, \boldsymbol{a})$ *and* $(\mathbf{B}, \boldsymbol{b})$.

Proof. Let $\boldsymbol{x} = (x_1, \ldots, x_k)$ be a k-tuple of distinct variables and assume that $(\mathbf{A}, \boldsymbol{a}) \equiv^{R}_{k,m,p} (\mathbf{B}, \boldsymbol{b})$. Suppose that Spoiler begins a round of the game by picking up pebbles labelled $i_1, \ldots, i_{2m}$, in that sequence, and write $\boldsymbol{i} = (i_1, \ldots, i_{2m})$. Then we show that Duplicator can respond with partitions that satisfy condition ($\star$) and which ensure that all game positions that can result will be $\equiv^{R}_{k,m,p}$-equivalent. Firstly, for each $\alpha \in \mathrm{Tp}(R^k_{m;p}; \tau, k)$ let

$$P_\alpha := \{(c_1, \ldots, c_{2m}) \in U(\mathbf{A})^{2m} \mid \mathrm{tp}(R^k_{m;p}; \mathbf{A}, \boldsymbol{a}\tfrac{c_1}{i_1} \cdots \tfrac{c_{2m}}{i_{2m}}) = \alpha\} \text{ and}$$
$$Q_\alpha := \{(d_1, \ldots, d_{2m}) \in U(\mathbf{B})^{2m} \mid \mathrm{tp}(R^k_{m;p}; \mathbf{B}, \boldsymbol{b}\tfrac{d_1}{i_1} \cdots \tfrac{d_{2m}}{i_{2m}}) = \alpha\}.$$

That is, each P_α consists of all $2m$-tuples that, when used to replace elements of $\boldsymbol{a}$ according to the index pattern $\boldsymbol{i}$, results in a tuple whose type over $\mathbf{A}$ is α (and similarly for each Q_α). The strategy of Duplicator is now to respond with

$$\mathbf{P} := \{P_\alpha \mid \alpha \in \Gamma \text{ and } P_\alpha \neq \emptyset\} \text{ and } \mathbf{Q} := \{Q_\alpha \mid \alpha \in \Gamma \text{ and } Q_\alpha \neq \emptyset\},$$

where we let $\Gamma := \mathrm{Tp}(R^k_{m;p}; \tau, k)$. To pair the two partitions together, Duplicator gives a mapping f on $\mathbf{P}$ defined by $P_\alpha \mapsto Q_\alpha$ for all non-empty P_α. It should be clear that $\mathbf{P}$ and $\mathbf{Q}$ are partitions of $U(\mathbf{A})^m \times U(\mathbf{A})^m$ and $U(\mathbf{B})^m \times U(\mathbf{B})^m$, respectively (just observe that each tuple of elements realises only one type).

We now establish, through a series of claims, that $\mathbf{P}$, $\mathbf{Q}$ and f satisfy the requirements ($\star$) of the game; in particular, that f is a bijection $\mathbf{P} \to \mathbf{Q}$.

Claim. f is a bijection $\mathbf{P} \to \mathbf{Q}$.

Proof of claim. To prove this claim, it suffices to show that $P_\alpha = \emptyset \Leftrightarrow Q_\alpha = \emptyset$ for all $\alpha \in \mathrm{Tp}(R^k_{m;p}; \tau, k)$. Suppose, towards a contradiction, that there is some α such that P_α is empty while Q_α is not empty (the other case is equivalent, by symmetry of the claim). By the definition of P_α, we know that

$$\mathrm{tp}(R^k_{m;p}; \mathbf{A}, \boldsymbol{a}\tfrac{\boldsymbol{c}}{\boldsymbol{i}}) \neq \alpha$$

for all $\boldsymbol{c} \in U(\mathbf{A})^{2m}$. This means that for each such tuple $\boldsymbol{c}$, there is some formula $\psi_{\boldsymbol{c}} \in \alpha$ such that $(\mathbf{A}, \boldsymbol{a}\tfrac{\boldsymbol{c}}{\boldsymbol{i}}) \not\models \psi_{\boldsymbol{c}}$. Fix such a formula $\psi_{\boldsymbol{c}}$ for each $\boldsymbol{c}$ and let $\Psi_\alpha := \bigwedge_{\boldsymbol{c}} \psi_{\boldsymbol{c}}$; since Ψ_α is defined by conjunction over a finite set of formulae from α, it follows that $\Psi_\alpha \in \alpha$. Since $P_\alpha = \emptyset$ it follows that

$$\|\{\boldsymbol{c} \in U(\mathbf{A})^{2m} \mid (\mathbf{A}, \boldsymbol{a}\tfrac{\boldsymbol{c}}{\boldsymbol{i}}) \models \Psi_\alpha\}\| = 0. \tag{2}$$

This condition can be formalised in $R^k_{m;p}$ as the formula

$$\theta := \mathrm{rk}_p^{=0}((x_{i_1}, \ldots, x_{i_m}), (x_{i_{m+1}}, \ldots, x_{i_{2m}})) \,.\, (\Psi_\alpha),$$

which asserts that the number of distinct $2m$-tuples $\boldsymbol{x}$ that realise Ψ_α over $(\mathbf{A}, \boldsymbol{a})$ is nil (more directly, that the matrix defined by Ψ_α is the all-zeroes matrix). By (2), it follows that $(\mathbf{A}, \boldsymbol{a}) \models \theta$. However, we have $(\mathbf{B}, \boldsymbol{b}) \not\models \theta$ since

$$\|\{\boldsymbol{d} \in U(\mathbf{B})^{2m} \mid (\mathbf{B}, \boldsymbol{b}\tfrac{\boldsymbol{d}}{\boldsymbol{i}}) \models \Psi_\alpha\}\| > 0,$$

by the assumption $Q_\alpha \neq \emptyset$. Observing that $\theta \in R^k_{m;p}$, we conclude that

$$(\mathbf{A}, \boldsymbol{a}) \not\equiv^R_{k,m,p} (\mathbf{B}, \boldsymbol{b}),$$

which contradicts the assumption of the lemma. $\lhd$

Claim. For each α for which both neither P_α nor Q_α are empty, there is a formula $\varphi_\alpha \in R^k_{m;p}$ (depending on both $\mathbf{P}$ and $\mathbf{Q}$) which isolates P_α in $\mathbf{P}$ and Q_α in $\mathbf{Q}$. That is, for all $\boldsymbol{c} \in P_\alpha$ and $\boldsymbol{d} \in Q_\alpha$,

$$(\mathbf{A}, \boldsymbol{a}\tfrac{\boldsymbol{c}}{\boldsymbol{i}}) \models \varphi_\alpha \quad \text{and} \quad (\mathbf{B}, \boldsymbol{b}\tfrac{\boldsymbol{d}}{\boldsymbol{i}}) \models \varphi_\alpha$$

and for all $\boldsymbol{c} \notin P_\alpha$ and $\boldsymbol{d} \notin Q_\alpha$,

$$(\mathbf{A}, \boldsymbol{a}\tfrac{\boldsymbol{c}}{\boldsymbol{i}}) \not\models \varphi_\alpha \quad \text{and} \quad (\mathbf{B}, \boldsymbol{b}\tfrac{\boldsymbol{d}}{\boldsymbol{i}}) \not\models \varphi_\alpha.$$

Proof of claim. Let P_β be non-empty and let $\boldsymbol{c} \in P_\beta$. For each $\alpha \neq \beta$ for which $P_\alpha \neq \emptyset$, fix a formula $\psi_{\boldsymbol{c}} \in \alpha$ for which $(\mathbf{A}, \boldsymbol{a}\tfrac{\boldsymbol{c}}{\boldsymbol{i}}) \not\models \psi_{\boldsymbol{c}}$. Such a formula must exist, since $\boldsymbol{c}$ does not realise the type α over $(\mathbf{A}, \boldsymbol{a})$ with respect to the index pattern $\boldsymbol{i}$. Doing this for all tuples in P_β, we let

$$\Psi^{\mathbf{P}}_{\alpha,\beta} := \bigwedge_{\boldsymbol{c} \in P_\beta} \psi_{\boldsymbol{c}}.$$

By this construction, it follows that $\Psi^{\mathbf{P}}_{\alpha,\beta} \in \alpha$. Now fix a type α with $P_\alpha \neq \emptyset$ and let

$$\varphi^{\mathbf{P}}_\alpha := \bigwedge_{\alpha \neq \beta \text{ and } P_\beta \neq \emptyset} \Psi^{\mathbf{P}}_{\alpha,\beta}.$$

Observe that for all $\boldsymbol{c} \in P_\alpha$ with $P_\alpha \neq \emptyset$, we have $(\mathbf{A}, \boldsymbol{a}\frac{\boldsymbol{c}}{\boldsymbol{i}}) \models \varphi^{\mathbf{P}}_\alpha$ and $(\mathbf{A}, \boldsymbol{a}\frac{\boldsymbol{c}}{\boldsymbol{i}}) \not\models \varphi^{\mathbf{P}}_\beta$ for all $\beta \neq \alpha$. In other words, $\varphi^{\mathbf{P}}_\alpha$ isolates P_α in $\mathbf{P}$.

By the same construction, we obtain for each α a formula $\varphi^{\mathbf{Q}}_\alpha$ that isolates Q_α in $\mathbf{Q}$. Then it follows that the formula $\varphi_\alpha := \varphi^{\mathbf{P}}_\alpha \wedge \varphi^{\mathbf{Q}}_\alpha \in \alpha$ isolates both P_α in $\mathbf{P}$ and Q_α in $\mathbf{Q}$. $\lhd$

Claim. $\mathrm{rk}(M^{\mathbf{P}}_\gamma) = \mathrm{rk}(M^{\mathbf{Q}}_{\gamma \circ f^{-1}})$ for all $\gamma : \mathbf{P} \to \mathsf{GF}_p$.

Proof of claim. Without loss of generality, we associate GF_p with the set $[0, p-1]$ in the canonical way throughout this proof. With that assumption, let $\gamma : \mathbf{P} \to [0, p-1]$ be a labelling. From the definition of $\mathbf{P}$, it can be seen that the collection of blocks labelled $c \in [0, p-1]$ by γ corresponds to a finite set of types $\Omega_c \subseteq \mathrm{tp}(R^k_{m;p}; \mathbf{A}, \boldsymbol{a})$, with each type $\alpha \in \Omega_c$ isolated by a formula $\varphi_\alpha \in R^k_{m;p}$, by the previous claim. That is, for each type α it holds that

$$\alpha \in \Omega_c \Leftrightarrow \gamma(P_\alpha) = \gamma(\mathrm{ext}^{\varphi_\alpha}_{\boldsymbol{l}}(\mathbf{A}, \boldsymbol{a})) = c.$$

For $c \in [0, p-1]$, let $\psi_c := \bigvee_{\alpha \in \Omega_c} \varphi_\alpha \in R^k_{m;p}$. It can now be seen that

$$\begin{aligned} M^{\mathbf{P}}_\gamma &:= \sum_{c=0}^{p-1} c \cdot \big(\sum_{\alpha \in \Omega_c} \mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\varphi_\alpha, \mathbf{A}, \boldsymbol{a})_p \big) \\ &= \sum_{c=1}^{p-1} c \cdot \mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\psi_c, \mathbf{A}, \boldsymbol{a})_p \\ &= \mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\psi_1, \dots, \psi_{p-1}, \mathbf{A}, \boldsymbol{a})_p, \end{aligned}$$

and

$$\begin{aligned} M^{\mathbf{Q}}_{\gamma \circ f^{-1}} &:= \sum_{c=1}^{p-1} c \cdot \big(\sum_{\alpha \in \Omega_c} \mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\varphi_\alpha, \mathbf{B}, \boldsymbol{b})_p \big) \\ &= \sum_{c=1}^{p-1} c \cdot \mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\psi_c, \mathbf{B}, \boldsymbol{b})_p \\ &= \mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\psi_1, \dots, \psi_{p-1}, \mathbf{B}, \boldsymbol{b})_p. \end{aligned}$$

By Lemma 12 we know that

$$\mathrm{rk}\big(\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\psi_1, \dots, \psi_{p-1}, \mathbf{A}, \boldsymbol{a})_p\big) = \mathrm{rk}\big(\mathrm{fmat}_{\boldsymbol{x},\boldsymbol{i}}(\psi_1, \dots, \psi_{p-1}, \mathbf{B}, \boldsymbol{b})_p\big).$$

Hence, $\mathrm{rk}(M^{\mathbf{P}}_\gamma) = \mathrm{rk}(M^{\mathbf{Q}}_{\gamma \circ f^{-1}})$ over GF_p, as required. $\lhd$

By the three preceding claims, it can be seen that for any block $P \in \mathbf{P}$, any choice of elements $(c_1, \dots, c_{2m}) \in P$ and $(d_1, \dots, d_{2m}) \in f(P)$ that Spoiler can make will result in tuples

$$\boldsymbol{a}\frac{c_1}{i_1} \cdots \frac{c_{2m}}{i_{2m}} \text{ and } \boldsymbol{b}\frac{d_1}{i_1} \cdots \frac{d_{2m}}{i_{2m}}$$

that realise the same $R^k_{m;p}$-type. Hence, $\equiv^R_{k,m,p}$-equivalence of game positions is maintained. This completes the proof of the lemma. □

B Detailed proof of Theorem 10

We give a proof of Theorem 10 that was omitted from Section 4. For convenience, we restate the theorem below.

Theorem 10. *For each vocabulary τ there is a deterministic algorithm that, given $(\mathbf{A}, \boldsymbol{a}), (\mathbf{B}, \boldsymbol{b}) \in \operatorname{fin}[\tau; k]$ (with $\|\mathbf{A}\| = \|\mathbf{B}\| = n$) and $m, p \in \mathbb{N}$ (with $2m \leq k$ and p prime), decides whether $(\mathbf{A}, \boldsymbol{a}) \approx^k_{m,p} (\mathbf{B}, \boldsymbol{b})$ in time $(np)^{\mathcal{O}(k)}$.*

Proof. Write $V_A := U(\mathbf{A})^k$, $V_B := U(\mathbf{B})^k$ and $V := V_A \dot\cup V_B$. We inductively refine the graph of the relation $\approx_{\mathbf{AB}}$, which is obtained by restricting $\approx^k_{p,m}$ to positions over $\mathbf{A}$ and $\mathbf{B}$:

$$\begin{aligned}\approx_{\mathbf{AB}} := \{(\boldsymbol{a}', \boldsymbol{a}'') \in V_A \times V_A \mid (\mathbf{A}, \boldsymbol{a}') \approx^k_{p,m} (\mathbf{A}, \boldsymbol{a}'')\} \;\cup \\ \{(\boldsymbol{a}', \boldsymbol{b}') \in V_B \times V_B \mid (\mathbf{A}, \boldsymbol{a}') \approx^k_{p,m} (\mathbf{B}, \boldsymbol{b}')\} \;\cup \\ \{(\boldsymbol{b}', \boldsymbol{b}'') \in V_A \times V_B \mid (\mathbf{B}, \boldsymbol{b}') \approx^k_{p,m} (\mathbf{B}, \boldsymbol{b}'')\}.\end{aligned}$$

The graph we inductively construct has two types of edges: *blue edges* on $V_A \times V_A$ and $V_B \times V_B$ (denoting intra-structure equivalence) and *red edges* on $V_A \times V_B \cup V_B \times V_A$ (denoting cross-structure equivalence). Initially, we add an edge between two tuples in V if they satisfy the same atomic type; this edge is coloured blue if the two tuples belong to the same structure and red otherwise. Let $G_0 = (V, B_0, R_0)$ denote the graph obtained in this way, where B_0 and R_0 denote the blue and red edge relations, respectively. Observe that each connected component of G_0 restricted to blue edges forms a clique, corresponding to the atomic-equivalence classes on the respective structures. We say that two such connected components C_1 and C_2 are *paired* if $R_0 \cap (C_1 \cup C_2) \neq \emptyset$. It necessarily follows from transitivity of atomic equivalence that if C_1 and C_2 are paired, then the graph induced by $C_1 \cup C_2$ on the red edges is a complete bipartite graph. Fix an enumeration $\alpha_1, \dots, \alpha_l$ of those connected components of (V_A, B_i) that are paired with some component on V_B and let β_i denote the component in (V_B, B_i) that is paired with α_i. Let X_A be the collection of connected components in (V_A, B_i) not included in the enumeration and similarly X_B for (V_B, B_i). Then we construct G_{i+1} as follows:

Blue edges. Let $\{\boldsymbol{c}, \boldsymbol{d}\} \subseteq V_A$ be a blue edge in B_i. Then we include $\{\boldsymbol{c}, \boldsymbol{d}\}$ in B_{i+1} if and only if

- for all $\alpha \in X_A$: $\operatorname{ext}^\alpha_{\boldsymbol{i}}(\mathbf{A}, \boldsymbol{c}) = \operatorname{ext}^\alpha_{\boldsymbol{i}}(\mathbf{A}, \boldsymbol{d}) = \emptyset$ for all $\boldsymbol{i} \in [k]^{2m}$; **and**
- for all $\boldsymbol{i} \in [k]^{2m}$, the matrix families $(C_1, \dots, C_l)$ and $(D_1, \dots, D_l)$ are simultaneously similar, where $C_i = \operatorname{extmat}^{\alpha_i}_{\boldsymbol{i}}(\mathbf{A}, \boldsymbol{c})$ and $D_i = \operatorname{extmat}^{\alpha_i}_{\boldsymbol{i}}(\mathbf{A}, \boldsymbol{d})$.

Blue edges $\{\boldsymbol{c}, \boldsymbol{d}\} \subseteq V_B$ are considered similarly.

Red edges. To decide which red edges of R_i to retain for R_{i+1}, we follow a procedure similar to that given for blue edges; the difference this time is that we compare matrices across the two structures, $\mathbf{A}$ and $\mathbf{B}$.

After all edges in G_i have been inspected in this way, we have constructed a graph $G_{i+1} = (V, B_{i+1}, R_{i+1})$. By applying transitivity of the matrix-similarity relation, it can be seen that each connected component in (V, B_{i+1}) forms a complete graph and that all paired classes in G_{i+1} form a complete bipartite graph. This completes the inductive step of the construction.

By Lemma 7, it can be seen that at each stage i, G_i is the graph of the i-round game relation $\sim_i$ restricted to $\mathbf{A}$ and $\mathbf{B}$. Hence, there is some $i \leq 2n^k$ such that G_i is the graph of the restricted relation $\approx_{\mathbf{AB}}$, which we set out to compute. The algorithm can detect when this stage has been reached by checking whether $B_i \backslash B_{i+1} = R_i \backslash R_{i+1} = \emptyset$ at the end of each refinement step. Finally, the algorithm outputs "yes" if there is a red edge between $\boldsymbol{a}$ and $\boldsymbol{b}$ in G_i and outputs "no" otherwise.

At each stage i there are at most $2n^k$ blue-edge connected components of G_i. Thus, each refinement step takes time polynomial in npm where the biggest contribution is the polynomial-time algorithm of Proposition 9 for testing simultaneous similarity of matrix families (of length at most n^k). The number of refinement steps is polynomial in n, as noted above, which gives us an overall running time that is bounded by $(npm)^{\mathcal{O}(k)}$ which is $(np)^{\mathcal{O}(k)}$, since $2m \leq k$ □