BJARKI HOLM

DESCRIPTIVE COMPLEXITY OF LINEAR ALGEBRA

DISSERATION SUBMITTED FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF CAMBRIDGE

MAGDALENE COLLEGE

2010

To the memory of my mother

Declaration

This dissertation is the result of my own work under the supervision of Professor Anuj Dawar of the University of Cambridge Computer Laboratory. It includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

This dissertation does not exceed the regulation length of 60,000 words, including tables and footnotes.

Summary

An important open question that has motivated a lot of work in finite model theory is that of finding a logical characterisation of polynomial-time computability (PTIME). Most attempts to answer this question have focused on finding suitable extensions of first-order logic that can describe exactly all properties decidable in PTIME. In this way, Immerman and Vardi independently showed that on inputs equipped with a linear order, inflationary fixed-point logic (IFP) expresses exactly the properties in PTIME. In the absence of an order, IFP is too weak to express all properties in PTIME. In particular, it fails to define very simple cardinality properties. This is easily solved by extending the logic with counting terms, which gives us inflationary fixed-point logic with counting (IFPC), which was at one time conjectured to be a logic for PTIME. However, Cai, Fürer and Immerman later showed that this logic still falls short of capturing PTIME. Since this result, a number of examples have been constructed of polynomial-time decidable properties that are not expressible in IFPC. Most recently, it was shown that the problem of determining the solvability of affine equations over any fixed finite Abelian group is not definable in this logic. In particular, this implies that over finite fields IFPC is not able to express *matrix rank*.

To address this deficiency, we define an extension of IFP by operators for expressing the rank of definable matrix relations over finite fields. We show that the resulting logic IFPR is strictly more expressive than IFPC. In fact, we show that an even weaker logic, the extension of first-order logic with rank operators (FOR), can already define many of the properties used to separate IFPC from PTIME, such as solvability of linear equations and the property defined by Cai et al. Over the class of ordered structures, we characterise the descriptive complexity of first-order rank logics and show that they correspond to natural logspace complexity classes. Moreover, we show that the rank logics FOR and IFPR have a strict arity hierarchy, where the arity of a rank operator is the number of distinct variables that it binds.

We also study the extent to which IFPC can express linear algebra. We show that IFPC can define the characteristic polynomial (and hence determinant) of any matrix over a finite field, over the ring of integers or over the field of rational numbers. Moreover, we show that for rational-valued matrices, IFPC can already define the rank and the minimal polynomial. It is therefore seen that the additional expressive power of the logic IFPR comes specifically from the ability to define matrix rank over *finite fields*.

Finally, we show that equivalence in logics with rank operators can be characterised in terms of pebble games based on set partitions. This gives us a game-based method for proving lower bounds for FOR and IFPR. As an illustration of the game method, we establish that over finite structures, $\mathrm{IFPR}_{p;2} \not\equiv \mathrm{IFPR}_{q;2}$ for distinct primes p and q, where $\mathrm{IFPR}_{p;m}$ is the restriction of IFPR that only has operators for defining rank of matrices of arity at most m over the finite field GF_p.

Acknowledgements

Any success I have had as a doctoral student owes credit to my supervisor, Anuj Dawar. Without his insight, encouragement and seemingly unlimited patience, none of the work I present here would have been possible. I am extremely thankful for all the support and guidance he has given me over the past four years and I look forward to our future collaboration.

I have also had the privilege to enjoy close collaboration with Martin Grohe and Bastian Laubner at Humboldt University. My time spent working with Bastian in Berlin, and his two visits to Cambridge, were some of the most fruitful and enjoyable times of my doctoral studies. Hopefully some day we will get the chance to work together again.

To my office mates past and present, Yuguo He, Christopher Thompson-Walsh and Timos Antonopoulos, I am thankful for all the lively discussions (and debates) we have had throughout the years. To Yuguo, especially, I am grateful for the time we have spent together at the Computer Laboratory. We have learned a lot together over the past four years. I also enjoyed my close collaboration with Cameron Hill, who visited the Computer Laboratory for a few months last year. Perhaps one day I can repay that visit.

I would like to thank the Engineering and Physical Sciences Research Council, the University of Cambridge Computer Laboratory, Magdalene College and the Cambridge Philosophical Society for their financial support. Without their generosity, none of this would have been possible.

Finally, I owe a debt of gratitude to my wife, Bryndís, for encouraging me to go down this path and sticking with me throughout the journey. Never once did I hear her complain about my long hours toiling away in the office, trying to get this thesis delivered, even though she had a big delivery to make herself. *Takk ástin mín*! In the end she narrowly beat me on the delivery date and special thanks go to my two-week old son Krummi, whose imminent arrival put the pressure on me to get the writing done in time.

Contents

Chapter 1

Introduction

Computational complexity theory is the programme of classifying computational problems based on how difficult they are to solve. In this context, complexity is measured by the amount of resources required by a computation, such as running time, memory, number of processors, and other measurable quantities of the computational model. From this study one formally defines classes of problems of related complexity such as PTIME, the collection of all decision problems that can be solved by deterministic polynomial-time algorithms. An alternative way of analysing the difficulty of solving computational problems is to apply techniques from logic and finite model theory. In the study of descriptive complexity, instead of considering the difficulty of *deciding* whether an input possesses a property, one studies the richness of the least logic needed to *define* that property. These two measures of complexity—the hardness of computation versus logical expressibility—turn out, in many cases, to be equivalent.

The study of descriptive complexity was essentially initiated by the work of Fagin [26], who showed that a class of finite relational structures is decidable in non-deterministic polynomial time (NP) if and only if it is definable in the existential fragment of second-order logic. This naturally raises the question whether there is a similar logical characterisation of PTIME. Specifically, is there 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? This question is still wide open and is considered to be one of the main open problems in both finite model theory and database theory.

1.1 Is there a logic for PTIME?

The question whether there is a logic that captures polynomial time was first raised by Chandra and Harel [13] in the context of database theory and later reformulated by Gurevich [35] who also stated the conjecture that no such logic exists. It asks for a logic, satisfying some basic technical requirements, in which precisely those properties of finite structures which are decidable in polynomial time are definable. The details of these technical requirements are not important in this context; essentially, the idea is to rule out the possibility of taking an arbitrary collection of properties (for instance, the set of all polynomial-time decidable properties) and letting that constitute a logic.

The programme of seeking a logic for PTIME is of fundamental theoretical significance, as it aims to characterise the structure of both logics and complexity classes. However, much

of the research in this area has also been motivated by applications in database theory. This is because a concrete logical characterisation of polynomial time would give rise to a database query language which could express precisely all the feasible database queries (that is, queries decidable in polynomial time). This was partly the motivation behind the work of Chandra and Harel and research in this area remains active to date (see for instance Nash, Remmel and Vianu [57] for new insights).

Most attempts to construct a logic for PTIME have focused on finding suitable extensions of first-order logic. It is easy to show that every class of structures defined by a first-order sentence is decidable in polynomial time. Similarly, it is not hard to show that there are polynomial-time properties that are not definable in this way. Specifically, it can be shown that first-order logic lacks the ability to express any non-trivial property based on *recursion*, such as transitive closure. Thus, a logic to capture PTIME must extend the expressive power of first-order logic with the power to define polynomial-time inductive properties.

Inflationary fixed-point logic (IFP) is a logic that combines first-order logic with the ability to formalise inductive definitions. By a result proved independently by Immerman [44] and Vardi [65], it is known that this logic expresses exactly the polynomial-time properties of *ordered* finite structures. Here, an ordered structure is a structure whose signature contains a special binary relation symbol $\leq$ that is interpreted as a total linear ordering of the underlying domain. Despite this result, IFP is too weak in the absence of ordering to express all polynomial-time properties. In particular, it fails to define very simple cardinality properties, such as whether the domain of a structure has an even number of elements. Clearly this property of "evenness" is decidable in polynomial time by a simple counting procedure.

Various attempts have been made to extend fixed-point logic with new arithmetical or logical features, in the hope of finding a logic which captures PTIME on all finite structures. In [44, 45], Immerman suggested adding a mechanism for *counting* to the logic IFP. Counting, apart from being a fundamental operation in numerous algorithms, exemplifies the limitations of fixed-point logic, as mentioned above. The resulting logic, inflationary fixed point logic with counting (IFPC), has been intensively studied over a number of years [58] and was at one time conjectured by Immerman to be a logic for PTIME (for further details, see [12]). This logic has been shown to capture polynomial time on many natural classes of structures, including planar graphs and structures of bounded tree-width [30, 32, 34, 46]. Most recently, it was shown by Grohe [33] that IFPC captures polynomial time on all classes of graphs with excluded minors, a result that generalises many of the previous partial capturing results. Furthermore, it can be shown that IFPC captures polynomial time on "almost all" finite structures in a precise technical sense [38].

Despite the many promising results, Immerman's conjecture was ultimately refuted by Cai, Fürer and Immerman [12], who constructed a query on a class of finite graphs that can be defined by a polynomial-time computation but not by any sentence of IFPC. Since then, other constructions that expose the limitations of IFPC have been given. Gurevich and Shelah [36] defined a class of finite rigid structures known as *multipedes*, and considered the task of uniformly defining a linear order over this class. They showed that this task, while computable in polynomial time, is not expressible by any fixed formula of IFPC. Blass, Gurevich and Shelah [9] later turned this construction into a decision problem and proved that IFPC is not able to tell whether two given multipedes (each with a designated vertex) are isomorphic or not; a problem which again is decidable in polynomial time.

Even though the work of Cai et al. shows that IFPC falls short of capturing PTIME on all finite structures, it can be argued that the graph query used in their construction is very artificial. The same can be said of the examples constructed by Gurevich and Shelah. Therefore, it was often remarked that possibly all *natural* polynomial-time properties of finite structures were still definable in IFPC. Recently, however, it was shown by Atserias, Bulatov and Dawar [4] that solvability of affine equations over any fixed finite Abelian group is not definable by any sentence of IFPC. In particular, this implies that IFPC is not expressive enough to define solvability of systems of linear equations over a fixed *finite field*; a problem which is easily decidable in polynomial time by Gaussian elimination. This gives an example of a natural problem in PTIME that is not expressible in IFPC.

1.2 The importance of linear algebra

In recent years, various studies have pointed to the importance of linear algebra over finite fields in marking the boundaries of logically-defined fragments of polynomial time. In [9], Blass, Gurevich and Shelah studied the problem of determining whether a square matrix has determinant zero (that is, determining whether or not a matrix is singular). They showed that for matrices over finite fields, this problem can be defined in IFPC but not in IFP. It was later observed by Rossman [7] that the actual value of the determinant of a matrix over a commutative ring of characteristic zero can be defined in the language of choiceless polynomial time with counting, which is another logic that has been studied as a candidate for capturing PTIME and subsumes IFPC. Blass and Gurevich [7] used this observation to show that the same logic can also express the determinant of any matrix over a finite field.

Another important problem in linear algebra is the problem of determining the solvability of a system of linear equations. Atserias, Bulatov and Dawar [4] showed that the solvability of affine equations over any fixed finite Abelian group is not definable in IFPC, as mentioned above. In particular, this shows that IFPC is not expressive enough to define solvability of linear equations over a fixed finite field. Recall that by elementary linear algebra, a system of linear equations $A\mathbf{x} = \mathbf{b}$ over a field is solvable if and only if $\mathrm{rank}(A \mid \mathbf{b}) = \mathrm{rank}(A)$, where $(A \mid \mathbf{b})$ is the matrix obtained from A by adding the column vector $\mathbf{b}$ on the right. This immediately shows that IFPC is not expressive enough to define the *rank* of a matrix over a finite field.

Of course, it can be seen that for all the linear-algebraic problems mentioned — determining solvability of linear equations, computing determinant or computing rank — there are well-known polynomial-time algorithms, such as Gaussian elimination. Since fixed-point logic is known to capture PTIME on the class of finite ordered structures, it follows that IFP and IFPC can already define each of these problems when given matrices indexed by ordered sets. Therefore, our interest is specifically in *unordered matrices*, whose rows and columns are indexed by arbitrary unordered sets.

More formally, we can view an unordered $I \times J$ matrix M over a field F as a function $M : I \times J \to F$, where I and J are finite non-empty sets that index the set of rows and the set of columns of M, respectively. By taking $I = \{1, \ldots, m\}$ and $J = \{1, \ldots, n\}$ we obtain the more familiar notion of an $m \times n$ matrix; that is, a rectangular array of scalar values from F, with m rows and n columns that are ordered by the natural ordering of the integers. Most natural matrix properties from linear algebra, such as determinant and rank, are invariant under simultaneous permutation of the rows and columns of the matrix and are therefore

well-defined for matrices indexed by unordered sets. This is because these matrix properties are in fact properties of the underlying linear map that the matrix represents and the linear map is invariant under a permutation of the chosen vector space bases.

1.3 Contributions of this thesis

The results mentioned above illustrate that IFPC is unable to express some quite natural linear-algebraic properties, such as solvability of linear equations over a finite field. This suggests that problems in linear algebra might be a possible source of new extensions to fixed-point logic, in an attempt to find a logical characterisation of PTIME. In this thesis we follow this line of inquiry by systematically studying the descriptive complexity of various polynomial-time problems in linear algebra.

The main body of this thesis consists of six chapters. After reviewing some preliminaries in Chapter 2 we consider in Chapter 3 the extent to which IFPC can express linear algebra. By the work of Atserias et al. [4] we know that the rank of matrices over finite fields is not definable by any sentence of IFPC. However, we show that many other natural matrix properties are in fact definable in this logic. Specifically, we show that IFPC can define the characteristic polynomial (and hence determinant) of any matrix over a finite field, over the ring of integers or over the field of rational numbers. Moreover, we show that for rational-valued matrices, IFPC can already define the rank and the minimal polynomial.

These results establish that the inability to define *matrix rank over finite fields* is a fundamental barrier that separates IFPC from PTIME, just like the inability to count is a fundamental property that separates IFP from PTIME. In fact, computing rank can be understood as a generalised form of counting which counts the dimension of a definable vector space rather than the cardinality of a definable set. This suggests that the key weakness of IFPC is that the form of counting it incorporates is too weak. To address this deficiency, we define in Chapter 4 an extension of inflationary fixed-point logic by operators for expressing the rank of definable matrix relations over finite fields of prime cardinality. The resulting logic IFPR is at least as expressive as IFPC. This is because counting can be simulated by rank operators using the observation that the rank of a diagonal matrix is precisely the number of non-zero entries along the main diagonal. Furthermore, we show that IFPR can define solvability of linear equations over *any* finite field. Together with the fact that IFPR has polynomial-time data complexity (that is, the queries it defines can be decided in PTIME), we establish that IFPC $\lneqq$ IFPR $\leqq$ PTIME. This illustrates that IFPR is a candidate logic for PTIME. Finally, we show that rank logics have a strict arity hierarchy, where the arity of a rank operator is the number of distinct variables that it binds. This contrasts IFPR with the counting logic IFPC, for which it can be shown that unary counting operators suffice to define counting in any arity [23]. For instance, a counting term $\#_{xy}\varphi(x, y)$, expressing the number of pairs that satisfy the formula $\varphi(x, y)$, is equivalent to the term

$$\sum_x \#_y \varphi(x, y),$$

which can be defined by a formula of IFPC.

To understand the inherent expressive power of rank operators, we study such operators in the presence of weaker logics in Chapter 5. First-order logic with rank (FOR) is the extension of first-order logic by rank operators over finite fields of prime cardinality. Despite

lacking the ability to formalise inductive definitions, it turns out that this logic is quite expressive. In particular, we show that two of the examples showing that IFPC $\lneqq$ PTIME—the problem of deciding the graph query of Cai, Fürer and Immerman and the problem of deciding isomorphism of multipedes—are both definable in FOR. This result illustrates that these two examples are really just clever ways of encoding systems of linear equations into complex combinatorial structures. We also consider the descriptive complexity of first-order rank logics over ordered finite structures by proving that for each prime p, FOR_p captures $\mathrm{MOD}_p\mathrm{L}$ and that $\mathrm{FOR}_{\mathbb{Q}}$ captures $\mathrm{L}^{\mathrm{C}=\mathrm{L}}$, which are natural complexity classes that characterise different levels of logarithmic space complexity. Here FOR_p is the fragment of FOR that only has rank operators over the prime field GF_p and $\mathrm{FOR}_{\mathbb{Q}}$ is the extension of first-order logic by rank operators for expressing the rank of rational-valued matrices.

In Chapter 6 we develop Ehrenfeucht-Fraïssé-style pebble games for rank logics, which gives us a game-based method for proving inexpressibility results for FOR and IFPR. The game protocol that we introduce is based on partitioning the game board into a number of disjoint regions, according to some linear-algebraic criteria, which then limits the possible placement of game tokens on the board. This method of partitioning the game board turns out to be quite flexible and can be used to give a game description of a very generic family of logics, as we illustrate.

In Chapter 7 we establish the first inexpressibility results for rank logics. Writing $\mathrm{IFPR}_{p;m}$ to denote the fragment of IFPR restricted to rank operators of arity at most m over GF_p, with p prime, we show that for all distinct primes p and q, $\mathrm{IFPR}_{p;2} \not\equiv \mathrm{IFPR}_{q;2}$ over finite structures. The proof of this result combines linear algebra with an application of the game method developed earlier, played on a pair of highly symmetric combinatorial structures.

Finally, we conclude our discussion in Chapter 8 by summarising our major results and highlighting some of the open problems and future work in this area.

1.4 Previously published work and collaborations

The work in several chapters of this thesis was done in collaboration and we conclude this introduction by acknowledging these contributions.

The definition and initial study of rank logics arose through collaboration with Bastian Laubner, Anuj Dawar and Martin Grohe, and was presented at the 24th IEEE Symposium on Logic in Computer Science [17]. Parts of this work appear in Chapter 3 (definability of the characteristic polynomial over $\mathbb{Z}$, $\mathbb{Q}$ and prime fields; definability of matrix rank over $\mathbb{Q}$), Chapter 4 (definition of rank operators and rank logics; solvability of linear equations given by terms or formulae; definability of deterministic and symmetric transitive closure operators; arity hierarchy of rank logics for characteristic two) and Chapter 5 (definability of the CFI graph query; capturing result for $\mathrm{MOD}_p\mathrm{L}$ on ordered structures).

An introduction to the rank-partition games was presented jointly with Anuj Dawar at a workshop on Logical Approaches to Barriers in Computing and Complexity [18] and a more general overview was also included in a chapter in *Studies in Weak Arithmetics* [19], which was also co-authored with Anuj Dawar.

Chapter 2

Definitions and preliminaries

In this chapter we provide the necessary background from mathematical logic, matrix theory, algebra, and complexity theory. Apart from §2.8, where we introduce matrices indexed by unordered sets, all the definitions we consider are standard. Readers familiar with the material might therefore only want to refer back to individual definitions at a later point.

2.1 Basic notation

We write $\mathbb{N}$ and $\mathbb{N}_0$ for the positive and non-negative integers, respectively. For $m, n \in \mathbb{N}_0$, let $[m, n] := \{l \in \mathbb{N}_0 \mid m \leq l \leq n\}$ and $[n] := [1, n]$. We often denote tuples $(v_1, \ldots, v_k)$ by $\vec{v}$ and denote their length by $\|\vec{v}\|$. It is assumed that the components of a k-tuple are indexed from $1, \ldots, k$. If $\vec{v} = (v_1, \ldots, v_k)$ is a k-tuple of elements from a set X, $i \in [k]$ and $w \in X$, then we write $\vec{v}\frac{w}{i}$ for the tuple obtained from $\vec{v}$ by replacing the i-th component with w; that is, $\vec{v}\frac{w}{i} = (v_1, \ldots, v_{i-1}, w, v_{i+1}, \ldots, v_k)$. This notation can be extended to describe the replacement of more than one component, so for instance $\vec{v}\frac{w}{i}\frac{u}{j} := (\vec{v}\frac{w}{i})\frac{u}{j}$ for $u, w \in X$ and $i, j \in [k]$. If $\vec{i} = (i_1, \ldots, i_m) \in [k]^m$ is a tuple of distinct integers, $m \leq k$, then we write $\vec{x} \restriction \vec{i}$ to denote the m-tuple $(x_{i_1}, \ldots, x_{i_m})$.

If $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_m)$ are tuples of elements, then we often write $\vec{x} \cup \vec{y}$ to denote the set of elements $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$.

If X is a set, then we write $\wp(X)$ to denote the power set of X; that is, the set of all subsets of X. Similarly, we write $\wp_{\text{fin}}(X)$ to denote the set of all *finite* subsets of X.

2.2 Logics and structures

A *vocabulary* (also called a *signature* or a *language*) τ is a finite sequence of relation and constant symbols $(R_1, \ldots, R_k, c_1, \ldots, c_l)$. Every relation symbol R_i has a fixed *arity* $\text{ari}(R_i) \in \mathbb{N}$. We consider both vocabularies that contain no constant symbols as well as vocabularies that contain no relation symbols; the empty vocabulary is one particular example. If σ and τ are vocabularies, then we write $\sigma \subseteq \tau$ to denote that every relation and constant symbol that appears in σ also appears in τ.

2.2.1 Structures

Let τ be a vocabulary. A *τ-structure* $\mathbf{A} = (U(\mathbf{A}), R_1^{\mathbf{A}}, \ldots, R_k^{\mathbf{A}}, c_1^{\mathbf{A}}, \ldots, c_l^{\mathbf{A}})$ consists of a non-empty set $U(\mathbf{A})$, called the *domain* of $\mathbf{A}$, together with relations $R_i^{\mathbf{A}} \subseteq U(\mathbf{A})^{\mathrm{ari}(R_i)}$ and constants $c_j^{\mathbf{A}} \in U(\mathbf{A})$ for each $1 \leq i \leq k$ and $1 \leq j \leq l$. The elements of the set $U(\mathbf{A})$ are called the *elements* of $\mathbf{A}$ and we define $\|\mathbf{A}\|$, the *cardinality* of $\mathbf{A}$, to be the cardinality of $U(\mathbf{A})$.

If τ and σ are vocabularies with $\sigma \subseteq \tau$, and $\mathbf{A}$ is a τ-structure, then we write $\mathbf{A}|\sigma$ to denote the *σ-reduct* of $\mathbf{A}$, which is the structure obtained from $\mathbf{A}$ by forgetting the interpretations of the symbols and constants that are in τ but not in σ.

Unless otherwise stated, all structures are assumed to be finite. We write $\mathrm{fin}[\tau]$ for the class of all finite structures of vocabulary τ. Following Otto [58], we also consider finite structures with an additional tuple of *parameters*. We denote the class of all τ-structures with fixed tuples of $r \in \mathbb{N}$ parameters by:

$$\mathrm{fin}[\tau; r] := \left\{(\mathbf{A}, \vec{a}) \mid \mathbf{A} \in \mathrm{fin}[\tau],\ \vec{a} \in U(\mathbf{A})^r\right\}.$$

2.2.2 Logics

A *logic* L consists of a mapping that assigns for each vocabulary τ a set of formulae $\mathrm{L}[\tau]$, and a *satisfaction relation* $\models_{\mathrm{L}}$ between structures and formulae (possibly with an assignment to any free variables, as we discuss in more detail below). We do not require a formal definition of 'a logic' here as we merely use the term in the abstract to generalise some common definitions that apply to the specific logics we consider in this dissertation; for instance, first-order and inflationary fixed-point logic[1].

We recall the definition of first-order logic (FO). A *term* of first-order logic over a signature τ is either a symbol from some countable collection of (first-order) variables, or a constant symbol from τ. We define the set $\mathrm{FO}[\tau]$ of first-order formulae over τ to be the smallest set containing the atomic formulae, $t_i = t_j$ and $R(t_1, \ldots, t_m)$, where each t_i is a term and R a relation symbol from τ of arity m, which is closed under the operations of negation, conjunction, disjunction and universal and existential quantification. All the logics we consider hereafter will be extensions of first-order logic, defined via explicit rules for formula-formation and matching rules for semantics.

In general, we consider logics whose formulae may contain both first-order and second-order variables (also called *relation variables*), where each second-order variable has a prescribed *arity*. We commonly use lower-case letters $x, y, z, \ldots$ to denote first-order variables and use upper-case letters $X, Y, Z, \ldots$ to denote second-order variables. We write $\mathrm{free}(\varphi)$ to denote the set of free variables of φ, where φ is a formula of a logic L. For a tuple $\vec{\varphi} = (\varphi_1, \ldots, \varphi_k)$, where each φ_i is a formula, we write $\mathrm{free}(\vec{\varphi}) := \bigcup_{i=1}^{k} \mathrm{free}(\varphi_i)$. A formula without free variables is a *sentence*. We commonly annotate formulae with tuples of variables $\varphi(x_1, \ldots, x_k, X_1, \ldots, X_l)$ to indicate that all the variables $x_1, \ldots, x_k$ and $X_1, \ldots, X_l$ are distinct and that $\mathrm{free}(\varphi) \subseteq \{x_1, \ldots, x_k, X_1, \ldots, X_l\}$. The same notation applies in exactly the same way to terms of L.

Remark. Throughout, we commonly write $(x \neq y)$ as a shorthand for the formula $\neg(x = y)$.

[1] The study of abstract extensions of first-order logic is known as *abstract model theory*. This field was initiated by Lindström [55], whose aim was to develop a single concept whose instances would be the various known extensions of first-order logic, such as infinitary logic and logics with inductive definitions. For more background on abstract model theory, see e.g. Barwise and Feferman [6].

2.2.3 Assignments

Let τ be a vocabulary and let $\mathbf{A}$ be a τ-structure. An *assignment in* $\mathbf{A}$ is a function α that associates an element $\alpha(x) \in U(\mathbf{A})$ with each first-order variable x and associates an n-ary relation $\alpha(X) \subseteq U(\mathbf{A})^n$ with each second-order variable X of arity n. Intuitively we think of α as assigning the meaning $\alpha(x)$ to a variable x. We extend α to a function over terms by setting $\alpha(c) = c^{\mathbf{A}}$ for each constant symbol $c \in \tau$. We also let $\alpha(\vec{x}) := (\alpha(x_1), \ldots, \alpha(x_k))$ for a k-tuple of variables $\vec{x} = (x_1, \ldots, x_k)$. Finally, if $\vec{a} \in U(\mathbf{A})^k$, then we write $\alpha\frac{\vec{a}}{\vec{x}}$ to denote the assignment obtained by setting $x_i \mapsto a_i$, for $i \in [k]$, and $y \mapsto \alpha(y)$ for all $y \notin \{x_1, \ldots, x_k\}$.

The semantics of a logic L over τ-structures is defined over pairs $(\mathbf{A}, \alpha)$, where $\mathbf{A}$ is a τ-structure and α is an assignment in $\mathbf{A}$. We write $\mathbf{A} \models_{\mathrm{L}} \varphi[\alpha]$ to denote that $\mathbf{A}$ *satisfies* the formula φ under the assignment α. Of course, the exact definition of the relation $\models_{\mathrm{L}}$ depends on the logic in question; as an example we note that for all the logics that we consider, an atomic formula $R\vec{x}$ is satisfied in $\mathbf{A}$ if $\alpha(\vec{x}) \in R^{\mathbf{A}}$. When the logic L is clear from the context, we omit the subscript to the satisfaction relation. Suppose $\mathrm{free}(\varphi) \subseteq \vec{x} \cup \vec{X}$, where $\vec{x} = (x_1, \ldots, x_k)$ and $\vec{X} = (X_1, \ldots, X_l)$ are tuples of first- and second-order variables, respectively. If $\vec{a} = (a_1, \ldots, a_k)$ are elements from $U(\mathbf{A})$ and $\vec{R} = (R_1, \ldots, R_l)$ are relations over $U(\mathbf{A})$, with $R_i \subseteq U(\mathbf{A})^{\mathrm{ari}(X_i)}$, then we write

$$\mathbf{A} \models \varphi[a_1/x_1, \ldots, a_k/x_k, R_1/X_1, \ldots, R_l/X_l],$$

or $\mathbf{A} \models \varphi[\vec{a}/\vec{x}, \vec{R}/\vec{X}]$ for short, to denote that φ is satisfied in $\mathbf{A}$ by assigning a_i for x_i and R_j for X_j, for each i and j. Often we omit the named variables when they are clear from the context, and simply write $\mathbf{A} \models \varphi[\vec{a}, \vec{R}]$. Finally, when $\mathrm{free}(\varphi) = \varnothing$ then we write $\mathbf{A} \models \varphi$ to denote $\mathbf{A} \models \varphi[\varnothing]$.

We write

$$\varphi(\vec{x})^{\mathbf{A}} := \{\vec{a} \in U(\mathbf{A})^{\|\vec{x}\|} \mid \mathbf{A} \models \varphi[\vec{a}]\} \subseteq U(\mathbf{A})^{|\vec{x}|}$$

to denote the relation defined by a formula $\varphi(\vec{x})$ in a structure $\mathbf{A}$, where $\mathrm{free}(\varphi) \subseteq \vec{x}$. Similarly, we write

$$t(\vec{x})^{\mathbf{A}} := \{(\vec{a}, t[\vec{a}]^{\mathbf{A}}) \mid \vec{a} \in U(\mathbf{A})^{\|\vec{x}\|}\} \subseteq U(\mathbf{A})^{|\vec{x}|+1}$$

to denote the graph of the function defined by a term $t(\vec{x})$ in a structure $\mathbf{A}$, where $\mathrm{free}(t) \subseteq \vec{x}$. We also consider relations defined by omitting named variables from a particular assignment over a structure $\mathbf{A}$. Let $\varphi(\vec{x})$ be a formula of $\mathrm{L}[\tau]$, where $\vec{x}$ is a k-tuple of variables. For an integer $m \leq k$, consider an m-tuple $\vec{i} = (i_1, \ldots, i_m) \in [k]^m$ of distinct elements, indexing variables in $\vec{x}$. Then for each finite τ-structure $\mathbf{A}$ and k-tuple $\vec{a}$ of elements from $U(\mathbf{A})$, define

$$\varphi[\vec{a}]^{\mathbf{A}} \restriction \vec{i} := \{(c_1, \ldots, c_m) \in U(\mathbf{A})^m \mid \mathbf{A} \models \varphi[\vec{a}\frac{c_1}{i_1} \cdots \frac{c_m}{i_m}]\}$$

We can also define $t[\vec{a}]^{\mathbf{A}} \restriction \vec{i}$ for an L-term t completely analogously.

2.2.4 Definable classes and queries

If φ is a sentence in vocabulary τ, then we write

$$\mathrm{Mod}(\varphi) := \{\mathbf{A} \mid \mathbf{A} \in \mathrm{fin}[\tau] \text{ and } \mathbf{A} \models \varphi\}$$

to denote the class of finite models of φ.

Definition 2.1 (Definable classes of structures)**.** Let τ be a vocabulary and let L be a logic. A class $\mathcal{K}$ of finite τ-structures is said to be *definable in* L if there is a sentence $\varphi \in \mathrm{L}[\tau]$ for which it holds that $\mathrm{Mod}(\varphi) = \mathcal{K}$. ■

We also consider *queries* defined by formulae. The following definition is adapted from Libkin [52].

Definition 2.2 (Queries)**.** An *m-ary query*, $m \geq 0$, on τ-structures is a mapping Q that associates with every $\mathbf{A} \in \mathrm{fin}[\tau]$ a subset of $U(\mathbf{A})^m$ and is closed under isomorphisms; that is, if $\mathbf{A}$ and $\mathbf{B}$ are finite τ-structures with $f : \mathbf{A} \cong \mathbf{B}$ an isomorphism, then $Q(\mathbf{B}) = f(Q(\mathbf{A}))$.

We say that an m-ary query Q over τ-structures is L-definable if there is an $\mathrm{L}[\tau]$-formula $\varphi(\vec{x})$ with $\|\vec{x}\| = m$, such that for every $\mathbf{A} \in \mathrm{fin}[\tau]$:

$$Q(\mathbf{A}) = \varphi(\vec{x})^{\mathbf{A}} = \{\vec{a} \in U(\mathbf{A})^m \mid \mathbf{A} \vDash \varphi[\vec{a}]\}.$$

■

It is often convenient to consider nullary (that is, 0-ary) queries separately. In that case we naturally identify nullary relations with Boolean values; that is, we consider $U(\mathbf{A})^0$ as a one-element set which has only two subsets, which we identify as *true* and *false*. In this sense, a nullary quary on τ-structures is a mapping $Q : \mathrm{fin}[\tau] \to \{\mathrm{true}, \mathrm{false}\}$ which is closed under isomorphism; that is, if $\mathbf{A}$ and $\mathbf{B}$ are finite τ-structures with $\mathbf{A} \cong \mathbf{B}$, then $Q(\mathbf{B}) = Q(\mathbf{A})$. Such queries are known as *Boolean queries*.

We commonly associate a Boolean query Q on τ-structures with the isomorphism-closed class of finite structures $\mathcal{C}_Q$ defined by

$$\mathcal{C}_Q := \{\mathbf{A} \in \mathrm{fin}[\tau] \mid Q(\mathbf{A}) = \mathrm{true}\}.$$

In this case, it can be seen that a Boolean query Q is defined by a sentence φ if and only if $\mathcal{C}_Q = \mathrm{Mod}(\varphi)$. In the following we often do not distinguish between a Boolean query Q and the associated class of structures $\mathcal{C}_Q$; that is, we will often identify a Boolean query with the associated class of structures.

Using the language of queries, we can now formally define relations between logics, indicating their relative expressive power over finite structures. We say that a logic L_1 is (at least) as expressive as a logic L_2, and write $\mathrm{L}_2 \leqq \mathrm{L}_1$, if every query definable in L_2 is also definable in L_1. We write $L_1 \equiv L_2$ if $L_1 \leqq L_2$ and $L_2 \leqq L_1$. Finally, we write $\mathrm{L}_2 \lneqq \mathrm{L}_1$ if $\mathrm{L}_2 \leqq \mathrm{L}_1$ and there is a query definable in L_1 which is not definable in L_2.

2.2.5 Interpretations and logical reductions

We frequently consider ways of defining one structure within another in a logic L. Recall that an equivalence relation $\sim$ is a *congruence* for some n-ary relation R if for all n-tuples $\vec{x}$ and $\vec{y}$: $\bigwedge_i x_i \sim y_i \to (R\vec{x} \leftrightarrow R\vec{y})$.

Definition 2.3 (Interpretations)**.** Consider two signatures σ and τ and a logic L. An *m-ary* L*-interpretation of* τ *in* σ is a sequence of formulae of L in vocabulary σ consisting of:

- a formula $\delta(\vec{x})$;

- a formula $\varepsilon(\vec{x}, \vec{y})$;
- for each relation symbol $R \in \tau$ of arity k, a formula $\varphi_R(\vec{x}_1, \ldots, \vec{x}_k)$; and
- for each constant symbol $c \in \tau$, a formula $\gamma_c(\vec{x})$,

where each $\vec{x}$, $\vec{y}$ or $\vec{x}_i$ is an m-tuple of free variables. We call m the *width* of the interpretation. We say that an interpretation Θ associates a τ-structure $\mathbf{B}$ to a σ-structure $\mathbf{A}$ if there is a surjective map h from the m-tuples $\{\vec{a} \in U(\mathbf{A})^k \mid \mathbf{A} \models \delta[\vec{a}]\}$ to $\mathbf{B}$ such that:

- $h(\vec{a}_1) = h(\vec{a}_2)$ if, and only if, $\mathbf{A} \models \varepsilon[\vec{a}_1, \vec{a}_2]$;
- $R^{\mathbf{B}}(h(\vec{a}_1), \ldots, h(\vec{a}_k))$ if, and only if, $\mathbf{A} \models \varphi_R[\vec{a}_1, \ldots, \vec{a}_k]$;
- $h(\vec{a}) = c^{\mathbf{B}}$ if, and only if, $\mathbf{A} \models \gamma_c[\vec{a}]$.

Note that an interpretation Θ associates a τ-structure with $\mathbf{A}$ only if ε defines an equivalence relation on $U(\mathbf{A})^m$ that is a congruence with respect to the relations defined by the formulae φ_R and γ_c. In such cases, however, $\mathbf{B}$ is uniquely defined up to isomorphism and we write $\Theta(\mathbf{A}) := \mathbf{B}$. ■

Hereafter we are only interested in interpretations that associate a τ-structure to every $\mathbf{A}$. We say that $\mathbf{B}$ is L-*definable over* $\mathbf{A}$ if there is an L-interpretation (which does not depend on either $\mathbf{A}$ or $\mathbf{B}$) that associates $\mathbf{B}$ with $\mathbf{A}$.

We can now define logical reductions from one class of structures to another.

Definition 2.4 (Logical reductions)**.** Let $\mathcal{C}$ be a class of σ-structures and $\mathcal{D}$ a class of τ-structures closed under isomorphisms. An L-interpretation Θ of τ in σ is said to be an L-*reduction from* $\mathcal{C}$ *to* $\mathcal{D}$ if for every σ-structure $\mathbf{A}$ it holds that $\mathbf{A} \in \mathcal{C}$ if and only if $\Theta(\mathbf{A}) \in \mathcal{D}$. ■

In the following, we will focus mostly on *first-order reductions*. In particular, most of the logics we consider in this thesis are closed under first-order reductions, in the following sense.

Definition 2.5 (Closure under first-order reductions)**.** Let L be a logic. We say that L is *closed under first-order reductions* if and only if the set of Boolean queries definable in L is closed under first-order reductions; that is, if $\mathcal{C}$ is a Boolean query definable in $\mathrm{L}[\sigma]$ and Θ a first-order interpretation of τ in σ, then the Boolean query $\{\Theta(\mathbf{A}) \mid \mathbf{A} \in \mathcal{C}\}$ is definable in $\mathrm{L}[\tau]$. ■

2.2.6 Types and equivalences

Let L be a logic and $\mathbf{A}$ a τ-structure. The $\mathrm{L}[\tau]$-*type* of a tuple $\vec{a} = (a_1, \ldots, a_k)$ of elements of $\mathbf{A}$ is the class of all L-formulae in k free variables that are satisfied by $\vec{a}$ in $\mathbf{A}$:

$$\mathrm{tp}(\mathrm{L}; \mathbf{A}, \vec{a}) := \left\{\varphi(\vec{x}) \in \mathrm{L}[\tau] \mid \mathbf{A} \models \varphi[\vec{a}]\right\},$$

where $\vec{x}$ is a k-tuple of variables. We often use Greek symbols $\alpha, \beta, \gamma, \ldots$ to denote types. We write $\mathrm{Tp}(\mathrm{L}; \tau, k)$ for the class of all $\mathrm{L}[\tau]$-types in k free variables over finite τ-structures, that is

$$\mathrm{Tp}(\mathrm{L};\tau,k) := \{\mathrm{tp}(\mathrm{L};\mathbf{A},\vec{a}) \mid (\mathbf{A},\vec{a}) \in \mathrm{fin}[\tau;k]\}.$$

Let $(\mathbf{A},\vec{a}),(\mathbf{B},\vec{b}) \in \mathrm{fin}[\tau;k]$, $k \geq 1$. We say $(\mathbf{A},\vec{a})$ and $(\mathbf{B},\vec{b})$ are L-*equivalent*, and write $(\mathbf{A},\vec{a}) \equiv^{\mathrm{L}} (\mathbf{B},\vec{b})$, if $\mathrm{tp}(\mathrm{L};\mathbf{B},\vec{a}) = \mathrm{tp}(\mathrm{L};\mathbf{B},\vec{b})$. In other words, $(\mathbf{A},\vec{a})$ and $(\mathbf{B},\vec{b})$ are L-equivalent if $\vec{a}$ and $\vec{b}$ satisfy exactly the same L-formulae over $\mathbf{A}$ and $\mathbf{B}$ respectively. Similarly, we write $\mathbf{A} \equiv^{\mathrm{L}} \mathbf{B}$ if $\mathbf{A}$ and $\mathbf{B}$ satisfy exactly all the same L-sentences.

If $\alpha \in \mathrm{Tp}(\mathrm{L};\tau,k)$ and $\vec{a}$ is a k-tuple of elements from a τ-structure $\mathbf{A}$, then we say that $\vec{a}$ *realises* α *in* $\mathbf{A}$ if $\mathrm{tp}(\mathrm{L};\mathbf{A},\vec{a}) = \alpha$. The *atomic type* of $\vec{a}$ over $\mathbf{A}$, $\mathrm{atp}(\mathbf{A},\vec{a})$, is the type $\mathrm{tp}(\mathrm{L};\mathbf{A},\vec{a})$ when L is taken to be the quantifier-free fragment of first-order logic.

2.2.7 Lindström quantifiers and extensions

Generalised quantifiers in the sense of Lindström [54] have been studied as a way to increase the expressiveness of a logic by a prescribed query. Let $\sigma = (R_1,\ldots,R_l)$ be a vocabulary where each relation R_i has arity n_i. Consider a class $\mathcal{K}$ of σ-structures that is closed under isomorphism; that is, for σ-structures $\mathbf{A}$ and $\mathbf{B}$, if $\mathbf{A} \in \mathcal{K}$ and $\mathbf{A} \cong \mathbf{B}$, then $\mathbf{B} \in \mathcal{K}$. With $\mathcal{K}$ we associate a *Lindström quantifier* $Q_{\mathcal{K}}$ whose *type* is the tuple $(n_1,\ldots,n_l)$. The *arity* of the quantifier $Q_{\mathcal{K}}$ is the value of $\max\{n_1,\ldots,n_l\}$. For a logic L, define the extension $\mathrm{L}(Q_{\mathcal{K}})$ by closing the set of formulae of L by introducing the following formula-formation rule:

> if $\varphi_1,\ldots,\varphi_l$ are formulae of $\mathrm{L}(Q_{\mathcal{K}})$, $\vec{x}_1,\ldots,\vec{x}_l$ tuples of variables where $\vec{x}_i$ has length n_i, then the expression $Q_{\mathcal{K}}\vec{x}_1\ldots\vec{x}_l\,(\varphi_1,\ldots,\varphi_l)$ is a formula of $\mathrm{L}(Q_{\mathcal{K}})$ with all occurrences of $\vec{x}_i$ in φ_i bound.

The semantics of the Lindström quantifier $Q_{\mathcal{K}}$ is defined such that

$$\mathbf{A} \vDash Q_{\mathcal{K}}\vec{x}_1\ldots\vec{x}_l\,(\varphi_1,\ldots,\varphi_l) \text{ if and only if } (U(\mathbf{A});\varphi_1(\vec{x}_1)^{\mathbf{A}},\ldots,\varphi_l(\vec{x}_l)^{\mathbf{A}}) \in \mathcal{K},$$

where $(U(\mathbf{A});\varphi_1(\vec{x}_1)^{\mathbf{A}},\ldots,\varphi_l(\vec{x}_l)^{\mathbf{A}})$ is interpreted as a σ-structure.

Example 2.6. The existential quantifier $\exists$ can be seen as the Lindström quantifier associated with the class of structures $\mathcal{K}$ over a signature with one unary relation symbol R, given by $\mathcal{K} := \{(A,R^{\mathbf{A}}) \mid R^{\mathbf{A}} \subseteq A \text{ and } R^{\mathbf{A}} \neq \varnothing\}$. ■

Similarly we can consider the extension of a logic L by a collection $\mathbf{Q}$ of Lindström quantifiers. The logic $\mathrm{L}(\mathbf{Q})$ is defined by adding a rule for constructing formulae with the quantifier Q, for each $Q \in \mathbf{Q}$, to the list of formula-formation rules for L. The semantics is defined by considering the semantics for each quantifier $Q \in \mathbf{Q}$, as above.

Often we consider families of quantifiers generated under some uniformity condition. Here we focus on the following notion of uniformity, due to Dawar [15]. Let $\mathcal{K}$ be a class of structures over vocabulary $\sigma = (R_1,\ldots,R_l)$. For each $k \in \mathbb{N}$, let σ_k be the vocabulary $(R_{k,1},\ldots,R_{k,l})$ where the arity of $R_{k,i}$ is $k \cdot n_i$. Let $\mathcal{K}_k$ be the class of σ_k-structures defined by

$$\mathcal{K}_k := \{(A,S_1,\ldots,S_l) \mid (A^k,S_1,\ldots,S_l) \in \mathcal{K}\},$$

where $(A^k;S_1,\ldots,S_l)$ is seen as a σ-structure with universe A^k. If Q_k is the Lindström quantifier associated with $\mathcal{K}_k$ then we say that the sequence $\{Q_k \mid k \in \mathbb{N}\}$ is *uniformly generated by* $\mathcal{K}$.

Definition 2.7 (Uniform sequences of quantifiers)**.** A countable collection $\mathbf{Q}$ of Lindström quantifiers is a *uniform sequence* if there is a class of structures $\mathcal{K}$ such that $\mathbf{Q}$ is uniformly generated by $\mathcal{K}$. ■

2.3 Logics with inductive definitions

We review some common extensions of first-order logic with operators for formalising inductive definitions. For a more detailed description of any of these logics, see e.g. Ebbinghaus and Flum [23] or Libkin [52].

2.3.1 Inflationary fixed-point logic

Let $\varphi(R, \vec{x})$ be a formula in the vocabulary τ, where R is a k-ary relation variable and $\vec{x}$ is a k-tuple of variables. Over a pair $(\mathbf{A}, \alpha)$, where $\mathbf{A}$ is a finite τ-structure and α an assignment in $\mathbf{A}$, the formula $\varphi(R, \vec{x})$ defines an operator

$$F_\varphi^{(\mathbf{A},\alpha)} : \wp(U(\mathbf{A})^k) \to \wp(U(\mathbf{A})^k)$$

which maps a relation $S \subseteq U(\mathbf{A})^k$ interpreting the relation variable R to the relation

$$F_\varphi^{(\mathbf{A},\alpha)}(S) := \{\vec{a} \in U(\mathbf{A})^k \mid \mathbf{A} \vDash \varphi[\alpha \tfrac{S}{R} \tfrac{\vec{a}}{\vec{x}}]\} \subseteq U(\mathbf{A})^k.$$

This allows us to define an *increasing sequence* of relations on $\mathbf{A}$:

$$\begin{aligned} X^0 &:= \varnothing, \\ X^{i+1} &:= X^i \cup F_\varphi^{(\mathbf{A},\alpha)}(X^i) \end{aligned}$$

The *inflationary fixed point of* $F_\varphi^{(\mathbf{A},\alpha)}$, written $\mathrm{ifp}(F_\varphi^{(\mathbf{A},\alpha)})$, is the limit of this sequence. It can be seen that if $\|\mathbf{A}\| = n$ then this limit will be reached after at most n^k stages.

The terms and formulae of *inflationary fixed-point logic* (IFP) in vocabulary τ are defined inductively by extending the rules of first-order logic with the following rule.

> Let $\varphi(R, \vec{x})$ be a formula, where R is k-ary and $\vec{x}$ is a k-tuple of variables. If $\vec{t}$ is a k-tuple of terms then
>
> $$[\mathbf{ifp}_{R,\vec{x}}\varphi](\vec{t})$$
>
> is a formula, with $\mathrm{free}([\mathbf{ifp}_{R,\vec{x}}\varphi](\vec{t})) := \big(\mathrm{free}(\varphi) \setminus (\vec{x} \cup R)\big) \cup \mathrm{free}(\vec{t})$.

The semantics of IFP in vocabulary τ is defined inductively for all pairs $(\mathbf{A}, \alpha)$, where $\mathbf{A}$ is a finite τ-structure and α an assignment in $\mathbf{A}$, by extending the semantics rules for FO with the following rule for the **ifp**-operator:

$$\mathbf{A} \vDash_{\mathrm{IFP}} \big([\mathbf{ifp}_{R,\vec{x}}\varphi](\vec{t})\big)[\alpha] \text{ iff } \alpha(\vec{t}) \in \mathrm{ifp}(F_\varphi^{(\mathbf{A},\alpha)}).$$

2.3.2 Transitive closure logics

Let $\varphi(\vec{x}, \vec{y})$ be a formula in vocabulary τ, where $\vec{x}$ and $\vec{y}$ are k-tuples of variables. Given a pair $(\mathbf{A}, \alpha)$, where $\mathbf{A}$ is a finite τ-structure and α an assignment in $\mathbf{A}$, write $\vec{G}^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}$ to denote the graph on vertex set $U(\mathbf{A})^k$ with the set of edges

$$\{(\vec{a}, \vec{b}) \mid \mathbf{A} \vDash \varphi[\alpha \tfrac{\vec{a}}{\vec{x}} \tfrac{\vec{b}}{\vec{y}}]\} \subseteq U(\mathbf{A})^k \times U(\mathbf{A})^k.$$

Similarly, write $G^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}$ and $\tilde{G}^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}$ to denote the symmetric closure and the deterministic part of $\vec{G}^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}$, respectively. To be precise, $\tilde{G}^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}$ is the graph obtained from $\vec{G}^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}$ by retaining only those edges (u, v) where u has out-degree one.

The terms and formulae of *transitive closure logic* (FO+TC) in vocabulary τ are defined inductively by extending the rules of first-order logic with the following rule.

> Let $\varphi(\vec{x}, \vec{y})$ be a formula, where $\vec{x}$ and $\vec{y}$ are k-tuples of variables. If $\vec{t}$ and $\vec{s}$ are k-tuples of terms then
>
> $$[\mathbf{tc}_{\vec{x},\vec{y}}\varphi](\vec{t}, \vec{s})$$
>
> is a formula, with $\text{free}([\mathbf{tc}_{\vec{x},\vec{y}}\varphi](\vec{t}, \vec{s})) := \big(\text{free}(\varphi) \setminus (\vec{x} \cup \vec{y})\big) \cup \text{free}(\vec{t}) \cup \text{free}(\vec{s})$.

The semantics of FO+TC in vocabulary τ is defined inductively for all pairs $(\mathbf{A}, \alpha)$, where $\mathbf{A}$ is a finite τ-structure and α an assignment in $\mathbf{A}$, by extending the semantics rules for FO with the following rule for the **tc**-operator:

$$\mathbf{A} \vDash_{\text{FO+TC}} \big([\mathbf{tc}_{\vec{x},\vec{y}}\varphi](\vec{t}, \vec{s})\big)[\alpha] \text{ iff } (\alpha(\vec{t}), \alpha(\vec{s})) \text{ is in the transitive closure of } \vec{G}^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}.$$

Similarly, we define *symmetric transitive closure logic* (FO+STC) and *deterministic transitive closure logic* (FO+DTC) in exactly the same way as FO+TC above, except that instead of formulae involving the **tc**-operator, we have formulae with operators **stc** and **dtc**, respectively. The semantics of these operators is defined like the semantics of the **tc**-operator, except now we consider reachability in the undirected graph $G^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}$ for the **stc**-operator and reachability in the deterministic graph $\tilde{G}^{(\mathbf{A},\alpha)}_{\varphi,\vec{x},\vec{y}}$ for the **dtc**-operator.

2.4 Many-sorted logics and structures

We occasionally consider structures with a number of distinct domains (called *sorts*) and strongly typed logics to match, in which the variables range over different domains. Generally, we try to avoid the notational overhead caused by the presence of sorts as much as possible, and only mention the typing of variables when it is not obvious from the context.

2.4.1 Many-sorted structures

An *m-sorted vocabulary* is a vocabulary $\tau = (R_1, \ldots, R_k, c_1, \ldots, c_l)$ where every relation or constant symbol $X \in \tau$ has an associated *type*, denoted by $\text{type}(X)$. That is, if $R \in \tau$ is a

relation symbol of arity n, then $\mathsf{type}(R) \in [m]^n$, and if $c \in \tau$ is a constant symbol then $\mathsf{type}(c) \in [m]$. An *m-sorted structure*

$$\mathbf{A} = ((S_1, \ldots, S_m), R_1^{\mathbf{A}}, \ldots, R_k^{\mathbf{A}}, c_1^{\mathbf{A}}, \ldots, c_l^{\mathbf{A}})$$

consists of m non-empty sets $S_1, \ldots, S_m$, together with

- relations $R^{\mathbf{A}} \subseteq S_{t_1} \times \cdots \times S_{t_n}$ for each relation symbol R of arity n and type $(t_1, \ldots, t_n) \in [m]^n$; and
- constants $c^{\mathbf{A}} \in S_t$ for each constant symbol c of type $t \in [m]$.

We write $U(\mathbf{A}) := \bigcup_{i=1}^{m} S_i$ for the domain of $\mathbf{A}$. When $m = 1$ then we simply write $\mathbf{A} = (U(\mathbf{A}), R_1^{\mathbf{A}}, \ldots, R_k^{\mathbf{A}}, c_1^{\mathbf{A}}, \ldots, c_l^{\mathbf{A}})$ instead of $\mathbf{A} = ((U(\mathbf{A})), R_1^{\mathbf{A}}, \ldots, R_k^{\mathbf{A}}, c_1^{\mathbf{A}}, \ldots, c_l^{\mathbf{A}})$, to be consistent with our notation from before.

2.4.2 Many-sorted logics

Let L be a logic. We can extend L to a *many-sorted logic* as follows. The m-sorted logic L_m is L together with a function type that associates every first-order variable x with an integer $\mathsf{type}(x) \in [m]$ and associates every second-order variable X of arity n with a tuple $\mathsf{type}(X) \in [m]^n$. The semantics of L_m, defined over m-sorted structures, are just like the semantics of L, except that any assignment α over an m-sorted structure $\mathbf{A}$ with sorts $S_1, \ldots, S_m$ must satisfy

$$\mathsf{type}(x) = t \Leftrightarrow \alpha(x) \in S_t,$$

for every first-order variable x and

$$\mathsf{type}(X) = (t_1, \ldots, t_n) \Leftrightarrow \alpha(X) \subseteq S_{t_1} \times \ldots, \times S_{t_n},$$

for every second-order variable X of arity n.

2.5 Logics with counting

In this section we define extensions of first-order and fixed-point logic with operators for expressing the cardinality of definable relations. In our definition of these *counting logics*, we follow the convention of Grohe [31]. That is, both counting logics have variables that range over the non-negative integers, as well as variables ranging over the elements of a structure, and terms and formulae are interpreted over countable-infinite structures that are obtained by extending a finite structure with a copy of the integers. To ensure that these logics define only decidable queries, all variables ranging over integers must be bound by terms when they are introduced. Other authors [23, 52] consider counting logics that have number variables which only range over a finite subset of the integers. By bounding quantification over the integers as described above, it can be seen that two formalisms are in fact equivalent over finite structures (see e.g. [31] for further details).

Because both counting logics and other 'numerical logics' of similar kind will play a prominent role in this thesis, the definitions that follow are provided in full detail.

2.5.1 First-order logic with counting

First-order logic with counting (FOC) has two kinds of variables: *element variables*, that range over the domain elements of a structure, and *number variables*, that range over the non-negative integers. We commonly use lower-case Latin symbols $x, y, z, \ldots$ to denote element variables and lower-case Greek symbols $\eta, \gamma, \upsilon, \ldots$ to denote number variables. Generally, we allow for many-sorted variants of FOC. For instance, in $(m+1)$-sorted FOC we have $m+1$ variable types, with number variables assigned type $(m+1)$ and variables of all other types collectively referred to as element variables (when $m > 1$ the actual typing of element variables will usually be clear from the context). In addition, we have second-order variables $X, Y, Z, \ldots$, where the type of a second-order variable X is defined as usual.

Let τ be a vocabulary which does not contain any of the symbols in $\{\leq, +, \cdot, 0_N, 1_N\}$ (otherwise, simply rename the conflicting symbols in τ). Terms of FOC of vocabulary τ are of two kinds: an *element term* is an element variable or a constant in τ, and a *number term* is a number variable, one of the constant symbols in $\{0_N, 1_N\}$, an application of the functions $+$ or $\cdot$ or a counting term, expressing the cardinality of a definable relation. The terms and formulae of FOC of vocabulary τ are defined inductively by the following rules.

E.1 All element variables x are element terms, where we let $\mathrm{free}(x) := \{x\}$.

E.2 All constants $c \in \tau$ are element terms, where we let $\mathrm{free}(c) := \varnothing$.

N.1 All number variables υ are number terms, where we let $\mathrm{free}(\upsilon) := \{\upsilon\}$.

N.2 The constants 0_N and 1_N are number terms, where we let $\mathrm{free}(0_N) := \mathrm{free}(1_N) = \varnothing$.

N.3 If s, t are number terms, then the expressions $(s + t)$ and $(s \cdot t)$ are number terms, where we let $\mathrm{free}(s * t) := \mathrm{free}(s) \cup \mathrm{free}(t)$ for $* \in \{+, \cdot\}$.

F.1 If s, t are number terms, then the expressions $s = t$ and $s \leq t$ are formulae, where we let $\mathrm{free}(s * t) := \mathrm{free}(s) \cup \mathrm{free}(t)$ for $* \in \{=, \leq\}$.

F.2 If s, t are element terms, then the expression $s = t$ is a formula, where $\mathrm{free}(s = t) := \mathrm{free}(s) \cup \mathrm{free}(t)$.

F.3 If $R \in \tau$ is a k-ary relation symbol and $\vec{t} = (t_1, \ldots, t_k)$ is a tuple of element terms, then $R\vec{t}$ is a formula, where we let $\mathrm{free}(R\vec{t}) := \mathrm{free}(\vec{t})$.

F.4 If X is a k-ary relation variable and $\vec{t} = (t_1, \ldots, t_k)$ are terms whose type matches that of X, then $X\vec{t}$ is a formula, where we let $\mathrm{free}(X\vec{t}) := \{X\} \cup \mathrm{free}(\vec{t})$.

F.5 If φ is a formula and x an element variable, then $\exists x.\varphi$ and $\forall x.\varphi$ are formulae, where we let $\mathrm{free}(\exists x.\varphi) := \mathrm{free}(\forall x.\varphi) = \mathrm{free}(\varphi) \setminus \{x\}$.

F.5 If φ is a formula, υ is a number variable and t is a number term, then $\exists \upsilon \leq t.\varphi$ and $\forall \upsilon \leq t.\varphi$ are formulae, where we let $\mathrm{free}(\exists \upsilon \leq t.\varphi) := \mathrm{free}(\forall \upsilon \leq t.\varphi) = (\mathrm{free}(\varphi) \setminus \{\upsilon\}) \cup \mathrm{free}(t)$.

F.6 If φ and ψ are formulae then the expressions $\neg\varphi$, $\varphi \wedge \psi$, and $\varphi \vee \psi$ are formulae, where we let $\mathrm{free}(\neg\varphi) := \mathrm{free}(\varphi)$ and $\mathrm{free}(\varphi * \psi) := \mathrm{free}(\varphi) \cup \mathrm{free}(\psi)$ for $* \in \{\wedge, \vee\}$.

C If φ is a formula and x is an element variable then $\#_x\varphi$ is a number term, where we let $\mathrm{free}(\#_x\varphi) := \mathrm{free}(\varphi) \setminus \{x\}$.

The semantics of FOC of vocabulary τ is defined over *numerical structures* of vocabulary τ, which are τ-structures expanded with a copy of the non-negative integers.

Definition 2.8 (Numerical structures)**.** Let $\tau = (R_1, \ldots, R_k, c_1, \ldots c_l)$ be a vocabulary which does not contain any of the symbols in $\{\leq, +, \cdot, 0_N, 1_N\}$. Write $\tau^\star := \tau \cup \{\leq, +, \cdot, 0_N, 1_N\}$, where $\leq$ is a binary relation, $+$ and $\cdot$ are binary functions and 0_N and 1_N are constants. For any m-sorted τ-structure $\mathbf{A}$, with sorts $S_1, \ldots, S_m$, we write $\mathbf{A}^\star$ to denote the $(m+1)$-sorted $\tau^\star$-structure

$$((S_1, \ldots, S_m, \mathbb{N}_0), R_1^{\mathbf{A}}, \ldots, R_k^{\mathbf{A}}, c_1^{\mathbf{A}}, \ldots, c_l^{\mathbf{A}}, \leq^{\mathbb{N}_0}, +^{\mathbb{N}_0}, \cdot^{\mathbb{N}_0}, 1_N^{\mathbb{N}_0}, 0_N^{\mathbb{N}_0}),$$

where we view $\mathbb{N}_0$ as the set of non-negative integers, disjoint from $U(\mathbf{A})$. Here $+^{\mathbb{N}_0}$ and $\times^{\mathbb{N}_0}$ denote addition and multiplication over $\mathbb{N}_0$, $\leq^{\mathbb{N}_0}$ is the standard ordering of the integers $\mathbb{N}_0$, and the constants $0_N^{\mathbb{N}_0}$ and $1_N^{\mathbb{N}_0}$ denote the first and second elements of $\mathbb{N}_0$, respectively. We refer to the domain $U(\mathbf{A})$ as the *element sort* of $\mathbf{A}^\star$ and the last sort $\mathbb{N}_0$ as the *number sort* of $\mathbf{A}^\star$. ■

Remark. To keep our vocabulary purely relational, we could instead define addition and multiplication over the number sort as ternary relations R_+ and $R_\cdot$, respectively, instead of functions $+$ and $\cdot$. Thus, we could write $R_+(x, y, z)$ instead of $x + y = z$ for all numeric terms x, y and z, and similarly for $R_\cdot$. However, the use of functions does simplify some of our exposition later and, apart from minor changes to the rules for bounded quantification, it can be seen the difference between the two representations is purely notational and has no bearing on the expressive power of our logics.

Formally, the semantics of FOC$[\tau]$ is defined over pairs $(\mathbf{A}^\star, \alpha)$, where α is a variable assignment in $\mathbf{A}^\star$. We write $\models^{\mathrm{num}}_{\mathrm{FOC}}$ for the satisfaction relation between numerical structures on the one hand and on the other hand FOC formulae and assignments. If t is a term, then we write $\alpha(t)$ to denote the value that is assigned to t over $\mathbf{A}^\star$[2]. Number terms are assigned values in $\mathbb{N}_0$ and element terms are assigned values in $U(\mathbf{A})$. The constants 0_N and 1_N are interpreted as the integers zero and one, respectively. For number terms s and t, we define $\alpha(s+t) := \alpha(s) + \alpha(t)$ and $\alpha(s \cdot t) := \alpha(s) \cdot \alpha(t)$, where the right-hand side of each equation denotes an arithmetic expression over the integers. We extend the definition of type from variables to terms, and define the *type* of a term t to be k if t takes values in the k-th sort of $\mathbf{A}^\star$. For a formula φ, the satisfaction relation $\mathbf{A}^\star \models^{\mathrm{num}}_{\mathrm{FOC}} \varphi[\alpha]$ is defined in the obvious way, with comparison of number terms interpreted by comparing the respective integer assignments over $\mathbb{N}_0$. In particular, when φ is of the form $\exists v \leq t.\psi(v)$, for some formula ψ and number term t, then we define $\mathbf{A}^\star \models^{\mathrm{num}}_{\mathrm{FOC}} (\exists v \leq t.\psi(v))[\alpha]$ if and only if there is an integer $m \in [0, \alpha(t)]$ such that

$$\mathbf{A}^\star \models^{\mathrm{num}}_{\mathrm{FOC}} \psi(v)[\alpha\tfrac{m}{v}].$$

We also commonly write $\exists v < t.\varphi(v)$ and $\forall v < t.\varphi(v)$ as shorthand for $\exists v \leq t.(v \neq t) \wedge \varphi(v)$ and $\forall v \leq t.(v \neq t) \rightarrow \varphi(v)$, respectively.

Finally, consider a *counting term* of the form $\#_x\varphi$, where φ is a formula and x an element variable. Here the intended semantics is that $\#_x\varphi$ denotes the number (i.e. the member of

[2]Occasionally, when t has no free variables, we write $t^{\mathbf{A}^\star}$ to denote $\alpha(t)$ where α is any assignment in $\mathbf{A}^\star$.

the number sort) of elements that satisfy the formula φ. More formally, the semantics of counting terms of FOC over vocabulary τ is defined as follows:

$$\alpha(\#_x\varphi) := \|\{a \in U(\mathbf{A}) \mid \mathbf{A} \vDash^{\text{num}}_{\text{FOC}} \varphi[\alpha\tfrac{a}{x}]\}\|.$$

While we interpret terms and formulae of FOC over countable-infinite $\tau^\star$-structures, we are ultimately interested in queries defined over finite τ-structures. For that purpose, we define the satisfaction relation $\vDash_{\text{FOC}}$ between formulae of $\text{FOC}[\tau]$ and structure-assignment pairs over vocabulary τ as follows.

Definition 2.9. Let τ be a vocabulary and let φ be a formula of $\text{FOC}[\tau]$. Suppose the free variables of φ contain no number variables and no second-order variables with a component of number type. Then for any τ-structure $\mathbf{A}$ and any assignment α over $\mathbf{A}$, we define

$$\mathbf{A} \vDash_{\text{FOC}} \varphi[\alpha] :\Leftrightarrow \mathbf{A}^\star \vDash^{\text{num}}_{\text{FOC}} \varphi[\alpha].$$

That is, φ is satisfied in $\mathbf{A}$ with assignment α if and only if φ is satisfied in the number expansion $\mathbf{A}^\star$ with assignment α. ■

Example 2.10. Over any finite structure, the number term $t_{\text{card}} \equiv \#_x(x = x)$ denotes the cardinality of the domain of the structure. The following sentence in the language of graphs now states that all vertices have an even degree:

$$\varphi \equiv \forall x\, \exists\mu \leq t_{\text{card}}\, (\#_y(E(x,y)) = \mu + \mu).$$

It follows that on the class of all finite graphs, φ defines exactly the class of Eulerian graphs. ■

Later we will make use of the following basic lemma, whose proof is trivial.

Lemma 2.11. *There is a formula* $\mathsf{prime}(v)$ *of* FOC*, where* v *is a number variable, such that for all structures* $\mathbf{A}$ *and all assignments* α *over* $\mathbf{A}^\star$*,*

$$\mathbf{A}^\star \vDash^{num}_{\text{FOC}} \mathsf{prime}(v)[\alpha] \;\Leftrightarrow\; \alpha(v) \textit{ is a prime number}.$$

Proof. The formula

$$\mathsf{comp}(v) \equiv \exists\mu_1 < v\, \exists\mu_1 < v.\big((1 < \mu_1) \wedge (\mu_1 \cdot \mu_2 = v)\big)$$

says that v has a proper factor. Hence, $\mathsf{prime}(v) \equiv \neg\mathsf{comp}(v)$. □

Finally, we note that the reason why primality can be expressed quite easily in FOC has more to do with the fact that integers described by number terms are *tiny* (i.e. represented in unary) than it has to do with the actual expressive power of the logic.

2.5.2 Fixed-point logic with counting

We also consider the logic obtained by extending FOC with inflationary fixed-point operators over numerical structures.

Definition 2.12 (Fixed points in numerical structures)**.** Let $\varphi(R, \vec{x})$ be a formula in vocabulary τ, where R is a k-ary relation variable of type $(h_1, \ldots, h_k)$ and $\vec{x}$ is a k-tuple of variables of types $h_1, \ldots, h_k$, respectively. Let $\vec{t}$ be an l-tuple of number terms, where l is the number of distinct number variables in $\vec{x}$. Let $f : \{x_1, \ldots, x_k\} \rightharpoonup [l]$ be the partial function which maps each number variable in $\vec{x}$ to its index amongst the number variables in $\vec{x}$; that is $f(x_i) = j$ if x_i is a number variable that occurs as the j-th number variable in $\vec{x}$. Given a finite τ-structure $\mathbf{A}$ and an assignment α in $\mathbf{A}^\star$, write $m_i = \alpha(t_i)$ for $i \in [l]$. Over $(\mathbf{A}^\star, \alpha)$, the pair $(\varphi(R, \vec{x}), \vec{t})$ defines an operator

$$F^{(\mathbf{A},\alpha)}_{\varphi,\vec{t}} : \wp\big(U(\mathbf{A}^\star)^k\big) \to \wp\big(U(\mathbf{A}^\star)^k\big)$$

which maps a relation $S \subset U(\mathbf{A}^\star)^k$ interpreting the relation variable R to the relation

$$F^{(\mathbf{A},\alpha)}_{\varphi,\vec{t}}(S) := \{\vec{a} \in B_1 \times \cdots \times B_k \mid \mathbf{A}^\star \vDash \varphi[\alpha \tfrac{S}{R} \tfrac{\vec{a}}{\vec{x}}]\},$$

where

$$B_i = \begin{cases} [0, m_{f(x_i)}] \subset \mathbb{N}_0 & \text{if } x_i \text{ is a number variable,} \\ U(\mathbf{A}) & \text{otherwise.} \end{cases}$$

Here the number terms $\vec{t}$ ensure that $F^{(\mathbf{A},\alpha)}_{\varphi,\vec{t}}(S)$ is a finite set. This now allows us to define an *increasing sequence* of relations on $\mathbf{A}^\star$:

$$\begin{aligned} X^0 &:= \varnothing, \\ X^{i+1} &:= X^i \cup F^{(\mathbf{A},\alpha)}_{\varphi,\vec{t}}(X^i) \end{aligned}$$

The *inflationary fixed point of* $F^{(\mathbf{A},\alpha)}_{\varphi,\vec{t}}$, written $\mathrm{ifp}(F^{(\mathbf{A},\alpha)}_{\varphi,\vec{t}})$, is the limit of this sequence. It can be seen that if $\|\mathbf{A}\| = n$ and $m = \max\{m_1, \ldots, m_l\}$ then this limit will be reached after at most $(n + m)^k$ stages. ■

We can now define the logic IFPC, the extension of FOC with operators for defining inflationary fixed points. The terms and formulae of IFPC of vocabulary τ are defined inductively by extending the rules of FOC with the following rule.

> Let $\varphi(R, \vec{x})$ be a formula, where R is k-ary of type $(h_1, \ldots, h_k)$ and $\vec{x}$ is a k-tuple of variables of types $h_1, \ldots, h_k$, respectively. Let $\vec{t}$ be an l-tuple of number terms, where l is the number of distinct number variables in $\vec{x}$. If $\vec{s}$ is a k-tuple of terms of types $h_1, \ldots, h_k$, respectively, then
>
> $$[\mathbf{ifp}_{R,\vec{x}\le\vec{t}}\varphi](\vec{s})$$
>
> is a formula. We let
>
> $$\mathrm{free}(\,[\mathbf{ifp}_{R,\vec{x}\le\vec{t}}\varphi](\vec{s})\,) := \big((\mathrm{free}(\varphi) \cup \mathrm{free}(\vec{t})) \setminus (\vec{x} \cup R)\big) \cup \mathrm{free}(\vec{s}).$$

Terms and formulae of IFPC are interpreted over pairs $(\mathbf{A}^\star, \alpha)$, just as with FOC before. We write $\vDash^{\mathrm{num}}_{\mathrm{IFPC}}$ for the satisfaction relation between numerical structures on the one hand and

on the other hand IFPC formulae and assignments. The relation $\models^{\text{num}}_{\text{IFPC}}$ extends $\models^{\text{num}}_{\text{FOC}}$, with the semantics of the **ifp**-operator defined as follows:

$$\mathbf{A}^\star \models^{\text{num}}_{\text{IFPC}} \big([\mathbf{ifp}_{R,\vec{x}\le\vec{t}}\varphi](\vec{s})\big)[\alpha] \text{ iff } \alpha(\vec{s}) \in \text{ifp}(F^{(\mathbf{A},\alpha)}_{\varphi,\vec{t}}).$$

As with FOC before, we define a satisfaction relation $\models_{\text{IFPC}}$ between non-numerical structures $\mathbf{A}$ and IFPC formulae with no free number variables by

$$\mathbf{A} \models_{\text{IFPC}} \varphi[\alpha] :\Leftrightarrow \mathbf{A}^\star \models^{\text{num}}_{\text{IFPC}} \varphi[\alpha]$$

for an assignment α in $\mathbf{A}$. Hereafter, we usually omit the subscripts from $\models_{\text{FOC}}$ and $\models_{\text{IFPC}}$ and simply write $\models$ where it will be clear from the context which logic we are considering.

2.6 Infinitary logics

We write FO^k to denote the fragment of first-order logic in which the only variables allowed are $x_1, \ldots, x_k$. The infinitary logic $\mathcal{L}^k$ is obtained by closing FO^k under conjunction and disjunction of arbitrary (possibly infinite) sets of formulae. That is, if Φ is any set of $\mathcal{L}^k$-formulae, then $\bigwedge\Phi$ and $\bigvee\Phi$ are formulae of $\mathcal{L}^k$ that denote the conjunction and disjunction of the formulae in Φ, respectively. The intended semantics is that $\bigwedge\Phi$ is satisfied when all the formulae in Φ are satisfied and $\bigvee\Phi$ is satisfied when at least one of the formulae in Φ is satisfied. We write $\mathcal{L}^\omega := \bigcup_{k<\omega}\mathcal{L}^k$ for the infinitary logic in which each formula has only finitely many variables, taken from the collection $\{x_i \mid i \in \mathbb{N}\}$. We often use variables $x, y, z, \ldots$ instead of $x_1, x_2, x_3, \ldots$ to make formulae more readable. For an excellent reference on finite-variable infinitary logics, see Otto's monograph [58].

It is noted by Otto [58] that the logic $\mathcal{L}^\omega$ can define arbitrarily complex queries. In fact, he shows that with only two variables, there are non-recursive queries on the class of linearly ordered structures that can be defined in $\mathcal{L}^2$. On the other hand, it can also be shown that there are queries of very low complexity which are not definable in $\mathcal{L}^\omega$. Such queries often involve counting in one form or another. For instance, it can be proved using a simple game argument that over the empty vocabulary, $\mathcal{L}^{k-1}$ cannot define the class of structures having at least k distinct elements.

Due to these limitations of $\mathcal{L}^\omega$, it is natural to consider the extension of infinitary logic with a collection of *counting quantifiers*, which are defined as follows. For each natural number i, we have a quantifier $\exists^{\ge i}$ which binds a single formula. A logic L extended with counting quantifiers has the following formula-formation rule, in addition to its usual rules: if φ is a formula and i a positive integer, then $\exists^{\ge i}x\,\varphi$ is a formula. The semantics of a counting quantifier is defined as follows:

> $\mathbf{A} \models \exists^{\ge i}x\,\varphi$ if and only if there are at least i distinct elements $a \in U(\mathbf{A})$ such that $(\mathbf{A}, a) \models \varphi(x)$.

We also write $\exists^{=i}x\,\varphi$ to denote the formula $\exists^{\ge i}x\,\varphi \wedge \neg\exists^{\ge i+1}x\,\varphi$. Similarly, we can define counting quantifiers $\exists^{\le i}$, $\exists^{<i}$ and $\exists^{>i}$. We write FOC^k to denote the k-variable fragment of first-order logic extended with counting quantifiers and write $\mathcal{C}^k$ to denote the corresponding infinitary logic. For each k, it can be shown that FOC^k is more expressive than FO^k (the k-variable fragment of first-order logic) and $\mathcal{C}^k$ is more expressive than $\mathcal{L}^k$, and indeed $\mathcal{C}^\omega := \bigcup_{k<\omega}\mathcal{C}^k$ contains formulae that are not equivalent to any formula of $\mathcal{L}^\omega$.

In each of the above cases, the proofs that certain properties are not expressible in the given logic are most clearly formulated in terms of games. Thus, we can show that there are properties not definable in $\mathcal{L}^\omega$ by means of a variant of the classic Ehrenfeucht-Fraïssé game, which allows for infinitely long plays but with a fixed number of tokens. We discuss game methods further in Chapter 6. Similarly, there is a game that gives us a method to prove that there are properties not definable in $\mathcal{C}^\omega$.

The interest in studying these infinitary logics, from the point of view of finite model theory, comes from the fact that they have proved useful in analysing the expressive power of fixed-point logics. This was illustrated by Kolaitis and Vardi [49], who observed that any sentence of IFP is equivalent to one of $\mathcal{L}^\omega$. Similarly, it can be shown that any sentence of IFPC is equivalent to one of $\mathcal{C}^\omega$ over finite structures (see e.g. [58] for details). Since queries definable in IFP and IFPC are in PTIME, while $\mathcal{L}^\omega$ and $\mathcal{C}^\omega$ can express queries of arbitrary complexity, it follows that both the inclusions are proper, as stated by the following theorem.

Theorem 2.13. IFP $\subsetneqq \mathcal{L}^\omega$ *and* IFPC $\subsetneqq \mathcal{C}^\omega$. □

2.7 Algebra

We recall some basic definitions from abstract algebra, linear algebra and graph theory.

2.7.1 Common algebraic structures

For reference, we give the definition of some of the basic algebraic structures we will consider in this dissertation. For further details on any of these topics, see e.g. Lang's textbook [51].

Groups. A *group* is a non-empty set G with one binary operation $\circ$ that satisfies the following axioms:

- *Closure.* If a and b are two elements in G, then $a \circ b$ is also in G;
- *Associativity.* The operation $\circ$ is associative, i.e. $(a \circ b) \circ c = a \circ (b \circ c)$ for any a, b and c in G;
- *Identity.* There is an element e in G, known as the *identity element*, such that $a \circ e = e \circ a = a$ for any a in G;
- *Inverse.* The operation $\circ$ admits inverse elements; that is for any a in G there exists an element a^{-1} in G, said to be *inverse* to a, such that $a \circ a^{-1} = a^{-1} \circ a = e$.

Often we use the standard symbols for addition $(+)$ and multiplication $(\cdot)$ to denote the group operation. In the former case we say that the group is written *additively* and we write 0 or 0_G for the identity element and $-a$ for the inverse to an element a in G. In the latter case we say that the group is written *multiplicatively* and we write 1 or 1_G for the identity element. A group G is said to be *Abelian* if its operation $\circ$ is commutative; that is, if $a \circ b = b \circ a$ for all a and b in G.

Rings. A *ring* is a set R with two binary operations, called addition ($+$) and multiplication ($\cdot$), for which it holds that (a) the set $(R, +)$ is an Abelian group with respect to addition (the *additive group* of the ring); (b) R is closed under multiplication, multiplication is associative and there exists an element 1_R in R such that $a \cdot 1_R = 1_R \cdot a = a$ for all $a \in R$; and (c) multiplication is related to addition by the distributive laws:

$$a \cdot (b + c) = a \cdot b + a \cdot c,$$
$$(a + b) \cdot c = a \cdot c + b \cdot c.$$

As with standard addition and multiplication of numbers, we often omit the multiplication symbol and write ab to denote $a \cdot b$, when it is clear from the context. We write $-a$ for the additive inverse to an element a in R and write $a - b$ as a shorthand for $a + (-b)$, for a and b in R. Also, we usually write 0_R for the additive identity element of R. Note that our definition of a ring is strictly a "ring with multiplicative identity" but the distinction will not be necessary in the following. A ring R is said to be *commutative* if its multiplication is commutative. For $m \in \mathbb{N}$, we write $\mathbb{Z}_m$ for the finite ring consisting of the m integers $\{0, \ldots, m-1\}$ where addition and multiplication is defined as the corresponding operation over the integers modulo m.

An *ideal* of a ring R is a subset $I \subseteq R$ that is an additive subgroup of R and is closed under multiplication by elements from R; that is, whenever a belongs to R and b belongs to I, then ab and ba belong to I. For example, in the ring of integers $\mathbb{Z}$, the set of even numbers $2\mathbb{Z} := \{2n \mid n \in \mathbb{Z}\}$ is an ideal: it forms a group under integer addition and the product of an arbitrary integer and an even number is always even.

Let I be an ideal of a ring R. Define an equivalence relation $\simeq_I$ on R by $a \simeq_I b$ if and only if $a - b \in I$. If a is an element of R then we write $[\![a]\!]_I$ for the equivalence class of a in $R/\simeq_I$. The *quotient ring of R modulo I* is the set $R/I := \{[\![a]\!]_I \mid a \in R\}$, with ring structure defined for all a and b in R by

$$[\![a]\!]_I + [\![b]\!]_I := [\![a + b]\!]_I \text{ and } [\![a]\!]_I \cdot [\![b]\!]_I := [\![a \cdot b]\!]_I.$$

It can be easily verified that this is a well-defined ring, with multiplicative identity $[\![1_R]\!]_I$. In particular, it can be seen that R/I is commutative if R is commutative. As an example, consider the ring $\mathbb{Z}$ and the ideal $2\mathbb{Z}$ of even numbers, as above. Then $\mathbb{Z}/(2\mathbb{Z})$ is a ring that contains two elements (equivalence classes): one for the even numbers and one for the odd numbers. More generally, for $m \in \mathbb{N}$, the quotient ring $\mathbb{Z}/(m\mathbb{Z})$ contains exactly m elements. It can be seen that for each $m \in \mathbb{N}$, $\mathbb{Z}/(m\mathbb{Z}) \cong \mathbb{Z}_m$ under the isomorphism that maps the equivalence class of $k \in \mathbb{Z}$ to the integer $k \mod m$.

Fields. A *field* is a commutative ring F that contains at least two elements and for which every non-zero element has a multiplicative inverse. In other words, a ring F is a field if the set $F^\times := F \setminus \{0_F\}$ of non-zero elements of F is a group with respect to multiplication. It can be seen that the ring of integers $\mathbb{Z}$ is not a field (there is no integer x such that $2x = 1$) whereas the set of rational numbers $\mathbb{Q}$ is a field, where each non-zero rational number a has multiplicative inverse $\frac{1}{a}$.

Let F be a field. If there is a natural number n such that for any a in F, the element

$$n \times a := \underbrace{a + a + \ldots a + a}_{n \text{ times}},$$

obtained by adding a to itself n times, is 0_F then the *characteristic of* F is the least such n; otherwise, if there is no such n then F is said to have characteristic zero. It is not hard to show that the characteristic of any field is either zero or a prime number.

A *subfield* F of a field E is a subset $F \subseteq E$ which itself is a field under the operations of addition and multiplication defined in E. A field E containing F as a subfield is called an *extension of* F. A field extension E of F can be regarded as a vector space over F in the obvious way (E is the set of vectors, F the field of scalars). The degree of E over F, written $[E : F]$, is the dimension of E as an F-vector space. A field extension E over F is said to be finite if $[E : F]$ is finite and infinite otherwise. As an example, the field of complex numbers $\mathbb{C}$ is a field extension of degree two over the field of real numbers $\mathbb{R}$. On the other hand, it can be seen by a countability argument that $\mathbb{R}$ has infinite degree as an extension of $\mathbb{Q}$: the set $\mathbb{Q}$ is countable and every finite-dimensional vector space over a countable set must be countable, which $\mathbb{R}$ is of course not.

Polynomials and polynomial rings. Let R be a commutative ring. We write $R[X]$ for the set of polynomials in indeterminate X with coefficients from R. It can be seen that $R[X]$ forms a ring under addition and multiplication of polynomials, called the *polynomial ring of R in indeterminate X*. A polynomial $f \in R[X]$ is said to be a *constant polynomial* if there is an element $a \in R$ such that $f = a$. When F is a field, we consider certain minimal elements of the polynomial ring $F[X]$, defined as follows.

Definition 2.14 (Irreducible polynomial)**.** Let F be a field. A polynomial $f \in F[X]$ is said to be *irreducible over* F if f has positive degree and for any $g, h \in F[X]$, $f = gh$ implies that either g or h is a constant polynomial. ■

2.7.2 Finite fields

A *finite field* is a field with finitely many elements. A finite field F will have p^d elements, where the prime p is the characteristic of F and d is the degree of F over its prime subfield. Moreover, it can be shown that for each prime p and each positive integer d, there exists a finite field with p^d elements which is unique up to isomorphism. For further details, see e.g. Lidl and Niederreiter [53, Chapter 2.1]. We write GF_{p^d} for *the* finite field with p^d elements.

We write $F^\times$ to denote the multiplicative group of nonzero elements of a finite field F. The multiplicative group of a finite field is always cyclic (see e.g. Lang [51, Theorem 5.3]). A generator of the cyclic group $F^\times$ is called a *primitive element* and there are exactly $\varphi(p-1)$ primitive elements in F, where φ is Euler's totient function (that is, $\varphi(n)$ is the number of integers in $[n]$ which are co-prime to n).

Now consider the *prime field* $F = \mathsf{GF}_p$, where p is prime. Then $F \cong \mathbb{Z}/(p\mathbb{Z}) \cong \mathbb{Z}_p$ where the field $\mathbb{Z}_p$ consists of the integers $\{0, \ldots, p-1\}$, with addition and multiplication carried out modulo p, as discussed before. However, when $d > 1$ it is not true that $\mathsf{GF}_{p^d} \cong \mathbb{Z}/(p^d\mathbb{Z})$, the quotient ring of $\mathbb{Z}$ by the ideal $p^d\mathbb{Z}$. To see this, note that the element $p \in \mathbb{Z}/(p^d\mathbb{Z})$ does not have a multiplicative inverse in $\mathbb{Z}/(p^d\mathbb{Z})$. Suppose, towards a contradiction, that q is the

inverse of p in $\mathbb{Z}/(p^d\mathbb{Z})$, i.e. $pq \equiv 1 \pmod{p^d}$. Then

$$p^{d-1}pq = p^{d-1}(pq) \equiv p^{d-1}(1) \equiv p^{d-1} \not\equiv 0 \pmod{p^d},$$

but at the same time

$$p^{d-1}pq = (p^{d-1}p)q) = p^d q \equiv 0 \pmod{p^d}.$$

Instead, a standard representation of the elements of GF_{p^d} is obtained as follows. Let $K = \mathsf{GF}_p$ be the finite field with p elements. Then for every $n \geq 1$, there exists a monic irreducible polynomial of degree n over K. Indeed, the number $N_p(n)$ of such polynomials can be given explicitly by

$$N_p(n) = \frac{1}{n}\sum_{k|n} \mu(k) p^{n/k},$$

where the Moebius function $\mu : \mathbb{N}_0 \to \mathbb{N}_0$ is defined by

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1, \\ (-1)^r & \text{if } n \text{ is the product of } r \text{ distinct primes}, \\ 0 & \text{if } n \text{ is divisible by a square of a prime.} \end{cases}$$

A rough estimate gives a lower bound of

$$N_p(n) \geq \frac{1}{n}(p^n - \frac{p^n - p}{p-1}) > 0.$$

For more details, see Lidl and Niederreiter [53, Chapter 2]. Now let $f \in K[X]$ be a monic irreducible polynomial $f(X)$ of degree d over K. Then $F \cong K[X]/(f(X))$, the quotient of the polynomial ring $K[X]$ and the ideal generated by $f(X)$. That is, we can consider the elements of F to be polynomials of degree less than d. Addition and multiplication in this representation is carried out by adding and multiplying together polynomials and reducing the result modulo $f(X)$. Note that in §4.2.2 we consider an alternative way to represent the elements of F, as $d \times d$ matrices with elements from the prime field K.

2.7.3 Graphs

A *directed graph* is a pair $G = (V, E)$ where V is a finite set of *vertices* and the set of *edges* E is an irreflexive binary relation on V. An edge $(v, w) \in E$ is considered to be directed from v to w. In this case, we say that w is a *direct successor* of v and v is a *direct predecessor* of w. We write $N_{\text{out}}(v) := \{w \in V \mid (v, w) \in E\} \subseteq V$ for the set of direct successors of v and $N_{\text{in}}(v) := \{w \in V \mid (w, v) \in E\} \subseteq V$ for the set of direct predecessors of v. The *out-degree* of a vertex v, $\deg_{\text{out}}(v)$, is the number $\|N_{\text{out}}(v)\|$ of its direct successors and the *in-degree* of v, $\deg_{\text{in}}(v)$, is the number $\|N_{\text{in}}(v)\|$ of its direct predecessors.

A *graph* is a pair $G = (V, E)$ where the set of edges E is a collection of two-element subsets of the vertex set V. Thus, a graph can be seen as a directed graph with a symmetric edge relation. Observe that we consider only simple graphs, that is graphs that are loop-free and without parallel edges. We often write vw instead of $\{v, w\}$ to denote an edge between v and w in a graph G. An edge $e \in E$ is said to be *incident to* v if v is one of the end points of e, that is if $v \in e$. The degree of v, $\deg(v)$, is the number of edges incident to v. If $v \in V$ is a

vertex of G, then we write $N(v) := \{w \in V \mid vw \in E\} \subseteq V$ for the set of neighbours of v and $E(v) := \{vw \mid w \in N(v)\} \subseteq E$ for the set of edges incident to v. If G is a graph (directed or undirected), then we write $V(G)$ and $E(G)$ for the vertex set and edge set of G, respectively.

An *orientation* of a graph $G = (V, E)$ is a directed graph $\vec{G} = (V, \vec{E})$ which is obtained by orienting the edges of G; that is, for each edge $vw \in E$, exactly one of (v, w) and (w, v) is in $\vec{E}$ and for every $(v, w) \in \vec{E}$, vw is an edge in E.

Definition 2.15 (Disjoint union of graphs)**.** Let $(G_i)_{i\in I}$ be a family of graphs, indexed by a non-empty finite set I. The *disjoint* union $\dot{\bigcup}_{i\in I} G_i$ of the graphs G_i is a graph defined by

$$V(\dot{\bigcup}_{i\in I} G_i) := \{(v, i) \mid v \in V(G_i), i \in I\} \text{ and}$$
$$E(\dot{\bigcup}_{i\in I} G_i) := \{(v, i)(w, i) \mid vw \in E(G_i), i \in I\}.$$

In particular, we write $G \dot{\cup} H$ for the disjoint union of a pair of graphs G and H. ■

2.8 Linear algebra

We review some basic linear algebra and introduce *unordered* matrices, whose rows and columns are indexed by arbitrary unordered sets. For more background on matrix theory and linear algebra see e.g. Horn and Johnson [42] and for further details on the study of unordered matrices see Blass et al. [9].

2.8.1 Matrices and linear maps

Let R be a commutative ring. An $m \times n$ *matrix over* R is a rectangular array of scalars from R, consisting of m rows and n columns. We write $m \times n$ for the *dimension* of A. An $m \times n$ matrix is said to be *square* (of order n) if $m = n$. We write $A = (a_{ij})_{1\le i\le m, 1\le j\le n}$ (or $A = (a_{ij})$ for short when the dimension of A is clear from the context) to denote the matrix

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}.$$

When $R = F$ is a field, a matrix over F can alternatively be seen to represent a linear map between two finite-dimensional vector spaces, given a basis for each vector space. To see this, we first recall some basic definitions from elementary linear algebra. Let U be an n-dimensional vector space over a field F and let $B = \{u_1, \ldots, u_n\}$ be an *ordered* basis for U, where $u_1 < u_2 < \cdots < u_n$. With respect to this basis, we can represent each element x of U as an n-tuple $[x]_B := (a_1, \ldots, a_n) \in F^n$, where $x = a_1u_1 + \cdots + a_nu_n$ is the unique expression of x in terms of the basis elements of B. The scalars a_i are called the *coordinates* of x with respect to the basis B and $[x]_B$ is the unique *B-coordinate representation of x*. It is not hard to see that the map $U \to F^n$, $x \mapsto [x]_B$ is an isomorphism of vector spaces.

Now consider an n-dimensional vector space U and an m-dimensional vector space V over the same scalar field F. Let $B_U = \{u_1, \ldots, u_n\}$ be a basis for U and let $B_V = \{v_1, \ldots, v_m\}$ be a basis for V. If $T : U \to V$ is a linear map then we construct an $m \times n$ matrix A as follows. For each $u_i \in B_U$, let $[T(u_i)]_{B_V} = a_{1i}v_1 + \cdots + a_{mi}v_m$ and let $A := (a_{ij})$ denote the matrix obtained by gathering all the coefficients a_{ij}. We call A the *matrix representing T with respect to bases B_U and B_V*. It can be seen that for any $x \in U$, $[T(x)]_{B_V} = A([x]_{B_U})$. Observe that

here the ordering of each basis is important. That is, changing the ordering of a given basis amounts to permuting the rows and columns of the associated matrix representation.

By this discussion, every linear map can be represented by a matrix, given suitable bases for both its domain and co-domain. Moreover, every matrix can be seen as a representation of some linear map. More specifically, let A be an $m \times n$ matrix over a field F. Then A is the matrix of the linear map $T : F^n \to F^m$, defined for all $x \in F^n$ by $T(x) := Ax$, where x is seen as a column vector of length n over F. The *image* of A is the image $\{Ax \mid x \in F^n\} \subseteq F^m$ of the associated linear map and the *null-space* (or *kernel*) of A is $\{x \in F^n \mid Ax = 0\} \subseteq F^n$. The *rank* of the matrix A is the dimension of its image and the *nullity* of A is the dimension of its null-space. A fundamental result of elementary linear algebra is the relation

$$n = \text{dimension of the image of } A + \text{dimension of the null-space of } A,$$

often referred to as the *rank-nullity theorem*.

2.8.2 Matrices indexed by unordered sets

Rank and nullity are two examples of matrix properties that are in fact properties of the underlying linear map that the matrix represents. The same holds for many common matrix properties that we focus on in linear algebra; for example determinant and singularity. It follows from the above discussion that such natural matrix properties are invariant under permutation of the rows and columns of the matrix, since the associated linear map is invariant under a permutation of the chosen vector space bases. With this in mind, it is natural to consider a more general notion of a matrix, where the rows and columns are indexed by arbitrary *unordered* sets.

Let R be a commutative ring and let I, J be finite, non-empty sets. An $I \times J$ *matrix* A over R is a function $A : I \times J \to R$. Here the rows of A are indexed by I and the columns of A are indexed by J. We write $A = (a_{ij})$ to denote that $A(i, j) = a_{ij}$ for all $i \in I$ and $j \in J$. If $I = J$ then A is called a *square matrix*. We write $M_{I\times J}(R)$ for the set of $I \times J$ matrices over R, and let $M_I(R) := M_{I\times I}(R)$. If A is an $I \times J$ matrix then the *dimension* of A is $\|I\| \times \|J\|$.

By taking $I = [m]$ and $J = [n]$ we recover the more familiar notion of an $m \times n$ matrix A from above; i.e. a rectangular array of elements consisting of m rows and n columns. Most natural matrix properties from linear algebra apply directly to matrices indexed by arbitrary sets as discussed above; in the following we review a few of the relevant ones, where we write R to denote a commutative ring.

Matrix addition and multiplication. Addition and multiplication of unordered matrices is defined in exactly the same way as for ordered matrices, except that we now have to ensure that the index sets of the two matrices, and not just their dimension, are matching. That is, if A and B are two $I \times J$ matrices then their sum $A + B$ is the $I \times J$ matrix defined for all $i \in I$ and $j \in J$ by $(A + B)(i, j) := A(i, j) + B(i, j)$. Similarly, if A is an $I \times K$ matrix and B is a $K \times J$ matrix, where all index sets are finite and non-empty, then the product of A and B is the $I \times J$ matrix AB defined for all $i \in I$ and $j \in J$ by $(AB)(i, j) := \sum_{k\in K} A(i, k)B(k, j)$.

Determinant and trace. Let I be a finite and non-empty set and consider an $I \times I$ square matrix A over R. The *determinant* of A, written $\det(A)$, is given by

$$\det(A) := \sum_{\sigma \in \mathrm{Sym}(I)} \mathrm{sgn}(\sigma) \prod_i a_{i\sigma(i)},$$

where the sum is taken over the symmetric group $\mathrm{Sym}(I)$ of all permutations of I. Here $\mathrm{sgn}(\sigma)$ denotes the sign of the permutation σ, defined by $\mathrm{sgn}(\sigma) := (-1)^m$ where m is the number of transpositions of pairs of elements that must be composed to build up the permutation σ. Note that the determinant of a matrix over R is an element in R. This definition agrees with the usual definition of the determinant of an $n \times n$ matrix. This is because if we linearly order the index set I, then we obtain an $\|I\| \times \|I\|$ matrix in the usual sense and the determinant of this matrix is independent of the ordering, since by changing the ordering we are effectively permuting the rows and columns in the same way, which preserves the value of the determinant[3].

The *trace* of an $I \times I$ matrix $A = (a_{ij})$ over a commutative ring R, written $\mathrm{tr}(A)$, is defined by $\mathrm{tr}(A) := \sum_{i \in I} a_{ii}$. That is, the trace of A is just the sum of the entries along the main diagonal.

Rank and singularity over a field. Let F be a field and consider a finite set I. We write F^I for the space of all functions $g : I \to F$. This can be given the structure of a vector space, where addition of two vectors g and f in F^I is the function $x \mapsto g(x) + f(x)$ in F^I and multiplication of a vector $g \in F^I$ by a scalar $a \in F$ is the function $x \mapsto a \cdot g(x)$ in F^I. When I is linearly ordered, this definition agrees with the standard notion of a Cartesian $\|I\|$-space over F.

Now consider an $I \times J$ matrix A over F. The *rank* of A, written $\mathrm{rank}(A)$, is defined as the dimension of the image $\{Ax \mid x \in F^J\} \subseteq F^I$, as a subspace of F^I. It can be seen that the rank of an unordered matrix in this sense agrees with the usual definition of matrix rank, since the rank of a matrix is invariant under permutation of its rows and columns. When $\|I\| = \|J\| = n$, then we say that A is *non-singular* if $\mathrm{rank}\, A = n$ and *singular* otherwise. By elementary linear algebra, it follows that A is non-singular if and only if there is a $J \times I$ matrix B for which it holds that AB is the $I \times I$ identity matrix (equivalently, such that BA is the $J \times J$ identity matrix). Such a matrix is also said to be *invertible* and we write A^{-1} to denote its inverse. For the case when $I = J$, it can be seen that a square $I \times I$ matrix A over F is singular if and only if $\det(A) = 0$.

Consider again non-empty finite sets I and J of the same cardinality. Two square matrices $A \in M_I(F)$ and $B \in M_J(F)$ are said to be *similar* if there exists a non-singular $I \times J$ matrix S

[3] Alternatively, we could have considered a determinant function which applies to arbitrary $I \times J$ matrices where $\|I\| = \|J\| = n$ (that is, I and J are not necessearily the same set). This problem was considered briefly by Blass, Gurevich and Shelah in [9]. However, it can be seen that when $I \neq J$ then the determinant of an unordered $I \times J$ matrix only corresponds with the determinant of an ordered $n \times n$ matrix *up to sign*. This is because if we linearly order the index sets I and J then the determinant of the corresponding $n \times n$ matrix depends on the chosen orderings since we are effectively permuting the rows and columns *independently*. To see how this affects the sign of the determinant, recall that for any $n \times n$ matrix M and $n \times n$ permutation matrices P and Q, $\det(PMQ) = \det(P)\det(M)\det(Q) = (-1)^i \det(M)(-1)^j$, where $i, j \geq 0$ are integers depending on P and Q, respectively (see e.g. Horn and Johnson [42]). Clearly when $Q = P^{-1}$ (that is, when we permute the rows and columns in the same way), then $\det(PMP^{-1}) = \det(M)$. Otherwise, however, $\det(PMQ)$ may differ in sign from $\det(M)$.

over F such that

$$B = S^{-1}AS.$$

The transformation $A \mapsto S^{-1}AS$ is called a *similarity transformation* by the *similarity matrix* S.

Definition 2.16 (Bijection matrices)**.** Let F be a field and let I and J be finite sets with $\|I\| = \|J\| > 0$. For a bijection $\pi : I \to J$, write B_π for the $I \times J$ *bijection matrix* over F, defined for all $i \in I$ and $j \in J$ by

$$B_\pi(i,j) := \begin{cases} 1 & \text{if } \pi(i) = j, \\ 0 & \text{otherwise.} \end{cases}$$

■

It can be easily verified that a bijection matrix B_π is invertible, with its inverse explicitly given by $B_\pi^{-1} = B_{\pi^{-1}}$. Multiplying a $J \times J$ matrix A on the left by an $I \times J$ bijection matrix B_π results in a relabeling of the rows of A according to π. That is, $B_\pi A$ is an $I \times J$ matrix given by $(B_\pi A)(i,j) = A(\pi(i), j)$. Similarly, multiplying A on the right by B_π^{-1} results in a relabeling of the columns of A according to π^{-1}. By simultaneously applying the bijection π to the rows and columns of A, we obtain the $I \times I$ matrix $B_\pi A B_\pi^{-1}$ given by $(B_\pi A B_\pi^{-1})(i,k) = A(\pi(i), \pi(k))$, for all $i, k \in I$. Finally, note that when $I = J$, then an $I \times I$ bijection matrix agrees with the usual notion of a permutation matrix on I.

2.9 Logics and complexity classes

We conclude this chapter by reviewing some common concepts in both computational and descriptive complexity theory. For further background on descriptive complexity see e.g. Ebbinghaus and Flum [23] while Papadimitriou's textbook [59] is an excellent reference on computational complexity.

2.9.1 Computational complexity

We briefly recall the definition of some of the common complexity classes we will frequently encounter in this thesis. We write PTIME to denote the set of languages decidable in deterministic polynomial time and write NP to denote the set of languages decidable in non-deterministic polynomial time. We also consider space-bounded computation, where the computational model is a Turing machine with a separate work tape. Since only the space used on the work tape is counted towards the space usage during a computation, this model allows us to consider sub-linear space complexity. In particular, we consider *logspace* computations, where the amount of space used is at most logarithmic in the input size. In this way, we write L to denote the class of languages decided by a deterministic logspace machine and write NL to denote the class of languages decided by a non-deterministic logspace machine.

2.9.2 Logics capturing complexity classes

Intuitively, a logic L captures a complexity class C on a class of finite structures $\mathcal{K}$ if the L-definable properties of structures in $\mathcal{K}$ are precisely those that are decidable in C. In order to define this relation more formally, we need to establish some notation for encoding finite relational structures as strings over $\Sigma := \{0,1\}$. Our presentation follows that of Libkin [52].

Let $\tau = (R_1, \ldots, R_s, c_1, \ldots, c_t)$ be a vocabulary, where the R_i are relation symbols and the c_i are constants. Let $\mathbf{A}$ be a τ-structure of size n and consider a linear ordering $<$ of $U(\mathbf{A})$. Write $U(\mathbf{A}) = \{a_1, \ldots, a_n\}$ where the elements of $U(\mathbf{A})$ are ordered $a_1 < a_2 < \cdots < a_n$ by $<$. A string encoding of $\mathbf{A}$, with respect to the ordering $<$, can now be defined as follows. For each k-ary relation symbol $R \in \tau$, consider an enumeration of all k-tuples of elements of $U(\mathbf{A})$, in the lexicographic ordering induced by $<$. That is, we enumerate k-tuples as

$$(a_1, \ldots, a_1), (a_1, \ldots, a_2), \ldots, (a_n, \ldots, a_{n-1}), (a_n, \ldots, a_n),$$

and write $\vec{a}_j$ for the j-th tuple in this enumeration. Then the relation $R^{\mathbf{A}}$ is encoded by an n^k-bit string $\mathsf{enc}(R^{\mathbf{A}}, <)$ where the j-th bit of $\mathsf{enc}(R^{\mathbf{A}}, <)$ is 1 if and only if $\vec{a}_j \in R^{\mathbf{A}}$ and 0 otherwise.

Constants can be encoded similarly, by viewing each constant as a unary relation containing exactly one element. Putting this all together, we write $\mathsf{enc}(\mathbf{A}, <)$ for the string encoding of $\mathbf{A}$ with respect to $<$ defined by

$$\mathsf{enc}(\mathbf{A}, <) := 0^n 1 \cdot \mathsf{enc}(R_1^{\mathbf{A}}, <) \cdots \mathsf{enc}(R_s^{\mathbf{A}}, <) \cdot \mathsf{enc}(c_1^{\mathbf{A}}, <) \cdots \mathsf{enc}(c_t^{\mathbf{A}}, <),$$

where $a \cdot b$ denotes the concatenation of strings a and b.

Let $\mathcal{K}$ be a class of finite τ-structures. Overloading our notation, we write $\mathsf{enc}(\mathcal{K}) \subseteq \Sigma^*$ to denote the language defined by

$$\mathsf{enc}(\mathcal{K}) := \{\mathsf{enc}(\mathbf{A}, <) \mid \mathbf{A} \in \mathcal{K} \text{ and } < \text{ a linear ordering of } U(\mathbf{A})\}.$$

We can now define the capturing relation between complexity classes and logics more formally as follows.

Definition 2.17 (Logics capturing complexity classes)**.** Let C be a complexity class, L a logic, and $\mathcal{K}$ a class of finite structures.

- We write $\mathrm{L} \leqq_{\mathcal{K}} \mathrm{C}$ if for every L-sentence φ, there is a language $A \in \mathrm{C}$ such that

  $$\mathsf{enc}(\mathrm{Mod}(\varphi)) \cap \mathsf{enc}(\mathcal{K}) = A \cap \mathsf{enc}(\mathcal{K}).$$

  In other words, $\mathrm{L} \leqq_{\mathcal{K}} \mathrm{C}$ if for every L-sentence φ, the problem of deciding if $\mathbf{A} \models \varphi$, given $\mathbf{A} \in \mathcal{K}$, belongs to C.

- We write $\mathrm{C} \leqq_{\mathcal{K}} \mathrm{L}$ if for every class $\mathcal{K}_P \subseteq \mathcal{K}$ for which there is a language $A \in \mathrm{C}$ with $A \cap \mathsf{enc}(\mathcal{K}) = \mathsf{enc}(\mathcal{K}_P)$, there is an L-sentence φ_P such that $\mathrm{Mod}(\varphi) \cap \mathcal{K} = \mathcal{K}_P$.

  In other words, $\mathrm{C} \leqq_{\mathcal{K}} \mathrm{L}$ if for every property P of structures from $\mathcal{K}$ which can be decided with complexity C, there is an L-sentence φ_P for which it holds that for every $\mathbf{A} \in \mathcal{K}$, $\mathbf{A} \models \varphi_P$ if and only $\mathbf{A}$ has the property P.

- Finally, we say that L *captures* C *on* $\mathcal{K}$, and write $\mathrm{L} \equiv_{\mathcal{K}} \mathrm{C}$, if $\mathrm{C} \leqq_{\mathcal{K}} \mathrm{L}$ and $\mathrm{L} \leqq_{\mathcal{K}} \mathrm{C}$.

■

Note that when $\mathcal{K}$ is the class of all finite structures, then we usually omit the subscript to the above relations, and simply write $\mathrm{L} \equiv \mathrm{C}$, $\mathrm{C} \leqq \mathrm{L}$ and $\mathrm{L} \leqq \mathrm{C}$.

Chapter 3

Linear algebra in fixed-point logic with counting

The results of Atserias, Bulatov and Dawar [4] show that the problem of deciding solvability of systems of linear equations over a finite field is not definable in IFPC. Recall that by elementary linear algebra, a system of linear equations $A\mathbf{x} = \mathbf{b}$ over a field is solvable if and only if $\text{rank}(A \mid \mathbf{b}) = \text{rank}(A)$, where $(A \mid \mathbf{b})$ is the matrix obtained from A by adding the column vector $\mathbf{b}$ on the right. It then follows that IFPC is not expressive enough to define the rank of a matrix over a finite field. However, this result does not directly imply the non-definability of other important matrix properties. In particular, is IFPC expressive enough to define the *determinant* of a matrix?

In this chapter we study the descriptive complexity of various problems in linear algebra. This follows up on the work of Blass, Gurevich and Shelah [9], who considered the problem of deciding if a matrix is singular. This problem lies in PTIME and it is shown that it can be expressed in IFPC for both integer and finite field matrices. Recall that over a field, a matrix A is singular if and only if its determinant $\det(A)$ is zero. Blass et al. [9] note that over the two-element field GF_2, the determinant can therefore be expressed in IFPC by testing for singularity. In this chapter we generalise this result, by showing that over any finite field the characteristic polynomial (and thereby, the determinant) of a matrix can be defined in IFPC. The same result is obtained for matrices with integer and rational entries. Moreover, we show that for matrices over the field of rationals, both the rank and the minimal polynomial can also be defined in IFPC. This demonstrates that it is really the inability of IFPC to define matrix rank over *finite* fields that separates the logic from PTIME.

We begin this chapter in §3.1 by defining a representation of matrices over $\mathbb{Z}$, $\mathbb{Q}$ and finite fields as finite relational structures. In §3.2 we show that various structural properties of finite fields, given explicitly by their addition and multiplication tables, can be defined in IFPC. In particular, we show that a linear ordering can be defined over any finite field and that over fields of non-prime cardinality, we can define a representation of the field elements in terms of polynomials in a certain polynomial ring.

We shift our attention to integer- and rational-valued matrices in §3.3, where we consider the definability of various arithmetic operations on matrices. We show that the product AB of matrices A and B, matrix powers A^m, for $m \geq 0$, and the trace $\text{tr}(A)$ can all be defined in IFPC. The results of the previous sections are combined in §3.4 to show that the characteristic

polynomial and the determinant of integer and rational matrices can be defined in IFPC. By an appropriate translation of the underlying field, this implies that the characteristic polynomial and the determinant of matrices over finite fields can also be defined. This extends the results of Blass, Gurevich and Rossman [7], who show that the characteristic polynomial can be defined in the formalism of choiceless polynomial time with counting, which subsumes IFPC.

Finally, §3.5 focuses on matrices with elements from the field of rationals. For such matrices, it is shown that both the rank and the minimal polynomial can be defined in IFPC.

3.1 Matrices as relational structures

We consider matrices whose rows and columns are indexed by arbitrary sets, not necessarily ordered. Let R be a commutative ring with a multiplicative identity and let I, J be finite, non-empty sets. We consider an $I \times J$ matrix A over R as a function $A : I \times J \to R$. Here the rows of A are indexed by I and the columns of A are indexed by J. We write $A = (a_{ij})$ to denote that $A(i, j) = a_{ij}$ for all $i \in I$ and $j \in J$. We write $M_{I\times J}(R)$ for the set of $I \times J$ matrices over R, and let $M_I(R) := M_{I\times I}(R)$.

In the following we consider matrices over three kinds of domain: finite fields, the ring of integers and the field of rationals.

3.1.1 Matrices over finite fields

Over a finite field F, we can represent a matrix $A = (a_{ij}) \in M_{I\times J}(F)$ as a finite, relational structure. We consider two different representations.

- **The field F is part of the vocabulary.** We consider A as a two-sorted structure $\mathbf{A}$ over the vocabulary $\tau_F = \{M_f \mid f \in F\}$, where M_f is a binary relation for each $f \in F$. The two sorts of $\mathbf{A}$ are the row sort I and the column sort J. The relations M_f are interpreted as
$$M_f^{\mathbf{A}} = \{(i, j) \in I \times J \mid a_{ij} = f\},$$
for each $f \in F$.

- **The field F is part of the structure.** We consider A as a three-sorted structure $\mathbf{A}$ with row sort I, column sort J and domain sort D. Here the last sort is interpreted as the elements of the field F. Write $\tau_{\text{field}} := \{+_{\text{f}}, \times_{\text{f}}, 0_{\text{f}}, 1_{\text{f}}\}$ for the vocabulary of fields. Then the vocabulary of $\mathbf{A}$ is $\tau_{\text{fmat}} := \{M\} \cup \tau_{\text{field}}$, where $(D, +_{\text{f}}^{\mathbf{A}}, \times_{\text{f}}^{\mathbf{A}}, 0_{\text{f}}^{\mathbf{A}}, 1_{\text{f}}^{\mathbf{A}})$ is the field F and the ternary relation M is interpreted as
$$M^{\mathbf{A}} = \{(i, j, d) \in I \times J \times D \mid a_{ij} = d\}.$$

Hereafter, we will assume that all matrices over finite fields are given as finite τ_{fmat}-structures in this way. The benefit of this representation is that it allows us to consider fields that are not fixed. Note that this is without any loss of generality, for it can be seen that for each finite field F, there is a first-order interpretation Γ of τ_{fmat} in τ_F, such that for every τ_F-structure $\mathbf{A}$, $\mathbf{A}$ and $\Gamma(\mathbf{A})$ represent the same matrix.

Remark. Both these representations could also be used to describe matrices over a finite *ring* R. However, we do not consider finite rings on their own (that is, other than as a part of a finite field) anywhere in this thesis, so this will not be studied further.

3.1.2 Integer and rational matrices

To represent unordered integer and rational matrices as finite structures, we follow the convention of Blass et al. [9] and write matrix entries in binary notation. Let $A = (a_{ij}) \in M_{I\times J}(\mathbb{Z})$ be an integer matrix and let $m = \max\{\mathrm{abs}(a_{ij}) \mid i \in I,\ j \in J\}$ be the maximum absolute value of integers appearing in A. Let $b = \lceil \log_2(m) \rceil$, $B = [0, b]$ and write $\mathrm{bit}(x, k)$ to denote the k-th least-significant bit in the binary expansion of $x \in \mathbb{Z}$. Then we consider A as a three-sorted structure $\mathbf{A}$, with row sort I, column sort J and bit sort B, over the vocabulary $\tau_{\mathbb{Z}} = \{M, P, \leq_B\}$. Here $\leq_B$ is interpreted as a linear ordering of B, $P^{\mathbf{A}} = \{(i, j) \in I\times J \mid a_{ij} \geq 0\}$ identifies the non-negative elements of A, and the ternary relation M is interpreted as

$$M^{\mathbf{A}} = \{(i, j, k) \in I \times J \times B \mid \mathrm{bit}(\mathrm{abs}(a_{ij}), k) = 1\}.$$

That is, $(i, j, k) \in M^{\mathbf{A}}$ when "the k-th bit in the binary expansion of $\mathrm{abs}(a_{ij})$ is 1". Observe that the role of $\leq_B$ is only to order the set of bit positions B, which we commonly view as an initial segment of the integers. In particular, the rows and columns of the matrix $M^{\mathbf{A}}$ are themselves unordered.

Matrices with rational entries can be treated similarly by handling numerators and denominators of matrix elements separately. That is, we consider matrices over the vocabulary $\tau_{\mathbb{Q}} = \{M_{\mathrm{n}}, M_{\mathrm{d}}, P, \leq_B\}$, where $\leq_B^{\mathbf{A}}$ and $P^{\mathbf{A}}$ are defined as before, and the ternary relations $M_{\mathrm{n}}^{\mathbf{A}}$ and $M_{\mathrm{d}}^{\mathbf{A}}$ define the numerators and denominators of elements in A, respectively.

3.2 Describing finite fields in IFPC

In our chosen representation of finite-field matrices, the underlying field is given explicitly as a part of the matrix structure by its addition and multiplication tables. In [9], Blass et al. consider a similar representation, where they assume that the field elements are linearly ordered. In this section we show that this assumption is not necessary in the current context, as we can already define a linear ordering over any finite field in FOC+DTC $\leqq$ IFPC. Here, FOC+DTC is the extension of FOC with operators for defining deterministic transitive closure. Moreover, we also show how to define in FOC+DTC or IFPC many important structural properties of finite fields. In particular, for a field F of cardinality p^d, with $d > 1$ and p prime, we define in IFPC a representation of elements of F as polynomials of degree less than d over $\mathbb{Z}_p$. This will play a crucial role in our construction of the characteristic polynomial in §3.4.

The remainder of this section is split into two parts. In §3.2.1 we consider fields F of prime cardinality p. Our main result is that an isomorphism $F \to \mathbb{Z}_p$ can be defined by a fixed formula of FOC+DTC over any field F of prime cardinality p. This in turn gives a way to canonically order the elements of F according to the natural ordering of the integers $\{0, \ldots, p-1\}$, as claimed. For the case when F has cardinality p^d, with $d > 1$, we show in §3.2.2 that there is a formula of FOC+DTC that defines the set of all primitive elements of F, which are the cyclic generators of the multiplicative group $F^\times$. It follows easily that for each primitive element $\alpha \in F$, there is an FOC+DTC-definable linear ordering of F, dependent

only on α. Since the number of primitive elements is generally greater than one, this of course does not give us a canonical ordering of F, but rather a collection of linear orderings. However, we generally only consider queries over ordered structures that are *order-invariant*, so this makes no difference with respect to definability as we can take the conjunction over the set of all orderings. Finally, we show that we can define the minimal polynomial of any primitive element by a formula of IFPC. This allows us to represent the elements of F as polynomials in a finite polynomial ring, as discussed above.

Remark. All the results here on definability in FOC+DTC could be stated directly in terms of IFPC, which is after all the focus of this chapter. However, the reason for emphasising FOC+DTC is that in Chapter 4 we will apply the results here also to extensions of first-order logic with rank operators, which subsume FOC+DTC but not IFPC.

3.2.1 Prime fields

Let F be a finite field with p elements, where p is prime. In this section we show how to define by a fixed formula of FOC+DTC an isomorphism $F \to \mathbb{Z}_p$. Here the field $\mathbb{Z}_p$ consists of the integers $\{0, \dots, p-1\}$, with addition and multiplication carried out modulo p. Specifically, we will prove the following lemma.

Lemma 3.1 (Isomorphism of prime fields)**.** *There is an* FOC+DTC *number term $\eta(z)$ in vocabulary $\tau^\star_{field}$, where z is an element variable, for which it holds that for any τ_{field}-structure* $\mathbf{F}$ *of prime cardinality p, the map*

$$\eta(z)^{\mathbf{F}^\star} := \{(f, \eta[f]^{\mathbf{F}^\star}) \mid f \in U(\mathbf{F})\}$$

is an isomorphism of fields $\mathbf{F} \to \mathbb{Z}_p$.

Proof. Given two prime fields F and K of the same cardinality p, we can explicitly construct an isomorphism $F \to K$ as follows. Since the field F has characteristic p, each element can be uniquely written in the form

$$k \cdot 1_{\mathrm{f}} = \underbrace{1_{\mathrm{f}} +_{\mathrm{f}} \cdots +_{\mathrm{f}} 1_{\mathrm{f}}}_{k \text{ times}},$$

where 1_{f} is the multiplicative identity of F and $0 \le k \le p-1$. Similarly, each element of G can be written uniquely in the form $k \cdot 1_G$, $0 \le k \le p-1$. It is easily verified that the map $\varphi : F \to G$, $k \cdot 1_{\mathrm{f}} \mapsto k \cdot 1_G$ is an isomorphism of fields.

Now suppose we have a formula $\psi(z, v)$ which relates an element variable z and a number variable v whenever $z = 1_{\mathrm{f}} \cdot v$, where 1_{f} denotes the multiplicative constant symbol of τ_{field}. By the above discussion, $\psi(z, v)$ is necessarily the graph of an injective function. We can then define the required number term $\eta(z)$ as follows:

$$\eta(z) \equiv \#_w(\exists v\, \exists \mu\, (\mu < v) \wedge \psi(z, v) \wedge \psi(w, \mu)),$$

which counts the number of elements w that appear before z in the sequence of elements $0_{\mathrm{f}}, 1_{\mathrm{f}}, 1_{\mathrm{f}} \cdot 2, 1_{\mathrm{f}} \cdot 3, \dots$.

It remains to show that we can define the formula $\psi(z, v)$ in FOC+DTC. Define a formula $\theta(x_1, v_1, x_2, v_2) \equiv (v_2 = v_1 + 1_N) \wedge (x_2 = x_1 +_{\mathrm{f}} 1_{\mathrm{f}})$, where we write + (without subscript)

for addition of integers over the number sort and write $+_{\mathrm{f}}$ for addition of field elements over the element sort. Viewed as a binary relation over pairs of elements of the type '(field element, number)', θ relates (x_1, v_1) to (x_2, v_2) exactly when x_2 is the successor of x_1, with respect to the ordering of the field elements, and v_2 is the successor of v_1, with respect to the ordering of the integers. It follows that $z = 1_{\mathrm{f}} \cdot \mu$ exactly when there is a path from $(0_{\mathrm{f}}, 0_G)$ to (z, μ) in the graph defined by θ. It can be seen that this graph is deterministic, and the desired formula is given by

$$\psi(z, v) \equiv [\mathbf{dtc}_{x_1\mu_1, x_2\mu_2}\theta](0_{\mathrm{f}}, 0_G, z, v).$$

□

As a direct corollary of this lemma, we see that there is a formula of FOC+DTC which defines a linear ordering over any prime field in vocabulary τ_{field}.

Corollary 3.2 (Linear ordering over prime fields)**.** *There is an* FOC+DTC*-formula $\varphi(x, y)$ in vocabulary $\tau^{\star}_{field}$, where x and y are element variables, for which it holds that for any τ_{field}-structure* $\mathbf{F}$ *of prime cardinality, the binary relation $\varphi(x, y)^{\mathbf{F}^{\star}}$ is a linear ordering of $U(\mathbf{F})$.* □

3.2.2 Prime-power fields

Let F be a finite field of cardinality p^d, where p is prime and $d > 1$ an integer, and write $K := \mathsf{GF}_p$ for the prime sub-field of F of cardinality p. The field F is commonly represented as a quotient ring $K[X]/(g(X))$ where $g(X)$ is a monic irreducible polynomial of degree d over K. This was explained in more detail in §2.7.2. One way to define a polynomial of this kind is to construct the minimal polynomial over K of some primitive element. Recall from §2.7.2 that a primitive element of F is any generator of the multiplicative group $F^{\times}$. The minimal polynomial for a primitive element $\alpha \in F$ over K is defined to be the least monic polynomial $f(X) \in K[X]$ such that $f(\alpha) = 0$. The polynomial $f(X)$ is irreducible over $K[X]$, by definition, and has degree d, as required (see e.g. Lidl and Niederreiter [53, Chapter 3]).

In this section we consider the definability of various properties of prime-power fields. In particular, we show that there is a formula of IFPC that defines over any τ_{field}-structure $\mathbf{F}$ a monic irreducible polynomial of degree d over GF_p, where $\|\mathbf{F}\| = p^d$. To do that, we first show that the collection of primitive elements of $\mathbf{F}$ can be defined by a formula of FOC+DTC. As a consequence of this construction, we obtain for each fixed primitive element α an FOC+DTC-definable ordering of the field $\mathbf{F}$. Finally, we show that for each primitive element α there is an IFPC formula that defines its minimal polynomial over $\mathbf{F}$.

The first step in our construction is to establish the following lemma, which shows that the operation of raising field elements to an integer power (that is, repeated multiplication in the field) can be defined in FOC+DTC.

Lemma 3.3 (Powering of field elements)**.** *Consider element variables x and y and a number variable v. There is an* FOC+DTC *formula* $\mathsf{pow}(x, v, y)$ *in vocabulary $\tau^{\star}_{field}$, for which it holds that for any τ_{field}-structure* $\mathbf{F}$ *and any $g, h \in U(\mathbf{F})$ and $m \in \mathbb{N}_0$,*

$$\mathbf{F}^{\star} \vDash \mathsf{pow}[h, m, g] \quad \Leftrightarrow \quad h^m := \underbrace{h \times_f^{\mathbf{F}} \cdots \times_f^{\mathbf{F}} h}_{m \text{ times}} = g.$$

The proof of this lemma is very similar to the proof of Lemma 3.1 above but we give the details for completeness.

Proof. Define a formula $\theta(z_1, \mu_1, z_2, \mu_2; x) \equiv (\mu_2 = \mu_1 + 1_N) \wedge (z_2 = z_1 \times_{\mathrm{f}} x)$, where we write + (without subscript) for addition of integers over the number sort and write $\times_{\mathrm{f}}$ for multiplication of field elements over the element sort. Here, x, z_1 and z_2 are element variables and μ_1 and μ_2 are number variables. Treating x as a parameter, we can view θ as a binary relation of type '(field element, number)' which relates (z_1, μ_1) to (z_2, μ_2) exactly when z_2 is z_1 multiplied by x and μ_2 is the successor of μ_1, with respect to the ordering of the integers. It follows that $y = x^v := x \times_{\mathrm{f}} \cdots \times_{\mathrm{f}} x$ (v times) exactly when either both y and v are zero (0_{f} and 0_N, respectively) or there is a path from $(x, 1_N)$ to (y, v) in the graph defined by θ with fixed parameter x. It can be seen that this graph is deterministic and the desired formula is given by

$$\mathsf{pow}(x, v, y) \equiv ((v = 0_N) \wedge (y = 0_{\mathrm{f}})) \vee [\mathbf{dtc}_{z_1\mu_1, z_2\mu_2}\theta](x, 1_N, z, v; x).$$

□

As a direct corollary of Lemma 3.3, we can see that the set of primitive elements of any finite field can be defined in FOC+DTC, simply by checking for each field element α if every other non-zero element can be expressed as a power of α. Clearly, this happens if and only if α is primitive.

Corollary 3.4 (Primitive elements)**.** *There is an* FOC+DTC *formula* $\mathsf{prim}(x)$ *in vocabulary* $\tau^\star_{field}$ *for which it holds that for any* τ_{field}*-structure* $\mathbf{F}$, $\mathsf{prim}(x)^{\mathbf{F}^\star}$ *is the collection of primitive elements of* $\mathbf{F}$. □

Let F be a finite field with multiplication written as $\times$ and multiplicative identity 1_F. For a primitive element $\alpha \in F$, we define the *α-order* of an element $g \in F^\times$ to be the integer m for which $\alpha^m = \alpha \times \cdots \times \alpha = g$. Here, α^0 is taken to be 1_F. Since the multiplicative group $F^\times$ is generated by α, the α-order is well-defined. The following corollary now follows immediately from Lemma 3.3 and Corollary 3.4.

Corollary 3.5 (α-order of field elements)**.** *Consider element variables x and y. There is an* FOC+DTC *number term* $\mathsf{ord}(x, y)$ *in vocabulary* $\tau^\star_{field}$*, for which it holds that for any* τ_{field}*-structure* $\mathbf{F}$*, any primitive element* $\alpha \in U(\mathbf{F})$ *and any* $g \in U(\mathbf{F})$, $\mathsf{ord}[\alpha, g]^{\mathbf{F}^\star} = m$ *is the α-order of g in* $\mathbf{F}$*; that is,*

$$\alpha^m := \underbrace{\alpha \times_f^{\mathbf{F}} \cdots \times_f^{\mathbf{F}} \alpha}_{m \text{ times}} = g.$$

□

For a fixed primitive element $\alpha \in F$, we can define a linear ordering $\leqslant_\alpha$ of F by setting $g \leqslant_\alpha h$ if and only if the α-order of g is at most the α-order of h, for all $g, h \in F^\times$, and setting $0_F \leqslant_\alpha g$ for all $g \in F$. Applying Corollary 3.5, this relation can clearly be defined in FOC+DTC, as stated in the following.

Corollary 3.6 (Linear ordering over prime-power fields)**.** *There is an* FOC+DTC*-formula* $\varphi(x, y, z)$ *in vocabulary* $\tau^{\star}_{field}$*, where* x*,* y *and* z *are element variables, for which it holds that for any* τ_{field}*-structure* $\mathbf{F}$ *and any primitive element* $\alpha \in U(\mathbf{F})$*, the binary relation*

$$\varphi(x, y, \alpha/z)^{\mathbf{F}^{\star}} = \{(g, h) \in U(\mathbf{F}) \mid \mathbf{F}^{\star} \vDash \varphi[g, h, \alpha]\}$$

is a linear ordering of $U(\mathbf{F})$*.* □

It remains to show that the minimal polynomial of a primitive element can be defined in IFPC. For that, we need to be able to encode polynomials over pairs $(\mathbf{F}, \alpha)$, where $\mathbf{F}$ is a τ_{field}-structure and $\alpha \in U(\mathbf{F})$ a primitive element. Here we crucially rely on being able to define a linear ordering $\leqslant_{\alpha}$ of $U(\mathbf{F})$ (with respect to α), as discussed above and described by Corollary 3.6. To simplify our notation we will hereafter assume that the universe of a τ_{field}-structure $\mathbf{F}$ of cardinality m, given a primitive element $\alpha \in U(\mathbf{F})$, consists of the integers $\{0, 1, \ldots, m-1\}$, with $\leqslant_{\alpha}$ interpreted as the standard ordering of the integers. This is without any loss of generality, for we can always define a bijection $\iota : U(\mathbf{F}) \to \{0, \ldots, m-1\}$ as a number term in FOC by

$$\iota(x) \equiv \#_{y}\big((y \leqslant_{\alpha} x) \wedge (y \neq x)\big).$$

Now consider a number term $\pi(x)$ in vocabulary τ_{field}, where x is an element variable. Given a τ_{field}-structure $\mathbf{F}$, a primitive element $\alpha \in U(\mathbf{F})$ and an integer m, we write $\text{poly}_{x}(\pi, \mathbf{F}, \alpha, m)$ to denote the integer polynomial $a_m X^m + \cdots + a_1 X + a_0$, where $a_i = \pi[i]^{\mathbf{F}^{\star}}$ for each $i \leq m$ in $U(\mathbf{F})$.

Lemma 3.7 (Minimal polynomials)**.** *There is an* IFPC *number term* $\mathsf{minpoly}(x, y)$ *in vocabulary* τ_{field} *for which it holds that for any* τ_{field}*-structure* $\mathbf{F}$ *and primitive element* $\alpha \in U(\mathbf{F})$*, the polynomial*

$$\text{poly}_{x}(\mathsf{minpoly}(x, \alpha/y), \mathbf{F}, \alpha, d) \bmod p$$

is the minimal polynomial of α *over* GF_p*, where* $\|\mathbf{F}\| = p^d$ *and* p *is a prime.*

Proof. Consider a finite field F of cardinality p^d, p prime, and write K to denote the prime field $\mathbb{Z}_p$. We first describe a general polynomial-time procedure for constructing the minimal polynomial over K of a primitive element $\alpha \in F$. Then we show how this procedure can be turned into a fixed-point formula over τ_{field}-structures.

To define the minimal polynomial of a primitive element α, we first generate a list Π of all polynomials in $K[X]$ of degree at most d. There are exactly p^{d+1} of those polynomials; alternatively, we can reduce this number to p^d by ignoring all polynomials of degree exactly d that are not monic. Let $\leqslant_{\text{poly}}$ be the ordering on Π defined by

$$\begin{aligned} &a_d X^d + \cdots + a_1 X + a_0 \leqslant_{\text{poly}} b_d X^d + \cdots + b_1 X + b_0 \\ :\Leftrightarrow\ &a_d a_{d-1} \cdots a_1 a_0 \leq_{\text{lex}} b_d b_{d-1} \cdots b_1 b_0, \end{aligned}$$

where $a_d a_{d-1} \cdots a_1 a_0$ and $b_d b_{d-1} \cdots b_1 b_0$ are strings over $\{0, \ldots, p-1\}^*$ and $\leq_{\text{lex}}$ is the standard lexicographic ordering on $\{0, \ldots, p-1\}^*$. To generate the list Π we enumerate the polynomials in increasing $\leqslant_{\text{poly}}$-order, starting with the constant polynomial $g(X) = 1$. Now we can construct the minimal polynomial of α over K by

(*1*) building the set Π of polynomials in $K[X]$ of degree at most d;

(*2*) defining the set $\Pi_{\text{ann}} \subset \Pi$ of monic polynomials in Π that annihilate α — i.e. each $g \in \Pi_{\text{ann}}$ satisfies $g(\alpha) = 0$; and

(*3*) finding the least element in Π_{ann} with respect to $\leqslant_{\text{poly}}$, which will be the minimal polynomial.

Constructing Π takes $\mathcal{O}(p^d)$ steps, while the number of steps required for constructing Π_{ann} is $\|\Pi\|$ times the number of steps to decide for each $g \in \Pi$ if α is a root. Evaluating $g(\alpha) = 0$ requires $\mathcal{O}(d^2)$ multiplications and $\mathcal{O}(d)$ additions of elements from K. Finally, finding the least element of Π_{ann} requires a linear number of $\leqslant_{\text{poly}}$-comparisons, each of which takes $\mathcal{O}(d)$ steps. The overall algorithm takes $\mathcal{O}(p^d d^2)$ steps, which is polynomial in $\|F\| = p^d$.

Let Min-Polynomial (F, i, α, k) be the problem of deciding whether the coefficient of X^i in the minimal polynomial $f(X)$ over K of a primitive element α is k, where i and k are integers and the field F is given explicitly by its multiplication and addition tables. This problem is in PTIME by our discussion above. We can define a linear ordering of structures of vocabulary τ_{field}, given a fixed primitive element, so by the Immerman-Vardi Theorem there is a formula $\mathsf{mincoeff}(x, y, z)$ of IFP for which it holds that for any τ_{field}-structure $\mathbf{F}$ and $i, \alpha, k \in U(\mathbf{F})$, with α a primitive element,

> $(\mathbf{F}, i, \alpha, k) \models \mathsf{mincoeff}(x, y, z)$ if, and only if, the coefficient of X^i in the minimal polynomial $f(X)$ of α over K is k.

Here we are assuming that $U(\mathbf{F})$ consists of the integers $\{0, \dots, p^d - 1\}$, as noted earlier. Finally, the required IFPC-formula $\mathsf{minpoly}(x, y)$ is given by

$$\mathsf{minpoly}(x, y) \equiv \#_w\big(\exists z\ (\mathsf{mincoeff}(x, y, z) \wedge (w \leq z) \wedge (w \neq z))\big).$$

□

Let F be a finite field of cardinality p^d with a primitive element $\alpha \in F$. Write $f(X)$ for the minimal polynomial of α over $K = \mathbb{Z}_p$. Then an isomorphism $\iota : F \cong K[X]/(f(X))$ can be explicitly given by

$$g \mapsto h(X) :\Leftrightarrow h(\alpha) = g,$$

for all $g \in F$. This is well-defined, since every element of F can be written uniquely as a linear combination over K of elements in $\{1_F, \alpha, \alpha^2, \dots, \alpha^{d-1}\}$. For instance, if $g \in F$ has α-order m, then g can be written as $\alpha^m \bmod f(\alpha)$ in this way. This expression has to be unique, since $f(X)$ is a minimal polynomial. By putting this together with Lemma 3.7, we obtain the following theorem, which says that the isomorphism ι can be defined in IFPC. This theorem will be crucial for our construction of characteristic polynomial of finite-field matrices in §3.4.

Corollary 3.8 (Isomorphism of prime-power fields)**.** *There is an* IFPC *number term $\eta(x, y)$ in vocabulary $\tau^\star_{field}$, where x and y are element variables, for which it holds that for any τ_{field}-structure $\mathbf{F}$ of cardinality p^d and any primitive element $\alpha \in U(\mathbf{F})$, the map defined for all $g \in U(\mathbf{F})$ by*

$$g \mapsto \text{poly}_x(\eta(x, g/z), \mathbf{F}, \alpha, d) \bmod p,$$

is an isomorphism of fields $\mathbf{F} \cong (\mathbb{Z}_p)[X]/(f(X))$*, where we write*

$$f(X) := \mathrm{poly}_x(\mathsf{minpoly}(x, \alpha/y), \mathbf{F}, \alpha, d) \bmod p.$$

□

3.3 Describing integer and rational matrices in IFPC

In [9], Blass et al. showed that standard arithmetic operations on matrices, such as multiplication and exponentiation, can be defined in IFPC for matrices over finite fields. In this chapter, we establish similar results for integer and rational matrices. These will be crucial later on for our construction of characteristic and minimal polynomials in §3.4 and §3.5.

In order to describe our results, we first establish in §3.3.1 some notation for specifying matrices by terms and formulae. In this notation, we can describe one matrix in terms of others by a substitution of formulae. In §3.3.2 we establish further notation, this time for working with integers in binary representation. Furthermore, we prove some basic technical results, showing that arithmetic over unordered collections of binary numbers can be expressed in IFPC. Using these results, we show in §3.3.3 how to define in IFPC the product AB, when A and B are matrices of the appropriate dimension over $\mathbb{Z}$ or $\mathbb{Q}$. Finally, we show in §3.3.4 that for any square matrix A over $\mathbb{Z}$ or $\mathbb{Q}$ and any integer $m \geq 0$, the matrix A^m and the trace $\mathrm{tr}(A)$ are definable in IFPC.

3.3.1 Specifying matrices over $\mathbb{Z}$ and $\mathbb{Q}$ by formulae

When describing matrices in the two-sorted logic IFPC, it can simplify our notation to have a standard specification given by formulae and number terms. In the following we define a specification of this kind, which will be used to describe arithmetic operations on matrices later on.

Let $\vec{x}$ and $\vec{y}$ be tuples of element variables and let v be a number variable. Consider a number term t and formulae φ_d, φ_n and ψ in vocabulary τ where $\mathrm{free}(t) = \varnothing$ and all the free variables of φ_n, φ_d and ψ are amongst the variables in $\vec{x} \cup \vec{y} \cup \{v\}$. Here the role of v is to index the binary expansion of matrix elements over the number sort, bounded by the number term t. Consider a τ-structure $\mathbf{A}$ and tuples $\vec{a} \in U(\mathbf{A})^{\|\vec{x}\|}$ and $\vec{b} \in U(\mathbf{A})^{\|\vec{y}\|}$. Let

$$\Gamma_n(\mathbf{A}, \vec{a}, \vec{b}) := \{i \leq t^{\mathbf{A}^\star} \mid \mathbf{A}^\star \vDash \varphi_n[\vec{a}, \vec{b}, i]\} \text{ and}$$
$$\Gamma_d(\mathbf{A}, \vec{a}, \vec{b}) := \{i \leq t^{\mathbf{A}^\star} \mid \mathbf{A}^\star \vDash \varphi_d[\vec{a}, \vec{b}, i]\}$$

denote the collections of integer assignments to v that satisfy $\varphi_n(\vec{a}/\vec{x}, \vec{b}/\vec{y}, v)$ and $\varphi_d(\vec{a}/\vec{x}, \vec{b}/\vec{y}, v)$ in $\mathbf{A}^\star$, respectively. Define integers

$$n_{\vec{a}\vec{b}} = \sum_{i \in \Gamma_n(\mathbf{A}, \vec{a}, \vec{b})} 2^i,$$
$$d_{\vec{a}\vec{b}} = \sum_{i \in \Gamma_d(\mathbf{A}, \vec{a}, \vec{b})} 2^i, \text{ and}$$
$$s_{\vec{a}\vec{b}} = \begin{cases} 1 & \text{if } \mathbf{A}^\star \vDash \psi[\vec{a}, \vec{b}] \\ -1 & \text{otherwise.} \end{cases}$$

If $d_{\vec{a}\vec{b}} \neq 0$ for all $\vec{a}$ and $\vec{b}$, then we write $\mathrm{mat}_{\vec{x},\vec{y},v}(\varphi_n, \varphi_d, \psi, t, \mathbf{A})$ to denote the $U(\mathbf{A})^{\|\vec{x}\|} \times U(\mathbf{A})^{\|\vec{y}\|}$ rational matrix $M = (m_{\vec{a}\vec{b}})$ whose entry at row index $\vec{a}$ and column index $\vec{b}$ is given by

$$m_{\vec{a}\vec{b}} := s_{\vec{a}\vec{b}} \cdot \frac{n_{\vec{a}\vec{b}}}{d_{\vec{a}\vec{b}}}.$$

Here the formulae φ_n and φ_d encode the numerators and denominators of elements of M, respectively, and ψ identifies the set of non-negative elements of M. In other words, the tuple $(\varphi_n(\vec{x},\vec{y},v), \varphi_d(\vec{x},\vec{y},v), \psi(\vec{x},\vec{y}), t)$ describes an interpretation of $\tau_{\mathbb{Q}}$ in τ of width $\max\{\|\vec{x}\|, \|\vec{y}\|\}$, where we omit the trivial domain-defining and equality-defining formulae δ and ε, respectively, and the linear ordering of bit positions is just the natural ordering over the number sort (see §2.2.5 for more details). Here the role of the integer $t^{\mathbf{A}^\star} \in \mathbb{N}_0$ is to denote the maximum bit length of all the matrix elements, so that

$$\forall v > t\, \neg(\exists \vec{x}\vec{y}\, (\varphi_n(\vec{x},\vec{y},v) \vee \varphi_d(\vec{x},\vec{y},v))).$$

Integer matrices can be described similarly, by setting all denominators to one. In this way, for a triple $(\varphi(\vec{x},\vec{y},v), \psi(\vec{x},\vec{y}), t)$ we write $\mathrm{mat}_{\vec{x},\vec{y},v}(\varphi, \psi, t, \mathbf{A})$ to denote the $U(\mathbf{A})^{\|\vec{x}\|} \times U(\mathbf{A})^{\|\vec{y}\|}$ integer matrix, defined like above by setting $d_{\vec{a}\vec{b}} = 1$ for all row indices $\vec{a}$ and column indices $\vec{b}$.

3.3.2 Binary arithmetic

Let τ be a vocabulary and consider a formula $\eta(v)$ and a number term t in IFPC$[\tau]$, where v is a number variable. Given a τ-structure $\mathbf{A}$, we write

$$(\eta(v), t)^{\mathbf{A}} := \{i \mid 0 \leq i \leq t^{\mathbf{A}^\star} \wedge \mathbf{A}^\star \vDash \eta[i]\}.$$

That is, the pair $(\eta(v), t)$ defines over $\mathbf{A}$ the $t^{\mathbf{A}^\star}$-bit binary encoding of an integer m, where $m = \sum_{i \in (\eta,t)^{\mathbf{A}}} 2^i$. We write $\mathrm{binenc}_v(\eta, t, \mathbf{A}) \in \mathbb{N}_0$ to denote the integer defined in this way by $(\eta(v), t)$ over $\mathbf{A}$.

Let $\gamma(\vec{x}, v)$ be an IFPC$[\tau]$-formula, where $\vec{x}$ are element variables and v is a number variable, and let t be an IFPC$[\tau]$-number term. The pair $(\gamma(\vec{x}, v), t)$ defines over $\mathbf{A}$ a collection of $\|U(\mathbf{A})\|^{\|\vec{x}\|}$ integers

$$\mathrm{binset}_{\vec{x},v}(\gamma, t, \mathbf{A}) := \{(\mathrm{binenc}_v(\gamma(\vec{a}/\vec{x}, v), t, \mathbf{A}) \mid \vec{a} \in U(\mathbf{A})^{\|\vec{x}\|}\} \subseteq \mathbb{N}_0,$$

where $\gamma(\vec{a}/\vec{x}, v)$ is obtained from $\gamma(\vec{x}, v)$ by replacing every occurrence of $\vec{x}$ with $\vec{a}$. The following lemma shows that there is a formula of IFPC which defines over any structure $\mathbf{A}$ the number which is the sum of all elements in the collection $\mathrm{binset}_{\vec{x},v}(\gamma, t, \mathbf{A})$.

Lemma 3.9 (Sums of binary numbers)**.** *Let $\gamma(\vec{x}, v)$ be an* IFPC$[\tau]$*-formula, where $\vec{x}$ are element variables and v is a number variable, and let t be an* IFPC$[\tau]$*-number term. There is an* IFPC*-formula* $\mathsf{sum}(v)$ *and a numeric* IFPC*-term s, such that for all τ-structures $\mathbf{A}$,*

$$\mathrm{binenc}_v(\mathsf{sum}, s, \mathbf{A}) = \sum_{m \in \mathrm{binset}_{\vec{x},v}(\gamma,t,\mathbf{A})} m.$$

Remark. In the statement of this lemma, the number term s gives an upper bound for the number of bits in $\sum_{m \in \mathrm{binset}_{\vec{x},v}(\gamma,t,\mathbf{A})} m$.

Proof. We consider a simple ripple-carry algorithm for simultaneously adding together a collection of integers in binary. Let bit : $\mathbb{N}_0 \times \mathbb{N}_0 \to \{0,1\}$ be the function that maps (i,m) to the i-th bit of m, for $i, m \in \mathbb{N}_0$. Let bitcount : $\mathbb{N}_0 \times \wp_{\text{fin}}(\mathbb{N}_0) \to \mathbb{N}_0$ be the function

$$\text{bitcount} : (i, M) \mapsto \|\{m \in M \mid \text{bit}(i,m) = 1\}\|,$$

and let quot : $\mathbb{N}_0 \to \mathbb{N}_0$ be the function $k \mapsto \lfloor k/2 \rfloor$. Define carry : $\mathbb{N}_0 \times \wp_{\text{fin}}(\mathbb{N}_0) \to \mathbb{N}_0$ by induction for all $M \subset_{\text{fin}} \mathbb{N}_0$ as

$$\begin{aligned} \text{carry}(0, M) &= \text{quot}(\text{bitcount}(0, M)), \\ \text{carry}(i+1, M) &= \text{quot}(\text{bitcount}(i+1, M) + \text{carry}(i, M)) \qquad \forall\, i \geq 1. \end{aligned}$$

In other words, $\text{carry}(i, M)$ is the number of bits carried over from bit position i to bit position $i+1$ when adding together all the elements in M. Now we can define sumbit : $\mathbb{N}_0 \times \wp_{\text{fin}}(\mathbb{N}_0) \to \mathbb{N}_0$, where $\text{sumbit}(i, M)$ denotes the i-th bit in the binary expansion of $\sum_{m \in M} m$, as follows.

$$\begin{aligned} \text{sumbit}(0, M) &= \text{bitcount}(0, M) \pmod 2, \\ \text{sumbit}(i+1, M) &= (\text{bitcount}(i+1, M) + \text{carry}(i, M)) \pmod 2 \qquad \forall\, i \geq 1. \end{aligned}$$

Let $\gamma(\vec{x}, v)$ be an IFPC$[\tau]$-formula, where $\vec{x}$ is a k-tuple of element variables, and let t be a number term of IFPC$[\tau]$. It is straightforward to turn the above algorithm into a formula, using the **ifp**-operator. Define

$$\begin{aligned} \varphi_{\text{bits}}(v) &\equiv \#_{\vec{x}}\gamma(\vec{x}, v), \\ \varphi_{\text{odd}}(\kappa) &\equiv \#_{v}(\exists \mu \leq \kappa . 2\mu + 1 = \kappa) \text{ and} \\ \varphi_{\text{quot}}(\kappa, \mu) &\equiv (2\kappa = \mu) \vee (2\kappa + 1 = \mu). \end{aligned}$$

Here, $\varphi_{\text{bits}}(v)$ denotes the number of "1" bits at position v over all $\vec{x}$ in $\gamma(\vec{x}, v)$; the term $\varphi_{\text{odd}}(\kappa)$ is one if κ denotes an odd number and zero otherwise; and $\varphi_{\text{quot}}(\kappa, \mu)$ is a formula that says that κ is the integer quotient of μ divided by two. Now let R be a binary relation symbol of type {number, number} and define the formulae

$$\begin{aligned} \theta_1(\iota, \kappa, R) \equiv &\big((\iota = 0) \wedge \varphi_{\text{quot}}(\kappa, \varphi_{\text{bits}}(\iota))\big) \vee \\ &\big(\exists \mu . R(\iota - 1, \mu) \wedge \varphi_{\text{quot}}(\kappa, \varphi_{\text{bits}}(\iota) + \mu)\big) \text{ and} \\ \varphi_1(v, \eta) \equiv &[\mathbf{ifp}_{R, \iota\kappa}\theta_1(\iota, \kappa, R)](v, \eta). \end{aligned}$$

If $\mathbf{A}$ is a τ-structure, then $\varphi_1(v,\eta)^{\mathbf{A}}$ defines the graph of the function $\text{carry}(i, M)$, where $M = \text{binset}_{\vec{x},v}(\gamma, t, \mathbf{A}))$. Let

$$t_{\text{carry}}(v) \equiv \#_{\mu}(\exists \eta . \varphi_1(v, \eta) \wedge (\mu < \eta))$$

be the corresponding number term; that is, for all i, $t_{\text{carry}}(v)^{(\mathbf{A}, i)} = \text{carry}(i, M)$, with M as above. Define

$$\begin{aligned} \theta_{\text{sum}}(\iota, \kappa, R) \equiv &\big((\iota = 0) \wedge (\kappa = \varphi_{\text{odd}}(\varphi_{\text{bits}}(\iota)))\big) \vee \\ &\big(\varphi_{\text{odd}}(\varphi_{\text{bits}}(\iota) + t_{\text{carry}}(\iota - 1, \eta)))\big) \text{ and} \\ \varphi_{\text{sum}}(v, \kappa) \equiv &[\mathbf{ifp}_{R, \iota\kappa}\theta_{\text{sum}}(\iota, \kappa, R)](v, \eta). \end{aligned}$$

Finally, define $\mathsf{sum}(v) \equiv \#_{\mu}(\exists \eta . \varphi_{\text{sum}}(v, \eta) \wedge (\mu < \eta))$, as required.

All that remains is to show that we can construct a number term s for which it holds that for any τ-structure $\mathbf{A}$, $s^{\mathbf{A}}$ is an upper bound for the the number of bits in the sum over all elements in $\mathrm{binset}_{\vec{x},v}(\gamma, t, \mathbf{A})$. Consider a τ-structure $\mathbf{A}$ of size $\|A\| = n$. Each element in the collection $\mathrm{binset}_{\vec{x},v}(\gamma, t, \mathbf{A})$ has at most $m = t^{\mathbf{A}}$ bits, so the number of bits in the sum over all elements in $\mathrm{binset}_{\vec{x},v}(\gamma, t, \mathbf{A})$ is at most

$$\log(2^m n^k) = m + k\log(n) \leq m + kn.$$

Therefore, it can be seen that to get an upper bound for the number of bits in this sum for any τ-structure, it suffices to take the number term

$$s \equiv t + k \cdot (\#_x(x = x)).$$

Here we write "$k \cdot (\#_x(x = x))$" to denote the number term obtained by adding together k terms $\#_x(x = x)$, where the constant k is the number of distinct variables in $\vec{x}$.

□

3.3.3 Product of matrices

Let $(\eta_1(v), t_1)$ and $(\eta_2(v), t_2)$ be IFPC-definable specifications of binary numbers over signature τ. It is not hard to verify that there is an IFPC$[\tau]$-formula $\mathsf{prod}(v)$ and an IFPC$[\tau]$-number term $t \equiv t_1 + t_2$ such that for all τ-structures $\mathbf{A}$,

$$\mathrm{binenc}_v(\mathsf{prod}, t, \mathbf{A}) = \mathrm{binenc}_v(\eta_1, t_1, \mathbf{A}) \cdot \mathrm{binenc}_v(\eta_2, t_2, \mathbf{A}).$$

This of course follows directly from the Immerman-Vardi theorem, as the bit positions are linearly ordered over $\mathbb{N}_0$. In fact, the product can even be expressed by an FOC-formula, by defining a lexicographic ordering of bit strings in FOC and applying Lemma 6.14 from Libkin [52]. From this observation, and the results of the previous section, we can now prove the following theorem.

Theorem 3.10 (Product of integer matrices)**.** *Let*

$$\Theta_1 = (\varphi_1(\vec{x}, \vec{z}, v), \psi_1(\vec{x}, \vec{z}), t_1) \text{ and}$$
$$\Theta_2 = (\varphi_2(\vec{z}, \vec{y}, v), \psi_2(\vec{z}, \vec{y}), t_2)$$

be IFPC*-definable specifications of integer matrices over τ-structures. Then there is an* IFPC*-definable matrix specification*

$$\Theta_\times(\Theta_1, \Theta_2) = (\varphi_\times(\vec{x}, \vec{y}, v), \psi_\times(\vec{x}, \vec{y}), t_\times),$$

such that for all τ-structures $\mathbf{A}$*,*

$$\mathrm{mat}_{\vec{x},\vec{y},v}(\varphi_\times, \psi_\times, t_\times, \mathbf{A}) = \mathrm{mat}_{\vec{x},\vec{z},v}(\varphi_1, \psi_1, t_1, \mathbf{A}) \cdot \mathrm{mat}_{\vec{z},\vec{y},v}(\varphi_2, \psi_2, t_2, \mathbf{A}).$$

Proof. By our previous observation, there is a formula-term pair $(\mathsf{prod}(\vec{x}, \vec{y}, \vec{z}, v), t)$ in IFPC (here with additional parameters) which describes the product of $(\varphi_1(\vec{x}, \vec{z}, v), t_1)$ and $(\varphi_2(\vec{x}, \vec{z}, v), t_2)$, with respect to v. The formula

$$\psi_{\mathrm{prod}}(\vec{x}, \vec{y}, \vec{z}) \equiv (\psi_1(\vec{x}, \vec{z}) \leftrightarrow \psi_2(\vec{z}, \vec{y}))$$

denotes the sign of the integer $\mathsf{prod}(\vec{x}, \vec{y}, \vec{z}, v)$. Now define formulae

$$\gamma_{\geq 0}(\vec{x}, \vec{y}, \vec{z}, v) \equiv \psi_{\text{prod}}(\vec{x}, \vec{y}, \vec{z}) \wedge \mathsf{prod}(\vec{x}, \vec{y}, \vec{z}, v) \text{ and}$$
$$\gamma_{<0}(\vec{x}, \vec{y}, \vec{z}, v) \equiv \neg\psi_{\text{prod}}(\vec{x}, \vec{y}, \vec{z}) \wedge \mathsf{prod}(\vec{x}, \vec{y}, \vec{z}, v),$$

which respectively denote the collection of nonnegative elements and the collection of negative elements $\mathsf{prod}(\vec{x}, \vec{y}, \vec{z}, v)$. Here each collection of integers is indexed by $\vec{z}$, with $\vec{x}$ and $\vec{y}$ treated as parameters. By Lemma 3.9, there are formula-term pairs $(\mathsf{sum}_{\geq 0}(\vec{x}, \vec{y}, v), s_{\geq 0})$ and $(\mathsf{sum}_{<0}(\vec{x}, \vec{y}, v), s_{<0})$ that denote the sum over $\gamma_{\geq 0}(\vec{x}, \vec{y}, \vec{z}, v)$ and $\gamma_{<0}(\vec{x}, \vec{y}, \vec{z}, v)$, respectively, with respect to $\vec{z}$.

Now we can define formulae $\varphi_\times(\vec{x}, \vec{y}, v)$ and $\psi_\times(\vec{x}, \vec{y})$ which for all $\vec{x}$, $\vec{y}$ denote the absolute value and sign, respectively, of the integer obtained by subtracting $\mathsf{sum}_{<0}(\vec{x}, \vec{y}, v)$ from $\mathsf{sum}_{\geq 0}(\vec{x}, \vec{y}, v)$. Finally, let $t_\times$ be the number term that defines the maximum of $s_{\geq 0}$ and $s_{<0}$. □

Using the above result, it is straightforward to define the product of rational matrices in IFPC. Suppose $A_1 \in M_{n\times q}(\mathbb{Q})$ and $A_2 \in M_{q\times m}(\mathbb{Q})$ are rational matrices whose product we want. We can rewrite each matrix as $A_i = N_i^{-1}B_i$, where for $i \in \{1, 2\}$, B_i is an integer matrix of the same dimension as A_i and N_i is the least common multiple of all integers appearing as denominators of elements in A_i. The product $A_1A_2 = (N_1N_2)^{-1}B_1B2$ can now be obtained by separately calculating the product of two positive integers and the product of two integer matrices.

Corollary 3.11 (Product of rational matrices)**.** *Let*

$$\Theta_1 = (\varphi_{1,n}(\vec{x}, \vec{z}, v), \varphi_{1,d}(\vec{x}, \vec{z}, v), \psi_1(\vec{x}, \vec{z}), t_1) \text{ and}$$
$$\Theta_2 = (\varphi_{2,n}(\vec{z}, \vec{y}, v), \varphi_{2,d}(\vec{z}, \vec{y}, v), \psi_2(\vec{z}, \vec{y}), t_2)$$

be IFPC*-definable specifications of rational matrices over τ-structures. Then there is an* IFPC*-definable matrix specification*

$$\Theta_\times(\Theta_1, \Theta_2) = (\varphi_{\times,n}(\vec{x}, \vec{y}, v), \varphi_{\times,d}(\vec{x}, \vec{y}, v), \psi_\times(\vec{x}, \vec{y}), t_\times),$$

such that for all τ-structures **A***,*

$$\text{mat}_{\vec{x},\vec{y},v}(\varphi_{\times,n}, \varphi_{\times,d}, \psi_\times, t_\times, \mathbf{A})$$
$$= \text{mat}_{\vec{x},\vec{z},v}(\varphi_{1,n}, \varphi_{1,d}, \psi_1, t_1, \mathbf{A}) \cdot \text{mat}_{\vec{z},\vec{y},v}(\varphi_{2,n}, \varphi_{2,d}, \psi_2, t_2, \mathbf{A}).$$

Proof. For each $i = 1, 2$, we can order the collection of denominators $\varphi_{i,d}(\vec{x}, \vec{z}, v)$ using a lexicographic ordering of binary numbers like the one we defined before. While this is not a linear ordering, as some of the denominators may be repeated, we can define over the number sort the corresponding collection of s_i distinct denominators, in strictly increasing order. Here s_i is a number term we can define in terms of $\varphi_{i,d}(\vec{x}, \vec{z}, v)$ and the lexicographic ordering. Let $\gamma_i(\mu, v)$ define this collection of distinct denominators, where $\mu \leq s_i$ and $v \leq t_i$ are number variables.

Because it is linearly ordered, we can now express any polynomial-time computation over the collection of numbers defined by $(\gamma_i(\mu, v), s_i)$ as an IFP formula. In particular, we can define the least common multiple of all numbers in the collection as a formula $\theta_i(v)$,

where $v \leq r_i$ and $r_i \equiv s_i t_i$ is an upper bound on the number of bits required. It is now straightforward to define formulae $\varphi_1(\vec{x}, \vec{z}, v)$ and $\varphi_2(\vec{z}, \vec{y}, v)$ so that for all τ-structures $\mathbf{A}$,

$$\mathrm{mat}_{\vec{x},\vec{z},v}(\varphi_{1,n}, \varphi_{1,d}, \psi_1, t_1, \mathbf{A}) = (\mathrm{binenc}_v(\theta_1, r_1, \mathbf{A}))^{-1} \cdot \mathrm{mat}_{\vec{x},\vec{z},v}(\varphi_1, \psi_1, t_1 + r_1, \mathbf{A}) \text{ and}$$
$$\mathrm{mat}_{\vec{x},\vec{z},v}(\varphi_{2,n}, \varphi_{2,d}, \psi_2, t_2, \mathbf{A}) = (\mathrm{binenc}_v(\theta_2, r_2, \mathbf{A}))^{-1} \cdot \mathrm{mat}_{\vec{x},\vec{z},v}(\varphi_2, \psi_2, t_2 + r_2, \mathbf{A}).$$

The proof now follows from Theorem 3.10 and our previous observation. □

3.3.4 Exponentiation and trace of matrices

We can now define formulae that express the exponentiation of any definable matrix by an element of the number sort. Let $\Theta = (\varphi(\vec{x}, \vec{y}, v), \psi(\vec{x}, \vec{y}), t)$ be a matrix specification in vocabulary τ and let κ be a number variable. A matrix A^m, $m \geq 2$, is defined only when A is square, so we assume $\|\vec{x}\| = \|\vec{y}\| = k$. Assume furthermore that all matrix entries are positive, that is $\psi(\vec{x}, \vec{y}) \equiv \top$. This is only to simplify the current presentation; matrices with negative entries can be handled similarly. Let R be a relation symbol of type (element^k, element^k, number, number) and let $\Theta_R = (R(\vec{x}, \vec{y}, v; \kappa), \psi(\vec{x}, \vec{z}), s)$ be a matrix representation, where $s \equiv \kappa t$ is a number term. Here, the number variable κ is treated as a parameter. Let $\Theta_\times(\Theta, \Theta_R) = (\varphi_\times(\vec{x}, \vec{y}, v; \kappa), \psi_\times(\vec{x}, \vec{y}), t_\times)$ denote the product of Θ and Θ_R, as in Theorem 3.10. Then the exponentiation of the matrix Θ by the number term κ is defined by

$$\begin{aligned}\mathsf{power}(\vec{x}, \vec{y}, v; \kappa) \equiv [&\mathbf{ifp}_{R,\vec{x}\vec{y}v\kappa}((\kappa = 0) \wedge (\vec{x} = \vec{y})) \vee \\ &\exists \mu \leq \kappa.(\kappa = \mu + 1) \wedge \varphi_\times(\vec{x}, \vec{y}, v; \mu)](\vec{x}, \vec{y}, v; \kappa),\end{aligned}$$

where we write $\vec{x} = \vec{y}$ to denote $\bigwedge_i (x_i = y_i)$. This construction resembles the one given by Dawar in [16], except there the formula power is constructed using the least-fixed-point operator, as opposed to **ifp** here. Now for every τ-structure $\mathbf{A}$ and any interpretation $m \in \mathbb{N}_0$ of the number variable κ,

$$\mathrm{mat}_{\vec{x},\vec{y},v}(\mathsf{power}, \psi_\times, s, (\mathbf{A}, m)) = \mathrm{mat}_{\vec{x},\vec{y},v}(\varphi, \psi, t, \mathbf{A})^m.$$

We can define the exponentiation of rational matrices very similarly.

Finally, we observe that we can define in IFPC the trace of integer and rational matrices. Recall that the trace of a square matrix $A = (a_{ij})$ is defined as $\mathrm{tr}(A) := \sum_i a_{ii}$. The trace of an integer matrix, denoted by a tuple $(\varphi(\vec{x}, \vec{y}, v), \psi(\vec{z}, \vec{y}), t)$, with $\|\vec{x}\| = \|\vec{y}\|$, is just the sum of all the binary numbers along the main diagonal, which can be defined in IFPC according to Lemma 3.9. Similarly, the trace of a rational matrix can be defined by first expressing the matrix as the product of a rational number and an integer matrix, as we have discussed before.

3.4 Characteristic polynomial over $\mathbb{Z}$, $\mathbb{Q}$ and finite fields

It has been observed by Rossman that Csanky's algorithm [14] for computing the characteristic polynomial (and thereby, the determinant) of a matrix over any commutative ring of characteristic zero is expressible in the logic of choiceless polynomial time with counting.

Blass and Gurevich [7] used this observation to show that the same logic can also express the determinant of any definable matrix over a finite field.

In this section we strengthen this result by showing that Le Verrier's method for finding the coefficients of the characteristic polynomial of a matrix, which is the main building block of Csanky's algorithm, can already be expressed in IFPC for both integer and rational matrices, as well as matrices over finite fields. We start by reviewing Le Verrier's method; for more details, see e.g. Faddeev and Faddeeva [25].

3.4.1 Overview of Le Verrier's method

Let M be an $n \times n$ matrix over a commutative ring R of characteristic zero, $n \geq 1$. The characteristic polynomial $\chi_M(x)$ of M is

$$\begin{aligned}\det(xI - M) &= x^n - p_1 x^{n-1} + p_2 x^{n-2} - \cdots + (-1)^n p_n \\ &= \prod_{i=1}^{n} (x - \lambda_i),\end{aligned}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of M, counted with multiplicities. The coefficients $p_1, \ldots, p_n \in R$ of the characteristic polynomial can be written in terms of the eigenvalues as follows:

$$p_k = \sum_{1 \leq i_1 < \cdots < i_k \leq n} \prod_{j=1}^{k} \lambda_{i_j},$$

for $k \in [n]$. That is, the k-th coefficient p_k is the sum of all products of k distinct elements from $\{\lambda_1, \ldots, \lambda_n\}$. In particular, $p_n = \det(M)$ and $p_1 = \sum_{i=1}^{n} \lambda_i = \operatorname{tr}(M)$, the trace of M.

We now derive a linear recurrence for the coefficients p_k. This recurrence can be solved to obtain the coefficients of $\chi_M(x)$ without actually knowing any of the eigenvalues. Let $m \geq 1$ and define $s_m := \operatorname{tr}(M^m)$, which can be written as

$$s_m = \operatorname{tr}(M^m) = \sum_{i=1}^{n} \lambda_i^m,$$

using the fact that M^m has eigenvalues $\lambda_1^m, \ldots, \lambda_n^m$ (see e.g. Horn and Johnson [42]). Also define for $k, m \geq 1$,

$$f_k^m := \sum_{\substack{1 \leq i_1 < \cdots < i_k \leq n \\ l \notin \{i_1, \ldots, i_k\}}} \Big(\prod_{j=1}^{k} \lambda_{i_j}\Big)(\lambda_l)^m.$$

Multiply together p_k and s_m and simplify to obtain

$$p_k s_m = \Big(\sum_{1 \leq i_1 < \cdots < i_k \leq n} \prod_{j=1}^{k} \lambda_{i_j} \Big) \Big(\sum_{i=1}^{n} \lambda_i^m \Big) = f_{k-1}^{m+1} + f_k^m.$$

Using this equation, we write down a telescoping series

$$\begin{aligned}&p_k s_0 - p_{k-1} s_1 + \cdots \mp p_1 s_{k-1} \pm s_k \\ &= (f_k^0 + f_{k-1}^1) - (f_{k-1}^1 + f_{k-2}^2) + \cdots \pm f_0^k \\ &= f_k^0 = (n-k) p_k \\ &= (s_0 - k) p_k.\end{aligned}$$

This gives us a linear recurrence for the coefficient p_k in terms of the coefficients $p_1, \ldots, p_{k-1}$:

$$p_k = \tfrac{1}{k}(p_{k-1}s_1 - p_{k-2}s_2 + \cdots \pm s_k).$$

Treating each s_j as a scalar coefficient and each p_k as a variable, we can write this linear recurrence as a system of linear equations

$$A\mathbf{x} = \mathbf{b}, \tag{$*$}$$

where $\mathbf{x} = (p_n, \ldots, p_1)^t$, $\mathbf{b} = (\pm\frac{s_n}{n}, \mp\frac{s_{n-1}}{n-1}, \ldots, s_1)^t$ and

$$A = \begin{pmatrix} 1 & -\frac{s_1}{n} & \ldots & \pm\frac{s_{n-2}}{n} & \mp\frac{s_{n-1}}{n} \\ 0 & 1 & \ldots & \mp\frac{s_{n-3}}{n-1} & \pm\frac{s_{n-2}}{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -\frac{s_1}{2} \\ 0 & 0 & \ldots & 0 & 1 \end{pmatrix}.$$

Le Verrier's method for finding the coefficients of $\chi_M(x)$ now consists of the following steps:

(*1*) for each k, compute s_k from the trace of the k-th power of M;

(*2*) construct the matrices A and $\mathbf{b}$; and

(*3*) solve for the p_k.

3.4.2 Characteristic polynomial over $\mathbb{Z}$ and $\mathbb{Q}$

Here we consider the characteristic polynomial of matrices with rational entries; by setting all denominators to 1, we can use the same approach to define the characteristic polynomial of integer matrices. Below we sketch a proof of the following theorem.

Theorem 3.12 (Characteristic polynomial over $\mathbb{Q}$)**.** *There are* IFPC*-formulae* $\theta^n_{char}(\mu, \upsilon)$ *and* $\theta^d_{char}(\mu, \upsilon)$ *in vocabulary* $\tau^\star_{\mathbb{Q}}$*, where* μ *and* υ *are number variables, which for all square* $\tau_{\mathbb{Q}}$*-matrices* $\mathbf{M}$ *satisfy:*

- $\mathbf{M} \vDash \theta^n_{char}[k, i]$ *iff the* i*-th bit of the* numerator *of the coefficient of* x^k *in the characteristic polynomial* $\chi_{\mathbf{M}}(x)$ *of* $\mathbf{M}$ *over* $\mathbb{Q}$ *is* 1*; and*
- $\mathbf{M} \vDash \theta^d_{char}[k, i]$ *iff the* i*-th bit of the* denominator *of the coefficient of* x^k *in the characteristic polynomial* $\chi_{\mathbf{M}}(x)$ *of* $\mathbf{M}$ *over* $\mathbb{Q}$ *is* 1*.*

Recall that for any $n \times n$ matrix M, the constant term of $\chi_M(x)$ takes value $(-1)^n \cdot \det(M)$.

Corollary 3.13 (Determinant over $\mathbb{Q}$)**.** *There are* IFPC*-formulae* $\theta^n_{det}(\upsilon)$ *and* $\theta^d_{det}(\upsilon)$ *in vocabulary* $\tau^\star_{\mathbb{Q}}$*, where* υ *is a number variable, which for all square* $\tau_{\mathbb{Q}}$*-matrices* $\mathbf{M}$ *satisfy:*

- $\mathbf{M} \vDash \theta^n_{det}[i]$ *iff the* i*-th bit of the* numerator *of the determinant of* $\mathbf{M}$ *over* $\mathbb{Q}$ *is* 1*; and*
- $\mathbf{M} \vDash \theta^d_{det}[i]$ *iff the* i*-th bit of the* denominator *of the determinant of* $\mathbf{M}$ *over* $\mathbb{Q}$ *is* 1*.*

□

To prove Theorem 3.12, it is enough to show that we can describe the linear system $(*)$ in IFPC over any square $\tau_{\mathbb{Q}}$-structure $\mathbf{M}$. As outlined in §3.3, we can express in IFPC (a) the product of two matrices; (b) a matrix raised to the power k, where k can be expressed as a number term; and (c) the trace of a matrix. From this, it should be clear that we can define the linear system $A\mathbf{x} = \mathbf{b}$ from equation $(*)$ in IFPC over $\mathbf{M}$. By the Immerman-Vardi theorem, we can express any polynomial-time property of this linear system, because the matrices A and $\mathbf{b}$ are defined on a *linearly ordered* subset of the number sort. In particular, we can express Gaussian elimination as a fixed-point formula, use that to solve the system for $\mathbf{x}$ and hence obtain the coefficients of $\chi_{\mathbf{M}}(x)$. Theorem 3.12 now follows.

3.4.3 Characteristic polynomial over finite fields

Le Verrier's method involves division by integers up to the size of the matrix, so it cannot be applied directly over finite fields F of positive characteristic. Instead, we first map the input matrix to a ring of characteristic zero, apply Le Verrier's method, and then reduce the result back to get the specification of the characteristic polynomial over F. This approach was suggested by Blass and Gurevich in [7]. We consider separately two cases, one when F is a prime field and the other when F is a prime-power field.

Prime fields. Let $\mathbf{M}$ be a square τ_{fmat}-matrix over a finite field $\mathbf{F}$, where $\mathbf{F}$ is a τ_{field}-structure with p elements. By Lemma 3.1, there is an IFPC-definable isomorphism $\iota : \mathbf{F} \to \mathbb{Z}_p$. Hence, we can assume without loss of generality that the elements of $\mathbf{M}$ are integers in the range $[0, p-1]$. To express the characteristic polynomial of $\mathbf{M}$ over $\mathbf{F}$, we (a) first map $\mathbf{M}$ to a matrix $\mathbf{M}^+$ over the ring of integers, (b) apply Le Verrier's method to $\mathbf{M}^+$ over $\mathbb{Z}$, and then (c) reduce the result modulo p to get the specification of the characteristic polynomial over $\mathbf{F}$, with integer coefficients in the range $[0, p-1]$. The binary expansion of each element of $\mathbf{M}$ can be described in FOC, thereby defining the matrix $\mathbf{M}^+$ (see e.g. Libkin [52, Theorem 6.12]). Here we use the fact that FOC has addition and multiplication over the number sort. Likewise, the binary representations $\theta^{\text{n}}_{\text{char}}(\mu, \upsilon)$ and $\theta^{\text{n}}_{\text{det}}(\upsilon)$ from Theorem 3.12 and Corollary 3.13, respectively, can be reduced modulo p to an integer in $[0, p-1]$ with a formula of IFPC. Here we use the fact that the bit positions are ordered and so we can express any polynomial-time computation in fixed-point logic.

Prime-power fields. Let $\mathbf{M}$ be a square τ_{fmat}-matrix over a finite field $\mathbf{F}$, where $\mathbf{F}$ is a τ_{field}-structure with $q = p^d$ elements, where p is prime and $d > 1$. Consider a primitive element $\alpha \in U(\mathbf{F})$. By Lemma 3.7, there is a formula of IFPC that defines over $(\mathbf{F}, \alpha)$ the minimal polynomial $f(X)$ of α over $\mathsf{GF}_p[X]$. As discussed in §3.2.2, this polynomial is monic and irreducible of degree d over $\mathsf{GF}_p[X]$. To describe the characteristic polynomial of M over $\mathbf{F}$, we follow these steps. First, we define a polynomial $g(X)$ over $\mathbb{Z}$ whose reduction modulo p is $f(X)$. This can be done trivially, for the coefficients of $f(X)$ are already given by integers in the range $[0, p-1]$, according to Lemma 3.7. Next we lift the matrix $\mathbf{M}$ to a matrix $\mathbf{M}^+$ over the commutative ring $R = \mathbb{Z}[X]/(g(X))$, according to the IFPC-definable isomorphism given by Corollary 3.8. Finally, we apply Le Verrier's method over R to the matrix $\mathbf{M}^+$ and then reduce the output modulo p to get the correct result. This last reduction is sound as we have $\mathbf{F} = R/(p\mathbb{Z})$.

Addition and multiplication of elements in R is carried out coefficient-by-coefficient, and can be expressed by formulae of IFPC. This follows from the Immerman-Vardi theorem

since the polynomial coefficients are linearly ordered. The same argument shows that we can define reduction of polynomials modulo $f(X)$ with a fixed-point formula. Multiplication of matrices over R can be defined in IFPC, by an argument similar to the one given in the proof of Theorem 3.10. It should now be clear the we can describe the characteristic polynomial over R. Reducing the result modulo p in the end is straightforward, as before.

Finally, note that we are here using a linear ordering which depends on the choice of primitive element α. However, as the outcome of each step that we describe above does not rely on the actual ordering of the field elements, it can be seen that the overall query is order-invariant. Therefore, the result stated below is obtained by quantifying over *all* primitive elements of **F**, which are definable in FOC+DTC by Corollary 3.4.

Putting all the above together we get the following theorem, which says that the characteristic polynomial, and hence the determinant, can be defined in IFPC over any finite-field matrix.

Theorem 3.14 (Characteristic polynomial over finite fields)**.** *There are* IFPC*-formulae* $\theta_{det}(z)$ *and* $\theta_{char}(z, v)$ *in vocabulary* $\tau^{\star}_{fmat}$*, where* z *is an element variable and* v *is a number variable, which for any square* τ_{fmat}*-matrices* **M** *over a finite field* **F** *satisfy:*

- $\mathbf{M}^{\star} \models \theta_{det}[d]$ *iff the determinant of* **M** *over* **F** *is* $d \in U(\mathbf{F})$*;*
- $\mathbf{M}^{\star} \models \theta_{char}[d, k]$ *iff the coefficient of* x^k *in the characteristic polynomial* $\chi_{\mathbf{M}}(x)$ *of* **M** *over* **F** *is* $d \in U(\mathbf{F})$*.*

□

3.5 Rank and minimal polynomial over the rationals

We conclude this chapter by studying properties of matrices over the field of rationals. Our main result is that both the rank and the minimal polynomial of rational matrices can be defined in IFPC. These results both rely on properties of certain inner products over $\mathbb{Q}$ (and more generally over $\mathbb{C}$ and $\mathbb{R}$) which do not hold over fields of positive characteristic, as we will explain in further detail below.

3.5.1 Rank over $\mathbb{Q}$

Let A be a matrix over $\mathbb{Q}$, not necessarily square, and write $A^* := A^{\mathrm{t}}A$. The matrix A^* is square and symmetric, for $(A^*)^{\mathrm{t}} = (A^{\mathrm{t}}A)^{\mathrm{t}} = A^{\mathrm{t}}A = A^*$. For the following lemma, we use the fact that for any $n \geq 1$, the dot product $\langle \cdot, \cdot \rangle : \mathbb{Q}^n \times \mathbb{Q}^n \to \mathbb{Q}$ defined by $\langle x, y \rangle := x^{\mathrm{t}}y$ is an inner product on the vector space $\mathbb{Q}^n$. In particular, $\langle x, x \rangle = 0$ if and only if $x = 0$.

Lemma 3.15. *For any matrix* A *over* $\mathbb{Q}$ *it holds that* $\operatorname{rank} A = \operatorname{rank} A^* = \operatorname{rank}(A^*A^*)$.

Proof. Let A be an $m \times n$ matrix over $\mathbb{Q}$. Recall that the kernel of A is the subspace of $\mathbb{Q}^n$ which consists of all vectors that are annihilated by A; that is,

$$\ker(A) := \{x \in \mathbb{Q}^n \mid Ax = 0\}.$$

Furthermore, we know that $\operatorname{rank}(A) = n - \dim \ker(A)$, by the rank-nullity theorem (see §2.8 for further details). We claim that $\ker A = \ker(A^{\mathrm{t}}A)$, which then implies that $\operatorname{rank}(A) =$

$\operatorname{rank}(A^*)$. To prove this claim, first observe that $\ker(A) \subseteq \ker(A^{\mathrm{t}}A)$, for if $Ax = 0$ then $A^{\mathrm{t}}Ax = A^{\mathrm{t}}(Ax) = 0$ too. For the other inclusion, consider $x \in \mathbb{Q}^n$. Then

$$\begin{aligned} x \in \ker(A^{\mathrm{t}}A) &\Leftrightarrow A^{\mathrm{t}}Ax = 0 \\ &\Rightarrow x^{\mathrm{t}}A^{\mathrm{t}}Ax = 0 \\ &\Rightarrow (Ax)^{\mathrm{t}}(Ax) = 0 \\ &\Rightarrow (Ax) = 0, \end{aligned}$$

which shows that $\ker(A^{\mathrm{t}}A) \subseteq \ker(A)$. Similarly, it can be shown that $\ker A^* = \ker(A^*A^*)$, using the fact that A^* is symmetric. The lemma now follows. □

With this in mind, the following lemma (see e.g. Kozen [50, Lemma 32.1]) tells us that the rank of a rational matrix A can be inferred directly from its characteristic polynomial.

Lemma 3.16. *Let A be an $n \times n$ matrix over any field. If* $\operatorname{rank} A = \operatorname{rank} A^2$*, then* $\operatorname{rank} A = n - k$ *where x^k is the highest power of x that divides the characteristic polynomial $\chi_A(x)$.* □

Now consider an $m \times n$ matrix A over $\mathbb{Q}$. The above results show that the rank of A can be computed in the following steps:

(*1*) compute the matrix $A^* = A^{\mathrm{t}}A$;

(*2*) calculate the characteristic polynomial $p_{A^*}(x)$ of A^*; and

(*3*) find x^d, the highest power of the x that divides $p_{A^*}(x)$. Then the rank of A is $n - d$, where A^* has dimension $n \times n$.

As all the computation steps outlined above can be described in IFPC, we get the following result.

Corollary 3.17 (Rank over the rationals)**.** *There is a numeric* IFPC*-term θ_{rank} of vocabulary $\tau^\star_{\mathbb{Q}}$ which for all finite $\tau_{\mathbb{Q}}$-structures $\mathbf{A}$ satisfies: $\theta^{\mathbf{A}^\star}_{rank} = r$ iff the rank of $\mathbf{A}$ over $\mathbb{Q}$ is r.* □

Finally, remark that the statement of Lemma 3.16 holds more generally for matrices over $\mathbb{R}$ or $\mathbb{C}$ if we take $A^\star := \overline{A}^{\mathrm{t}}A$, where $\overline{A}^{\mathrm{t}}$ is the transpose of A with every entry replaced its complex conjugate. However, the statement does not hold over finite fields. For instance, over GF_p, the $p \times p$ all-ones matrix J_p has rank one, but $J_p^{\mathrm{t}}J_p = 0$ has rank zero. Essentially, the problem here is that the vector space map $(x, y) \mapsto x^{\mathrm{t}}y$ is *not* an inner product when the underlying field has positive characteristic (the condition "$x^{\mathrm{t}}x = 0$ iff $x = 0$" is violated).

3.5.2 Minimal polynomial over $\mathbb{Q}$

The minimal polynomial of a square matrix A is the monic polynomial $m_A(x)$ of smallest degree m such that

$$m_A(A) = A^m + a_{m-1}A^{m-1} + \cdots + a_1A + a_0I = 0.$$

The minimal polynomial divides any polynomial $q(x)$ with $q(A) = 0$. In particular, it divides the characteristic polynomial $\chi_A(x)$ (see e.g. Horn and Johnson [42]). Hoang and Thierauf [41] give a polynomial-time algorithm for computing the coefficients of the minimal polynomial, which crucially does not require a computation of the eigenvalues of A. In this section we briefly review this algorithm, and show that it can be expressed in IFPC.

Let A be an $J \times J$ matrix over $\mathbb{Q}$, where J is a finite set of cardinality $n > 0$. Then by definition of the minimal polynomial it follows that A has a minimal polynomial of degree m if and only if

(*1*) there is a monic polynomial $p(x)$ of degree m for which it holds that $p(A) = 0$; and

(*2*) for every monic polynomial $q(x)$ of degree $k < m$, $q(A) \neq 0$.

For an $r \times s$ matrix B, we write $\mathrm{vec}(B)$ to denote the column vector of length $r \cdot s$ obtained by stacking the columns of B one below the other. For $i = 1, \ldots, n$, let $\mathbf{v}_i = \mathrm{vec}(A^i)$. Then it can be seen that condition (*1*) from above is equivalent to saying that there exist $x_0, \ldots, x_{m-1} \in \mathbb{Q}$ such that

$$\mathbf{v}_m + x_{m-1}\mathbf{v}_{m-1} + \cdots + x_0\mathbf{v}_0 = 0, \tag{†}$$

which states that the vectors $\{\mathbf{v}_0, \ldots, \mathbf{v}_m\}$ should be linearly *dependent* over $\mathbb{Q}$. Similarly, condition (*2*) that all monic polynomials $q(x)$ of degree $k < m$ should have $q(A) \neq 0$, is equivalent to saying that for all $k < m$ and all $x_0, \ldots, x_{k-1} \in \mathbb{Q}$ it holds that

$$\mathbf{v}_k + x_{k-1}\mathbf{v}_{k-1} + \cdots + x_0\mathbf{v}_0 \neq 0, \tag{‡}$$

which states that the vectors $\{\mathbf{v}_0, \ldots, \mathbf{v}_k\}$ should be linearly *independent* over $\mathbb{Q}$. Hence, we see that the set of coefficients $\mathbf{a} = (a_0, \ldots, a_{m-1})$ of the minimal polynomial $m_A(x)$ is a solution to equation (†), in unknowns $x_0, \ldots, x_{m-1}$, for the least m for which it has a solution. Indeed, for this value of m such a solution will be unique. This gives us an algorithm to compute $m_A(x)$:

(*a*) Determine the least $m \leq n$ such that the vectors $\{\mathbf{v}_0, \ldots, \mathbf{v}_m\}$ are linearly dependent and the vectors $\{\mathbf{v}_0, \ldots, \mathbf{v}_{m-1}\}$ are linearly independent. This m will be the degree of $m_A(x)$.

(*b*) Solve the linear system $\mathbf{v}_m + x_{m-1}\mathbf{v}_{m-1} + \cdots + x_0\mathbf{v}_0 = 0$, in unknowns $x_0, \ldots, x_{m-1}$.

The linear system in step (*b*) above can be written as $B_m\mathbf{x} = -\mathbf{v}_m$, where $\mathbf{x} = (x_{m-1}, \ldots, x_0)^t$ and $B_m = (\mathbf{v}_0 \mid \ldots \mid \mathbf{v}_{m-1})$ is a matrix indexed by $J^2 \times [m]$, for $m \in [n]$. Since the columns $\{\mathbf{v}_0, \ldots, \mathbf{v}_{m-1}\}$ are independent by assumption, it follows that the matrix B_m has full column rank. Hence, the system will have a unique solution, as expected.

The algorithm we have described here can be expressed in IFPC as follows. Firstly, note that $\mathrm{rank}(M^{\mathrm{t}}M) = \mathrm{rank}(M)$ for any rational matrix M, as stated by Lemma 3.15. Hence, the vectors $\{\mathbf{v}_0, \ldots, \mathbf{v}_{m-1}\}$ are independent if and only if $\mathrm{rank}(B_m^{\mathrm{t}}B_m) = m$. In other words, $\{\mathbf{v}_0, \ldots, \mathbf{v}_{m-1}\}$ are independent if and only if the square matrix M_m has full rank, where $M_m \coloneqq B_m^{\mathrm{t}}B_m$. Thus,

$$\{\mathbf{v}_0, \ldots, \mathbf{v}_{m-1}\} \text{ are independent } \Leftrightarrow \det(M_m) \neq 0.$$

This test can be expressed in IFPC, for each $m = 1, \ldots, n$, using Corollary 3.13. To find the degree of the minimal polynomial in (*a*), we simply have to iterate this until we find m where $\det(M_m) \neq 0$ and $\det(M_{m+1}) = 0$. Having found this value of m, in step (*b*) we want to solve the system

$$B_m\mathbf{x} = -\mathbf{v}_m. \tag{§}$$

Define a new system $C\mathbf{x} = \mathbf{b}$, where $C = B_m^{\mathrm{t}} B_m$ is an $[m] \times [m]$ matrix and $\mathbf{b} = -B_m^{\mathrm{t}} \mathbf{v}_m$. In particular, note that the rows and columns of C are linearly ordered over the integers. Since $C = M_m$ is non-singular, we solve the system (§) by taking

$$\mathbf{x} = C^{-1}\mathbf{b}.$$

This step can also be expressed in IFPC; the matrix C can be defined in IFPC as we described in §3.3 and since the rows and columns of C are linearly ordered, its inverse C^{-1} can be defined in IFPC (even IFP) by the Immerman-Vardi theorem. We conclude with the following theorem.

Theorem 3.18 (Minimal polynomial over $\mathbb{Q}$)**.** *There are formulae $\theta^n_{min}(\mu, \upsilon)$ and $\theta^d_{min}(\mu, \upsilon)$ in* IFPC$[\tau^\star_{\mathbb{Q}}]$*, where μ and υ are number variables, which for all square $\tau_{\mathbb{Q}}$-matrices $\mathbf{A}$ satisfy:*

- $\mathbf{A}^\star \vDash \theta^n_{min}[k, i]$ *iff the i-th bit of the* numerator *of the coefficient of x^k in the minimal polynomial $m_{\mathbf{A}}(x)$ of $\mathbf{A}$ over $\mathbb{Q}$ is* 1*; and*
- $\mathbf{A}^\star \vDash \theta^d_{min}[k, i]$ *iff the i-th bit of the* denominator *of the coefficient of x^k in the minimal polynomial $m_{\mathbf{A}}(x)$ of $\mathbf{A}$ over $\mathbb{Q}$ is* 1*.*

□

Chapter 4

Logics with matrix rank operators

It has been observed in recent years that many of the problems separating IFP and IFPC from PTIME relate to the inability of these logics to express certain basic properties from linear algebra. For instance, it has been shown that over finite fields, IFP is unable to determine whether or not a square matrix is singular [9] and IFPC is unable to define matrix rank [4, 16]. Both of these matrix properties are computable in polynomial time by Gaussian elimination, for instance.

In order to address these shortcomings of the two logics, it is natural to consider extensions of fixed-point logic with operators for defining basic linear-algebraic properties. Here our focus is on well-defined properties of *unordered* matrices, whose rows and columns are indexed by arbitrary sets. This is because on ordered matrices, every polynomial-time computable property can already be defined in IFP, by the Immerman-Vardi theorem. In particular, there is a fixed-point formula that, by performing Gaussian elimination, defines the row-reduced echelon form of any ordered matrix, from which both the rank and singularity can be deduced. By 'well-defined' we mean matrix properties that are invariant under simultaneous permutation of the rows and columns. It can be readily seen that *singularity*, *determinant* and *rank* are all well-defined matrix properties in this sense.

In [9], Blass et al. showed that the class of square singular matrices can be defined in IFPC over finite fields, over the ring of integers and over the field of rational numbers. In the previous chapter, we showed that over each of the three aforementioned domains, IFPC can express the characteristic polynomial—and hence the determinant—of any square matrix. Furthermore, we showed that the rank of rational-valued matrices can already be defined in IFPC. Together, these results focus attention specifically on matrix rank over *finite fields* as an algebraic property that separates IFPC from PTIME.

To address this shortcoming of IFPC, we introduce in this chapter an extension of fixed-point logic with terms to express the rank of definable matrix relations over a finite field. In this setting, we identify a binary relation $R \subseteq A \times A$ over a set A with a $(0,1)$-matrix $M = (m_{ij})$ by letting $m_{ij} = 1$ if $(i, j) \in R$ and $m_{ij} = 0$ otherwise. A *rank term* is an expression of the form $\mathbf{rk}_p(\vec{x}, \vec{y}).\varphi$, where the rank operator $\mathbf{rk}_p$ binds the tuples of variables $\vec{x}$ and $\vec{y}$ in the formula φ and denotes over a finite structure $\mathbf{A}$ the number that is the rank of the binary relation $\varphi(\vec{x}, \vec{y})^{\mathbf{A}}$ interpreted as a $(0,1)$-matrix over the finite field GF_p, where p is a prime number. More generally, we consider rank operators that bind number terms instead of formulae, so that we can describe the rank of definable matrices that contain entries other

than just zero and one. The logic IFPR is defined by extending fixed-point logic with rules for forming rank terms of this kind. We show that this logic can both simulate counting and define solvability of systems of linear equations over any finite field. It follows that IFPR is strictly more expressive than IFPC. Moreover, since matrix rank can be computed in polynomial time, it follows that IFPR has polynomial-time data complexity, which is to say that all properties of finite structures definable in IFPR are decidable in polynomial time. Together, this implies that IFPC $\subsetneqq$ IFPR $\leqq$ PTIME.

Apart from extensions of fixed-point logic with rank operators over finite fields, we also consider extensions of first-order logic with finite-field rank operators and logics with rank operators over the field of rational numbers. This will be the topic of §4.1, where we define each of these rank logics and study some of their basic properties. In particular, we show that each type of rank operator can simulate counting. In §4.2 we study the problem of deciding solvability of systems of linear equations. We show that for each prime p, the class of solvable linear systems over GF_p can be defined in FOR_p, the extension of first-order logic with rank operators of the form $\mathbf{rk}_p$. Furthermore, we show that IFPR_p, the extension of fixed-point logic with rank operators of the form $\mathbf{rk}_p$, can define solvability of linear systems over GF_{p^d} for *any* exponent $d \in \mathbb{N}$. We then study *arity hierarchies* of the rank logics FOR and IFPR in §4.3. Here, we define the arity of a rank term $\mathbf{rk}_p(\vec{x}, \vec{y}).\varphi$ to be the total number of distinct variables in $\vec{x} \cup \vec{y}$. Writing $\mathrm{FOR}_{p;m}$ and $\mathrm{IFPR}_{p;m}$ to denote the sublogic of FOR_p and IFPR_p, respectively, obtained by allowing only rank terms of arity at most m, we show that the arity hierarchies $\mathrm{FOR}_{p;2} \leqq \mathrm{FOR}_{p;3} \leqq \ldots$ and $\mathrm{IFPR}_{p;2} \leqq \mathrm{IFPR}_{p;3} \leqq \ldots$ are strict for each prime p. This contrasts with the counting logic IFPC, for which it can be shown that unary counting operators suffice to define counting in any arity. Finally, we conclude by giving a summary in §4.4 of all the rank logics defined in this chapter, illustrating their relations with other rank logics as well as some of the other logics we have studied previously.

4.1 Rank logics

In this section we introduce extensions of first-order and fixed-point logic with operators that express the rank of a definable matrix. Here we focus on three kinds of rank logics. Firstly, in §4.1.2 we consider numerical extensions of first-order and fixed-point logic with rank operators for matrices over finite fields. In §4.1.3 we define similar numerical logics with operators for defining the rank of rational-valued matrices. Finally, in §4.1.4 we define (non-numerical) extensions of finite-variable infinitary logic with rank *quantifiers* over finite fields. These infinitary rank logics subsume both first-order and fixed-point logic with rank operators over finite fields, as we will see.

The rank operators and rank quantifiers we define apply to matrices described by number terms or formulae. This kind of notation was defined in Chapter 3 for matrices over $\mathbb{Z}$ and $\mathbb{Q}$. We begin our discussion by establishing in §4.1.1 the corresponding notation for describing matrices over finite fields.

4.1.1 Specifying matrices over GF_p by number terms or formulae

Here we define notation for specifying matrices over finite fields in a two-sorted (numerical) logic. More specifically, for prime p we introduce two alternative ways to describe matrices over GF_p: one by giving a single number term, which is reduced modulo p at every position,

and another by giving a $(p-1)$-tuple of formulae, specifying which matrix entries are non-zero field elements.

Definition 4.1 (Matrices over GF_p described by number terms)**.** Let $\vec{x}$ and $\vec{y}$ be tuples of element variables and consider a number term η in vocabulary τ where $\mathrm{free}(\eta) \subseteq \vec{x} \cup \vec{y}$.

- Given a finite τ-structure $\mathbf{A}$, we write $\mathrm{mat}_{\vec{x},\vec{y}}(\eta, \mathbf{A}) := \eta(\vec{x}, \vec{y})^{\mathbf{A}^\star}$ to denote the $U(\mathbf{A})^{\|\vec{x}\|} \times U(\mathbf{A})^{\|\vec{y}\|}$ integer matrix defined by $\eta(\vec{x}, \vec{y})$ over $\mathbf{A}^\star$. That is, if we write $\mathrm{mat}_{\vec{x},\vec{y}}(\eta, \mathbf{A}) = (m_{\vec{a}\vec{b}})$ then
$$m_{\vec{a}\vec{b}} = \eta[\vec{a}, \vec{b}]^{\mathbf{A}},$$
for all $\vec{a} \in U(\mathbf{A})^{\|\vec{x}\|}$ and $\vec{b} \in U(\mathbf{A})^{\|\vec{y}\|}$.
- Let p be prime. Given a finite τ-structure $\mathbf{A}$, we write
$$\mathrm{fmat}_{\vec{x},\vec{y}}(\eta, \mathbf{A})_p := \mathrm{mat}_{\vec{x},\vec{y}}(\eta, \mathbf{A}) \pmod p$$
to denote the matrix over GF_p obtained from $\mathrm{mat}_{\vec{x},\vec{y}}(\eta, \mathbf{A})$ by reducing each matrix entry modulo p.

■

Overloading our notation, we also consider matrices defined in this way by formulae, rather than number terms.

Definition 4.2 (Matrices over GF_p described by formulae)**.** Consider a prime p and let $\vec{x}$ and $\vec{y}$ be tuples of element variables.

- Consider a formula φ in vocabulary τ where $\mathrm{free}(\varphi) \subseteq \vec{x} \cup \vec{y}$. Given a finite τ-structure $\mathbf{A}$, we write $\mathrm{fmat}_{\vec{x},\vec{y}}(\varphi, \mathbf{A})_p$ to denote the $U(\mathbf{A})^{\|\vec{x}\|} \times U(\mathbf{A})^{\|\vec{y}\|}$ $(0,1)$-matrix over GF_p defined for all $\vec{a} \in U(\mathbf{A})^{\|\vec{x}\|}$ and $\vec{b} \in U(\mathbf{A})^{\|\vec{y}\|}$ by
$$(\vec{a}, \vec{b}) \mapsto 1 \Leftrightarrow \mathbf{A} \models \varphi[\vec{a}, \vec{b}].$$
- Let $\Phi = (\varphi_1, \ldots, \varphi_{p-1})$ be a tuple of formulae in vocabulary τ, with $\mathrm{free}(\varphi_i) \subseteq \vec{x} \cup \vec{y}$ for all i. Given a finite τ-structure $\mathbf{A}$, we write $\mathrm{fmat}_{\vec{x},\vec{y}}(\Phi, \mathbf{A})_p$ for the $U(\mathbf{A})^{\|\vec{x}\|} \times U(\mathbf{A})^{\|\vec{y}\|}$ matrix over GF_p defined by
$$\mathrm{fmat}_{\vec{x},\vec{y}}(\Phi, \mathbf{A})_p := \sum_{i=1}^{p-1} i \cdot \mathrm{fmat}_{\vec{x},\vec{y}}(\varphi_i, \mathbf{A})_p \pmod p.$$

■

Example 4.3. For any prime field GF_p, the formula $\neg(x = y)$ defines a square matrix in which the entries outside the main diagonal are one and all the diagonal entries are zero. Similarly, for any formula $\varphi(x)$, $(x = y \wedge \varphi(x))$ interpreted in a structure $\mathbf{A}$ defines a square diagonal matrix, with 1 in position $(a, a) \in A \times A$ on the diagonal if, and only if, $(\mathbf{A}, a) \models \varphi$. ■

4.1.2 Logics with rank operators over prime fields

First-order logic with variable rank ($\mathrm{FOR}_{\mathrm{var}}$) is a numerical logic with operators for defining the rank of matrices over prime fields. The terms and formulae of this logic are defined inductively in exactly the same way as the terms and formulae of FOC (see §2.5), except that we replace the rule for forming counting terms of the kind $\#_x\varphi$ with the following rule for constructing *rank terms* over prime fields:

> For all $\mathrm{FOR}_{\mathrm{var}}$ number terms t and all tuples of element variables $\vec{x}$ and $\vec{y}$, if η is a number term or a formula of $\mathrm{FOR}_{\mathrm{var}}$ then $\mathbf{rk}(\vec{x},\vec{y}).(\eta,t)$ is a number term of $\mathrm{FOR}_{\mathrm{var}}$. We let $\mathrm{free}(\mathbf{rk}(\vec{x},\vec{y}).(\eta,t)) := (\mathrm{free}(\eta) \setminus (\vec{x}\cup\vec{y})) \cup \mathrm{free}(t)$.

The semantics of rank terms of $\mathrm{FOR}_{\mathrm{var}}$ over vocabulary τ are defined for all pairs $(\mathbf{A}^\star,\alpha)$, where $\mathbf{A}$ is a finite τ-structure, as follows:

$$\alpha(\mathbf{rk}(\vec{x},\vec{y}).(\eta,t)) := \begin{cases} \mathrm{rank}(\mathrm{fmat}_{\vec{x},\vec{y}}(\eta,\mathbf{A})_p) & \text{if } \alpha(t)=p \text{ is a prime number ,} \\ 0 & \text{otherwise.} \end{cases}$$

For prime p, we also consider the logic FOR_p which is defined like $\mathrm{FOR}_{\mathrm{var}}$ except that we replace the above rule for forming rank terms with a rule for constructing terms of the following kind:

> If η is a number term or a formula of FOR_p and $\vec{x}$, $\vec{y}$ tuples of element variables, then $\mathbf{rk}_p(\vec{x},\vec{y}).\eta$ is a number term of FOR_p. We let $\mathrm{free}(\mathbf{rk}_p(\vec{x},\vec{y}).\eta) := (\mathrm{free}(\eta)\setminus(\vec{x}\cup\vec{y}))$.

The semantics of rank terms of FOR_p are defined like for $\mathrm{FOR}_{\mathrm{var}}$, where now all matrices are defined over GF_p. Finally, we write FOR to denote the numerical extension of first-order logic with all the rank operators $\mathbf{rk}_p$, for prime p.

We also consider extensions of fixed-point logic with rank operators. For prime p, the rank logic IFPR_p is obtained by extending IFP in the numerical setting with the rank operator $\mathbf{rk}_p$, just like we obtained IFPC by extending IFP with counting operators before. We write IFPR for the numerical extension of IFP with all the rank operators $\mathbf{rk}_p$, for prime p. Similarly, we write $\mathrm{IFPR}_{\mathrm{var}}$ for the numerical extension of IFP with rank operators over fields of variable characteristic.

It can be seen that for each prime p and each formula $\varphi \in \mathrm{FOR}_p$, there is a formula $\varphi' \in \mathrm{FOR}_{\mathrm{var}}$ which is logically equivalent to φ over finite structures. For instance, φ' can be obtained from φ by replacing every occurence of a rank term $\mathbf{rk}_p(\vec{x},\vec{y}).\eta$ in φ with the term $\mathbf{rk}(\vec{x},\vec{y}).(\eta,t_p)$, where $t_p \equiv 1_N + \cdots + 1_N$ is the number term defined by adding together p copies of the constant 1_N. Similar obversations can be made about the other rank logics we have considered, as stated by the following lemma.

Lemma 4.4. *For prime p,* $\mathrm{FOR}_p \leqq \mathrm{FOR} \leqq \mathrm{FOR}_{var}$ *and* $\mathrm{IFPR}_p \leqq \mathrm{IFPR} \leqq \mathrm{IFPR}_{var}$. □

For a formula $\varphi \in \mathrm{FOR}_{\mathrm{var}}$ of vocabulary τ, let

$$T(\varphi) := \{t \mid \text{a rank term } \mathbf{rk}(\vec{x},\vec{y}).(\eta,t) \text{ occurs in } \varphi\}$$

be the set of all number terms that define the prime characteristic of some rank term in φ. Given a finite τ-structure $\mathbf{A}$, let $\Pi(\varphi,\mathbf{A}) := \{t^{\mathbf{A}^\star} \mid t \in T(\varphi)\}$ denote the interpretation of

all the number terms in $T(\varphi)$ over $\mathbf{A}$ and set $\Pi(\varphi) := \bigcup_{\mathbf{A}\in\mathrm{fin}[\tau]} \Pi(\varphi, \mathbf{A})$. Now it can be seen that the key difference between the logics $\mathrm{FOR}_{\mathrm{var}}$ and FOR (and likewise for the logics $\mathrm{IFPR}_{\mathrm{var}}$ and IFPR) is that for each formula $\varphi \in \mathrm{FOR}$ there is a logically equivalent formula $\varphi' \in \mathrm{FOR}_{\mathrm{var}}$ for which the the set $\Pi(\varphi')$ is *finite*. This is generally not true for formulae in $\mathrm{FOR}_{\mathrm{var}}$, where each rank operator is applied to a field of prime characteristic p which may depend on the size of the underlying structure. However, the following lemma shows that the range of primes available to formulae in $\mathrm{FOR}_{\mathrm{var}}$ and $\mathrm{IFPR}_{\mathrm{var}}$ is not arbitrary.

Lemma 4.5. *Consider a vocabulary τ. For any number term t in $\mathrm{IFPR}_{var}[\tau]$ with $\mathrm{free}(t) = \varnothing$, there is a polynomial $q : \mathbb{N}_0 \to \mathbb{N}_0$ so that $t^{\mathbf{A}^\star} \leq q(\|\mathbf{A}\|)$ for any finite τ-structure $\mathbf{A}$.*

Proof. We prove this by induction over terms. Clearly, the statement holds for the number constants 0 and 1. Suppose now that s and t are number terms whose value is bounded above by polynomials q_s and q_t, respectively. Then the terms $s \cdot t$ and $s + t$ are bounded by polynomials $q_s q_t$ and $q_s + q_t$, respectively. Finally, consider a number term $s \equiv \mathbf{rk}(\vec{x}, \vec{y}).(\eta, t)$. Since the rank of any matrix is bounded above by both its row and column dimension, it follows that the value of s is bounded above by the polynomial n^k where $k = \min\{\|\vec{x}\|, \|\vec{y}\|\}$. □

Corollary 4.6. *Consider a vocabulary τ. For any formula $\varphi \in \mathrm{IFPR}_{var}[\tau]$ there is a polynomial $q : \mathbb{N}_0 \to \mathbb{N}_0$ so that $m \leq q(\|\mathbf{A}\|)$ for all $\mathbf{A} \in \mathrm{fin}[\tau]$ and $m \in \Pi(\varphi, \mathbf{A})$.* □

We have seen that rank logics are numerical logics defined in exactly the same way as counting logics, except that rules for forming counting terms are replaced with rules for forming rank terms. Alternatively, instead of *replacing* rules in this way we could have defined rank logics by *adding* the rules for constructing rank terms to the set of rules for the corresponding counting logic. This, however, would have made no difference in terms of expressive power, as the following theorem shows.

Theorem 4.7. *For prime p,* $\mathrm{FOC} \leqq \mathrm{FOR}_p$ *and* $\mathrm{IFPC} \leqq \mathrm{IFPR}_p$.

Proof. Consider a formula $\psi(x)$ in vocabulary τ. Define the formula $\varphi(x, y) \equiv (x = y) \wedge \psi(x)$. As in Example 4.3, it can be seen that for any prime p and finite τ-structure $\mathbf{A}$, $\mathrm{fmat}_{x,y}(\varphi, \mathbf{A})_p$ is a square diagonal $(0,1)$-matrix, with one in position $(a, a) \in U(\mathbf{A}) \times U(\mathbf{A})$ on the main diagonal if and only if $\mathbf{A} \vDash \varphi[a, a]$. Moreover, by the definition of φ, it holds that $\mathbf{A} \vDash \varphi[a, a]$ if and only if $\mathbf{A} \vDash \psi[a]$, for all $a \in U(\mathbf{A})$. The rank of the matrix $\mathrm{fmat}_{x,y}(\varphi, \mathbf{A})_p$ is just the number of non-zero entries along the main diagonal, which is the same as the number of satisfying assignments to ψ from $\mathbf{A}$, by the above. Hence, for any prime p it holds that

$$(\mathbf{rk}_p(\vec{x}, \vec{y}).\varphi)^{\mathbf{A}^\star} = (\#_x \psi)^{\mathbf{A}^\star},$$

for all finite τ-structures $\mathbf{A}$. Hence, counting terms can be simulated by rank terms. The theorem now follows by a simple induction on formulae. □

Finally, we note that all the rank logics $\mathrm{IFPR}_{\mathrm{var}}$, $\mathrm{FOR}_{\mathrm{var}}$, IFPR, FOR, IFPR_p and FOR_p (for prime p) are closed under Boolean operations as well as applications of rank operators (and thereby first-order quantification). It follows readily that all these logics are closed under first-order reductions.

4.1.3 Logics with rank operators over $\mathbb{Q}$

For completeness, we also define a logic with operators for expressing the rank of rational-valued matrices. The terms and formulae of *first-order logic with rank over* $\mathbb{Q}$ ($\mathrm{FOR}_{\mathbb{Q}}$) are defined inductively in exactly the same way as the terms and formulae of FOC, except that we replace the rule for forming counting terms of the kind $\#_x\varphi$ with the following rule for constructing rank terms over the rationals:

> Let $\vec{x}$ and $\vec{y}$ be tuples of element variables and let v be a number variable. If φ_n, φ_d and ψ are formulae of $\mathrm{FOR}_{\mathbb{Q}}$ then $\mathbf{rk}_{\mathbb{Q}}(\vec{x},\vec{y}).(\varphi_n,\varphi_d,\psi,t)$ is a number term of $\mathrm{FOR}_{\mathbb{Q}}$. We let
>
> $$\begin{aligned}&\mathrm{free}(\mathbf{rk}_{\mathbb{Q}}(\vec{x},\vec{y}).(\varphi_n,\varphi_d,\psi,t)) :=\\ &((\mathrm{free}(\varphi_n)\cup\mathrm{free}(\varphi_d)\cup\mathrm{free}(\psi))\setminus(\vec{x}\cup\vec{y}))\cup\mathrm{free}(t).\end{aligned}$$

The semantics of rank terms of $\mathrm{FOR}_{\mathbb{Q}}$ over vocabulary τ are defined for all pairs $(\mathbf{A}^\star,\alpha)$, where $\mathbf{A}$ is a τ-structure, as follows:

$$\alpha(\mathbf{rk}_{\mathbb{Q}}(\vec{x},\vec{y}).(\varphi_n,\varphi_d,\psi,t)) := \begin{cases}\mathrm{rank}(M) & \text{if } M=\mathrm{mat}_{\vec{x},\vec{y},v}(\varphi_n,\varphi_d,\psi,t,\mathbf{A}) \text{ is defined},\\ 0 & \text{otherwise,}\end{cases}$$

where $\mathrm{mat}_{\vec{x},\vec{y},v}(\varphi_n,\varphi_d,\psi,t,\mathbf{A})$ denotes the matrix with entries from $\mathbb{Q}$ we defined in §3.3.1. Recall from Corollary 3.17 that the rank of rational matrices can be expressed in IFPC. Also, we can see that the simulation of counting terms by rank terms defined above for rank operators $\mathbf{rk}_p$ (Theorem 4.7) is valid for rank operators over $\mathbb{Q}$ as well. Therefore, the logic $\mathrm{IFPR}_{\mathbb{Q}}$, obtained by extending IFP in the numerical setting with rules for constructing rank terms over $\mathbb{Q}$, coincides exactly with IFPC over finite structures. We summarise these observations as follows.

Corollary 4.8. *Over finite structures,* $\mathrm{FOR}_{\mathbb{Q}} \leqq \mathrm{IFPR}_{\mathbb{Q}} = \mathrm{IFPC}$. □

4.1.4 Infinitary logic with rank quantifiers

In this section we consider extensions of finite-variable infinitary logic with quantifiers for expressing matrix rank. These *rank quantifiers*, which can be seen as special types of the Lindström quantifiers which we discussed in §2.2.7, are defined as follows.

For each integer $i \geq 0$ and prime p, define an m-ary rank quantifier $\mathrm{rk}_p^{\geq i}$ which binds exactly m variables and $(p-1)$-formulae. For each prime p and integers $k, m \geq 1$, with $m \leq k$, we write $\mathcal{R}^k_{p;m}$ to denote *k-variable infinitary rank logic of arity m over* GF_p. This logic is obtained by extending the formula-formation rules for k-variable infinitary logic $\mathcal{L}^k$ with the following rule:

> If $\varphi_1,\ldots,\varphi_{p-1}$ are formulae, $\vec{x}$ and $\vec{y}$ are non-empty tuples of distinct variables with $\|\vec{x}\cup\vec{y}\| = m$ and $i \geq 0$, then $\mathrm{rk}_p^{\geq i}(\vec{x},\vec{y}).(\varphi_1,\ldots,\varphi_{p-1})$ is a formula. We let $\mathrm{free}(\mathrm{rk}_p^{\geq i}(\vec{x},\vec{y}).(\varphi_1,\ldots,\varphi_{p-1})) := (\bigcup_i \mathrm{free}(\varphi_i)\setminus(\vec{x}\cup\vec{y}))$.

The semantics of rank quantifiers of $\mathcal{R}^k_{p;m}$ over vocabulary τ are defined for all pairs $(\mathbf{A}^\star, \alpha)$, where $\mathbf{A}$ is a finite τ-structure, as follows:

$$\mathbf{A} \vDash \mathrm{rk}^{\geq i}_p(\vec{x}, \vec{y}).(\varphi_1, \ldots, \varphi_{p-1}) \text{ if and only if } \operatorname{rank} \mathrm{fmat}_{\vec{x},\vec{y}}((\varphi_1, \ldots, \varphi_{p-1}), \mathbf{A})_p \geq i.$$

Additionally, we can define rank quantifiers $\mathrm{rk}^{=i}_p$, $\mathrm{rk}^{\leq i}_p$, $\mathrm{rk}^{<i}_p$ and $\mathrm{rk}^{>i}_p$, by a simple combination of $\mathrm{rk}^{\geq i}_p$-quantifiers.

We write $\mathcal{R}^\omega$ to denote finite-variable infinitary logic with rank operators of any arity and over any prime field. For $m \geq 2$, we write $\mathcal{R}^\omega_{*;m}$ to denote the fragment of $\mathcal{R}^\omega$ in which each formula uses only rank operators arity at most m, and for prime p, we write $\mathcal{R}^\omega_{p;m}$ to denote the fragment of $\mathcal{R}^\omega_{*;m}$ in which each formula uses only rank operators over GF_p. We also define similar restrictions of first-order and fixed-point rank logics. That is, for each $m \geq 2$ we write $\mathrm{IFPR}_{*;m}$ for the class of all those $\mathrm{IFPR}_{\mathrm{var}}$-formulae in which all occurrences of rank operators are of arity at most m. Also, for $m \geq 2$ and prime p, we write $\mathrm{IFPR}_{p;m}$ for the restriction of $\mathrm{IFPR}_{*;m}$ where each formula has only rank operators of the form $\mathbf{rk}_p$. The corresponding restrictions of $\mathrm{FOR}_{\mathrm{var}}$ ($\mathrm{FOR}_{*;m}$ and $\mathrm{FOR}_{p;m}$) are defined in exactly the same way.

Our main interest in studying the infinitary logics $\mathcal{R}^k_{p;m}$ is to analyse the expressive power of first-order and fixed-point logics with operators for matrix rank. We will see examples of this later in §4.3, where we show that rank logics form a strict arity hierarchy, and in Chapter 6, where we develop a game-theoretic proof method for proving non-definability results for rank logics. Recall that by Theorem 2.13, we have $\mathrm{IFP} \leqq \mathcal{L}^\omega$ and $\mathrm{IFPC} \leqq \mathcal{C}^\omega$. In other words, both fixed-point logic and fixed-point logic with counting are subsumed by the corresponding infinitary logic. Below we establish a similar correspondence between fixed-point rank logics and infinitary rank logics.

First though, we need an intermediate lemma, to translate from rank terms binding a single number term to rank terms binding a tuple of formulae. That is, we write $\mathrm{IFPR}^\star_{p;m}$ for the logic defined in exactly the same way as $\mathrm{IFPR}_{p;m}$, except that the rule for forming rank terms is replaced with the following rule, where p is prime:

> If $\varphi_1, \ldots, \varphi_{p-1}$ are formulae of $\mathrm{IFPR}^\star_{p;m}$, $\vec{x}$ and $\vec{y}$ are non-empty tuples of distinct element variables with $\|\vec{x} \cup \vec{y}\| = m$, and $\Phi = (\varphi_1, \ldots, \varphi_{p-1})$, then $\mathbf{rk}_p(\vec{x}, \vec{y}).\Phi$ is a number term of $\mathrm{IFPR}^\star_{p;m}$.

Here the semantics are defined exactly like before, this time by considering the matrix defined by the tuple of formulae Φ.

Lemma 4.9. *For each integer $m \geq 2$ and prime p, $\mathrm{IFPR}_{p;m} \equiv \mathrm{IFPR}^\star_{p;m}$ over finite structures.*

Proof. Let $m \geq 2$ and p be prime. To show that $\mathrm{IFPR}_{p;m} \leqq \mathrm{IFPR}^\star_{p;m}$, consider a rank term of $\mathrm{IFPR}_{p;m}$ of the form $\mathbf{rk}_p(\vec{x}, \vec{y}).\eta$, where η is a number term. For each $i \in [p-1]$, define a formula φ_i by

$$\varphi_i(\vec{x}, \vec{y}) \equiv \exists \mu \leq \eta(\vec{x}, \vec{y})\, (\eta(\vec{x}, \vec{y}) = i + \mu \cdot p).$$

In other words, $\varphi_i(\vec{x}, \vec{y})$ defines the predicate "$\eta(\vec{x}, \vec{y}) \equiv i \pmod p$". Let $\Phi = (\varphi_1, \ldots, \varphi_{p-1})$. It is now clear that for any finite structure $\mathbf{A}$,

$$(\mathbf{rk}_p(\vec{x}, \vec{y}).\eta)^{\mathbf{A}^\star} = (\mathbf{rk}_p(\vec{x}, \vec{y}).\Phi)^{\mathbf{A}^\star},$$

as required. The remainder of the induction is straightforward. The other direction, $\mathrm{IFPR}^{\star}_{p;m} \leqq \mathrm{IFPR}_{p;m}$, can be proved similarly. □

Theorem 4.10. *For each integer $m \geq 2$ and prime p,* $\mathrm{IFPR}_{p;m} \lneqq \mathcal{R}^{\omega}_{p;m}$.

Sketch proof. To prove this theorem, we need to show that every formula of $\mathrm{IFPR}_{p;m}$ without free number variables can be translated into a formula of $\mathcal{R}^{\omega}_{p;m}$. Following the proof of Lemma 3.2 in [29], we show that all occurrences of number variables and generation of fixed-points can be expanded uniformly with respect to the cardinality of the underlying structure. That is, the formulae we construct in $\mathcal{R}^{\omega}_{p;m}$ will be of the form

$$\bigvee_{n<\omega} \exists^{=n} x\,(x = x) \wedge \varphi_n,$$

where φ_n captures the meaning of the formula over structures of size n. Note here that the logic $\mathcal{R}^{\omega}_{p;m}$ does not have actual counting quantifiers (and therefore we write these above only as shorthand) but we can simulate counting quantifiers with rank quantifiers, in a similar way as w simulated counting terms with rank terms before.

The expansion of first-order and fixed-point operators can be considered by standard means; for instance, see the proof of Corollary 1.30 in Otto [58]. Number variables and number terms can be dealt with in a similar way as in the proof of $\mathrm{IFPC} \lneqq \mathcal{C}^{\omega}$ by Grädel and Otto [29, Lemma 3.2], by replacing the translation of counting terms into counting quantifiers with a translation of rank terms into rank quantifiers. For completeness, we retrace the main argument here.

Consider a formula $\varphi(x, \upsilon)$ of $\mathrm{IFPR}^{\star}_{p;m} \equiv \mathrm{IFPR}_{p;m}$ (Lemma 4.9), where x is an element variable and υ is a number variable. We translate $\varphi(x, \upsilon)$ with respect to υ to a sequence of formulae $(\varphi_k(x))_{k<\omega}$, where for each $k < \omega$ the relation defined by the formula $\varphi_k(x)$ (with respect to x) agrees with the relation defined by the formula $\varphi(x, \upsilon)$ when υ is assigned integer value k. The induction argument is similar to the proof of Lemma 3.2 in [29], with the exception of formulae containing rank terms, which we explain here.

To give an example of the induction step with formulae involving rank operators, we consider an $\mathrm{IFPR}^{\star}_{p;m}$-formula $\varphi(x, \upsilon) \equiv \upsilon \leq \mathbf{rk}_p(\vec{y}, \vec{z}).(\psi_1, \ldots, \psi_{p-1})$, with $\|\vec{y} \cup \vec{z}\| = m$, the maximum arity, and each $\psi_i(x, \vec{y}, \vec{z})$ a formula. By the induction hypothesis, suppose that $\psi_{i,n}(x, \vec{y}, \vec{z})$ captures the meaning of $\psi_i(x, \vec{y}, \vec{z})$ on structures of size n. Then the uniform family for φ is defined with respect to n and k by

$$\varphi_{n,k}(x) \equiv \mathrm{rk}^{\geq k}_p(\vec{y}, \vec{z}).(\psi_{1,n}, \ldots, \psi_{p-1,n}),$$

where $k \leq n^m$. Here n is the parameter for the size of the structure and k is the parameter for the number variable υ. The rest of the induction proceeds as in the proof of Grädel and Otto.

Finally, since queries definable in $\mathrm{IFPR}_{p;m}$ are in PTIME, while $\mathcal{R}^{\omega}_{p;m}$ can express even non-recursive queries (see §2.6), it follows that the inclusion of $\mathrm{IFPR}_{p;m}$ in $\mathcal{R}^{\omega}_{p;m}$ is proper. □

Corollary 4.11. $\mathrm{IFPR}_{var} \lneqq \mathcal{R}^{\omega}$ *and* $\mathrm{IFPR}_{*;m} \lneqq \mathcal{R}^{\omega}_{*;m}$ *for each integer* $m \geq 2$. □

4.2 Systems of linear equations

Let G be group, written additively with identity 0_G. An *equation* over G is an expression of the form

$$v_1 + v_2 + \cdots + v_m = 0_G$$

where each v_i is either a variable, an inverted variable or a constant group element. The expression can be *satisfied* if there is an assignment of values from G to the variables so that the equality holds. A system of equations over G is a collection of such equations. A system of equations is said to be *solvable* is there is an assignment of values which simultaneously satisfies each equation.

More often, we consider equations where the variables are allowed to take values in a field instead of a group. Let F be a field and write $+$ and $\cdot$ for addition and multiplication in F, respectively. A *linear equation* over F is an expression of the form

$$a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots + a_m \cdot x_m = b,$$

where b is a constant element from F, each x_i is a variable and each scalar coefficient a_i is a constant element from F. A *system of linear equations* over F (or *linear system*, for short) is a collection of such expressions; the system is said to be solvable if there is an assignment of the variables to elements in F that simultaneously satisfies each equation.

The complexity of determining the solvability of a system of equations varies according to the domain that the variables are assigned values from. It is known that the problem of deciding solvability of a system of equations over a fixed finite group is in PTIME if the group is Abelian and NP-complete otherwise [28]. When we consider linear equations over a field we can write the system as a matrix equation $A\mathbf{x} = \mathbf{b}$ and apply methods from linear algebra to its study. Such a system is solvable if and only if $\mathbf{b}$ is contained in the span of the column vectors of A; or in other words if and only if the two matrices A and $(A|\mathbf{b})$ have the same *rank*. This shows that the solvability of a system of linear equations over a field can be decided in PTIME since matrix rank can be computed in polynomial time by Gaussian elimination, say.

Atserias, Bulatov and Dawar [4] considered the problem of defining solvable systems of equations over a finite Abelian group. They showed that for any fixed finite Abelian group G with at least two elements, the class of solvable systems is not definable in finite-variable infinitary logic with counting. When the group G arises as the additive group of a finite field F, the problem of deciding solvability of a linear system over G can be trivially reduced to the problem of deciding solvability over F. That is, a linear system over $G = (F, +)$ can simply be seen as a linear system over F where all scalar coefficients appearing in the linear equations are either 1_F or -1_F. This immediately shows that the class of solvable linear systems over a fixed finite field is not definable in finite-variable infinitary logic with counting and hence also not in IFPC.

In this section we consider the problem of defining solvable systems of linear equations over a finite field. From the basic characterisation of solvability in terms of matrix rank, it follows easily that the rank logic FOR_p can express solvability of linear systems over GF_p, for each prime p. With a little more work, we can also show that for each prime p and $d \in \mathbb{N}$, IFPR_p can decide solvability of linear systems over GF_{p^d}. Before making these statements more specific, we define our chosen representation of linear systems as finite relational structures.

A system of linear equations over a finite field can be represented as a three-sorted finite structure in vocabulary $\tau_{\text{sys}} := \{A, B\} \cup \tau_{\text{field}}$, where A is a ternary relation symbol of type $(1, 2, 3)$ and B is a binary relation symbol of type $(1, 3)$. Here $\tau_{\text{field}} = \{+_{\text{f}}, \times_{\text{f}}, 0_{\text{f}}, 1_{\text{f}}\}$ denotes the signature of fields defined in §3.1.1, with all relation and constant symbols restricted to the third sort. A τ_{sys}-structure $\mathbf{S}$ with sorts I, J and F, in that order, describes the system of linear equations

$$A^{\mathbf{S}}\mathbf{x} = B^{\mathbf{S}} \text{ over } \mathbf{F},$$

where the field $\mathbf{F} = (F, +_{\text{f}}^{\mathbf{F}}, \times_{\text{f}}^{\mathbf{F}}, 0_{\text{f}}^{\mathbf{F}}, 1_{\text{f}}^{\mathbf{F}})$ is obtained from the reduct $\mathbf{S}|\tau_{\text{field}}$ by retaining only the elements of the third sort F, $A^{\mathbf{S}}$ is an $I \times J$ matrix over $\mathbf{F}$, $B^{\mathbf{S}}$ is a column vector indexed by I over $\mathbf{F}$ and $\mathbf{x} = (x_j)_{j \in J}$ is a row vector of distinct variables, indexed by J. The system $\mathbf{S}$ is solvable over $\mathbf{F}$ if there is a column vector $\mathbf{c}$ indexed by J over $\mathbf{F}$ such that

$$A^{\mathbf{S}}\mathbf{c} = B^{\mathbf{S}}.$$

Here, the multiplication of matrices is with respect to the field operations of $\mathbf{F}$. For $m = p^d$, where p is prime and $d \in \mathbb{N}_0$, write $\mathsf{Solvable}_m$ for the class of solvable τ_{sys}-structures over a field of cardinality m. Moreover, for prime p write $\mathsf{Solvable}_p^{\text{pow}} := \bigcup_{d>1} \mathsf{Solvable}_{p^d}$ for the class of solvable τ_{sys}-structures over a non-prime field of characteristic p. Our main result in this section is the following.

Theorem 4.12. *For each prime p the following hold:*

(i) $\mathsf{Solvable}_p$ *is definable in* FOR_p*; and*

(ii) $\mathsf{Solvable}_p^{pow}$ *is definable in* IFPR_p.

It can be seen that there is a number term of IFPC that defines the field characteristic over any structure of vocabulary τ_{field}. Writing $\mathsf{Solvable} := \bigcup_{p \text{ prime}} \big(\mathsf{Solvable}_p \cup \mathsf{Solvable}_p^{\text{pow}}\big)$, we therefore get the following corollary.

Corollary 4.13. $\mathsf{Solvable}$ *is definable in* IFPR_{var}. □

Keeping in mind the result of Atserias et al. [4] and our previous discussion, Theorem 4.12 immediately implies the separation of IFPC and IFPR_p for each prime p. Furthermore, since matrix rank can be computed in polynomial time, it follows that IFPR_{var} has polynomial-time data complexity, which is to say that all properties of finite structures definable in IFPR_{var} are decidable in polynomial time. This shows that both IFPR and IFPR_{var} are candidate logics for PTIME, as stated in the following.

Corollary 4.14. *For each prime p,* $\text{IFPC} \lneqq \text{IFPR}_p \leqq \text{IFPR} \leqq \text{IFPR}_{var} \leqq \text{PTIME}$. □

The proof of Theorem 4.12 is given in the next two subsections, where we separately consider linear equations over prime fields and linear equations over prime-power fields.

4.2.1 Linear equations over prime fields

In this section we consider systems of linear equations over a finite field GF_p, where p is prime. Our aim is to prove part (i) of Theorem 4.12; that is, to show that for each prime p, the query $\mathsf{Solvable}_p$ is definable in FOR_p.

The proof we give is based on the following elementary result from linear algebra. Consider a system of linear equations $A\mathbf{x} = \mathbf{b}$ over a field F. Such a system is solvable if and only if $\mathbf{b}$ is contained in the span (over F) of the column vectors of A. In other words, the system is solvable if and only if adding $\mathbf{b}$ as a new column to A does not increase the rank of the matrix. This relates the question of solvability to the calculation of matrix rank. However, the rank operators we introduced in §4.1 apply only to matrices specified by a number term or a formula of a certain kind. In other words, these operators cannot be used directly on matrices described by a structure of vocabulary τ_{sys}. Therefore, our main effort in this section is to show that there are number terms of FOR_p that translate any linear system given by a τ_{sys}-structure to an equivalent linear system on which we can apply rank operators. Once this is done, it is straightforward to check if the original system of equations is solvable by comparing the rank of two matrices, as described above.

More specifically, our proof of Theorem 4.12 (i) consists of three main steps. Firstly, we consider systems of linear equations given by a pair of number terms $\alpha(\vec{x}, \vec{y})$ and $\beta(\vec{x})$ which are interpreted over a finite structure $\mathbf{A}$. We show that there is a sentence of FOR_p, depending only on α and β, that defines exactly the class of finite structures $\mathbf{A}$ where the system

$$\text{mat}_{\vec{x},\vec{y}}(\alpha, \mathbf{A})_p \cdot \mathbf{x} = \text{mat}_{\vec{x}}(\beta, \mathbf{A})_p$$

is solvable over GF_p. This follows more or less directly from the definition of the rank operator $\mathbf{rk}_p$. Here, we write $\text{mat}_{\vec{x}}(\beta, \mathbf{A})_p$ to denote the GF_p-column vector defined by β over $\mathbf{A}$ in exactly the same way we described matrices in §4.1. Next, we use this result to show that basic problems of graph reachability (symmetric and deterministic transitive closure) can be described in FOR_p, by a reduction to the problem of deciding solvability of linear systems over GF_p, defined by number terms as above. As a corollary, we establish that $\text{FO+STC} \leqq \text{FOR}_p$ and $\text{FO+DTC} \leqq \text{FOR}_p$. Finally, we show that there is a pair of number terms of $\text{FOR}_p[\tau_{\text{sys}}]$ which over any linear system $\mathbf{S}$, with prime field $\mathbf{F}$, describe a system of linear equations equivalent to $\mathbf{S}$. Here we crucially rely on Lemma 3.1, which states that there is an FOC+DTC-definable isomorphism $\mathbf{F} \cong \mathbb{Z}_p$ which associates each element of $\mathbf{F}$ with an integer in the range $[0, p-1]$. Since $\text{FOC} \leqq \text{FOR}_p$ and $\text{FO+DTC} \leqq \text{FOR}_p$, this isomorphism is also definable in FOR_p. Putting all these results together, we can finally show that the class of finite solvable τ_{sys}-systems can be defined in FOR_p, which gives us the proof of Theorem 4.12 (i).

Lemma 4.15. *Consider a prime p let $\alpha(\vec{x}, \vec{y})$ and $\beta(\vec{x})$ be number terms in $\text{FOR}_p[\tau]$, where $\vec{x}$ and $\vec{y}$ are tuples of element variables. Then there is an $\text{FOR}_p[\tau]$-sentence φ for which it holds that for any τ-structure $\mathbf{A}$: $\mathbf{A} \models \varphi$ if and only if the linear system described by α and β over $\mathbf{A}$ is solvable over GF_p.*

Proof. Consider a τ-structure $\mathbf{A}$ along with number terms $\alpha(\vec{x}, \vec{y})$ and $\beta(\vec{x})$ of vocabulary τ. Together, α, β and $\mathbf{A}$ describe the system of linear equations $A_\alpha \mathbf{x} = \mathbf{b}_\beta$ over GF_p, where $A_\alpha := \text{mat}_{\vec{x},\vec{y}}(\alpha, \mathbf{A})_p$ is a matrix indexed by $U(\mathbf{A})^{\|\vec{x}\|} \times U(\mathbf{A})^{\|\vec{y}\|}$ and $\mathbf{b}_\beta := \text{mat}_{\vec{x}}(\beta, \mathbf{A})_p$ is a column vector indexed by $U(\mathbf{A})^{\|\vec{x}\|}$. This system is solvable if and only if adding $\mathbf{b}_\beta$ as a new column to A_α does not increase the rank of the matrix, as discussed before. We show that this condition can be expressed by a sentence of FOR_p. To do that, first consider the number term defined by

$$\gamma(\vec{x}, \vec{y}\vec{y}'; z) \equiv (1 - \eta_{\text{equal}}(z, y')) \cdot \beta + \eta_{\text{equal}}(z, y') \cdot \alpha,$$

where the number term $\eta_{\text{equal}}(z, y') \equiv \#_w((w = z) \wedge (z = y'))$ takes value $1^{\mathbf{A}} \in \mathbb{N}_0$ if $z = y'$ and takes value $0^{\mathbf{A}} \in \mathbb{N}_0$ otherwise. Here z is a parameter that allows us to identify one particular element of $\mathbf{A}$ and we assume that y', w and z are variables distinct from those in $\vec{x}$ and $\vec{y}$. Over $\mathbf{A}$, $\gamma(\vec{x}, \vec{y}y')$ describes a matrix indexed by $U(\mathbf{A})^{\|\vec{x}\|} \times U(\mathbf{A})^{\|\vec{y}\|+1}$, which consists of one copy of A_α and $\|A\|^{\|\vec{y}\|} \cdot (\|A\| - 1)$ copies of the column vector $\mathbf{b}_\beta$, stacked side-by-side. Based on this, and the preceding discussion, it can now be seen that the following sentence of FOR_p defines solvability of the system $A_\alpha \mathbf{x} = \mathbf{b}_\beta$ over $\mathbf{A}$:

$$\forall z \left(\mathbf{rk}_p(\vec{x}, \vec{y}y').\gamma = \mathbf{rk}_p(\vec{x}, \vec{y}).\alpha \right).$$

Note that the matrix defined by γ on the left-hand side of the equality will contain multiple copies of the column vector $\mathbf{b}_\beta$, which of course does not alter the solvability of the system. □

We now consider the definability in FOR_p of certain graph-reachability problems. Our aim is to show that $\mathrm{FO{+}DTC} \lneq \mathrm{FOR}_p$ for any prime p, as discussed earlier. The first problem we consider is *symmetric* (s, t)*-reachability*, which, given a graph G with distinguished vertices s and t, asks whether there is a path from s to t in G. We show that this problem is definable in FOR_p.

Let $G = (V, E)$ be a graph and let s and t be two vertices in V. For a prime p, let $\mathcal{S}_{G,s,t}$ be the system of linear equations over GF_p with variables x_v for all $v \in V$ and equations:

- $x_u - x_v = 0$, for every edge $e = (u, v) \in E$;
- $x_s = 1$ and $x_t = 0$.

We observe that the edge equations of $\mathcal{S}_{G,s,t}$ force variables x_u and x_v to take the same value if u and v are in the same connected component of G. This gives us the following lemma.

Lemma 4.16. *The linear system $\mathcal{S}_{G,s,t}$ is solvable over GF_p if and only if there is no path between s and t in the graph G.*

Proof. For one direction, suppose $\mathbf{x} \in \mathsf{GF}_p^V$ is a solution to the system $\mathcal{S}_{G,s,t}$. Label each vertex $v \in V$ with $x_v \in \mathsf{GF}_p$. By equations $x_u - x_v = 0$, it follows that all vertices in the same connected component of G must be assigned the same label. Equations $x_s = 1$ and $x_t = 0$ then imply that s and t belong to different connected components of G, and are hence not reachable from one another.

For the other direction, suppose there is no path between s and t in G. Then a solution to $\mathcal{S}_{G,s,t}$ is obtained by setting $x_v = 1$ for all vertices v in the connected component of G containing s, and $x_v = 0$ for all other v. □

The matrix of the system $\mathcal{S}_{G,s,t}$ is easily defined in the graph G by a numeric term $\eta(x_1x_2, y)$ taking the value 1 at (ss, s) and (tt, t), and for edges $(u, v) \in E$, taking the value 1 at (uv, u) and -1 at (uv, v). Note that every edge equation is stated twice in equivalent ways, which of course does not affect the solvability of the system. This shows that there is a first-order reduction from symmetric (s, t)-reachability to the problem of deciding solvability of linear systems over GF_p. By applying Lemma 4.15, we get the following result.

Lemma 4.17 (Symmetric transitive closure)**.** *Symmetric (s, t)-reachability is definable in FOR_p for all primes p.* □

The above method for defining reachability fails in general when applied to directed graphs. We can, however, consider an important special case, which is graphs whose vertices have out-degree at most one. Specifically, let $\vec{G} = (V, \vec{E})$ be a directed graph. Define the *deterministic* part $E_d \subseteq \vec{E}$ as all those edges (u, v) in $\vec{E}$ for which u has out-degree one. That is,

$$E_d := \{(u, v) \in \vec{E} \mid \forall w\,(\vec{E}(u, w) \to (v = w))\} \subseteq \vec{E}.$$

Given $\vec{G}$ and vertices $s, t \in V$, the *deterministic* (s, t)*-reachability* problem asks whether there is a path from s to t in the deterministic graph $G_d := (V, E_d)$.

Lemma 4.18 (Deterministic transitive closure)**.** *Deterministic* (s, t)*-reachability is definable in* FOR_p *for all primes* p*.*

Proof. Let G_d^* be the *undirected* graph obtained from G_d by removing any outgoing edge from t and then taking the symmetric closure of E_d. Clearly, if there is a directed path from s to t in G_d, the same path connects s and t in G_d^*. Conversely, if P is an undirected s, t-path in G_d^*, following P backwards from t to s we always use edges from G_d in the reverse direction since all vertices have out-degree at most one and t has no outgoing edge. Thus, there is a path from s to t in G_d if and only if there is an undirected (s, t)-path in G_d^*. Observe that the graph G_d^* is first-order definable over $\vec{G}$; that is, there is a first-order unary interpretation that associates G_d^* to $\vec{G}$. Hence there is a first-order reduction from deterministic (s, t)-reachability to symmetric (s, t)-reachability, and the lemma follows. □

Corollary 4.19. *On the class of all finite structures,* $\mathrm{FO{+}STC} \lneqq \mathrm{FOR}_p$ *and* $\mathrm{FO{+}DTC} \lneqq \mathrm{FOR}_p$ *for all primes* p*.*

Proof. Let p be prime. We have shown that both symmetric and deterministic (s, t)- reachability can be expressed in FOR_p. By treating s and t as parameters to the respective reachability queries, it follows that both the symmetric transitive closure and the deterministic transitive closure of any formula $\varphi(\bar{x}, \bar{y})$ can be expressed in FOR_p. For a separating example, note that FO+STC and FO+DTC do not have the ability to count, and cannot express that a set has an even number of elements (see e.g. Ebbinghaus [23]). □

For the proof of Theorem 4.12 (i), it remains to show that linear systems represented by τ_{sys}-structures can be described by number terms, as in the statement of Lemma 4.15. First note that we can express both counting and deterministic transitive closure in FOR_p for any prime p, as shown above. Hence, $\mathrm{FOC{+}DTC} \leqq \mathrm{FOR}_p$. It then follows from Lemma 3.1 that there is a formula of FOR_p that maps any linear system over a field of cardinality p to an equivalent linear system where the field is described by number terms. This allows us to prove the following result.

Lemma 4.20. *For prime* p*, there are number terms* $\alpha(x, y)$ *and* $\beta(x)$ *of* $\mathrm{FOR}_p[\tau_{sys}^\star]$ *for which it holds that for any* τ_{sys}*-structure* $\mathbf{S}$ *over a field of cardinality* p*,* $\mathbf{S} \in \mathsf{Solvable}_p$ *if and only if the linear system* $\mathrm{mat}_{x,y}(\alpha, \mathbf{S})_p \cdot \mathbf{x} = \mathrm{mat}_x(\beta, \mathbf{S})_p$ *is solvable over* GF_p*.*

Proof. Let $\eta(z)$ be a number term as in Lemma 3.1. Then for any linear system $\mathbf{S}$ of vocabulary τ_{sys}, over a field $\mathbf{F}$ of cardinality p, it follows that $\eta(z)^{\mathbf{F}^\star}$ defines a field isomorphism $\iota : \mathbf{F} \to \mathbb{Z}_p$, where every element of $U(\mathbf{F})$ is mapped to an integer in the range $[0, p-1]$. Now consider element variables x and y that range over the 'row sort' and 'column sort' of $\mathbf{S}$,

respectively. Then we define the required number terms $\alpha(x, y)$ and $\beta(x)$ by mapping each matrix element occurring in the linear system $\mathbf{S}$ according to the isomorphism ι, as follows:

$$\alpha(x, y) \equiv \#_w\big(\exists z\ (A(x, y, z) \wedge (\eta(w) < \eta(z)))\big) \text{ and}$$
$$\beta(x) \equiv \#_w\big(\exists z\ (B(x, z) \wedge (\eta(w) < \eta(z)))\big).$$

Here w is an element variable that ranges over the 'field sort' of $\mathbf{S}$ and the counting operators can be simulated by rank operators $\mathbf{rk}_p$, as shown before. □

Finally, the proof of Theorem 4.12 (i) follows by combining Lemma 4.20 with Lemma 4.15.

4.2.2 Linear equations over prime-power fields

In this section we study the solvability of linear equations over prime-power fields. Our aim is to prove part (ii) of Theorem 4.12; that is, to show that for each prime p, the query $\mathsf{Solvable}_p^{\mathrm{pow}}$ is definable in IFPR_p. When combined with the results of the previous section (concerning definability of $\mathsf{Solvable}_p$ in FOR_p), this concludes the proof of Theorem 4.12.

The idea behind our proof is as follows. Let p be prime and $d > 1$ an integer. The finite field of non-prime cardinality p^d is commonly represented as a quotient ring $\mathsf{GF}_p[X]/(g(X))$ where $g(X)$ is a monic irreducible polynomial of degree d over GF_p. This was discussed further in §2.7.2 and then §3.2. Another way to represent the elements of GF_{p^d} is to consider a certain ring of invertible $d \times d$ matrices over GF_p. This kind of representation has the nice property that the field operations over GF_{p^d} are simply the corresponding operations on matrices (in particular, the inverse of an element in GF_{p^d} is obtained by taking the inverse of the corresponding matrix). This approach was described in a short note by Wardlaw [66] and is mentioned briefly by Lidl and Niederreiter [53, Chapter 2]. After reviewing the construction described by Wardlaw, we show how the resulting matrix representation can be expressed over τ_{field}-structures in IFPR_p. This in turn allows us to translate a linear system over GF_{p^d}, given as a structure in vocabulary τ_{sys}, to a (slightly larger) linear system over GF_p which is solvable if and only if the original system is solvable. The proof then follows by applying Lemma 4.15.

We start by reviewing some of the theory behind our construction. Write $K = \mathsf{GF}_p$ for the prime field with p elements and consider a monic (not necessarily irreducible) polynomial $g(X) = X^m + a_{m-1}X^{m-1} + \cdots + a_1X + a_0$, where the a_i are scalar coefficients from K. The *companion matrix* of $g(X)$ is the $m \times m$ matrix

$$B := \begin{pmatrix} 0 & \cdots & \cdots & 0 & -a_0 \\ 1 & \ddots & & \vdots & \vdots \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & -a_{m-1} \end{pmatrix}$$

over K. That is, B is obtained by taking an $(m-1) \times (m-1)$ identity matrix, adding a zero row of length $m-1$ at the top and then appending the column vector $-(a_0\ a_1 \cdots a_{m-1})^t$ to the right of the resulting matrix. It is well known in linear algebra that the minimal polynomial and characteristic polynomial of B are both equal to $g(X)$ (see e.g. Horn and Johnson [42, Theorem 3.3.14]). Thus, $g(B) = 0$.

Now let $f(X)$ be a monic irreducible polynomial of degree d over K, let B be its companion matrix and write $F = \mathsf{GF}_{p^d}$. Because $f(B) = 0$ and the minimal polynomial of B is f, it follows that there is no polynomial g of degree less than d for which $g(B) = 0$. Therefore the ring $K[X]/(f(X))$ is isomorphic to the ring of matrices $K[B]/(f(B))$, via the map that sends a polynomial $g \in K[X]/(f(X))$ to the matrix $g(B) \in K[B]/(f(B))$. Thus

$$F \cong K[X]/(f(X)) \cong K[B]/(f(B)) = K[B]/(0) \cong K[B],$$

which shows that the matrix ring $K[B]$ is a representation of the field F. Here $K[B]$ is the ring that consists of all sums of powers of B over K, with multiplication and addition obtained by directly multiplying and adding the matrix elements. In particular, note that each matrix in $K[B]$ is invertible.

Since $F \cong K[B]$, it follows that $K[B]$ has exactly p^d elements, one of which is the zero matrix 0 and another which is the identity matrix I. In general, it is not the case that $K[B] = \{0, I, B, B^2, \ldots, B^{p^{d-2}}\}$. However, since the multiplicative group $F^\times$ is always cyclic (see §2.7.2), we know there is a matrix $M \in K[B]$ such that $K[B]$ is generated by M — that is, a matrix M for which it holds that

$$K[B] = \langle M \rangle = \{0, I, M, M^2, \ldots, M^{p^{d-2}}\} = K[M].$$

One way to construct a monic irreducible polynomial of degree d over K is to consider the minimal polynomial of a primitive element $\alpha \in F$. This was discussed in more detail in §3.2. Consider a polynomial $f(X)$ of this form and let B be its companion matrix. Then an isomorphism ι between the fields $K[B]$ and F is given explicitly by $\iota : g(B) \mapsto g(\alpha)$, for all $g(B) \in K[B]$, where g is a polynomial of degree less than d. That is, if $g(B) = c_{d-1}B^{d-1} + \cdots + c_1 B + c_0$, where each $c_i \in K$, then

$$g(\alpha) = c_{d-1}\alpha^{d-1} + \cdots + c_1\alpha + c_0 \in F.$$

It necessarily follows that B must be a cyclic generator of $K[B]$. For otherwise, there must be distinct integers k, m such that $B^m = B^k$. But then $\alpha^m = \iota(B^m) = \iota(B^k) = \alpha^k$, which implies $m = k$ as α is a cyclic generator of $F^\times$.

Our aim is now to show that we can define in IFPC a matrix representation of the elements of a finite field of the form described above. Here we critically rely on Lemma 3.7, which says that the minimal polynomial of a primitive element can be defined in IFPC. In order to formally state our result, we need to introduce some new notation. Consider a τ_{field}-structure $\mathbf{F}$, an integer $m \leq \|F\|$ and a number term $\eta(x, y)$ in vocabulary τ_{field}, where x and y are element variables. If $\leqslant$ is a linear ordering of $U(\mathbf{F})$, then we write $\text{submat}_{x,y}(\eta, \mathbf{F}, m, \leqslant)$ to denote the integer matrix of dimension $m \times m$ obtained from $\text{mat}_{x,y}(\eta, \mathbf{F})$ by retaining only rows and columns indexed by those elements $a \in U(\mathbf{F})$ where the position of a in the ordering $\leqslant$ is at most m. Also, for a primitive element $\alpha \in U(\mathbf{F})$ we write $\leqslant_\alpha$ for the ordering of $U(\mathbf{F})$ induced by α. This ordering is definable in FOC+DTC by Corollary 3.6.

Lemma 4.21 (Definable field isomorphism)**.** *There is an* IFPC *number term $\eta(x, y, z)$ in vocabulary τ_{field} for which the following holds. Consider a τ_{field}-structure $\mathbf{F}$ and a primitive element $\alpha \in U(\mathbf{F})$. Let $\|\mathbf{F}\| = p^d$, where p is prime, and write B for the $d \times d$ companion matrix of the minimal polynomial of α over $\mathbb{Z}_p$. Here we assume that the elements of B are integers from the set $\{0, \ldots, p-1\}$. Then for any field element $g \in U(\mathbf{F}) \setminus \{0_f^{\mathbf{F}}\}$,*

the matrix $M_g := \mathrm{submat}_{x,y}(\eta(x, y, g/z), \mathbf{F}, d, \leqslant_\alpha)$ *is equal to* B^i*, where* i *is the* α*-order of* g *in* $\mathbf{F}$.

Furthermore, let M_g *be the all-zero matrix when* $g = 0_f^{\mathbf{F}}$*. It then follows that the map* $g \mapsto M_g$ *defined by* $\eta(x, y, z)$ *over* $\mathbf{F}$ *is an isomorphism* $\mathbf{F} \to \mathbb{Z}_p[B]$*, where we view each* M_g *as a matrix over* $\mathbb{Z}_p$.

Proof. Consider a τ_{field}-structure $\mathbf{F}$ of cardinality p^d and let α be a primitive element of $\mathbf{F}$. By Lemma 3.7, we can define the minimal polynomial of α over $\mathbb{Z}_p$ by a number term of IFPC. From the term defining the minimal polynomial, it is straightforward to construct a number term defining the corresponding companion matrix B. By Corollary 3.5 we can define in FOC+DTC the α-order of each element in $\mathbf{F}$ and the matrix powers B^i can be defined in IFPC by a result of Blass, Gurevich and Shelah [9] (see also discussion in Chapter 3). This finally allows us to consider the map $f \mapsto B^i$ where i is the α-order of an element $f \in \mathbf{F}$. It should be clear by the above that this map can be defined by a formula of IFPC, as required. □

We conclude this section with the following lemma. Theorem 4.12 (ii) then follows by combining the result with Lemma 4.15.

Lemma 4.22. *For prime* p*, there are* IFPR_p *number terms* $\alpha(x, z, y, w)$ *and* $\beta(x, z)$ *in vocabulary* τ_{sys} *for which it holds that for any* τ_{sys}*-structure* $\mathbf{S}$ *with underlying field* $\mathbf{F}$ *of characteristic* p*,* $\mathbf{S}$ *is solvable over* $\mathbf{F}$ *if and only if the linear system* $\mathrm{mat}_{xz,yw}(\alpha, \mathbf{S})_p \cdot \mathbf{x} = \mathrm{mat}_{xz}(\beta, \mathbf{S})_p$ *is solvable over* GF_p.

The proof of this lemma is based on the following idea. Consider a finite field $F = \mathsf{GF}_{p^d}$ and its prime field $K = \mathsf{GF}_p$. Let $\mathcal{S} : A\mathbf{x} = \mathbf{b}$ be a system of linear equations over F, where A is an $n \times m$ matrix and $\mathbf{b}$ is a column vector of length n. Let B be a $d \times d$ matrix over K such that there is a field isomorphism $\iota : F \to K[B]$. We can translate $\mathcal{S}$ to a system of linear equations over K as follows. First, expand each of the variables in $\mathbf{x}$ to a $d \times d$ block of distinct variables. Each of these variables will be assigned a value from K. Let $\mathbf{x}_p$ denote the resulting block of $d \times d$ variable matrices. Second, expand each element a of the matrix A to $\iota(a)$, a $d \times d$ matrix of elements from K. This gives us an $nd \times md$ matrix A_p over K. Likewise, we can expand each element of $\mathbf{b}$ in this way, which gives us an $n \times d$ matrix $\mathbf{b}_p$ over K. Now it can be seen, as ι is an isomorphism of fields, that the system of linear equations $A_p\mathbf{x}_p = \mathbf{b}_p$ has a solution over K if and only if $A\mathbf{x} = \mathbf{b}$ has a solution over F. Of course, the matrix equation $A_p\mathbf{x}_p = \mathbf{b}_p$ is technically not a linear system in the traditional sense, as $\mathbf{b}_p$ is a proper matrix and not a vector. However, it can be turned into a linear system as follows:

For each linear equation $a_1x_1 + \ldots a_mx_m = c$ of $\mathcal{S}$ over F, write

$$\iota(a_1)X_1 + \ldots \iota(a_m)X_m = \iota(c)$$

for the corresponding matrix equation over K, where each X_i is a $d \times d$ matrix of distinct variables. The expression on the left-hand side can be written as a $d \times d$ matrix $M = (m_{ij})$ where each entry m_{ij} is a linear polynomial in variables $\vec{x}$ coming from the elements of $X_1, \ldots, X_m$. Writing $\iota(c) = (d_{ij})$, it follows that $\iota(a_1)X_1 + \ldots \iota(a_m)X_m = \iota(c)$ can be seen as a system of d^2 linear equations, with equation (i, j) given by $m_{ij}(\vec{x}) = d_{ij}$.

Proof of Lemma 4.22. Consider a τ_{sys}-structure $\mathbf{S}$ with underlying field $\mathbf{F}$ of characteristic p. By Corollary 3.4, the primitive elements of $\mathbf{F}$ can be defined by a formula of FOC+DTC $\leqslant$ FOR_p. Let $\gamma \in U(\mathbf{F})$ be one such primitive element. By Lemma 4.21, we can define in IFPC $\leqslant$ IFPR_p a field isomorphism $\iota : \mathbf{F} \to \mathbb{Z}_p[B]$, where B is the companion matrix of the minimal polynomial of γ. Now we can follow the steps outlined above, and reduce the linear system $\mathbf{S}$ over $\mathbf{F}$ to a system of linear equations over the prime field $\mathbb{Z}_p$. It can be seen that this system can be described by a pair of number terms $\alpha(x, z, y, w)$ and $\beta(x, z)$. Here the variables z and w, ranging over the field (third) sort of $\mathbf{S}$, are needed to describe the expansion of the field of $\mathbf{F}$ to matrix elements from the ring $\mathbb{Z}_p[B]$, using the ordering $\leqslant_\gamma$ induced by γ. Explicitly describing this construction is rather tedious, but fairly straightforward.

Now it should be clear from our earlier discussion that the system described by α and β has a solution over $\mathbb{Z}_p$ if and only if $\mathbf{S}$ has a solution over $\mathbf{F}$. The statement of the lemma now follows by quantifying over all primitive elements γ. □

4.3 Arity hieararchy of rank logics

For $n \in \mathbb{N}$, we write $\mathcal{L}^\omega(\mathcal{Q}_n)$ to denote the logic obtained by augmenting finite-variable infinitary logic with all Lindström quantifiers of arity at most n. It was proved by Hella [37] that for any $n \in \mathbb{N}$, the logic $\mathcal{L}^\omega(\mathcal{Q}_n)$ is not expressive enough to define all PTIME queries on the class of finite structures. More specifically, Hella shows that for each $n \geq 1$, there is a vocabulary τ_{n+1} and a class of finite τ_{n+1}-structures which is decidable in polynomial time but not definable by any sentence of $\mathcal{L}^\omega(\mathcal{Q}_n)$. Since IFPC $\leqslant \mathcal{L}^\omega(\mathcal{Q}_1)$, this result extends the result of Cai, Fürer and Immerman [12] discussed in §1.1.

Our aim in this section is to show that the arities of rank operators yield a strict hierarchy. For that purpose, we consider for each prime p and integer $n \geq 2$ the rank logics $\text{FOR}_{p;n}$ and $\text{IFPR}_{p;n}$ defined in §4.1, where

$$\text{FOR}_{p;n} \leqq \text{IFPR}_{p;n} \leqslant \mathcal{R}^\omega_{p;n} \leqq \mathcal{L}^\omega(\mathcal{Q}_n).$$

Our main result is the following.

Theorem 4.23 (Strictness of the rank-arity hierarchy)**.** *For any integer $n \geq 2$ and prime p there is a vocabulary τ_{n+1} and a class of finite τ_{n+1}-structures which is definable by a sentence of $\text{FOR}_{p;n+1}$ but not definable by any sentence of $\mathcal{L}^\omega(\mathcal{Q}_n)$. Thus, $\text{IFPR}_{p;n} \leqslant \text{IFPR}_{p;n+1}$ and $\text{FOR}_{p;n} \leqslant \text{FOR}_{p;n+1}$ for any $n \geq 2$ and prime p.*

We prove this theorem in two parts. In §4.3.1 we consider the original queries defined by Hella to separate $\mathcal{L}^\omega(\mathcal{Q}_n)$ from PTIME and show that for each $n \geq 2$, the corresponding query on τ_{n+1}-structures can be expressed using a linear system over GF_2 of arity $n+1$. This shows the strictness of the $\mathbf{rk}_2$-arity hierarchy[1]. In §4.3.2 we briefly describe how Hella's construction can be extended to work for all primes. As a result, we show the strictness of the arity hierarchy of $\mathbf{rk}_p$ for every prime p.

[1] Keep in mind that the minimum arity of rank operators is two, so we only consider the logics $\text{FOR}_{p;n}$ and $\text{IFPR}_{p;n}$ for $n \geq 2$.

4.3.1 Hella's construction for characteristic two

Throughout, we assume that all graphs are finite, undirected and connected. The following construction is due to Hella [37].

Definition 4.24 (Building blocks)**.** For $n \geq 2$, let $C_n := \{c_1, \dots, c_n, d_1, \dots, d_n\}$ denote a set of size $2n$. We equip C_n with the preorder $\prec_n$ defined by

$$x \prec_n y :\Leftrightarrow \text{there are some } i, j \in [n] \text{ with } i < j \text{ such that } x \in \{c_i, d_i\} \text{ and } y \in \{c_j, d_j\}.$$

Let $P_n := \{d_1, \dots, d_n\} \subset C_n$ and define n-ary relations R_n^+ and R_n^- by

$$(a_1, \dots, a_n) \in R_n^+ :\Leftrightarrow a_1 \prec_n \cdots \prec_n a_n \text{ and } \|\{i \mid a_i \in P_n\}\| \equiv 0 \pmod 2,$$
$$(a_1, \dots, a_n) \in R_n^- :\Leftrightarrow a_1 \prec_n \cdots \prec_n a_n \text{ and } \|\{i \mid a_i \in P_n\}\| \equiv 1 \pmod 2.$$

■

For $n \geq 2$, let $\tau_n = (R_n, E, <)$ be a vocabulary where R_n is n-ary and both E and $<$ are binary. We note that τ_n depends on the integer n as it contains a relation symbol of arity n.

Definition 4.25 (Hella structures)**.** Let $n \geq 2$ and assume $G = (V, E^G, <^G)$ is a graph which is regular of degree n, and $<^G$ is a strict linear order on V. For every vertex $u \in V$, fix an enumeration $h_u : \{v \mid (u, v) \in E^G\} \to [n]$ of its n neighbours. Then for any $S \subseteq V$, we define the τ_n-structure $\mathbf{D}_n(G, S)$ as follows, where we let $D_G = U(\mathbf{D}_n(G, S))$:

- $D_G := V \times C_n$;
- $R_n^{\mathbf{D}_n(G,S)}$ is the set of all tuples $((u, a_1), \dots, (u, a_n))$ in D_G so that either $u \notin S$ and $(a_1, \dots, a_n) \in R_n^+$, or $u \in S$ and $(a_1, \dots, a_n) \in R_n^-$;
- $E^{\mathbf{D}(G,S)}$ is the set of all pairs $((u, c_i), (v, c_j))$ and $((u, d_i), (v, d_j))$ in $(D_G)^2$ such that $(u, v) \in E^G$, $i = h_u(v)$, and $j = h_v(u)$; and
- $(u, a) <^{\mathbf{D}_n(G,S)} (v, b)$ if and only if $u <^G v$ or $(u = v) \wedge (a \prec_n b)$.

■

Notice that the ordering $<^{\mathbf{D}(G,S)}$ has width two, as for every $(u, a) \in D_G$, there is exactly one $(u, b) \in D_G$ with neither $(u, a) <^{\mathbf{D}(G,S)} (u, b)$ nor $(u, b) <^{\mathbf{D}(G,S)} (u, a)$. We call such (u, a) and (u, b) an *incomparable pair*. Hella [37] proves the following:

Lemma 4.26. *Let $n \geq 2$ and assume $G = (V, E^G, <^G)$ is an ordered and n-regular graph. Then for all $S, T \subseteq V$, the structures $\mathbf{D}_n(G, S)$ and $\mathbf{D}_n(G, T)$ are isomorphic if and only if $\|S\| \equiv \|T\| \pmod 2$.* □

By this lemma, there are exactly two non-isomorphic structures $\mathbf{D}_n(G, S)$ for any n-regular ordered graph G. Write $\mathbf{A}_n(G) := \mathbf{D}_n(G, \varnothing)$ and $\mathbf{B}_n(G) := \mathbf{D}_n(G, \{u\})$ for some $u \in V$. One of the main results of [37] is the following theorem, which shows that there is no fixed sentence of $\mathcal{L}^\omega(\mathcal{Q}_n)$ that can distinguish between all $\mathbf{A}_{n+1}(G)$ and $\mathbf{B}_{n+1}(G)$, with G ordered and $(n+1)$-regular.

Theorem 4.27 (Non-definability of Hella structures)**.** *For any $n \in \mathbb{N}$, there is a family of ordered $(n+1)$-regular graphs G_k with $\|G_k\| = \mathcal{O}(k^2)$, so that for any $\mathcal{L}^\omega(\mathcal{Q}_n)$-sentence φ there is $k_\varphi \in \mathbb{N}$ such that $\mathbf{A}_{n+1}(G_k) \models \varphi \Leftrightarrow \mathbf{B}_{n+1}(G_k) \models \varphi$, for all $k \geq k_\varphi$.* □

In contrast, it can be shown that for any ordered n-regular graph G, with $n \geq 2$, the two structures $\mathbf{A}_n(G)$ and $\mathbf{B}_n(G)$ can be distinguished by a polynomial-time algorithm. Theorem 4.27 therefore implies that $\mathrm{IFP}(\mathcal{Q}_n)$ does not capture PTIME for any $n \in \mathbb{N}$.

We now show that for $n \geq 3$, $\mathbf{A}_n(G)$ and $\mathbf{B}_n(G)$ can be distinguished by a system of linear equations over GF_2. More specifically, we show that given a τ_n-structure of the form $\mathbf{D}_n(G, S)$, with $n \geq 3$, there is a first-order definable linear system of arity n which is solvable if and only if $\|S\|$ is even. Here the arity of a linear systems $A\mathbf{x} = \mathbf{b}$ is simply the number of variables needed to describe the matrix A and the column vector $\mathbf{b}$ over $\mathbf{D}_n(G, S)$. Moreover, we show there is a sentence of FOR_2 using rank operators of arity at most n that can determine whether the system is solvable. Combined with Theorem 4.27 and $\mathrm{FOR}_{2;n} \nleqq \mathcal{L}^\omega(\mathcal{Q}_n)$, this gives us the following theorem.

Theorem 4.28 (Strictness of the $\mathbf{rk}_2$-arity hierarchy)**.** *For any $n \geq 2$ there is an $\mathrm{FOR}_{2;n+1}$-definable query that is not definable in $\mathcal{L}^\omega(\mathcal{Q}_n)$.*

Now consider a generic τ_n-structure $\mathbf{T} = (V \times C_n, R_n^{\mathbf{T}}, E^{\mathbf{T}}, <^{\mathbf{T}})$, $n \geq 3$. Let $\mathcal{S}_{\mathbf{T}}$ be the system of linear equations over GF_2 with variables $x_{(u,a)}$ for every $(u, a) \in V \times C_n$ and the following equations.

- *Incomparable pair equations.* For every incomparable pair (u, a), (u, b) we have the equation:
$$x_{(u,a)} + x_{(u,b)} = 1.$$
- *Edge equations.* For each $((u, a), (v, b)) \in E^{\mathbf{T}}$ we have the equation:
$$x_{(u,a)} + x_{(v,b)} = 0.$$
- *R-equations.* Finally, for every n-tuple $((u, a_1), \ldots, (u, a_n)) \in R_n^{\mathbf{T}}$ we have the equation:
$$x_{(u,a_1)} + \ldots + x_{(u,a_n)} = 0.$$

The following is not hard to establish.

Lemma 4.29. *There are first-order formulae $\alpha(\vec{x}, y)$ and $\beta(\vec{x})$ in vocabulary τ_n, where $\vec{x}$ is an $(n-1)$-tuple of variables, which over any τ_n-structure $\mathbf{T}$ define the linear system $\mathcal{S}_{\mathbf{T}}$.*

Proof. Write $A\mathbf{x} = \mathbf{b}$ for the system of linear equations $\mathcal{S}_{\mathbf{T}}$ over GF_2. In the following, we define formulae $\alpha(\vec{x}, y)$ and $\beta(\vec{x})$ which describe the $(0,1)$-matrix A and the $(0,1)$-vector $\mathbf{b}$ over $\mathbf{T}$, respectively. Here, $\vec{x}$ is an $(n-1)$-tuple of element variables and $n \geq 3$. First, we define the set of R-equations. Note that for every $(n-1)$-tuple

$$\vec{a} = ((u, a_1), \ldots, (u, a_{n-1}))$$

with $a_1 <_n \ldots <_n a_{n-1}$, there is at most one element (u, a_n) such that $((u, a_1), \ldots, (u, a_n)) \in R_n^{\mathbf{T}}$. Given a tuple $\vec{a}$ of this form, let $\alpha(\vec{a}/\vec{x}, y)$ express the linear equation $x_{(u,a_1)} + \ldots +$

$x_{(u,a_n)} = 0$. That is, for all $b \in U(\mathbf{T})$, $\alpha(\vec{x}, y)$ is defined so that $\mathbf{T} \vDash \alpha[\vec{a}, b]$ if and only if $b = (u, a_i)$ for some $1 \leq i \leq n$. Similarly, $\beta(\vec{x})$ is defined so that $\mathbf{T} \vDash \neg\beta[\vec{a}]$. Clearly, this can be expressed in first-order logic.

The edge equations can be defined at row indices $v_1 \dots v_{n-2}, v_{n-1} = v$ for which $v_1 = \dots = v_{n-2} = w$ and where $(w, v) \in E^{\mathbf{T}}$. That is, given an $(n-1)$-tuple $\vec{a} = (w, \dots, w, v)$ of this form, $\alpha(\vec{x}, y)$ is defined so that for all $b \in U(\mathbf{T})$ it holds that $\mathbf{T} \vDash \alpha[\vec{a}, b]$ if and only if $b \in \{v, w\}$. Moreover, for a tuple $\vec{a}$ of this form, $\beta(\vec{x})$ is defined so that $\mathbf{T} \vDash \neg\beta[\vec{a}]$.

Finally, the incomparable pair equations can be defined at row indices $v_1 \dots v_{n-2}, v_{n-1} = v$ for which $v_1 = \dots = v_{n-2} = w$ and where (w, v) is an incomparable pair. More specifically, given a tuple $\vec{a} = (w, \dots, w, v)$ of this form, $\alpha(\vec{x}, y)$ is defined so that for all $b \in U(\mathbf{T})$ it holds that $\mathbf{T} \vDash \alpha[\vec{a}, b]$ if and only if $b \in \{v, w\}$ and $\beta(\vec{x})$ is defined so that $\mathbf{T} \vDash \beta[\vec{a}]$. □

The previous lemma shows that the linear system $\mathcal{S}_{\mathbf{T}}$ is first-order definable over any τ_n-structure $\mathbf{T}$. In the following we shift our attention to τ_n-structures of the form $\mathbf{D}_n(G, S)$.

Lemma 4.30. *Let $n \geq 3$ and consider an n-regular ordered graph $G = (V, E^G, <^G)$. Then for any $S \subseteq V$, the system $\mathcal{S}_{\mathbf{D}_n(G,S)}$ is solvable over GF_2 if and only if $\|S\| \equiv 0 \pmod 2$.*

Proof. We first show that the system is solvable when $S = \varnothing$. In this case a solution can be constructed explicitly by setting

$$x_{(u,a)} = \begin{cases} 0 & \text{if } a = c_i \text{ for some } i, \\ 1 & \text{if } a = d_i \text{ for some } i, \end{cases}$$

for all $(u, a) \in V \times C$. Since $\mathcal{S}_{\mathbf{D}_n(G,S)}$ is definable by a first-order interpretation Θ by Lemma 4.30 and Θ is invariant under isomorphism, it then follows that $\mathcal{S}_{\mathbf{D}_n(G,S)}$ is solvable whenever $\|S\| \equiv 0 \pmod 2$.

Claim 1. *Any solution $\mathbf{x}$ of $\mathcal{S}_{\mathbf{D}_n(G,S)}$ induces an isomorphism $\iota : \mathbf{D}_n(G, S) \to \mathbf{A}_n(G)$ by letting*

$$\iota(u, a_i) = \begin{cases} (u, c_i) & \text{if } x_{(u,a_i)} = 0, \\ (u, d_i) & \text{if } x_{(u,a_i)} = 1. \end{cases}$$

Proof. The map ι is well-defined since $\mathcal{S}_{\mathbf{D}_n(G,S)}$ ensures that in every incomparable pair $(u, c_i), (u, d_i)$, exactly one of the corresponding variables is set to 1. It is immediately clear that ι is an isomorphism with respect to E and $<$. Now if $((u, a_1), \dots, (u, a_n)) \in R_n^{\mathbf{D}_n(G,S)}$, we must have $x_{(u,a_i)} = 1$ for an even number of $i \in [n]$ since otherwise one of the R-equations would be violated. Thus, $(\iota(u, a_1), \dots, \iota(u, a_n)) \in R_n^{\mathbf{A}_n(G)}$ by the definition of $R_n^{\mathbf{A}_n(G)}$ and ι. If $((u, a_1), \dots, (u, a_n)) \notin R_n^{\mathbf{D}(G,S)}$ we must have $x_{(u,a_i)} = 1$ for an odd number of $i \in [n]$, since replacing any (u, a_i) with its incomparable partner gives a tuple in $R_n^{\mathbf{D}_n(G,S)}$ whose sum is forced to be even (by one of the the R-equations). Therefore $(\iota(u, a_1), \dots, \iota(u, a_n)) \notin R_n^{\mathbf{A}_n(G)}$ and ι is an isomorphism. □

By this claim, the system $\mathcal{S}_{\mathbf{D}_n(G,S)}$ is *not* solvable whenever $\|S\| \equiv 1 \pmod 2$, which completes the proof. □

Combining Lemma 4.30 and Lemma 4.29, we see that there is a first-order reduction from the problem of deciding if $\|S\|$ is even, given a τ_n-structure of the form $\mathbf{D}_n(G,S)$, to the problem of deciding the solvability of a system of linear equations over GF_2. By Lemma 4.15, there is a sentence φ of FOR_2 that determines exactly when such a system $A\mathbf{x} = \mathbf{b}$ is solvable, by comparing the rank of the matrix A and the augmented matrix $(A \mid \mathbf{b})$. However, a close look at the proof of Lemma 4.15 reveals that this is obtained by using rank operators whose arity is one greater than the arity of the formula defining the linear system. This would put $\varphi \in \mathrm{FOR}_{2;n+1}$, which does not give us the strictness result that we want. That is, in order to prove Theorem 4.23, we have to determine whether $\mathcal{S}_{\mathbf{D}_n(G,S)}$ has a solution *without* increasing the arity of the linear system, which is $(n-1)+1 = n$. For this, we need the following lemma, which is easy to prove. Here, if A is a matrix, $\mathbf{c}$ a column vector of A, and $\mathbf{b}$ a column vector of the same dimension as $\mathbf{c}$, then we write $A_{\mathbf{c},\mathbf{b}}$ to denote the matrix obtained from A by replacing the column $\mathbf{c}$ with the column $\mathbf{c} + \mathbf{b}$.

Lemma 4.31. *Let A be a matrix that does not have full column rank over some field F. Then the linear system $A\mathbf{x} = \mathbf{b}$ is solvable if and only if* $\operatorname{rank} A_{\mathbf{c},\mathbf{b}} \leq \operatorname{rank} A$ *for all columns $\mathbf{c}$ of A.*

Proof. If $A\mathbf{x} = \mathbf{b}$ is solvable, then $\mathbf{b}$ is in the span of the columns $\mathbf{c}_i$ of A, which means that there are $a_i \in F$ such that $\sum_i a_i \mathbf{c}_i = \mathbf{b}$. Fix any column $\mathbf{c}$. By a column basis of A we mean a set of column vectors of A that is a basis for the vector space spanned by column vectors of A. First, assume there is a column basis B of A that does not contain $\mathbf{c}$. Then B is also a column basis for $A_{\mathbf{c},\mathbf{b}}$ since $\mathbf{c} + \mathbf{b}$ is in the span of B, and hence $\operatorname{rank} A = \operatorname{rank} A_{\mathbf{c},\mathbf{b}}$. Next, assume that all column bases of A contain $\mathbf{c}$ and let B be such a column basis. Let B' be obtained from B by exchanging $\mathbf{c}$ with $\mathbf{c} + \mathbf{b}$ and suppose there is a column $\mathbf{c}'_i$ of $A_{\mathbf{c},\mathbf{b}}$ that is not in the span of B'. Then $(B' \setminus \{\mathbf{c} + \mathbf{b}\}) \cup \{\mathbf{c}'_i\}$ is a linearly independent set of columns from A with the same cardinality as B, hence a column basis of A not containing $\mathbf{c}$. This contradicts our assumption, and therefore B' spans the column vector space of $A_{\mathbf{c},\mathbf{b}}$. Hence, $\operatorname{rank} A_{\mathbf{c},\mathbf{b}} = \|B'\| = \|B\| = \operatorname{rank} A$. Since column bases always exist, these two cases are exhaustive and we conclude that $\operatorname{rank} A_{\mathbf{c},\mathbf{b}} \leq \operatorname{rank} A$.

For the converse direction, suppose that $A\mathbf{x} = \mathbf{b}$ is not solvable, so $\mathbf{b}$ is not in the column span of A. By assumption, A does not have full column rank. Let B be a column basis of A and let $\mathbf{c}$ be a column which is not in B. Then $\mathbf{c} + \mathbf{b}$ is not in the span of B since $\mathbf{c}$ is, but $\mathbf{b}$ is not, and therefore $\operatorname{rank} A_{\mathbf{c},\mathbf{b}} = 1 + \operatorname{rank} A > \operatorname{rank} A$. □

Putting everything together, we can finally prove the main theorem of this section.

Proof of Theorem 4.28. Consider an n-regular graph $G = (V, E^G, <^G)$, with $n \geq 3$, and let $S \subseteq V$. Write $A\mathbf{x} = \mathbf{b}$ for the system of linear equations $\mathcal{S}_{\mathbf{D}_n(G,S)}$ over GF_2. By Lemma 4.29, A and $\mathbf{b}$ can be defined by first-order formulae $\alpha(\vec{x}, y)$ and $\beta(\vec{x})$ over $\mathbf{D}_n(G,S)$, respectively, where $\vec{x}$ is an $(n-1)$-tuple of variables.

Since G is n-regular and $n \geq 3$, it follows that G contains a cycle. Let H be such a cycle and let J be the collection of all $(u,a) \in V \times C_n$ where $u \in H$ and u has a neighbour $v \in H$ for which there is some $b \in C_n$ with $((u,a),(v,b)) \in E^{\mathbf{D}_n(G,S)}$. Then it is readily verified that on every row, the sum of the entries in the columns indexed by J is zero in GF_2. Thus, the matrix A does not have full column rank. Using Lemma 4.31, it is easy to construct a sentence $\Psi \in \mathrm{FOR}_{2;n}$ which determines the solvability of $\mathcal{S}_{\mathbf{D}_n(G,S)}$. The theorem now follows from Lemma 4.30 and Theorem 4.27. □

4.3.2 General construction for any prime characteristic

The results of the previous section illustrate that the construction of Hella is essentially a clever encoding of linear equations over GF_2. In this section we describe briefly how this construction can be extended to work over GF_p, for any prime p. Most of the proofs herein, which can be obtained by adapting the corresponding proofs from Hella [37], are omitted.

Definition 4.32 (Generalised building blocks)**.** For $n \geq 2$ and prime p, let

$$C_n^p := \{c_i^r \mid i \in [n], r \in [0, p-1]\}$$

denote a set, equipped with the preorder $\prec_n^p$ defined by

$$\begin{aligned} x \prec_n^p y :\Leftrightarrow \quad & \text{there are some } i, j \in [n] \text{ with } i < j \text{ such that} \\ & x \in \{c_i^r \mid r \in [0, p-1]\} \text{ and } y \in \{c_j^r \mid r \in [0, p-1]\}. \end{aligned}$$

Let $\rho : C_n^p \to [0, p-1]$ be the function defined by $\rho : x \mapsto s$ if and only if $x = c_i^s$ for some $i \in [n]$ and $s \in [0, p-1]$. Define n-ary relations R_n^r for every $r \in [0, p-1]$ by

$$(a_1, \ldots, a_n) \in R_n^r :\Leftrightarrow a_1 \prec_n^p \cdots \prec_n^p a_n \text{ and } \sum_{i=1}^{n} \rho(a_i) \equiv r \pmod{p}.$$

■

Clearly, for $p = 2$ we obtain Hella's building blocks from above, i.e. $C_n = C_n^2$. For $n \geq 2$ and prime p, let $\tau_{n,p} := (R_n, E_0, \ldots, E_{p-1}, <)$ be a vocabulary where R_n is n-ary and all other relations are binary.

Definition 4.33 (Generalised Hella structures)**.** Let $n \geq 2$ and assume $G = (V, E^G, <^G)$ is a graph which is regular of degree n, and $<^G$ is a strict linear order on V. For every vertex $u \in V$, fix an enumeration $h_u : \{v \mid (u, v) \in E^G\} \to [n]$ of its n neighbours. Then for any prime p and function $\gamma : V \to [0, p-1]$, we define the $\tau_{n,p}$-structure $\mathbf{D}_n^p(G, \gamma)$ as follows, where we let $D_G := U(\mathbf{D}_n^p(G, \gamma))$:

- $D_G := V \times C_n^p$;
- $R_n^{\mathbf{D}_n^p(G,\gamma)}$ is the set of all n-tuples $((u, a_1), \ldots, (u, a_n))$ in $(D_G)^n$ for which it holds that $(a_1, \ldots, a_n) \in R_n^r$, with $r = \gamma(u)$.
- For each $k \in [0, p-1]$, $E_k^{\mathbf{D}_n^p(G,\gamma)}$ is the set of all pairs $((u, c_i^r), (v, c_j^s))$ in $D_G \times D_G$, with $i, j \in [n]$ and $r, s \in [0, p-1]$, such that $(u, v) \in E^G$, $i = h_u(v)$, $j = h_v(u)$ and $r + s \equiv k \pmod{p}$;
- $(u, a) <^{\mathbf{D}_n^p(G,\gamma)} (v, b)$ if and only if $u <^G v$ or $(u = v) \wedge (a \prec_n^p b)$.

■

It can be seen that there is a first-order interpretation Θ of $\tau_{n,2}$ in τ_n such that for any ordered n-regular graph $G = (V, E^G, <^G)$ and $S \subseteq V$, we have $\Theta(\mathbf{D}_n(G, S)) = \mathbf{D}_n^2(G, \chi_S)$, where $\chi_S : S \to \{0, 1\}$ is the characteristic function of S. The following lemma classifies the generalised Hella structures up to isomorphism. The proof, which is similar to the proof of Lemma 4.26 above from [37], is omitted. Here, if $\gamma : V \to [0, p-1]$ is a function, then we write $\gamma(V) := \sum_v \gamma(v)$.

Lemma 4.34. *Let $p, n \geq 2$ with p prime and assume $G = (V, E^G, <^G)$ is an ordered and n-regular graph. Then for all $\gamma, \sigma : V \to [0, p-1]$, the structures $\mathbf{D}^p_n(G, \gamma)$ and $\mathbf{D}^p_n(G, \sigma)$ are isomorphic if and only if $\gamma(V) \equiv \sigma(V) \pmod p$.* □

It can be shown that the analogue of Theorem 4.27 also holds for the generalised Hella structures. That is, it can be shown that for each $n \in \mathbb{N}$, prime p and $r \in [0, p-1]$, there is no sentence of $\mathcal{L}^\omega(\mathcal{Q}_n)$ that defines the class of all structures of the form $\mathbf{D}^p_{n+1}(G, \gamma)$ with $G = (V, E^G, <^G)$ ordered and $(n+1)$-regular and $\gamma(V) \equiv r \pmod p$. This can be proved by viewing the pair (G, γ) as a circuit, where each vertex v is given a charge $\gamma(v)$, and showing that in a certain two-player game on $\mathbf{D}^p_{n+1}(G, \gamma)$, one of the players has a strategy to hide the total amount of charge $\gamma(V)$ from the other player. This strategy can be derived by playing a nested "cops-and-robber game" on the circuit (G, γ). We omit the details here, but note that a similar idea is used in Chapter 7 to construct a winning strategy in a different kind of game.

Finally, given a structure $\mathbf{D}^p_{n+1}(G, \gamma)$, it is not hard to construct for each $r \in [0, p-1]$ a first-order definable linear system over GF_p of arity $n+1$ which is solvable if and only if $\gamma(V) \equiv r$. Putting all this together, we get a proof of Theorem 4.23 for all primes p.

4.4 Relationships between rank logics

We conclude this chapter by summarising the known relationships between the rank logics we have defined as well as some of the other logics we considered in this chapter.

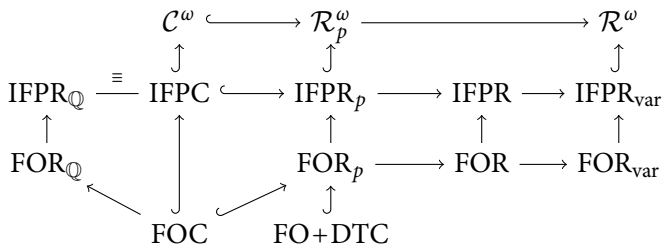


Figure 4.1: Relationships between rank logics. The direction of arrows indicates (semantic) inclusion of the respective logics over finite models; curved arrows (↪) denote proper inclusion. The separation of FOC and IFPC can be deduced e.g. from Etessami [47].

Chapter 5

First-order logic with rank

By extending fixed-point logic with operators for expressing the rank of definable matrix relations, we obtain a logic which is strictly more expressive than IFPC and is potentially a logic for PTIME. In order to understand both the strengths and limitations of this logic, it is important to study how much of its expressive power relates to the inherent capabilities of rank operators and how much of its expressive power relates to the interplay of rank terms and inductive definitions. A natural starting point in that study is to consider rank operators in the context of first-order logic.

In this chapter we study the extension FOR of first-order logic by rank operators and, for every prime p, its fragment FOR_p that only has rank operators over the field GF_p. It turns out that these rank logics are surprisingly expressive. By a simple comparison of rank terms, FOR is able to define the solvability of systems of linear equations over a finite field of prime cardinality, as we discussed in Chapter 4. From the work of Atserias et al. [4], it then follows that FOR $\not\equiv$ IFPC. In this chapter we give further proof of this result, by showing that the two other examples showing that IFPC $\lneqq$ PTIME—the problem of computing the parity of CFI graphs and the problem of deciding isomorphism of multipedes—are both also definable in FOR. This is the subject of §5.1. In §5.2 we establish the descriptive complexity of first-order rank logics over ordered finite structures by proving that for each prime p, FOR_p captures $\mathrm{MOD}_p\mathrm{L}$ and that $\mathrm{FOR}_{\mathbb{Q}}$ captures $\mathrm{L}^{\mathrm{C}_=\mathrm{L}}$, which are natural complexity classes that characterise different levels of logarithmic space complexity. These results further cement the status of first-order rank logics as objects worthy of study in themselves.

5.1 Expressive power of FOR

In [12], Cai, Fürer and Immerman showed that IFPC does not capture PTIME on the class of all finite structures, thereby settling what had been an important open problem in descriptive complexity theory. For the proof, they constructed a query on a class of graphs that can be defined by a polynomial-time computation but not by any sentence of IFPC. Since then, other constructions that expose the limitations of IFPC have been given. Gurevich and Shelah [36] defined a class of finite rigid structures known as *multipedes*, and considered the problem of uniformly defining a linear order over this class. They showed that this problem, while computable in polynomial time, is not expressible by any fixed formula of IFPC. Blass, Gurevich and Shelah [9] later turned this construction into a decision problem and proved

that IFPC is not able to tell whether two given multipedes (each with a designated vertex) are isomorphic or not; a problem which again is decidable in polynomial time.

In this section we show that both these decision problems separating IFPC from PTIME can be expressed in the logic FOR_2, by considering first-order definable systems of linear equations over GF_2. This gives us yet another separation of the fixed-point logics IFPC and IFPR, in addition to results concerning solvability of linear equations over a finite field. Throughout, all graphs are assumed to be undirected, unless otherwise noted.

5.1.1 Cai-Fürer-Immerman graphs

We recall the definition of Cai-Fürer-Immerman (CFI) graphs, which were constructed by Cai et al. [12] to define the query separating IFPC from PTIME. The following presentation of the graphs is adapted from Dawar et al. [21], who show that the CFI query can be expressed in the logic of choiceless polynomial time. Note that, unlike the presentation in [21], we do not require an ordering on the underlying graph G.

Definition 5.1 (CFI graphs)**.** Let $G = (V, E)$ be a connected graph with at least two vertices and let $T \subseteq V$. The *CFI graph* $\mathbf{G}^T = (V^*, E^*, C^*)$ is a two-coloured graph with vertex set V^*, edge relation E^* and a unary relation C^*, denoting the colour, which are defined as follows.

- *Vertices.* Denote the set of edges incident to $v \in V$ by $E(v)$. For each vertex $v \in V$, let $I(v)$ be the collection defined by

$$I(v) := \begin{cases} \{v_Z : Z \subseteq E(v) \text{ and } |Z| \equiv 1 \pmod 2\} & \text{if } v \in T, \\ \{v_Z : Z \subseteq E(v) \text{ and } |Z| \equiv 0 \pmod 2\} & \text{if } v \in V \setminus T. \end{cases}$$

  Define three collections of elements $\widehat{V} := \bigcup_{v \in V} I(v)$, $\widehat{E} := \{e_0, e_1 \mid e \in E\}$ and $\widehat{C} := \{e_c \mid e \in E\}$. Finally, set $V^* := \widehat{V} \cup \widehat{E} \cup \widehat{C}$.

- *Edges.* Define the edge relation $E^* \subseteq V^* \times V^*$ by

$$\begin{aligned} E^* := &\big\{\{v_Z, e_1\} : v \in V, v_Z \in I(v) \text{ and } e \in Z\big\} \cup \\ &\big\{\{v_Z, e_0\} : v \in V, v_Z \in I(v) \text{ and } e \in E(v) \setminus Z\big\} \cup \\ &\big\{\{e_i, e_c\} : e \in E \text{ and } i \in \{0,1\}\big\}. \end{aligned}$$

- *Colour relation.* Finally, define the unary colouring relation $C^* := \widehat{C} \subset V^*$. That is, all the vertices e_c are coloured in the same way and differently from all the other vertices.

■

Note that the rôle of the e_c vertices and the relation C^* in the definition of $\mathbf{G}^T$ is only to allow the nodes e_i to recognize their respective partners e_{1-i}. We refer to the sets of vertices $\widehat{C}$, $\widehat{E}$ and $\widehat{V}$ as the *colour nodes*, *outer nodes* and *inner nodes* of $\mathbf{G}^T$, respectively. The *parity* of a CFI graph $\mathbf{G}^T$ is the parity of $\|\mathbf{T}\|$. We say $\mathbf{G}^T$ is *even* if it has even parity and *odd* if it has odd parity. An example of a CFI graph is given in Figure 5.1.

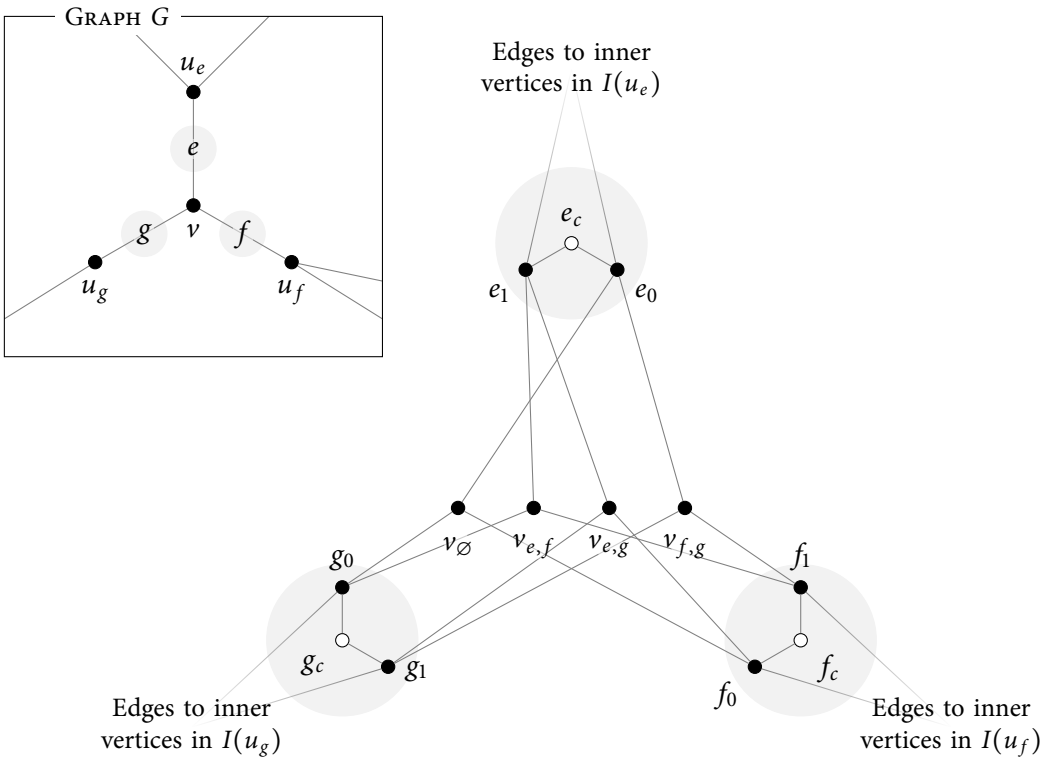


Figure 5.1: A fragment of a CFI graph $\mathbf{G}^T$ constructed for a vertex $v \notin T \subseteq V$ of degree three, where $G = (V, E)$ and $T \subseteq V$. Here $E(v) = \{e, f, g\}$ and $N(v) = \{u_e, u_f, u_g\}$. The vertex v is shown in the inset with the three edges in $E(v)$ labelled. The main figure shows the inner nodes $I(v)$ and the outer nodes constructed from the edges $E(v)$. Because $v \notin T$, all the inner nodes are connected to an *even* number of outer nodes with a "1" subscript. Note that there will be edges connecting each of the outer nodes to the inner nodes constructed from the respective neighbour of v in G. For instance, in $\mathbf{G}^T$ there will be edges connecting e_0 and e_1 to the inner nodes of $I(u_e)$, and so on.

In [12], Cai et al. show the following.

- For a connected graph G, where every vertex has degree at least two, and all $T, S \subseteq V(G)$, the graphs $\mathbf{G}^T$ and $\mathbf{G}^S$ are isomorphic if and only if they have the same parity. Hence there are exactly two non-isomorphic structures $\mathbf{G}^T$ for any graph G.
- While there is a polynomial-time algorithm that can distinguish between the odd and even CFI graphs of any graph G, there is no fixed sentence of IFPC that can do the same.

We now show that the odd and even CFI graphs of any graph G can be distinguished in FOR_2. Let $G = (V, E)$ be a connected graph where every vertex has degree at least two, let $T \subseteq V$ be a collection of vertices and let $\mathbf{G}^T$ be a CFI graph constructed from G and T. Let $\mathcal{S}_{\mathbf{G}^T}$ be a system of linear equations over GF_2 with variables x_{e_i} for all $e_i \in \widehat{E}$ and x_{v_Z} for all $v_Z \in \widehat{V}$, and the following equations.

- *Outer node equations.* For each $e_i \in \widehat{E}$ we have the equation:

$$x_{e_i} + x_{e_{1-i}} = 1.$$

- *Inner node equations.* For each $v_Z \in \widehat{V}$ we have the equation:

$$\sum_{e \in Z} x_{e_1} + \sum_{e \in E(v) \setminus Z} x_{e_0} = \sum_{v_Y \in I(v)} x_{v_Y}.$$

- *Parity equation.* Finally, we have the following equation:

$$\sum_{v_Z \in \widehat{V}} x_{v_Z} = 0.$$

The intuition behind this construction is as follows. Firstly, the outer node equations ensure that for each $e \in E$, the pair of variables x_{e_1} and x_{e_0} must take opposite values in GF_2. Secondly, observe that for any two inner nodes $v_Z, v_X \in \widehat{V}$ derived from the same vertex $v \in V$, the right-hand side $\sum_{v_Y \in I(v)} x_{v_Y}$ of the two corresponding inner node equations are the same. Furthermore, the variables appearing in the sum on the right-hand side appear only in those inner node equations that are derived from the vertex v. As a consequence, any assignment of values to these variables ensures that the right-hand side of all the inner node equations derived from v have the same value, which we call x_v. The idea here is that $x_v = 1$ if $v \in T$ and $x_v = 0$ otherwise. Finally, the parity equation sums up all the variables x_{v_Z} appearing on the right-hand side of the inner node equations. From the above, it is clear this amounts to summing up all the distinct x_v. By the above, this sum should then equal the parity of $\|\mathbf{T}\|$.

The following is not hard to establish.

Lemma 5.2. *$\mathcal{S}_{\mathbf{G}^T}$ is first-order definable over $\mathbf{G}^T$.*

Proof. Write $\tau_{\mathrm{CFI}} = \{R_E, R_C\}$ for the vocabulary of $\mathbf{G}^T$, where $R_E^{\mathbf{G}^T} = E^*$ and $R_C^{\mathbf{G}^T} = C^*$. If $x \in U(\mathbf{G}^T)$ and $Y \subseteq U(\mathbf{G}^T)$ is a non-empty set then we say that *x is connected to Y* if there is at least one vertex $y \in Y$ such that $(x, y) \in E^*$.

To describe the system $\mathcal{S}_{\mathbf{G}^T}$ in first-order logic, we first note that there are first-order formulae $\varphi_c(x)$, $\varphi_o(x)$ and $\varphi_i(x)$ that define the sets $\widehat{C}$, $\widehat{E}$ and $\widehat{V}$, respectively. More specifically,

$$\begin{aligned}
&\varphi_c(x) \equiv R_C(x),\\
&\varphi_o(x) \equiv \exists y\,(\varphi_c(y) \wedge R_E(x, y)), \text{ and}\\
&\varphi_i(x) \equiv \neg(\varphi_c(x) \vee \varphi_o(x)).
\end{aligned}$$

Similarly, there is a first-order formula

$$\theta_o(x, y) \equiv (x \neq y) \wedge \varphi_o(x) \wedge \varphi_o(y) \wedge \exists z\,(\varphi_c(z) \wedge R_E(x, z) \wedge R_E(y, z)),$$

that says that x and y are distinct outer nodes derived from the same edge $e \in E$.

We also need a first-order formula $\theta_i(x, y)$ that relates a pair of distinct inner nodes x and y if and only if x and y are derived from the same vertex $v \in V$. To define $\theta_i(x, y)$, observe that for any pair of (not necessarily distinct) inner nodes v_Y and v_Z derived from the same vertex $v \in V$, there are *at least two* distinct edges $e, f \in E$ such that both v_Y and v_Z are connected to $\{e_0, e_1\}$ and both v_Y and v_Z are connected to $\{f_0, f_1\}$ (see Figure 5.1 for an illustration). Conversely, if u_W is an inner node derived from a vertex $u \in V$ and $u \neq v$, then

there is *at most one* edge $e \in E$ such that both v_Y and u_W are connected to $\{e_0, e_1\}$. Indeed, if such an edge $e \in E$ exists then necessarily $e = \{u, v\}$. This observation can be used to define the formula $\theta_i(x, y)$. To see that, first define a formula

$$\Psi_{\text{conn}}(x, z, w) \equiv \varphi_i(x) \wedge \theta_0(z, w) \wedge (R_E(x, z) \vee R_E(x, w)),$$

which says that x is an inner node, z and w are distinct outer nodes derived from the same edge and x is connected to $\{z, w\}$. In other words, for all $v_Y \in \widehat{V}$ and $e_j, e_{1-j} \in \widehat{E}$, where $j \in \{0, 1\}$, it holds that

$$\mathbf{G}^T \models \Psi_{\text{conn}}[v_Y, e_j, e_{1-j}] \quad \text{if and only if} \quad v \in e.$$

We also define

$$\Psi_{\text{share}}(x, y, z, w) \equiv (x \neq y) \wedge \Psi_{\text{conn}}(x, z, w) \wedge \Psi_{\text{conn}}(y, z, w),$$

which states that x and y are distinct inner nodes both connected to the set $\{z, w\}$ of outer nodes. Finally, we let

$$\theta_i(x, y) \equiv \exists z_1, z_2, z_3, z_4 \bigwedge_{j \neq k} (z_j \neq z_k) \wedge \Psi_{\text{share}}(x, y, z_1, z_2) \wedge \Psi_{\text{share}}(x, y, z_3, z_4),$$

which has the desired properties.

The system $\mathcal{S}_{\mathbf{G}^T}$ can now be defined by formulae $\varphi(x, y)$ and $\beta(x)$ over $\mathbf{G}^T$ in the following way. The equations for the outer nodes e_i are defined at row indices a for which $(\mathbf{G}^T, a) \models \varphi_o$. Similarly, the equations for inner nodes v_Z are defined at row indices a for which $(\mathbf{G}^T, a) \models \varphi_i$, using $\theta_i(x, y)$ and the fact that the set of e_1 with $e \in Z$ is exactly the neighbourhood of v_Z in $\mathbf{G}^T$, and the set of e_0 with $e \in E(v) \setminus Z$ can be defined similarly. Finally, the equation that sums all the x_{v_Z} can be defined at row indices a for which $(\mathbf{G}^T, a) \models \varphi_c$; there will be multiple copies of this equation, which of course does not affect the solvability of the system. The definition of $\beta(x)$ follows similarly. □

Lemma 5.3. *The system $\mathcal{S}_{\mathbf{G}^T}$ is solvable if and only if $\mathbf{G}^T$ is even.*

Proof. We show that the system is solvable when $\|\mathbf{T}\| = 0$ and not solvable when $\|\mathbf{T}\| = 1$. Since $\mathcal{S}_{\mathbf{G}^T}$ is definable by a first-order interpretation Θ by Lemma 5.2 and Θ is invariant under isomorphism, it then follows that $\mathcal{S}_{\mathbf{G}^T}$ is solvable if and only if $\|\mathbf{T}\|$ is even.

First suppose $T = \varnothing$. In this case it is readily verified that any assignment that puts $x_{e_i} = i$ for all $e_i \in \widehat{E}$ and $\sum_{v_Y \in I(v)} x_{v_Y} = 0$ for all $v \in V$ is a solution to $\mathcal{S}_{\mathbf{G}^T}$. Next suppose $T = \{u\}$ where u is an arbitrary vertex in V. Fix one edge $f \in E(u)$ and consider the following equations from $\mathcal{S}_{\mathbf{G}^T}$:

- for every $v \in V \setminus \{u\}$: $\sum_{e \in E(v)} x_{e_0} = \sum_{v_Y \in I(v)} x_{v_Y}$,
- for u: $\left(\sum_{e \in E(u) \setminus \{f\}} x_{e_0}\right) + x_{f_1} = \sum_{u_Y \in I(u)} x_{u_Y}$.

In this subsystem, there is exactly one equation for each $v \in V$. It follows that for all $e \in E \setminus \{f\}$, the variable x_{e_0} occurs exactly twice on the left-hand side of the system, as each edge is connected to two vertices $v \in V$. However, for the edge f, we get both x_{f_0} and x_{f_1}

on the left-hand side. Summing up all the above equations we therefore get on the left-hand side:

$$\begin{aligned}&\Big(\sum_{v\in V\setminus\{u\}}\sum_{e\in E(v)} x_{e_0}\Big)+\Big(\sum_{e\in E(u)\setminus\{f\}} x_{e_0}\Big)+x_{f_1}\\ &=2\cdot\Big(\sum_{e\in E\setminus\{f\}} x_{e_0}\Big)+x_{f_0}+x_{f_1}\\ &=x_{f_0}+x_{f_1}=1,\end{aligned}$$

where the last equality comes from the outer node equation derived from f in $\mathcal{S}_{\mathbf{G}^T}$ and the summation is over GF_2. However, at the same time we get on the right-hand side:

$$\sum_{v\in V}\sum_{v_Y\in I(v)} x_{v_Y}=\sum_{v_Z\in\widehat{V}} x_{v_Z}=0,$$

where the last equality comes from the parity equation in $\mathcal{S}_{\mathbf{G}^T}$ (the last equation). Therefore, the system $\mathcal{S}_{\mathbf{G}^T}$ is inconsistent and has no solution. □

The preceding lemmas now establish that there is a first-order reduction from the problem of distinguishing odd and even CFI graphs to the problem of deciding solvability of linear systems over GF_2, which can be defined in FOR_2 by Theorem 4.12. Since FOR_2 is closed under first-order reductions, we get the following theorem.

Theorem 5.4 (CFI query in FOR_2)**.** *There is a sentence $\varphi_{CFI}\in\mathrm{FOR}_2$ that holds in structures $\mathbf{G}^T$ when $\|\mathbf{T}\|$ is even but not in structures $\mathbf{G}^T$ when $\|\mathbf{T}\|$ is odd.* □

5.1.2 Isomorphism of multipedes

In [36], Gurevich and Shelah showed that there is a first-order axiomatisable class of finite rigid structures, known as *multipedes*, for which there is no formula of $\mathcal{C}^\omega$ that can define a linear ordering on all class members. As mentioned earlier, Blass, Gurevich and Shelah [9] later turned this construction into a decision problem and showed that no fixed sentence of $\mathcal{C}^\omega$, let alone IFPC, can distinguish between a pair of similar but non-isomorphic multipedes.

In this section we show that the problem of deciding isomorphism of multipedes can be expressed in FOR_2, by exhibiting a first-order definable reduction to the problem of deciding solvability of linear equations over GF_2. This is based on the same idea as the reduction we presented in the previous section, although the construction here is more involved. We start by recalling the definition of multipedes from [36].

Definition 5.5 (Multipedes)**.** Let $\tau_{\mathrm{mpede}} := \{F, S, H, P, \leq_M, R, c\}$ be a vocabulary where S and F are unary relation symbols, H and P are ternary relation symbols, $\leq_M$ and R are binary relation symbols and c is a constant symbol. A *multipede* is a finite structure $\mathbf{M}$ in vocabulary τ_{mpede} which satisfies the following conditions.

- The universe $U(\mathbf{M})$ is partitioned into two parts $F^{\mathbf{M}}$ and $S^{\mathbf{M}}$. We refer to the elements of $F^{\mathbf{M}}$ as the *feet* of $\mathbf{M}$ and the elements of $S^{\mathbf{M}}$ as the *segments* of $\mathbf{M}$. For each segment, there are exactly two feet; i.e. $\|F^{\mathbf{M}}\| = 2\|S^{\mathbf{M}}\|$.
- $\leqslant_M^{\mathbf{M}}$ is a total order on the set of segments.

- $R^{\mathbf{M}} \subseteq F \times S$ is the graph of a function $\rho : F \to S$ from feet to segments such that each segment is the image of exactly two feet.

  Abusing notation, we extend ρ to a map $\rho : \wp(F) \mapsto \wp(S)$ by setting $\rho(X) := \{\rho(x) : x \in X\}$, for all $X \in \wp(F)$.

- The constant $c^{\mathbf{M}} \in F$ denotes the unique 'foot with a shoe'. The segment in S which is connected to the shoe $c^{\mathbf{M}}$ should always be the first element in the ordering $\leqslant^{\mathbf{M}}$.

- $H^{\mathbf{M}} \subseteq S \times S \times S$ is a totally irreflexive and symmetric relation that encodes a family $\Sigma \subseteq \wp(S)$ of three-element subsets of S, called *hyperedges*.

- $P^{\mathbf{M}} \subseteq F \times F \times F$ is a totally irreflexive and symmetric relation that encodes a family $\Pi \subseteq \wp(F)$ of three-element subsets of F, called *positive triples*. For each positive triple $p \in \Pi$, the image $\rho(p) \subseteq S$ is a hyperedge.

- If $h \in \Sigma$ is a hyperedge, then there are exactly eight triples of feet mapped by ρ onto h. Out of these eight triples, exactly four are positive. Moreover, if $X, Y \subseteq \Pi$ are positive triples of feet with $\rho(X) = \rho(Y) = h$, then $\|X \mathbin{\triangle} Y\| \equiv 0 \pmod 2$, where $X \mathbin{\triangle} Y$ denotes the symmetric difference of X and Y. In other words, if $\rho(X) = \rho(Y) = h$ then Y can be obtained from X by interchanging the two feet of an even number of segments in h. ■

Remark. Gurevich and Shelah consider more than one type of multipede in their paper [36]. In particular, they refer to the multipedes we consider here as '3-multipedes, to distinguish them from 1-, 2- and 4-multipedes. As we will only consider 3-multipedes in the present discussion, this distinction will be unnecessary. The 3-multipedes of Gurevich and Shelah did not actually have a designated vertex with a shoe; that definition comes from Blass et al. [9], who described the multipede isomorphism problem we consider here. Finally, note that the number '3' in the term 3-multipede does not refer to the fact that each hyperedge in Σ has size three.

The isomorphism problem for multipedes takes as input a pair of multipedes and asks whether the two multipedes are isomorphic. This problem can be turned into a Boolean query which consists of all pairs of multipedes $(\mathbf{M}_1, \mathbf{M}_2)$ with $\mathbf{M}_1 \cong \mathbf{M}_2$; here a pair of multipedes can be represented as a single finite structure as we will explain below. In [9] Blass et al. show the following.

- The isomorphism problem for multipedes can be decided in polynomial time.

- The isomorphism problem for multipedes is not definable in $\mathcal{C}^\omega$ and hence not in IFPC either.

In the following we fix a pair of multipedes $\mathbf{M}_1$ and $\mathbf{M}_2$ presented as a single relational structure $\mathbf{M} := \mathbf{M}_1 \dot\cup \mathbf{M}_2$ in vocabulary $\{F_1, S_1, H_1, P_1, \leq_1, R_1, c_1\} \cup \{F_2, S_2, H_2, P_2, \leq_2, R_2, c_2\}$, where $\leq_1$ and $\leq_2$ denote the two respective linear orders of segments, as discussed earlier. We write Π_i, Σ_i and ρ_i for the set of positive triples, the set of hyperedges and the function mapping feet to segments, respectively, implicit in the definition of each $\mathbf{M}_i$, where $i \in \{1, 2\}$. Write $\rho := \rho_1 \cup \rho_2$ for the function whose domain is the union of $F_1^{\mathbf{M}_1}$ and $F_2^{\mathbf{M}_2}$.

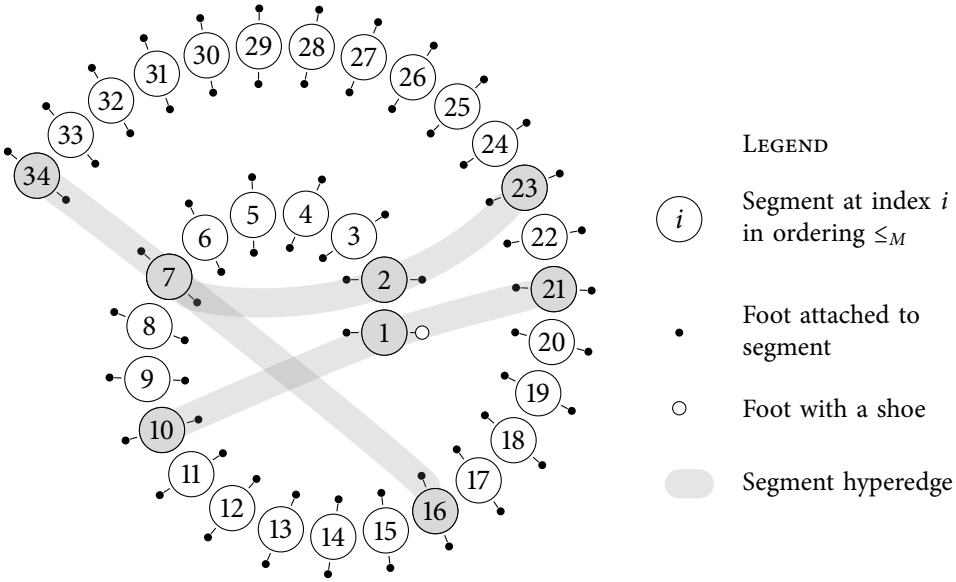


Figure 5.2: A multipede with 34 segments and a shoe. The hypergraph over the segments has three hyperedges: $\{34, 7, 16\}$, $\{10, 1, 21\}$ and $\{7, 2, 23\}$. It follows that the multipede has 12 positive triples, four for every hyperedge (although these are not shown in the figure). Lines connecting feet to segments represent the function ρ; note that each segment is connected to exactly two feet.

For $i \in \{1, 2\}$, write $\theta_i(x, y) \equiv S_i(x) \wedge S_i(y) \wedge (y \leq_i x) \wedge (y \neq x)$ for the formula that says that x and y are segments belonging to the same multipede, and y is strictly less than x in the ordering $\leq_i$. Define the FOC-term

$$\mathsf{ord}(x) \equiv \#_y(\theta_1(x, y) \vee \theta_2(x, y))$$

for which it holds that whenever either $S_1(x)$ or $S_2(x)$ holds, $\mathsf{ord}(x)$ denotes the position of x in the respective segment ordering $\leq_1$ or $\leq_2$. Then the formula

$$\eta(\vec{x}, \vec{y}) \equiv \bigwedge_{i=1}^{3}(S_1(x_i) \wedge S_2(y_i) \wedge (\mathsf{ord}(x_i) = \mathsf{ord}(y_i))) \qquad (*)$$

defines over $\mathbf{M}$ the graph of a function $(S_1^{\mathbf{M}})^3 \to (S_2^{\mathbf{M}})^3$ that sends $\vec{s} \mapsto \vec{t}$ if and only if the triples $\vec{s}$ and $\vec{t}$ occur at the same position in the lexicographic ordering of triples induced by $\leq_1^{\mathbf{M}}$ and $\leq_2^{\mathbf{M}}$, respectively. Here $\vec{x} = (x_1, x_2, x_3)$ and $\vec{y} = (y_1, y_2, y_3)$ are triples of distinct variables. Finally, define a sentence of FOC

$$\mathsf{hyperiso} \equiv (\#_x(S_1(x)) = \#_x(S_2(x))) \wedge \forall \vec{x}\, \vec{y}\, \eta(\vec{x}, \vec{y}) \to (H_1(\vec{x}) \leftrightarrow H_2(\vec{y})).$$

Then $\mathbf{M} \models \mathsf{hyperiso}$ if and only if the two hypergraphs

$$(S_1^{\mathbf{M}_1}, H_1^{\mathbf{M}_1}, \leq_1^{\mathbf{M}_1}) \text{ and } (S_2^{\mathbf{M}_2}, H_2^{\mathbf{M}_2}, \leq_2^{\mathbf{M}_2})$$

are isomorphic. Assume hereafter that hyperiso is satisfied in $\mathbf{M}$ and let

$$\iota : (S_1^{\mathbf{M}_1}, H_1^{\mathbf{M}_1}, \leq_1^{\mathbf{M}_1}) \to (S_2^{\mathbf{M}_2}, H_2^{\mathbf{M}_2}, \leq_2^{\mathbf{M}_2})$$

be the (unique) isomorphism of the two hypergraph structures defined by $\eta(\vec{x}, \vec{y})$, above. Now all that remains to decide if $\mathbf{M}_1 \cong \mathbf{M}_2$ is to check whether the feet of the two multipedes can be matched up to preserve (a) the shoe constant, (b) the function that associates feet with segments and (c) positivity of feet triples. By our discussion above, it is clear that this problem is not expressible in $\mathcal{C}^\omega$. However, we show that it is expressible in FOR_2, as follows.

Consider the set $V := F_1^{\mathbf{M}_1} \dot\cup F_2^{\mathbf{M}_2} \dot\cup \Sigma_1$ and let $\mathcal{S}_{\mathbf{M}}$ be the system of linear equations over GF_2 with variables x_v for all $v \in V$ and the following equations.

- *Segment equations.* For each segment $s \in S_1^{\mathbf{M}_1} \cup S_2^{\mathbf{M}_2}$, with a pair of feet $\rho^{-1}(s) = \{e, f\}$, we add the equation
$$x_e + x_f = 1.$$
That is, x_e and x_f must take opposite values in GF_2.

- *Hyperedge equations.* For each hyperedge $h \in \Sigma_1$ we add exactly 2×8 equations. Firstly, we add one of the following equations for each three-element set of feet $\{e, f, g\} \subseteq F^{\mathbf{M}_1}$ that ρ_1 maps onto h:
$$(x_e + x_f + x_g) + x_h = 1 \text{ if } \{e, f, g\} \in \Pi_1,$$
$$(x_e + x_f + x_g) + x_h = 0 \text{ if } \{e, f, g\} \notin \Pi_1.$$
Secondly, we add one of the following equations for each three-element set of feet $\{e, f, g\} \subseteq F^{\mathbf{M}_2}$ that ρ_2 maps onto $\iota(h) \in \Sigma_2$:
$$(x_e + x_f + x_g) + x_h = 1 \text{ if } \{e, f, g\} \in \Pi_2,$$
$$(x_e + x_f + x_g) + x_h = 0 \text{ if } \{e, f, g\} \notin \Pi_2.$$
Note that we use the same variable x_h for the equations defined by sets of feet from $F^{\mathbf{M}_1}$ as well as the equations defined by sets of feet from $F^{\mathbf{M}_2}$.

- *Shoe equation.* Finally, we add the following equation for the pair of shoes $e = c_1^{\mathbf{M}_1}$, $f = c_2^{\mathbf{M}_2}$:
$$x_e + x_f = 0.$$
That is, x_e and x_f must take the same value in GF_2.

We can index the equations in $\mathcal{S}_{\mathbf{M}}$ by the set

$$E := S_1^{\mathbf{M}_1} \cup S_2^{\mathbf{M}_2} \cup \left(F_1^{\mathbf{M}_1} \times F_1^{\mathbf{M}_1} \times F_1^{\mathbf{M}_1} \times \Sigma_1\right) \cup \left(F_2^{\mathbf{M}_2} \times F_2^{\mathbf{M}_2} \times F_2^{\mathbf{M}_2} \times \Sigma_1\right).$$

Observe that the set of hyperedges Σ_2 does not appear in E.

Lemma 5.6. *$\mathcal{S}_{\mathbf{M}}$ is first-order definable over* $\mathbf{M}$.

The proof of this lemma is straightforward though rather tedious.

Proof. Throughout this proof, we will use lower-case Latin characters $x, y, z, \ldots$ to denote variables that range over feet and we will use lower-case Greek characters $\upsilon, \mu, \gamma, \ldots$ to denote variables that range over segments. Note that this is purely for notational purposes; all variables ranging over $U(\mathbf{M})$ are untyped. We also write $\vec{x}, \vec{y}, \ldots$ and $\vec{\upsilon}, \vec{\mu}, \ldots$ to denote *triples* of variables, where the individual components are indexed as $\vec{x} = (x_1, x_2, x_3)$, for example.

In our definition of the system $\mathcal{S}_{\mathbf{M}}$ we will have to find a unique triple (e, f, g) to represent each hyperedge $\{e, f, g\} \in \Sigma_1$, to ensure the correctness of the linear system (that is, we want only one variable x_h for each hyperedge h). For that purpose, we consider for $i = 1, 2$ the formula

$$\theta_{\text{hyp},i}(\vec{\upsilon}) \equiv \bigwedge_j S_i(\upsilon_j) \wedge H_i(\vec{\upsilon}) \wedge (\upsilon_1 \leq_i \upsilon_2) \wedge (\upsilon_2 \leq_i \upsilon_3),$$

for which it holds that for all hyperedges $\{e, f, g\} \in \Sigma_i$, $(\mathbf{M}, e, f, g) \vDash \theta_{\text{hyp},i}(\vec{\upsilon})$ if and only if e, f, g are listed in increasing order with respect to $\leq_i^{\mathbf{M}}$. To define the system $\mathcal{S}_{\mathbf{M}}$ we now consider each type of equation separately. For tuples of variables $(x_1, \ldots, x_m)$ and $(y_1, \ldots, y_m)$, we will write $(x_1, \ldots, x_m) = (y_1, \ldots, y_m)$ as a shorthand for $\bigwedge_i (x_i = y_i)$.

- The segment equations can be defined by formulae

$$\begin{aligned}\alpha_{\text{seg}}(\upsilon; y) \equiv& (S_1(\upsilon) \wedge F_1(y) \wedge R_1(y, \upsilon)) \vee (S_2(\upsilon) \wedge F_2(y) \wedge R_2(y, \upsilon)) \text{ and}\\ \beta_{\text{seg}}(\upsilon) \equiv& S_1(\upsilon) \vee S_2(\upsilon).\end{aligned}$$

- We use the formula

$$\gamma_1(\vec{x}, \vec{\upsilon}) \equiv \theta_{\text{hyp},1}(\vec{\upsilon}) \wedge \bigwedge_i R_1(x_i, \upsilon_i)$$

  to pick out those rows that are indexed by a triple of feet $(e, f, g) \in (F_1^{\mathbf{M}_1})^3$ and a hyperedge $h \in \Sigma_1$ with $\rho_1(\{e, f, g\}) = h$. Similarly, we use the formula

$$\gamma_2(\vec{x}, \vec{\upsilon}) \equiv \exists \vec{\mu} \left(\theta_{\text{hyp},1}(\vec{\upsilon}) \wedge \theta_{\text{hyp},2}(\vec{\mu}) \wedge \bigwedge_i (\mathsf{ord}(\mu_i) = \mathsf{ord}(\upsilon_i)) \wedge \bigwedge_i (R_2(x_i, \mu_i))\right)$$

  to pick out those rows that are indexed by a triple of feet $(e, f, g) \in (F_2^{\mathbf{M}_2})^3$ and a hyperedge $h \in \Sigma_1$ with $\rho_2(\{e, f, g\}) = \iota(h)$. Here ι is the isomorphism of hypergraphs we have fixed before. Observe that the isomorphism ι simply maps each segment in $S_1^{\mathbf{M}_1}$ to the segment in $S_2^{\mathbf{M}_2}$ at the same position in the respective segment ordering; that is, for all segments $s \in S_1^{\mathbf{M}}$, $\iota(s) = t$ if and only if $\mathsf{ord}(x)^{(\mathbf{M}_1, s)} = \mathsf{ord}(x)^{(\mathbf{M}_2, t)}$.

  We also define a formula

$$\theta_{\text{ref}}(\vec{\upsilon}) \equiv (\upsilon_1 = \upsilon_2) \wedge (\upsilon_1 = \upsilon_3) \wedge S_1(\upsilon_1) \wedge (\mathsf{ord}(\upsilon_1) = 1_N),$$

  for which it holds that $\mathbf{M} \vDash \theta_{\text{ref}}[s_1, s_2, s_3]$ if and only if $s_1 = s_2 = s_3 = s \in S_1^{\mathbf{M}_1}$ and s is the first segment in the ordering $\leq_1^{\mathbf{M}_1}$. That is, we treat (s, s, s) as a fixed "reference triple" of segments, which explains the naming of the formula. This formula will be used to ensure that variables in the hyperedge equations won't be repeated, as we see below. In particular, note that (s, s, s) is not a hyperedge.

  The hyperedge equations over $\mathbf{M}_1$ can now be defined by formulae

$$\alpha_{\text{hyp},1}(\vec{x}, \vec{\mu}; y, \vec{\lambda}) \equiv \gamma_1(\vec{x}, \vec{\mu}) \wedge \left(\left(\left(\bigvee_{i=1}^{3} (y = x_i)\right) \wedge \theta_{\text{ref}}(\vec{\lambda})\right) \vee \left((\vec{\mu} = \vec{\lambda}) \wedge (y = c_1)\right)\right)$$

and

$$\beta_{\text{hyp},1}(\vec{x},\vec{\mu}) \equiv \gamma_1(\vec{x},\vec{\mu}) \wedge P_1(\vec{x}).$$

Note that in the definition of $\alpha_{\text{hyp},1}(\vec{x},\vec{\mu};y,\vec{\lambda})$, we used the shoe constant c_1 to ensure that each equation contains exactly one occurence of a "hyperedge variable" x_h.

The hyperedge equations over $\mathbf{M}_2$ can be similarly defined by formulae $\alpha_{\text{hyp},2}(\vec{x},\vec{\mu};\vec{y},\vec{\lambda})$ and $\beta_{\text{hyp},2}(\vec{x},\vec{\mu})$.

- The shoe equation can be defined by formulae

$$\alpha_{\text{shoe}}(x_1,x_2;y) \equiv (x_1 = c_1) \wedge (x_2 = c_2) \wedge (x_1 = y_1) \wedge ((y = x_1) \vee (y = x_2)) \text{ and}$$
$$\beta_{\text{shoe}}(x_1,x_2) \equiv (x_1 \neq x_1).$$

Finally, the system $\mathcal{S}_{\mathbf{M}}$ can be defined by formulae $\alpha(v,\vec{x},\vec{\mu};y,\vec{\lambda})$ and $\beta(v,\vec{x},\vec{\mu})$ over $\mathbf{M}$, where

$$\begin{aligned}\alpha(v,\vec{x},\vec{\mu};y,\vec{\lambda}) \equiv &\big((x_1 = x_2) \wedge (x_2 = x_3) \wedge \theta_{\text{ref}}(\vec{\lambda}) \wedge \alpha_{\text{seg}}(v;y)\big)\\ &\vee\big((x_1 \neq x_2) \wedge (x_2 = x_3) \wedge \theta_{\text{ref}}(\vec{\lambda}) \wedge \alpha_{\text{shoe}}(x_1,x_2;y)\big)\\ &\vee\big((x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_1 \neq x_3) \wedge (\alpha_{\text{hyp},1}(\vec{x},\vec{\mu};y,\vec{\lambda}) \vee \alpha_{\text{hyp},2}(\vec{x},\vec{\mu};y,\vec{\lambda}))\big),\end{aligned}$$

and $\beta(v,\vec{x},\vec{\mu})$ is defined similarly. Here the intended meaning is that the segment equations are indexed by tuples $v\vec{x}\vec{\mu}$ when all the components of $\vec{x}$ are equal, the shoe equation is indexed by a tuple $v\vec{x}\vec{\mu}$ when $x_1 \neq x_2 = x_3$, and the hyperedge equations are indexed by tuples $v\vec{x}\vec{\mu}$ when all the components of $\vec{x}$ are distinct. Note that there is redundancy in this description as some of the equations will be repeated a number of times. However, this does of course not affect the solvability of the system. □

Recall that we assume that $\mathbf{M} \models \mathsf{hyperiso}$ and that there is an isomorphism ι of the two disjoint segment hypergraphs in $\mathbf{M}$.

Lemma 5.7. *The system $\mathcal{S}_{\mathbf{M}}$ is solvable if and only if $\mathbf{M}_1 \cong \mathbf{M}_2$.*

Proof. First, suppose $\mathcal{S}_{\mathbf{M}}$ is solvable and let $T : V \to \mathsf{GF}_2$ be an assignment of values to the variables $(x_v)_{v\in V}$ that satisfies $\mathcal{S}_{\mathbf{M}}$. Define a map $\gamma : F_1^{\mathbf{M}_1} \to F_2^{\mathbf{M}_2}$ that pairs together each foot of $\mathbf{M}_1$ with a foot of $\mathbf{M}_2$ as follows:

$$\gamma(f) = g :\Leftrightarrow \iota(\rho_1(f)) = \rho_2(g) \text{ and } T(f) = T(g),$$

for all $f \in F_1^{\mathbf{M}_1}$ and $g \in F_2^{\mathbf{M}_2}$. That is, for each segment $s \in \mathbf{M}_1$, γ maps the two feet attached to s to the two feet attached to $\iota(s)$ in $\mathbf{M}_2$, where ι is the isomorphism of segment hypergraphs we have fixed earlier. There are two possible injective mappings from the pair of feet $\rho_1^{-1}(s)$ to the pair of feet $\rho_1^{-1}(\iota(s))$ and γ is defined by choosing the one mapping that agrees with the truth assignment T; that is, a foot f is mapped to a foot g if, and ony if, variables x_f and x_g are assigned the same value by T.

Combining ι and γ, we define a map $\Phi : \mathbf{M}_1 \to \mathbf{M}_2$ by $\Phi := \iota \cup \gamma$. That is, Φ maps segments according to ι and maps feet according to γ. We claim that Φ is an isomorphism of multipedes. To see this, first observe that Φ preserves the hypergraph structure and the ordering of segments as it extends the hypergraph isomorphism ι. Also by the definition of γ, it is clear that Φ preserves the relation that associates feet with segments. Furthermore:

- According to the segment equations of $\mathcal{S}_{\mathbf{M}}$, γ (and hence Φ) is bijective.
- According to the shoe equation of $\mathcal{S}_{\mathbf{M}}$, $T(c_1^{\mathbf{M}_1})$ and $T(c_2^{\mathbf{M}_2})$ must be equal. Hence, Φ maps the shoe of $\mathbf{M}_1$ to the shoe of $\mathbf{M}_2$.
- Suppose $\{e, f, g\} \subseteq F_1^{\mathbf{M}_1}$ is a three-element set of feet that ρ_1 maps onto a hyperedge $h \in H_1^{\mathbf{M}_1}$. Thus,

  $$(x_e + x_f + x_g) + x_h = (x_{\gamma(e)} + x_{\gamma(f)} + x_{\gamma(g)}) + x_h,$$

  over GF_2. It follows from the definition of the hyperedge equations of $\mathcal{S}_{\mathbf{M}}$ that

  $$\{e, f, g\} \in \Pi_1 \Leftrightarrow \{\gamma(e), \gamma(f), \gamma(g)\} \in \Pi_2,$$

  which shows that Φ maps positive triples to positive triples and thus preserves the positivity relation.

Together, this settles the claim.

Now suppose $\Phi : \mathbf{M}_1 \to \mathbf{M}_2$ is an isomorphism of multipedes. We will show that there is an assignment of values to the variables $(x_v)_{v \in V}$ that satisfies $\mathcal{S}_{\mathbf{M}}$.

Firstly, for each segment $s \in S_1^{\mathbf{M}_1}$ with an associated pair of feet $\rho_1^{-1} = \{e, f\}$, let T assign opposite values to the two feet e and f. That is, $T(f) = 1 - T(e)$ and $T(e) = 1 - T(f)$. This choice of assignment can be arbitrary. Doing this for all segments defines T on $F_1^{\mathbf{M}_1}$. Now for each hyperedge $h \in \Sigma_1$ consider a positive triple $\{e, f, g\} \in \Pi_1$. Assign a value to the variable x_h such that $(x_e + x_f + x_g) + x_h = 1$. This choice does not depend on which positive tuple we consider, as all the four positive triples mapped onto h are related by the condition $\|X \triangle Y\| \equiv 0 \pmod 2$, and permuting an even number of feet of any tuple (i.e. assigning opposite values to an even number of the three variables x_e, x_f and x_g) always preserves the parity of the sum $x_e + x_f + x_g$ in GF_2.

Now all that remains is to define T on $F_2^{\mathbf{M}_2}$. We do that according to the isomorphism Φ; that is, for each $f \in F_1^{\mathbf{M}_1}$ we set $T(\Phi(f)) := T(f)$. This concludes the definition of the assignment T. It remains to argue that this is a satisfying assignment to the system of linear equations $\mathcal{S}_{\mathbf{M}}$. But this should be clear from our construction of T. In particular,

- segment equations are satisfied because T assigns opposite values to the two feet attached to any segment;
- hyperedge equations are satisfied because of the way T assigns value to each hyperedge variable x_h; and
- the shoe equation is satisfied because T respects the isomorphism Φ. □

The preceding lemmas, along with the FOC-sentence hyperiso, now establish that there is a first-order (with counting) reduction from the problem of deciding isomorphism of multipedes to the problem of deciding solvability of linear systems over GF_2. The latter problem can be defined in FOR_2 by Theorem 4.12. Hence, we get the following result.

Theorem 5.8 (Multipede isomorphism in FOR_2)**.** *There is a sentence $\varphi_{MI} \in \mathrm{FOR}_2$ such that for all structures $\mathbf{M} = \mathbf{M}_1 \dot\cup \mathbf{M}_2$, where $\mathbf{M}_1$ and $\mathbf{M}_2$ are multipedes, it holds that $\mathbf{M} \models \varphi_{MI}$ if and only if $\mathbf{M}_1 \cong \mathbf{M}_2$.* □

5.2 Descriptive complexity

In descriptive complexity theory, it is known that extensions of first-order logic with various fixed-point operators capture different complexity classes on the class of finite ordered structures. For instance, on ordered structures FO+TC captures non-deterministic logspace (NL) while IFP captures PTIME. In this section we show such a natural correspondence for first-order rank logics on ordered structures, by proving that $\mathrm{FOR}_{\mathbb{Q}}$ captures the complexity class $\mathrm{L}^{\mathrm{C_=L}}$ and that for prime p, FOR_p captures the complexity class $\mathrm{MOD}_p\mathrm{L}$. While these classes are perhaps not as well known as some of the more established complexity classes (and we do give the formal definition of each one later), they do correspond to natural levels of complexity and have been extensively studied in the literature over the past couple of decades. In particular, $\mathrm{MOD}_2\mathrm{L}$ is better known under the name "parity logspace", denoted by $\oplus\mathrm{L}$, and $\mathrm{L}^{\mathrm{C_=L}}$ equals $\mathrm{C_=LH}$, the exact logspace counting hierarchy.

Put in the context of other known capturing statements, these results give us the following picture of logspace descriptive complexity on ordered structures. Here, the capturing result for FOC is from Etessami [47], where ThC^0 is the class of languages that can be decided by Boolean circuits with constant depth and polynomial size, containing only unbounded-fanin AND gates, OR gates, and threshold gates [59]. Both the capturing results for FO+DTC and FO+TC are from Immerman [43]. The inclusion $\mathrm{NL} \subseteq \mathrm{L}^{\mathrm{C_=L}}$ is from [1]. As evident from the diagram, relations between many of the logspace complexity classes are not fully understood. In particular, it is an open problem whether NL and $\oplus\mathrm{L}$ are directly comparable, although there is some evidence that $\mathrm{NL} \subseteq \oplus\mathrm{L}$ (see e.g. Allender [3]). In light of our capturing results, it is now conceivable that this important open problem can be settled by purely logical methods.

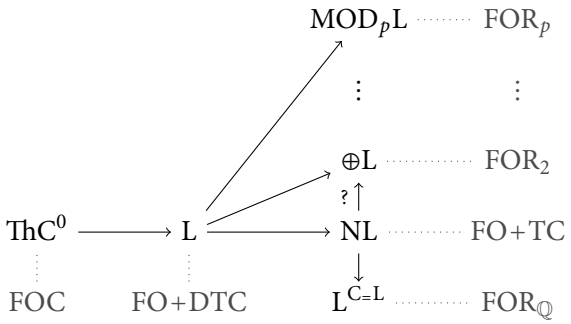


Figure 5.3: Descriptive complexity of logarithmic space classes. Complexity classes are typed in black, logics in grey. Arrows between complexity classes indicate inclusion and dotted lines between logics and complexity classes denote capturing on ordered structures.

Before we describe the actual capturing results, we first review some notions from both descriptive and computational complexity in the next two sections. Throughout, all structures are assumed to be finite.

5.2.1 Encoding ordered structures as strings

For a vocabulary τ with $\leq \in \tau$, we call a τ-structure $\mathbf{A}$ an *ordered structure* if $\mathbf{A}$ interprets $\leq$ as a total linear order of its universe. We identify the linearly ordered universe of $\mathbf{A}$ with

$[0, \|\mathbf{A}\| - 1] \subset \mathbb{N}_0$. An ordered τ-structure $\mathbf{A}$ can be encoded as a word over $\Sigma := \{0,1\}$ in a canonical way as follows. This encoding is similar to the encoding we described in §2.9.2, except now we consider the built-in ordering of the structure.

Assume $\tau = \{R_1, \ldots, R_s, c_1, \ldots, c_t\}$, where the R_i are relation symbols and the c_i are constants, and let $U(\mathbf{A}) = \{0, \ldots, n-1\}$. For each k-ary relation symbol $R \in \tau$, the relation $R^{\mathbf{A}}$ is encoded by an n^k-bit string $\mathsf{enc}(R^{\mathbf{A}})$ where the j-th bit of $\mathsf{enc}(R^{\mathbf{A}})$ is 1 if and only if $\vec{a} \in R^{\mathbf{A}}$ for the k-tuple $\vec{a}$ for which $\sum_{i=0}^{k-1} a_i n^i = j$. Constants can be encoded similarly, by viewing each constant as a unary relation containing exactly one element. Putting this all together, we write $\mathsf{enc}(\mathbf{A})$ for the canonical encoding of $\mathbf{A}$ defined by

$$\mathsf{enc}(\mathbf{A}) := 0^n 1 \cdot \mathsf{enc}(R_1^{\mathbf{A}}) \cdots \mathsf{enc}(R_s^{\mathbf{A}}) \cdot \mathsf{enc}(c_1^{\mathbf{A}}) \cdots \mathsf{enc}(c_t^{\mathbf{A}}),$$

where $a \cdot b$ denotes the concatenation of strings a and b. For further details, see e.g. Ebbinghaus and Flum [23] or Libkin [52]. It is not hard to see that there is a deterministic logarithmic-space algorithm, depending only on τ, that decides whether a given string Σ^* is a valid encoding of a τ-structure.

If $\mathcal{K}$ is a class of ordered τ-structures, then we say that a Turing machine M decides $\mathcal{K}$ if for any ordered τ-structure $\mathbf{A}$,

$$M(\mathsf{enc}(\mathbf{A})) \begin{cases} \text{accepts} & \text{if } \mathbf{A} \in \mathcal{K}, \\ \text{rejects} & \text{if } \mathbf{A} \notin \mathcal{K}. \end{cases}$$

Here we assume that the input alphabet of M contains Σ. Since it can be decided in logarithmic space whether a given string in Σ^* is a valid encoding of a τ-structure, M can be turned into a machine that decides $\{\mathsf{enc}(\mathbf{A}) \mid \mathbf{A} \in \mathcal{K}\} \subseteq \{0,1\}^*$, assuming M is not restricted to use less than logarithmic amount of work space. For a complexity class C we write $\mathcal{K} \in \mathrm{C}$ to mean $\{\mathsf{enc}(\mathbf{A}) \mid \mathbf{A} \in \mathcal{K}\} \in \mathrm{C}$.

Definition 5.9 (Capturing complexity on ordered structures)**.** Given a complexity class C, we say that a logic L *captures* C *on ordered structures* if for any vocabulary τ with $\leq \in \tau$ and any class $\mathcal{K}$ of ordered τ-structures, $\mathcal{K} \in \mathrm{C}$ if and only if there is a sentence $\varphi_{\mathcal{K}}$ of $\mathrm{L}[\tau]$ that defines $\mathcal{K}$. ■

5.2.2 Logspace-bounded Turing machines

All the Turing machines we consider in this and the following sections are non-deterministic logspace machines that use Σ as both their input and work tape alphabet. Let M be one such machine, with space bound $d \cdot \log n$, where $d \in \mathbb{N}$. Given a string $x \in \Sigma^*$, we write $G_{M,x}$ for the *configuration graph* of M on input x. Recall that $G_{M,x}$ is the directed graph whose vertices are all possible configurations of M with x on the input tape and the work tape having at most $d \cdot \log(|x|)$ symbols, and there is an edge from configuration c_1 to configuration c_2 if and only if the machine M can make the transition from c_1 to c_2 in one step. Since each configuration can be described using at most $e \cdot \log|x|$ bits, where $e \geq d$ is a constant depending only on d and M, it can be seen that the size of the graph $G_{M,x}$ (i.e. the number of possible configurations of M on input x) is bounded above by $p_M(|x|)$, where p_M is a polynomial depending only on M and d.

Also note that $G_{M,x}$ can generally be assumed to be free of cycles; if it is not, then we can consider instead the configuration graph of the Turing machine M' obtained from M by

keeping a time mark on the work tape that is increased by 1 at every transition. The maximum time mark required is simply the number of possible configurations of M on input x, and since this number is at most $p_M(|x|)$, it is clear that the machine M' also requires only logarithmic workspace.

Finally, we can also assume, without loss of generality, that M has only one accepting configuration; that is, a configuration in the accepting state of M. Otherwise, if M on input x has more than one accepting configuration, then we consider the machine M' obtained from M by (a) adding a new state s'_{acc}, (b) making s'_{acc} the accepting state of M' and (c) modifying the transition table of M so that whenever the machine reaches state s_{acc}, it will clear the work tape, move the work tape and input tape heads to the initial position and go to the new accepting state s'_{acc}. This ensures that the corresponding configuration is the unique accept configuration of M' on input x.

Write $\sigma_{\text{graph}} := \{E\}$ for the signature of graphs, where E is a binary relation symbol. Ebbinghaus and Flum [23] show that for each vocabulary τ containing the binary relation symbol $\leq$, there is an FO+DTC interpretation Φ of σ_{graph} in τ for which it holds that for any τ-structure $\mathbf{A}$, $\Phi(\mathbf{A})$ is the configuration graph of M on input $\mathsf{enc}(\mathbf{A})$. Here there is a fixed encoding of the configurations of M on input $\mathsf{enc}(\mathbf{A})$ by e-tuples of elements from $\mathbf{A}$, where $e \geq d$ is a constant depending only on d and M, as above. A close look at the proof of this statement illustrates that a similar interpretation can also be defined using formulae of FOC without any recursion, as we state more formally below.

Lemma 5.10 (Configuration graph in FOC)**.** *Let M be a non-deterministic logarithmic-space Turing machine with q states and a space bound $d \cdot \log n$, where $d \in \mathbb{N}$. Then there is a constant e, depending only on M and d, and* FOC*-formulae $\chi_{start}(\vec{x})$, $\chi_{accept}(\vec{x})$, and $\chi_{succ}(\vec{x}, \vec{y})$ such that for all ordered τ-structures $\mathbf{A}$ with $\|A\| > \max\{d \cdot \log\|A\|, q\}$ and $\vec{a} \in U(\mathbf{A})^e$,*

- $(\mathbf{A}, \vec{a}) \vDash \chi_{start}(\vec{x})$ *if and only if $\vec{a}$ encodes the start configuration of M on input* $\mathsf{enc}(\mathbf{A})$*;*
- $(A, \vec{a}) \vDash \chi_{accept}(\vec{x})$ *if and only if $\vec{a}$ encodes the accepting configuration of M on input* $\mathsf{enc}(\mathbf{A})$*; and*
- $\mathbf{A} \vDash \chi_{succ}(\vec{a}, \vec{b})$ *if and only if $\vec{a}$ and $\vec{b}$ encode valid configurations of M on input* $\mathsf{enc}(\mathbf{A})$ *and $\vec{b}$ is a successor configuration of $\vec{a}$.*

Here, $\vec{x}$ and $\vec{y}$ are assumed to be e-tuples of distinct variables.

Sketch proof. The formulae $\chi_{\text{start}}(\vec{x})$ and $\chi_{\text{accept}}(\vec{x})$ can be expressed in first-order logic, by Lemma 7.3.7 in [23]. That same lemma shows that the formula $\chi_{\text{succ}}(\vec{x}, \vec{y})$ can be expressed in FO+DTC over vocabulary τ. A close look at the proof of that lemma shows that the **dtc**-operator is required only for defining formulae $\varphi_+(x, y, z)$, $\varphi_\cdot(x, y, z)$, $\varphi_2(x, y)$ and $\varphi_{\log(\text{universe})}(x)$ (see Lemma 7.3.11 in [23]), for which it holds that for any ordered τ-structure $\mathbf{A}$ with $U(\mathbf{A}) = \{0, \ldots, n-1\}$ and any $a, b, c \in U(\mathbf{A})$,

$$
\begin{aligned}
(\mathbf{A}, a, b, c) \vDash \varphi_+ &\Leftrightarrow a + b = c,\\
(\mathbf{A}, a, b, c) \vDash \varphi_\cdot &\Leftrightarrow a \cdot b = c,\\
(\mathbf{A}, a, b) \vDash \varphi_2 &\Leftrightarrow 2^a = b, \text{ and}\\
(\mathbf{A}, a) \vDash \varphi_{\log(\text{universe})} &\Leftrightarrow a = \log\|A\|.
\end{aligned}
$$

Clearly, the formulae $\varphi_+(x, y, z)$ and $\varphi_\cdot(x, y, z)$ can be defined in FOC by mapping the elements of **A** into the number sort. From the proof of Theorem 6.12 in [52], the formula $\varphi_2(x, y)$ is expressible over ordered structures in FO$(+, \cdot)$, first-order logic with addition and multiplication over the domain elements. Finally, $\varphi_{\log(\text{universe})}(x)$ can be expressed with a simple application of $\varphi_2(x, y)$, as in the proof of Lemma 7.3.11 in [23]. □

For some of the complexity classes we define later, we need to consider Turing machines with access to an oracle. Loosely speaking, an oracle machine is a Turing machine which may pose questions (or *queries*) to a function $f : \Sigma^* \to \Sigma^*$, called the *oracle*. Apart from an input tape and a work tape, an oracle machine has an additional *oracle tape* which it uses to communicate with the oracle. To do that, the machine writes a query string x to the oracle tape and then tells the oracle to execute. In a single step, the oracle computes its function, erases the input, and writes its output $f(x)$ to the oracle tape. We write M^f to denote the oracle Turing machine M with access to the oracle f. Frequently, we consider oracles that are the characteristic function χ_A of some language A; in that case we write M^A to denote the machine M with access to the oracle χ_A.

To simulate oracle access for logspace-bounded machines, we follow the Ruzzo-Simon-Tompa oracle access mechanism described in [2] and [1] (see also Ruzzo et al. [61] for the original definitions). According to this mechanism, a logspace-bounded machine M is required to write its queries on the oracle tape in a deterministic manner. The number of possible configurations before the machine starts writing on the oracle tape is at most polynomial. It follows that for any given input string x, the number of queries M can submit to its oracle is at most polynomial in the size of x. Moreover, all these queries can be written in sequence on the oracle tape even before the machine starts reading its input (knowing only the size of the input). By pre-computing all oracle queries in this way, it can be ensured that M does not have to query its oracle for the remainder of the computation. Thus, in this context, an oracle-access machine M can be seen as an ordinary logspace-bounded machine with an additional polynomial-size "advise string" given as input.

Finally, we write $\|M(x)\|$ to denote the number of accepting computation paths of a nondeterministic machine M with input string x. The same notation will be used for machines with oracle access.

5.2.3 FOR_p captures $\mathrm{MOD}_p\mathrm{L}$ on ordered structures

In [11], Buntrock et al. investigated the logspace analogues of polynomial-time counting classes. In particular, they showed that many of the standard problems of linear algebra are complete for the logspace modulo-counting classes $\mathrm{MOD}_k\mathrm{L}$, which are defined as follows.

Definition 5.11 (Complexity class $\mathrm{MOD}_k\mathrm{L}$)**.** Let $k \in \mathbb{N}$. A language $L \subseteq \Sigma^*$ belongs to $\mathrm{MOD}_k\mathrm{L}$ if there is a non-deterministic logspace machine M, such that for every $x \in \Sigma^*$: $x \in L$ if and only if $\|M(x)\| \not\equiv 0 \bmod k$. ■

Remark. Note that $\mathrm{MOD}_2\mathrm{L}$ is better known under the name "parity logspace", usually denoted by $\oplus\mathrm{L}$.

Theorem 5.12. *Let p be prime. Then FOR_p captures $\mathrm{MOD}_p\mathrm{L}$ on ordered structures.*

The proof of this theorem consists of two parts. Firstly, we show that for any sentence $\varphi \in \mathrm{FOR}_p$, the class of finite ordered models of φ can be decided in $\mathrm{MOD}_p\mathrm{L}$. More specifically, we show that for any φ there is a non-deterministic logspace machine M_φ such that for any structure $\mathbf{A}$, $\|M_\varphi(\mathsf{enc}(\mathbf{A}))\| \not\equiv 0 \bmod p$ if and only if $\mathbf{A} \models \varphi$. Secondly, given a non-deterministic logspace machine M deciding a class of finite structures $\mathcal{K} \in \mathrm{MOD}_p\mathrm{L}$, we construct a sentence φ_M that holds in a structure $\mathbf{A}$ if and only if $\mathbf{A} \in \mathcal{K}$ (equivalently, if and only if $\|M(\mathsf{enc}(\mathbf{A}))\| \not\equiv 0 \bmod p$).

For the first part, assume that τ is a vocabulary with $\leq \in \tau$, and that φ is a $\mathrm{FOR}_p[\tau]$-sentence. In order to deal with rank operators occurring in φ, we need two results on $\mathrm{MOD}_p\mathrm{L}$-complexity. The first one says that the rank of a matrix over GF_p can be verified in $\mathrm{MOD}_p\mathrm{L}$.

Lemma 5.13 (Buntrock et al. [11])**.** *Let p be prime. Then the problem which takes as input an integer $r \in \mathbb{N}_0$ and a matrix $A \in \mathsf{GF}_p^{m\times n}$ and decides whether* $\operatorname{rank} A = r$ *is in* $\mathrm{MOD}_p\mathrm{L}$. □

The second result states that non-deterministic logspace machines deciding languages in $\mathrm{MOD}_p\mathrm{L}$ making oracle queries to a $\mathrm{MOD}_p\mathrm{L}$ problem can be simulated in $\mathrm{MOD}_p\mathrm{L}$ without oracle queries.

Lemma 5.14 (Hertrampf et al. [39])**.** *Let p be prime. Then* $\mathrm{MOD}_p\mathrm{L}^{\mathrm{MOD}_p\mathrm{L}} = \mathrm{MOD}_p\mathrm{L}$. □

It is left to show that the language

$$L_\varphi := \{\mathsf{enc}(\mathbf{A}) \mid \mathbf{A} \in \mathrm{fin}[\tau] \text{ and } \mathbf{A} \models \varphi\}$$

is in $\mathrm{MOD}_p\mathrm{L}$ by means of a non-deterministic logspace machine M_φ. The proof is by induction on the structure of FOR_p number terms and formulae. That is, we show that

- for each FOR_p-formula $\theta(\vec{x})$ of vocabulary τ with k free variables, the language

  $$L_{\theta(\vec{x})} := \{\mathsf{enc}(\mathbf{A}, \vec{a}) \mid \mathbf{A} \in \mathrm{fin}[\tau], \vec{a} \in U(\mathbf{A})^k \text{ and } (\mathbf{A}, \vec{a}) \models \theta(\vec{x})\}$$

  is in $\mathrm{MOD}_p\mathrm{L}$ by means of a non-deterministic logspace machine $M_{\theta(\vec{x})}$, where we write $\mathsf{enc}(\mathbf{A}, \vec{a})$ for the string obtained by extending $\mathsf{enc}(\mathbf{A})$ with an $\|\mathbf{A}\|^k$-bit string representing the tuple of elements $\vec{a}$; and

- for any number term $\eta(\vec{x})$ of $\mathrm{FOR}_p[\tau]$, there is a deterministic logspace machine $M_{\eta(\vec{x})}$, with access to $\mathrm{MOD}_p\mathrm{L}$-oracles, that when given as input a string $\mathsf{enc}(\mathbf{A}, \vec{a})$, accepts and halts with the integer $\eta(\vec{x})^{(\mathbf{A},\vec{a})}$ on the work tape.

Since existential quantifiers can be expressed using rank operators, it is enough to show the following cases.

- Atomic formulae $R\vec{x}$ and $x = y$ can be decided deterministically by lookup in $\mathsf{enc}(\mathbf{A})$ on the input tape. Similarly for formulae involving constant symbols in τ.
- Computation of the number constants $0_N^{\mathbf{A}}$ and $1_N^{\mathbf{A}}$ can be carried out in constant time by writing the integers zero or one on the work tape, respectively.
- Addition and multiplication of number terms is carried out deterministically in logspace.

- Given number terms s and t, formulae $\theta \equiv s = t$ and $\theta \equiv s \leq t$ are decided deterministically from the values computed for $s^{\mathbf{A}}$ and $t^{\mathbf{A}}$ by machines M_s and M_t, respectively.
- Now consider a formula θ. If $\theta \equiv \neg\psi$, then M_θ makes an oracle query to L_ψ and accepts if and only if the oracle rejects the input. If $\theta \equiv \psi_1 \wedge \psi_2$, then M_θ makes oracle queries to L_{ψ_1} and L_{ψ_2} and accepts if both queries succeed, and rejects otherwise. In both cases M_θ is a deterministic logspace-bounded machine with access to $\mathrm{MOD}_p\mathrm{L}$-oracles. As $\mathrm{L} \subseteq \mathrm{MOD}_p\mathrm{L}$ for any prime p, the language $L_\theta \in \mathrm{L}^{\mathrm{MOD}_p\mathrm{L}}$ decided by M_θ is in $\mathrm{MOD}_p\mathrm{L}$, by Lemma 5.14.
- Finally, consider a number term $\eta \equiv \mathbf{rk}_p(\vec{x}, \vec{y}).\gamma$, with $k = \min\{\|\vec{x}\|, \|\vec{y}\|\}$. A machine computing η proceeds by checking for every integer r with $0 \leq r \leq \|\mathbf{A}\|^k$ if $\mathbf{rk}_p(\vec{x}, \vec{y}).\gamma = r$ by making oracle queries to the $\mathrm{MOD}_p\mathrm{L}$-language from Lemma 5.13. Instead of querying its input tape for matrix entries $\gamma(\vec{a}, \vec{b})$, M_η computes $\gamma(\vec{a}, \vec{b})$ deterministically if γ is a term or makes oracle queries to L_γ with input $\mathsf{enc}(\mathbf{A}, \vec{a}\vec{b})$ if γ is a formula. Since $r \leq \|A\|^k$, all possible rank values can be written down in space logarithmic in the size of $\mathbf{A}$. It follows that at most polynomially many oracle queries have to be made by M_η. Once all these oracle queries have been processed, the deterministic oracle machine M_η goes through the list to find the right rank value and then writes it down on its work tape, as required.

For any FOR_p-sentence φ there is therefore, by repeatedly applying Lemma 5.14, a nondeterministic logspace machine M_φ such that for any structure $\mathbf{A}$, $\|M_\varphi(\mathsf{enc}(\mathbf{A}))\| \not\equiv 0 \bmod p$ if and only if $\mathbf{A} \models \varphi$. Hence, L_φ is in $\mathrm{MOD}_p\mathrm{L}$.

For the other part, consider a non-deterministic machine M with space bound $d \cdot \log n$ that decides a class of τ-structures $\mathcal{K} \in \mathrm{MOD}_p\mathrm{L}$. As noted before, we can assume that M has only one accepting configuration. We construct a formula φ_M that defines $\mathcal{K}$. In the following, we restrict ourselves to structures $\mathbf{A}$ so that $\|\mathbf{A}\| > N := \max\{d \cdot \log\|\mathbf{A}\|, q\}$ where q is the number of states of M. For those structures $\mathbf{B}$ whose size is less than N we can write down a fixed formula which checks whether $\mathsf{enc}(\mathbf{B}) \in \mathcal{K}$ by comparing $\mathbf{B}$ with a finite number of small structures that belong to $\mathcal{K}$.

Given a τ-structure $\mathbf{A}$, write $G_{M,\mathbf{A}}$ for the configuration graph of M on input $\mathsf{enc}(\mathbf{A})$. Let $e > d$ be a constant, depending only on M, such that all configurations of M may be encoded by e-tuples of elements from $\mathbf{A}$, as discussed earlier. If s and t denote the unique start and accept configurations of M on input $\mathsf{enc}(\mathbf{A})$, respectively, then it can be seen that $\mathbf{A} \in \mathcal{K}$ if and only if the number of paths from s to t in $G_{M,\mathbf{A}}$ is $\not\equiv 0 \bmod p$. By Lemma 5.10, there is a formula $\chi_{\mathrm{succ}}(\vec{x}, \vec{y})$ of FOC (and hence of FOR_p) which defines over any τ-structure $\mathbf{A}$ the configuration graph $G_{M,\mathbf{A}}$ of M on input $\mathsf{enc}(\mathbf{A})$, where $\vec{x}$ and $\vec{y}$ are e-tuples of variables. In other words, $\chi_{\mathrm{succ}}(\vec{x}, \vec{y})^{\mathbf{A}}$ defines the adjacency matrix A of $G_{M,\mathbf{A}}$. Let I denote the identity matrix of the same dimension as A. Then $I - A$ is definable in FOR_p by a term $\eta(\vec{x}, \vec{y})$, and the term

$$\eta^*(\vec{x}, \vec{y}) \equiv \big(\neg\chi_{\mathrm{accept}}(\vec{x}) \wedge \neg\chi_{\mathrm{start}}(\vec{y})\big) \cdot \eta(\vec{x}, \vec{y})$$

defines $I - A$ with row t and column s set to 0. Here, $\chi_{\mathrm{accept}}(\vec{x})$ and $\chi_{\mathrm{start}}(\vec{y})$ are obtained from Lemma 5.10. The formula $\varepsilon(\vec{x}, \vec{y}) \equiv (\vec{x} = \vec{y}) \wedge \neg\chi_{\mathrm{start}}(\vec{x})$ defines the identity matrix of the same dimension as A with row s set to 0. Let

$$\varphi_M \equiv \mathbf{rk}_p(\vec{x}, \vec{y}).\varepsilon = \mathbf{rk}_p(\vec{x}, \vec{y}).\eta^*.$$

The following completes the proof of Theorem 5.12.

Lemma 5.15. *For any ordered τ-structure $\mathbf{A}$ with $\|A\| > \max\{d \cdot \log\|A\|, q\}$, $\mathbf{A} \models \varphi_M$ if and only if $\mathbf{A} \in \mathcal{K}$.*

Proof. As $G_{M,\mathbf{A}}$ is cycle-free and has n^e vertices, there is no path of length larger than $n^e =: m$, hence $A^m = 0$. Here, e is the number of variables needed to encode configurations of M over $\mathbf{A}$, as above. Thus, $I - A$ is non-singular over GF_p, with the inverse explicitly given by $(I-A)^{-1} := I + A + A^2 + \ldots + A^{m-1}$, where all arithmetic is over GF_p. To see this, note that

$$\begin{aligned}
&(I-A)(I + A + A^2 + \ldots + A^{m-1}) \\
=&(I + A + A^2 + \ldots + A^{m-1}) - (A + A^2 + \ldots + A^{m-1} + A^m) \\
=&(I + A + A^2 + \ldots + A^{m-1}) - (A + A^2 + \ldots + A^{m-1} + 0) \\
=&I,
\end{aligned}$$

as required. Notice that for $k \in \mathbb{N}_0$, the (i,j)-th entry of A^k equals the number of paths modulo p of length k from i to j in $G_{M,\mathbf{A}}$. Thus, $(I-A)^{-1}$ is the matrix of the total numbers of paths modulo p. Recall that s and t denote the start and accept configuration, respectively. Then $\mathbf{A} \in \mathcal{K}$ if and only if $(I-A)^{-1}(s,t) \neq 0$ over GF_p.

It can be shown [42] that for any invertible matrix B, the entries b^{-1}_{ij} of its inverse B^{-1} are given by

$$b^{-1}_{ij} = (-1)^{i+j} \det B_{ji} \big/ \det B,$$

where B_{ji} is B with the j-th row and the i-th column deleted (this expression is known as the *adjugate rule*). To check if $(I-A)^{-1}(s,t) \neq 0$, it is therefore enough to test if $(I-A)_{ts}$ has full rank, which is exactly what φ_M does. □

5.2.4 $\mathrm{FOR}_{\mathbb{Q}}$ captures $\mathrm{L}^{\mathrm{C_=L}}$ on ordered structures

Allender and Ogihara [2] introduced the complexity class $\mathrm{C_=L}$ (pronounced "exact logspace counting") to characterise the complexity of the class of singular matrices. The following is one of several equivalent ways of defining this class.

Definition 5.16 (Complexity class $\mathrm{C_=L}$)**.** A language $L \subseteq \Sigma^*$ belongs to $\mathrm{C_=L}$ if there is a non-deterministic logspace machine M, such that for every $x \in \Sigma^*$: $x \in L$ if and only if M on input x has exactly the same number of accepting and rejecting paths. ■

Allender and Ogihara [2] also consider $\mathrm{L}^{\mathrm{C_=L}}$, the class of languages decided by a deterministic logspace-bounded machine with access to an $\mathrm{C_=L}$ oracle. In this section we prove the following theorem.

Theorem 5.17. $\mathrm{FOR}_{\mathbb{Q}}$ *captures* $\mathrm{L}^{\mathrm{C_=L}}$ *on ordered structures.*

The proof of this theorem is quite similar to the proof of Theorem 5.12. First we recall a couple of results on $\mathrm{C_=L}$ computation. The first result concerns the *exact logspace counting hierarchy* $\mathrm{C_=LH}$, which is defined as follows. Define $\mathrm{C_=LH}_1$ to be $\mathrm{C_=L}$ and let $\mathrm{C_=LH}_{i+1}$ be the class of languages L for which there is a logspace-bounded non-deterministic oracle Turing machine M and a language $A \in \mathrm{C_=LH}_i$ such that for any string $x \in \Sigma^*$: $x \in L$ if and only if M^A on input x has the same number of accepting and rejecting computation paths. Finally, set $\mathrm{C_=LH} := \bigcup_i \mathrm{C_=LH}_i$. Allender and Ogihara [2] show that the exact counting logspace hierarchy collapses to $\mathrm{L}^{\mathrm{C_=L}}$.

Lemma 5.18 (Allender and Ogihara [2]). $\mathrm{C_{=}L}^{\mathrm{C_{=}L}} = \mathrm{L}^{\mathrm{C_{=}L}}$. *Hence* $\mathrm{C_{=}LH} = \mathrm{L}^{\mathrm{C_{=}L}}$. □

The second result that we need says that the rank of matrices with entries from the field of rational numbers can be verified in $\mathrm{C_{=}L}$.

Lemma 5.19 (Allender and Ogihara [2]). *There is a non-deterministic logspace machine* M_{rk} *for which it holds that when given as input an integer* $r \in \mathbb{N}_0$ *and a matrix* $A \in \mathbb{Q}^{m \times n}$, M_{rk} *has exactly the same number of accepting and rejecting paths if and only if* $\operatorname{rank} A = r$. □

We can now show that formulae of $\mathrm{FOR}_{\mathbb{Q}}$ can be evaluated in $\mathrm{L}^{\mathrm{C_{=}L}}$ on ordered structures.

Lemma 5.20 ($\mathrm{FOR}_{\mathbb{Q}} \subseteq \mathrm{L}^{\mathrm{C_{=}L}}$). *Let* τ *be a vocabulary containing the symbol* $\leq$. *For every sentence* $\varphi \in \mathrm{FOR}_{\mathbb{Q}}$ *of vocabulary* τ *the class of finite ordered models of* φ *can be decided in* $\mathrm{L}^{\mathrm{C_{=}L}}$.

Proof. Assume τ is a vocabulary with $\leq \in \tau$ and let φ be a $\mathrm{FOR}_{\mathbb{Q}}$-sentence of vocabulary τ. We show that there is a deterministic logspace-bounded machine M and a language $B \in \mathrm{C_{=}L}$, such that for every ordered τ-structure $\mathbf{A}$: $\mathbf{A} \models \varphi$ if and only if M^B accepts $\mathsf{enc}(\mathbf{A})$. As discussed in Theorem 5.12, it can be shown by induction that all logical and arithmetical operations other than application of rank operators can be evaluated in $\mathrm{L}^{\mathrm{C_{=}L}}$. All that remains is to show that rank operators of the form $\mathbf{rk}_{\mathbb{Q}}(\vec{x}, \vec{y}).(\varphi_n, \varphi_d, \psi, t)$ can be evaluated in $\mathrm{L}^{\mathrm{C_{=}L}}$. But by Lemma 5.19, and an argument similar to the one given in the proof of Theorem 5.12, this should be clear. □

We define Path-Difference to be the function problem that takes as input a directed acyclic graph $G = (V, E)$ and vertices $s_1, t_1, s_2, t_2 \in V$ and computes the value of $\#\mathrm{Path}_G(s_1, t_1) - \#\mathrm{Path}_G(s_2, t_2)$, where $\#\mathrm{Path}_G(u, v)$ is the number of paths from vertex $u \in V$ to vertex $v \in V$. Toda [63] shows that the problem Path-Difference can be logspace many-to-one reduced to the problem of computing the determinant of an $(0,1)$-integer matrix (although his statement of the path difference problem is slightly more generic). The following lemma shows that this reduction can in fact be turned into a first-order reduction.

Lemma 5.21. *Path-Difference* $\leq_{fo}$ *Zero-One-Det.*

Proof. Let $G = (V, E)$ be a directed acyclic graph and consider vertices $s_1, t_1, s_2, t_2 \in V$. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two disjoint copies of G, and write $x_1, y_1 \in V_1$ and $x_2, y_2 \in V_2$ to denote the vertices corresponding to s_1, t_1 and s_2, t_2 in each of the two copies of G, respectively. We construct a graph H from G_1 and G_2 as follows.

- The vertex set of H contains all vertices in V_1 and V_2. In addition, for each edge $e = (u, v)$ in G_1 or G_2, we add a new vertex w_e to H. Finally, we add two special vertices a and b.
- For each edge $e = (u, v)$ in G_1 or G_2, we add the two directed edges (u, w_e) and (w_e, v)
- We add the following edges to the two special vertices a and b:
  - $a \to x_1$, $y_1 \to b$, $b \to a$, and
  - $a \to x_2$, $y_2 \to a$.
- Finally, we add self-loops to all vertices except a.

Write B for the adjacency matrix of H. Clearly, H (and hence B) is first-order definable over G.

Our claim is now that the number $\#\mathrm{Path}_G(s_1, t_1) - \#\mathrm{Path}_G(s_2, t_2)$ is equal to the determinant of the matrix B over $\mathbb{Z}$. For completeness, we recall the proof of this claim from [63]. Consider an $n \times n$ matrix $A = (a_{ij})$. The determinant of A is given by

$$\det(A) := \sum_{\sigma \in \mathrm{Sym}(n)} \mathrm{sgn}(\sigma) \prod_i a_{i\sigma(i)},$$

where the sum is taken over the symmetric group $\mathrm{Sym}(n)$ of all permutations of $[n] := \{1, \ldots, n\}$. Here $\mathrm{sgn}(\sigma)$ denotes the sign of the permutation σ, defined by $\mathrm{sgn}(\sigma) := (-1)^m$ where m is the number of transpositions of pairs of elements that must be composed to build up the permutation σ. In particular, if $\sigma \in \mathrm{Sym}(n)$ is a cyclic permutation then $\mathrm{sgn}(\sigma) = 1$ if n is odd and $\mathrm{sgn}(\sigma) = -1$ otherwise. In general, when σ is not necessarily cyclic, we can decompose σ into a product of cyclic permutations. It can be seen that this cycle decomposition of σ, when interpreted as a graph on vertices $[n]$, induces a partition of the vertex set into disjoint cycles.

The matrix A can be seen as a weighted directed graph G_A on n vertices, where the weight of an edge from i to j is a_{ij}. Recall that a *cycle cover* of G_A is a set of cycles which are subgraphs of G_A and which collectively contain all vertices of G_A. Here we consider only cycle covers where the cycles are disjoint. Clearly, each cycle cover of A corresponds to a permutation σ of $[n]$, where the partition of vertices into cycles corresponds to the cycle decomposition of σ. It is noted by Toda [63] that each cycle cover of A corresponds to a permutation σ whose additive term in $\det(A)$ is non-zero (clearly, since each vertex in the cycle cover must have an edge to at least one other vertex in the cycle). Similarly, it can be shown that every permutation in $\mathrm{Sym}(n)$ whose additive term in $\det(A)$ is non-zero corresponds to a cycle cover of A.

Now consider the graph H constructed above, with $m \times m$ adjacency matrix $B = (b_{ij})$, where $m > n$ is the number of vertices in H. From the construction of H it can be seen that each cycle cover of H consists of (a) one large cycle that includes the special vertex a and (b) a number of self-loops, one for each vertex not on the big cycle. It can be seen from the construction of H that the big cycle can take only one of two forms:

(C1) either it contains the edge $a \to x_1$, in which case it must include a path from x_1 to y_1 and come back to a via the path $y_1 \to b \to a$; or

(C2) it contains the edge $a \to x_2$, in which case it must include a path from x_2 to y_2 and come back to a via the path $y_2 \to a$.

Hence, we see that the number of cycle covers containing (C1) is the same as the number of distinct paths in G_1 from x_1 to y_1, and likewise that the number of cycle covers containing (C2) is the same as the number of distinct paths in G_2 from x_2 to y_2. It is clear that any path from either x_1 to y_1 or from x_2 to y_2 in H must be of even length, where by length of a path we mean the number of edges it contains. This is due to the intermediate vertices w_e, inserted in between two endpoints of an edge in G_1 or G_2. Consequently, it can be seen that the two big cycles (C1) and (C2) described above must have odd and even length, respectively.

Now consider all the permutations in $\mathrm{Sym}(m)$ whose additive term in the expression for $\det(B)$ is non-zero. Divide these permutations into two sets: P_1, the set of all permutations whose corresponding cycle cover on H contains the cycle (C1), and P_2, the set of all permutations whose corresponding cycle cover on H contains the cycle (C2). By the above these

two sets are disjoint. Also, we see that $\operatorname{sgn}(\sigma) = 1$ for all $\sigma \in P_1$ and $\operatorname{sgn}(\sigma) = -1$ for all $\sigma \in P_2$. Now we can write

$$\begin{aligned}\det(B) &= \sum_{\sigma \in \mathrm{Sym}(m)} \operatorname{sgn}(\sigma) \prod_i b_{i\sigma(i)} \\ &= \Big(\sum_{\sigma \in P_1} \prod_i b_{i\sigma(i)}\Big) - \Big(\sum_{\sigma \in P_2} \prod_i b_{i\sigma(i)}\Big) \\ &= \|\mathbf{P}_1\| - \|\mathbf{P}_2\|,\end{aligned}$$

where the last equality comes from the fact that B is a $(0,1)$-matrix. It follows that $\det(B) = \#\mathrm{Path}_{G_1}(x_1, y_1) - \#\mathrm{Path}_{G_2}(x_2, y_2)$. Since $\#\mathrm{Path}_{G_i}(x_i, y_i) = \#\mathrm{Path}_G(s_i, t_i)$ for $i \in \{1, 2\}$, the lemma now follows. □

Lemma 5.22 ($\mathrm{C_=L} \subseteq \mathrm{FOR}_{\mathbb{Q}}$)**.** *Let $\mathcal{K} \in \mathrm{C_=L}$ be a class of finite τ-structures decided by a non-deterministic logspace-bounded machine M. Then there is a sentence $\varphi_M \in \mathrm{FOR}_{\mathbb{Q}}[\tau]$ such that for any τ-structure $\mathbf{A}$: $\mathbf{A} \models \varphi_M$ if and only if $\mathbf{A} \in \mathcal{K}$.*

Proof. Consider a non-deterministic logspace-bounded machine M that decides a class of τ-structures $\mathcal{K} \in \mathrm{C_=L}$. That is, for any τ-structure $\mathbf{A}$, $\mathbf{A} \in \mathcal{K}$ if and only if M on input $\mathsf{enc}(\mathbf{A})$ has the same number of accepting and rejecting computation paths. We show that there is a formula of $\mathrm{FOR}_{\mathbb{Q}}[\tau]$ that defines $\mathcal{K}$.

As in the proof of Theorem 5.12, let $\chi_{\mathrm{succ}}(\vec{x}, \vec{y})$ be the formula that defines over a given τ-structure $\mathbf{A}$ the adjacency matrix of the configuration graph $G := G_{M,\mathbf{A}}$ of M on input $\mathsf{enc}(\mathbf{A})$. We can assume, as before, that M on input $\mathsf{enc}(\mathbf{A})$ has only one accepting configuration and one rejecting configuration. Write s_{init},t_{acc} and t_{rej} for the start configuration, accepting configuration and rejecting configuration of G, respectively. By the above, we know that $\mathbf{A} \in \mathcal{K}$ if and only if the number $D := \#\mathrm{Path}_G(s_{\mathrm{init}}, t_{\mathrm{acc}}) - \#\mathrm{Path}_G(s_{\mathrm{init}}, t_{\mathrm{rej}})$ is zero.

By Lemma 5.21, there is a first-order reduction from the problem of deciding if D is zero to the problem of determining whether a square integer matrix is singular over $\mathbb{Q}$ (that is, has determinant zero). This, in turn, can be first-order reduced to the problem of checking whether a square matrix has full rank over $\mathbb{Q}$. Since the logic $\mathrm{FOR}_{\mathbb{Q}}$ is closed under first-order reductions, the statement of the lemma now follows. □

Proof of Theorem 5.17. The proof of this theorem now follows directly from Lemma 5.20 and by combining Lemma 5.22 with the fact that queries in L can be defined in first-order logic on ordered structures. □

Chapter 6

Ehrenfeucht-Fraïssé games for rank logics

In order to analyse the expressive power of rank logics over finite structures, it is important to develop methods for proving non-definability. In this context, the restriction to finite structures means that many of the classical tools of model theory, such as the compactness theorem, are not available. Instead, we consider extensions of pebble games—variations of Ehrenfeucht-Fraïssé games for first-order logic—which have assumed a central role in the study of both infinitary and fixed-point logics.

A *pebble game* is a two-player model-comparison game where each player has a finite number of tokens ('pebbles') for placing on the game board. Intuitively, the finite collection of tokens each player is equipped with corresponds with the finite supply of variables that can be used to construct sentences of the corresponding logic. Pebble games were essentially described by Barwise [5] though versions were later independently presented by Immerman [44] and Poizat [60]. The k-pebble game can be shown to characterise definability in k-variable infinitary logic ($\mathcal{L}^k$). This correspondence gives a purely combinatorial game method for proving inexpressibility results for $\mathcal{L}^k$ in general and IFP in particular. Immerman and Lander [46] and Hella [37] later introduced separate versions of the k-pebble game for analysing the expressiveness of k-variable infinitary counting logic ($\mathcal{C}^k$) over finite models.

In this chapter we give a game characterisation of finite-variable infinitary logic with operators for defining matrix rank ($\mathcal{R}^k_{p;m}$). This gives us a game-based method for proving lower bounds (inexpressibility results) for FOR and IFPR. The game protocol that we introduce is based on partitioning the game board into a number of disjoint regions, according to some linear-algebraic criteria, which then limits the possible placement of pebbles on the board. This method of partitioning the game board turns out to be quite flexible and can be used to give a game description of finite-variable infinitary logic equipped with *any* set of generalised quantifiers. To give some examples of this approach, we will conclude the chapter by describing new partition-based games suitable for $\mathcal{L}^k$ and $\mathcal{C}^k$.

The remainder of this chapter is divided into three main sections. In §6.1 we give an overview of standard pebble games and some of their variations (counting and bijection games) and describe their relationship to fixed-point and infinitary logics with and without counting. In §6.2 we introduce a new model-comparison game, based on set partitions, that charac-

terises expressivity in logics that can define matrix rank. Finally, in §6.3 we indicate how the partition-based design can be used to obtain games that characterise definability in $\mathcal{L}^k(\mathbf{Q})$ for any set of generalised quantifiers $\mathbf{Q}$. To illustrate this idea, we finish our discussion by showing how new games for infinitary logic (with and without counting) can be obtained by putting certain restrictions on the protocol of this general partition game.

6.1 Pebble games for $\mathcal{L}^k$ and $\mathcal{C}^k$

Combinatorial games in model theory invariably involve comparing a pair of game positions over one or more structures. In order to formally compare game positions, we need the following definition.

Definition 6.1 (Partial isomorphism)**.** Let $\mathbf{A}$ and $\mathbf{B}$ be structures over the same vocabulary τ. A partial map $f : U(\mathbf{A}) \to U(\mathbf{B})$ is a *partial isomorphism* from $\mathbf{A}$ to $\mathbf{B}$ if

- f is injective;
- for every relation symbol $R \in \tau$ of arity k and all $a_1, \ldots, a_k \in U(\mathbf{A})$:

$$(a_1, \ldots, a_k) \in R^{\mathbf{A}} \Leftrightarrow (f(a_1), \ldots, f(a_k)) \in R^{\mathbf{B}};$$

- for every constant symbol $c \in \tau$: $c^{\mathbf{A}} \in \mathrm{dom}(f)$ and $f(c^{\mathbf{A}}) = c^{\mathbf{B}}$,

where we write $\mathrm{dom}(f) \subseteq U(\mathbf{A})$ for the domain of f. ■

Definability in k-variable infinitary logic is elegantly characterised in terms of two-player games based on a game style originally developed by Ehrenfeucht and Fraïssé [27, 24]. These games were essentially given by Barwise [5] though versions were also independently presented by Immerman [44] and Poizat [60]. The game board of the *k-pebble game* consists of two structures $\mathbf{A}$ and $\mathbf{B}$ over the same vocabulary and k pebbles for each of the two structures, labelled $1, \ldots, k$. The game has two players, Spoiler and Duplicator. At each round of the game, the following takes place.

1. Spoiler picks up a pebble in one of the structures (either an unused pebble or one that is already on the board) and places it on an element of the corresponding structure. For instance he[1] might take the pebble labelled by i in $\mathbf{B}$ and place it on an element of $\mathbf{B}$.

2. Duplicator must respond by placing the matching pebble in the opposite structure. In the above example, she must place the pebble labelled by i on an element of $\mathbf{A}$.

Assume at the end of the round that r pebbles have been placed and let $\{(a_i, b_i) \mid 1 \leq i \leq r\} \subseteq U(\mathbf{A}) \times U(\mathbf{B})$ denote the r pairs of pebbled elements, such that for each i the label of the pebble on element a_i is the same as the label of the pebble on element b_i. If the partial map $f : U(\mathbf{A}) \to U(\mathbf{B})$ given by

$$f := \{(a_i, b_i) \mid 1 \leq i \leq r\} \cup \{(c^{\mathbf{A}}, c^{\mathbf{B}}) \mid c \in \tau \text{ a constant}\}$$

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

is not a partial isomorphism, then Spoiler has won the game; otherwise it can continue for another round. We say that Duplicator has a winning strategy in the k-pebble game if she can play the game forever, maintaining a partial isomorphism at the end of each round.

We also consider the situation where the game starts with some of the pebbles initially placed on the game board. Formally, we refer to a placement of pebbles over one of the structures as a *position*. If $\vec{a}$ and $\vec{b}$ are r-tuples of elements from $U(\mathbf{A})$ and $U(\mathbf{B})$ respectively, $r \leq k$, then the game starting with positions $(\mathbf{A}, \vec{a})$ and $(\mathbf{B}, \vec{b})$ is played as above, except that pebbles $1, \ldots, r$ in $\mathbf{A}$ are initially placed on the elements $a_1, \ldots, a_r$ of $\vec{a}$ and pebbles $1, \ldots, r$ in $\mathbf{B}$ are initially placed on the elements $b_1, \ldots, b_r$ of $\vec{b}$. We will focus on this variant of the game in the following, with the understanding that by taking $r = 0$ we recover the situation where all the pebbles are initially off the game board. The result that links the k-pebble game with definability in $\mathcal{L}^k$ is the following.

Theorem 6.2. *Duplicator has a winning strategy in the k-pebble game starting with positions $(\mathbf{A}, \vec{a})$ and $(\mathbf{B}, \vec{b})$ if and only if $(\mathbf{A}, \vec{a}) \equiv^{\mathcal{L}^k} (\mathbf{B}, \vec{b})$.* □

One direction of the above is easy to show. That is, given a formula φ with k variables that distinguishes the pair $(\mathbf{A}, \vec{a})$, $(\mathbf{B}, \vec{b})$, it is straightforward to construct a finite-round winning strategy for Spoiler in the k-pebble game. This shows that the equivalence defined by the game is no coarser than that defined by the logic. The other direction, which would show that the equivalence is also no finer, requires a more careful argument; for details, see e.g. Ebbinghaus and Flum [23].

While the k-pebble game gives a complete characterisation of the infinitary logic $\mathcal{L}^k$, it also proves useful for analysing the expressive power of fixed-point logic. This is illustrated by Theorem 2.13, which states that any sentence of IFP is equivalent to one of $\mathcal{L}^\omega$. In particular, for each sentence φ of IFP there is a k such that the models of φ are invariant under the equivalence relation $\equiv^{\mathcal{L}^k}$. Hence we obtain the following game-based method for proving non-definability of queries in IFP:

> To show that a property P of finite structures is not definable in IFP, it suffices to show that for each $k < \omega$ there is a pair of structures $\mathbf{A}_k$ and $\mathbf{B}_k$ for which it holds that
>
> 1. $\mathbf{A}_k$ has property P but $\mathbf{B}_k$ does not; and
> 2. Duplicator has a winning strategy in the k-pebble game on $\mathbf{A}_k$ and $\mathbf{B}_k$. ■

We now turn our attention to infinitary logic with counting quantifiers. The relations $\equiv^{\mathcal{C}^k}$ were first given a game characterisation by Immerman and Lander [46]. This is a pebble game as before, played on a pair of structures $\mathbf{A}$ and $\mathbf{B}$, each with k pebbles labelled $1, \ldots, k$. In each round of the *k-pebble cardinality game* the following takes place:

1. Spoiler chooses a pebble label $i \in [k]$ and picks a subset of the universe of one of the two structures (say $X \subseteq U(\mathbf{B})$).
2. Duplicator must respond by choosing a subset of the universe of the other structure (say $Y \subseteq U(\mathbf{A})$) *of the same cardinality*.

3. Spoiler then places the pebble with label i on an element of Y and Duplicator must respond by placing the matching pebble in the opposite structure on an element of X.

This completes one round in the game. If, at any stage, the partial map from $\mathbf{A}$ to $\mathbf{B}$ defined by the pebbled positions (plus constants) is not a partial isomorphism, then Spoiler has won the game. Otherwise it can continue for another round. We say that Duplicator has a winning strategy in the game on $\mathbf{A}$ and $\mathbf{B}$ if she can ensure that it can be played forever. We also consider the case when the game starts with some of the pebbles initially placed on elements of the two structures, just like before. Immerman and Lander [46] prove the following equivalence.

Theorem 6.3. *Duplicator has a winning strategy in the k-pebble cardinality game starting with positions $(\mathbf{A}, \vec{a})$ and $(\mathbf{B}, \vec{b})$ if and only if $(\mathbf{A}, \vec{a}) \equiv^{\mathcal{C}^k} (\mathbf{B}, \vec{b})$.* □

An alternative game characterisation of the equivalence $\equiv^{\mathcal{C}^k}$ was given by Hella [37], who describes what we call a *k-pebble bijection game*. As before, the game is played on structures $\mathbf{A}$ and $\mathbf{B}$, each with k pebbles labelled $1, \ldots, k$, by Spoiler and Duplicator. If $\|\mathbf{A}\| \neq \|\mathbf{B}\|$, Spoiler wins the game immediately. Otherwise, each round of the game proceeds as follows:

1. Spoiler picks up a pebble from $\mathbf{A}$ and the matching pebble from $\mathbf{B}$.
2. Duplicator has to respond by choosing a bijection $h : U(\mathbf{A}) \to U(\mathbf{B})$.
3. Spoiler then places the pebble chosen from $\mathbf{A}$ on some element $a \in U(\mathbf{A})$ and places the matching pebble from $\mathbf{B}$ on $h(a)$.

This completes one round in the game. If, after this round, the partial map from $\mathbf{A}$ to $\mathbf{B}$ defined by the pebbled positions (plus constants) is not a partial isomorphism, then Spoiler has won the game. Otherwise it can continue for another round.

Observe that in any winning strategy for Duplicator, it is implicit that at every round in the game, the bijection $h : U(\mathbf{A}) \to U(\mathbf{B})$ has to respect the partial map defined by the currently pebbled elements *excluding* the two pebbles that were just picked up by Spoiler. That is, suppose at some round in the game that the tuples $\vec{a}$ and $\vec{b}$ describe the current pebble positions over $\mathbf{A}$ and $\mathbf{B}$, respectively. Then the mapping h given by Duplicator in response to Spoiler choosing pebbles with label i must satisfy $h(a_j) = b_j$ for all $j \neq i$. To see this, suppose instead that there is some $j \in [k]$, $j \neq i$, such that $h(a_j) \neq b_j$. Then Spoiler can immediately win the game in response to this choice of bijection, by placing the pebble labelled i on the element a_j and the matching pebble in $\mathbf{B}$ on $h(a_j)$. The resulting game positions will have $a_i = a_j$ but $b_i \neq b_j$. Hence, the mapping $\mathbf{A} \to \mathbf{B}$ defined by the pebbled elements is not a partial isomorphism, as it violates equality.

At first glance, compared to the Immerman-Lander cardinality game, the bijection game appears to be weighted in favour of Spoiler, as Duplicator has to come up with a response to every subset that Spoiler might possibly choose. However, as Hella shows, Duplicator still has a winning strategy as long as $\mathbf{A}$ and $\mathbf{B}$ are $\mathcal{C}^k$-equivalent. Thus the game has the same discriminating power as the Immerman-Lander game. However, it is often easier to express winning strategies in Hella's game, which is useful when presenting non-trivial game proofs.

6.2 Pebble game for $\mathcal{R}^k_{p;m}$

In order to delimit the expressive power of rank logics, we would like to have a characterisation of the logics in terms of suitable pebble games. One way to define a pebble game that characterises equivalence in $\equiv^{\mathcal{R}^k_{p;m}}$ is to extend the idea behind the Immerman-Lander counting game. For instance, for the case of quantifiers $\mathrm{rk}^{\geq l}_2$ of arity two, one could define a game played between Spoiler and Duplicator using k pairs of pebbles as follows. At each round, Spoiler chooses two pebble labels i and j. He then picks up a set of pairs of elements from the universe of one of the structures (say Spoiler chooses $X \subseteq U(\mathbf{B}) \times U(\mathbf{B})$) and Duplicator must respond with a similar set from the other structure (say $Y \subseteq U(\mathbf{A}) \times U(\mathbf{A})$) such that the $(0,1)$-matrices induced by these sets have the *same rank* over GF_2. Spoiler then places the two pebbles labelled i and j in $\mathbf{A}$ on an element of Y and Duplicator must respond by placing the matching pebbles in $\mathbf{B}$ on an element of X. It is possible to show that if Duplicator has a winning strategy in this game, then the two structures cannot be distinguished by any formula of $\mathcal{R}^k_{2;m}$ of arity at most two. But, for a converse, it seems that one has to restrict Spoiler to play on *definable* sets, which seems a rather unsatisfactory solution.

Another possibility to consider is whether the Hella bijection games can be modified, perhaps by replacing bijections with *invertible linear maps*. This seems natural, considering that these maps are exactly the ones that preserve dimension of vector spaces, just as bijections preserve cardinality of sets. It is straightforward to show that a winning strategy for Duplicator in such a game is sufficient to ensure that the underlying positions cannot be distinguished in infinitary rank logic, but again it is not clear that this is necessary.

In this section we describe a game design which is not based on either choosing arbitrary sets or picking invertible linear maps. Instead, the pebble games we consider are based on *partitioning* the game board into a number of disjoint regions, according to some linear-algebraic criteria, which then limits the possible placement of pebbles on the board. It turns out that this approach gives a complete characterisation of $\equiv^{\mathcal{R}^k_{p;m}}$ that is not itself based on the notion of definability, which is exactly what we are aiming for.

Before describing these games, we first establish some notation. Let X be a finite set and let $\mathbf{P}$ be a partition of X. That is, $\mathbf{P}$ is a collection of non-empty and mutually disjoint subsets of X (called *blocks*) whose union is X. For $x \in X$, we write $[\![x]\!]_{\mathbf{P}}$ to denote the $\mathbf{P}$-block containing x. For the next definition, recall that for prime p and sets I and J, we identify functions $I \times J \to [0, p-1]$ with matrices over GF_p, as discussed in Chapter 4.

Definition 6.4 (Matrices defined by set partitions)**.** Consider finite sets I and J. Let $\mathbf{P}$ be a set partition of $I \times J$ and let $\gamma : \mathbf{P} \to [0, p-1]$ be a map, with p prime. Then we write $M^{\mathbf{P}}_{\gamma}$ to denote the $I \times J$ matrix over GF_p defined for all $i \in I$ and $j \in J$ by

$$M^{\mathbf{P}}_{\gamma} : (i,j) \mapsto \gamma([\![(i,j)]\!]_{\mathbf{P}}) \in [0, p-1].$$

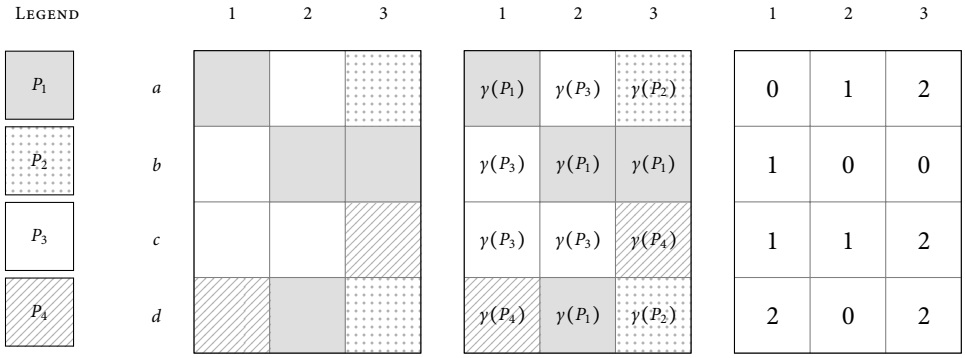
■

In this definition, the map γ can be seen as a labelling of the blocks in $\mathbf{P}$ with elements of GF_p. This view is further illustrated with the following example.

Example 6.5. Consider sets $I = \{a, b, c, d\}$ and $J = \{1, 2, 3\}$. Let

$$P_1 = \{(a,1), (b,2), (b,3), (d,2)\}$$
$$P_2 = \{(a,3), (d,3)\}$$
$$P_3 = \{(a,2), (b,1), (c,1)(c,2)\}$$
$$P_4 = \{(c,3), (d,1)\}$$

be subsets of $I \times J$ and put $\mathbf{P} = \{P_1, P_2, P_3, P_4\}$. The partition $\mathbf{P}$ can be visualised in block form in figure (a), below. Now take $p = 5$ and consider a labelling $\gamma : \mathbf{P} \to [0, 4]$ of the partition $\mathbf{P}$ defined by $P_1 \mapsto 0$, $P_2 \mapsto 2$, $P_3 \mapsto 1$, $P_4 \mapsto 2$. Figure (b) illustrates the situation where the labelling γ is applied to the partition $\mathbf{P}$. Finally, by evaluating $\gamma(P)$ for every block $P \in \mathbf{P}$, we obtain the matrix $M^{\mathbf{P}}_{\gamma}$ over GF_p displayed in figure (c). Here, we have $\operatorname{rank}(M^{\mathbf{P}}_{\gamma}) = 3$.



(a) Partition $\mathbf{P}$ of of $I \times J$. **(b)** The labeling γ applied to $\mathbf{P}$. **(c)** The matrix $M^{\mathbf{P}}_{\gamma}$.

■

Definition 6.6 (k-pebble m-ary rank-partition game)**.** Let k, m and p be positive integers with $m \leq k$ and p prime. The game board of the *k-pebble m-ary rank-partition game over* GF_p consists of two structures $\mathbf{A}$ and $\mathbf{B}$ of the same vocabulary, each with k pebbles labelled $1, \ldots, k$. There are two players, Spoiler and Duplicator, as before. At the beginning of each round, Spoiler chooses two positive integers s and t with $s + t = m$. The remainder of the round is as follows.

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

   for which it holds that for all labellings $\gamma : \mathbf{P} \to [0, p-1]$,

$$\operatorname{rank}(M^{\mathbf{P}}_{\gamma}) = \operatorname{rank}(M^{\mathbf{Q}}_{\gamma \circ f^{-1}}). \qquad (*)$$

Here the composite map $\gamma \circ f^{-1} : \mathbf{Q} \to [0, p-1]$ denotes a labelling of $\mathbf{Q}$ and $M^{\mathbf{P}}_{\gamma}$ and $M^{\mathbf{Q}}_{\gamma\circ f^{-1}}$ are interpreted as $U(\mathbf{A})^s \times U(\mathbf{A})^t$ and $U(\mathbf{B})^s \times U(\mathbf{B})^t$ matrices over GF_p, respectively.

3. Spoiler next picks a block $P \in \mathbf{P}$ and places the m chosen pebbles from $\mathbf{A}$ on the elements of some tuple in P (in the order they were chosen earlier) and the corresponding m 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 has won the game; otherwise it can continue for another round. ■

A variant of the game, whereby $r \leq k$ of the pebbles are initially placed on elements of each structure, can be defined similar to before. The following theorem relates definability in $\mathcal{R}^k_{p;m}$ with a winning strategy for Duplicator in the rank-partition game.

Theorem 6.7. *Duplicator has a winning strategy in the k-pebble m-ary rank-partition game over GF_p starting with positions $(\mathbf{A}, \vec{a})$ and $(\mathbf{B}, \vec{b})$ if and only if $(\mathbf{A}, \vec{a}) \equiv^{\mathcal{R}^k_{p;m}} (\mathbf{B}, \vec{b})$.*

By considering initial positions $(\mathbf{A}, \vec{a})$ and $(\mathbf{B}, \vec{b})$ where $\vec{a}$ and $\vec{b}$ are empty tuples, we get the following corollary.

Corollary 6.8. *Duplicator has a winning strategy in the k-pebble m-ary rank-partition game over GF_p on $\mathbf{A}$ and $\mathbf{B}$ if and only if $\mathbf{A} \equiv^{\mathcal{R}^k_{p;m}} \mathbf{B}$.*

Compared with the pebble games we saw earlier, it requires much more effort to describe a winning strategy for Duplicator in the k-pebble rank-partition game. Based only on the pebbles chosen by Spoiler at the beginning of a round, Duplicator has to partition the two sides of the game board in a way that both satisfies the rank condition $(*)$ and which gives a satisfying response to *any* placement of pebbles by Spoiler in the subsequent move. Note in particular that once Duplicator has specified the partitions, she has no further input for the remainder of that game round. Also note that it is implicit in the definition of the game that at every round, the bijection f chosen by Duplicator has to respect the current pebble positions, excluding the m pairs of pebbles picked up earlier by Spoiler. This follows an argument similar to the one we gave in our discussion of the bijection game in §6.1.

From the viewpoint of finite model theory, the interest in studying the infinitary logics $\mathcal{R}^k_{p;m}$ is mainly to analyse the expressive power of fixed-point logics with operators for matrix rank. In this context, the correspondence between $\equiv^{\mathcal{R}^k_{p;m}}$ and the k-pebble rank-partition game gives us a game-based method for proving non-definability of queries in $\mathrm{IFPR}_{p;m}$. This proof method is however complicated by the fact that we need to consider two additional parameters—the prime characteristic p and the quantifier arity m—in addition to the number of variables k employed in the game:

> To show that a property P of finite structures is not definable in IFPR, it suffices to show for each $k < \omega$, $m \leq k$ and prime p that there is a pair of structures $\mathbf{A}_{k,m,p}$ and $\mathbf{B}_{k,m,p}$ for which it holds that
>
> 1. $\mathbf{A}_{k,m,p}$ has property P but $\mathbf{B}_{k,m,p}$ does not; and

2. Duplicator has a winning strategy in the k-pebble m-ary rank-partition game over GF_p game on $\mathbf{A}_{k,m,p}$ and $\mathbf{B}_{k,m,p}$.

Similarly, to show that a property P of finite structures is not definable in IFPR_p for a prime p, it suffices to follow the procedure above with p fixed. ■

For the discussion of game strategies, we will need to formally define the quantifier rank of infinitary rank formulae.

Definition 6.9 (Quantifier rank)**.** The quantifier rank of a formula in $\mathcal{R}^k_{p;m}$ is an ordinal-valued function qr that is defined inductively as follows:

- $\mathrm{qr}(\varphi) = 0$ for atomic φ;
- $\mathrm{qr}(\neg\varphi) = \mathrm{qr}(\varphi)$;
- $\mathrm{qr}(\bigvee \Phi) = \mathrm{qr}(\bigwedge \Phi) = \sup\{\mathrm{qr}(\varphi) \mid \varphi \in \Phi\}$;
- $\mathrm{qr}(\exists x\, \varphi) = \mathrm{qr}(\forall x\, \varphi) = \mathrm{qr}(\varphi) + 1$;
- $\mathrm{qr}(\mathrm{rk}^{\geq i}_p(\vec{x}, \vec{y}).(\varphi_1, \ldots, \varphi_{p-1})) = \sup\{\mathrm{qr}(\varphi_i) \mid i \in [p-1]\} + 1$. ■

We also need the following lemma on definability of types in $\mathcal{R}^k_{p;m}$, which is a direct corollary of Lemma 1.33 in [58].

Lemma 6.10. *Let $k, m, p \geq 2$, with p prime, and consider a vocabulary τ. Then for all $\alpha \in \mathrm{Tp}(\mathcal{R}^k_{p;m}; \tau, k)$ there is a formula $\varphi_\alpha(x_1, \ldots, x_k) \in \mathcal{R}^k_{p;m}[\tau]$ such that for all $(\mathbf{A}, \vec{a}) \in \mathrm{fin}[\tau; k]$: $\mathrm{tp}(\mathcal{R}^k_{p;m}; \mathbf{A}, \vec{a}) = \alpha \Leftrightarrow \mathbf{A} \vDash \varphi_\alpha[\vec{a}]$.* □

The remainder of this section is devoted to proving Theorem 6.7. Before we can give the proof, we need to introduce some new notation. Throughout, let τ be a vocabulary and L a logic. To simplify our notation (and the proof), we will consider only positions $(\mathbf{A}, \vec{a})$ and $(\mathbf{B}, \vec{b})$ with $\|\vec{a}\| = \|\vec{b}\| = k$; that is, positions where all the pebbles are initially placed on the board. The argument for the case when the tuples $\vec{a}$ and $\vec{b}$ 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[2]. This has the effect of making the proof non-uniform, without actually providing any new insight.

Definition 6.11. Let $\varphi(\vec{x})$ be a formula of $\mathrm{L}[\tau]$, $\vec{x}$ a k-tuple of variables, and consider an m-tuple $\vec{i} = (i_1, \ldots, i_m) \in [k]^m$ of distinct integers, $m \leq k$. The tuple $\vec{i}$ can be seen to index a

[2]Note that it is possible to obtain a proof for $\|\vec{a}\| = \|\vec{b}\| = r \in [k-1]$ as a direct corollary of the situation when $\|\vec{a}\| = \|\vec{b}\| = k$. In this case, given r-tuples $\vec{a}$ and $\vec{b}$, one would consider the game with pebble positions $\vec{a}'$ and $\vec{b}'$, where the k-tuple $\vec{a}'$ is obtained from $\vec{a}$ 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 $\vec{b}'$. 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 [23] for an application of this idea.

sub-tuple of variables from $\vec{x}$. Then for each finite τ-structure $\mathbf{A}$ and tuple $\vec{a} \in U(\mathbf{A})^k$, define a relation

$$\varphi[\vec{a}]^{\mathbf{A}} \restriction \vec{i} := \{(c_1, \ldots, c_m) \in U(\mathbf{A})^m \mid \mathbf{A} \vDash \varphi[\vec{a}\tfrac{c_1}{i_1} \cdots \tfrac{c_m}{i_m}]\} \subseteq U(\mathbf{A})^m.$$

■

In §4.1.1 we considered matrices defined by tuples of formulae (viz. Definition 4.2). Here we extend that notation to allow some of the named variables to be interpreted by a fixed assignment.

Definition 6.12. Let k, p, m, s and t be positive integers, with $m \leq k$, p prime, and $s + t = m$. Let $\vec{x} = (x_1, \ldots, x_k)$ and consider tuples $\vec{i} = (i_1, \ldots, i_s) \in [k]^s$ and $\vec{j} = (j_1, \ldots, j_t) \in [k]^t$ of distinct integers indexing variables in $\vec{x}$.

1. Consider a $\mathrm{L}[\tau]$-formula $\varphi(\vec{x})$. Then for each finite τ-structure $\mathbf{A}$ and $\vec{a} \in U(\mathbf{A})^k$, write
$$\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\varphi, \mathbf{A}, \vec{a})_p : U(\mathbf{A})^s \times U(\mathbf{A})^t \to \mathsf{GF}_p$$
to denote the $(0,1)$-matrix over GF_p defined by
$$\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\varphi, \mathbf{A}, \vec{a})_p : (\vec{b}, \vec{c}) \mapsto \begin{cases} 1 & \text{if } (\vec{b}, \vec{c}) \in \varphi[\vec{a}]^{\mathbf{A}} \restriction \vec{i}\vec{j}, \\ 0 & \text{otherwise} \end{cases},$$
for $\vec{b} \in U(\mathbf{A})^s$ and $\vec{c} \in U(\mathbf{A})^t$.

2. Consider a tuple $\Phi = (\varphi_1, \ldots, \varphi_{p-1})$ of $\mathrm{L}[\tau]$-formulae and suppose that all the formulae occurring in Φ have free variables amongst $\vec{x}$. Then for each finite τ-structure $\mathbf{A}$ and $\vec{a} \in U(\mathbf{A})^k$, write
$$\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\Phi, \mathbf{A}, \vec{a})_p : U(\mathbf{A})^s \times U(\mathbf{A})^t \to \mathsf{GF}_p$$
to denote the matrix over GF_p defined by
$$\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\Phi, \mathbf{A}, \vec{a})_p := \sum_{i=1}^{p-1} i \cdot \mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\varphi_i, \mathbf{A}, \vec{a})_p \pmod{p}.$$
That is, $\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\Phi, \mathbf{A}, \vec{a})_p$ is a linear combination of $(0,1)$-matrices $\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\varphi_i, \mathbf{A}, \vec{a})_p$, with scalar coefficients defined by the position of each formula φ_i in the tuple Φ.

■

Lemma 6.13. *Suppose* $(\mathbf{A}, \vec{a}) \equiv^{\mathcal{R}^k_{p;m}} (\mathbf{B}, \vec{b})$ *and let* $\vec{x}$ *be a tuple of variables whose length matches that of* $\vec{a}$ *and* $\vec{b}$*. Let* s *and* t *be positive integers with* $s+t = m$*. Then for all* $\varphi_1, \ldots, \varphi_{p-1} \in \mathcal{R}^k_{p;m}$*, with* $\mathrm{free}(\varphi_i) \subseteq \vec{x}$*, and all tuples* $\vec{i} \in [k]^s$*,* $\vec{j} \in [k]^t$ *with* $\|\vec{i} \cup \vec{j}\| = m$*, it holds that*

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

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

Proof. Let $(\mathbf{A}, \vec{a}) \in \mathrm{fin}[\tau; k]$. Then for all tuples $\Phi = (\varphi_1, \ldots, \varphi_{p-1})$ of $\mathcal{R}^k_{p;m}$-formulae, with $\mathrm{free}(\varphi_i) \subseteq \vec{x}$, and all $\vec{i} \in [k]^s$, $\vec{j} \in [k]^t$ with $\|\vec{i} \cup \vec{j}\| = m$, the formula

$$\mathrm{rk}_p^{=l}((x_{i_1}, \ldots, x_{i_s}), (x_{j_1}, \ldots, x_{j_s})).(\varphi_1, \ldots, \varphi_{p-1})$$

is in $\mathrm{tp}(\mathcal{R}^k_{p;m}; \mathbf{A}, \vec{a})$ exactly for the number $l := \mathrm{rank}(\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\Phi, \mathbf{A}, \vec{a})_p)$. The statement of the lemma now follows by considering that $\mathrm{tp}(\mathcal{R}^k_{p;m}; \mathbf{A}, \vec{a}) = \mathrm{tp}(\mathcal{R}^k_{p;m}; \mathbf{B}, \vec{b})$. □

We are now ready to prove Theorem 6.7. The proof is given in two separate lemmas, one for each implication.

Lemma 6.14. *If* $(\mathbf{A}, \vec{a}) \not\equiv^{\mathcal{R}^k_{p;m}} (\mathbf{B}, \vec{b})$ *then Spoiler has a winning strategy in the k-pebble m-ary rank-partition game over* GF_p *starting with positions* $(\mathbf{A}, \vec{a})$ *and* $(\mathbf{B}, \vec{b})$.

Proof. If $(\mathbf{A}, \vec{a}) \not\equiv^{\mathcal{R}^k_{p;m}} (\mathbf{B}, \vec{b})$ then there is a formula $\varphi(\vec{x}) \in \mathcal{R}^k_{p;m}$ of quantifier rank ζ such that $\mathbf{A} \models \varphi[\vec{a}]$ but $\mathbf{B} \models \neg\varphi[\vec{b}]$. If $\zeta = 0$ then the mapping $\mathbf{A} \to \mathbf{B}$ defined by the pebbled elements $\vec{a} \mapsto \vec{b}$ is not a partial isomorphism and Spoiler has won the game. For the inductive step, suppose that $\zeta > 0$. We show that Spoiler can in one round force the game into positions $(\mathbf{A}, \vec{a}')$ and $(\mathbf{B}, \vec{b}')$ where $(\mathbf{A}, \vec{a}')$ and $(\mathbf{B}, \vec{b}')$ can be distinguished by a formula of quantifier rank $\zeta' < \zeta$. By a repeated application of such moves we get a strictly decreasing sequence of ordinal-valued quantifier ranks, which must have finite length. This gives Spoiler a strategy to win the game in a finite number of steps, as claimed.

We can assume without loss of generality that φ is of the form

$$\mathrm{rk}_p^{=l}((x_{i_1}, \ldots, x_{i_s}), (x_{j_1}, \ldots, x_{j_t})).(\varphi_1, \ldots, \varphi_{p-1})$$

for some $l \geq 0$, $s, t \geq 1$ and $s + t = m$. Other cases reduce to this one through the symmetry of the claim (we are considering an equivalence relation) and, if necessary, by replacing φ by one of its Boolean constituents. Set $\vec{i} = (i_1, \ldots, i_s)$, $\vec{j} = (j_1, \ldots, j_t)$ and $\Phi = (\varphi_1, \ldots, \varphi_{p-1})$. Then by the assumption on φ,

$$\mathrm{rank}(\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\Phi, \mathbf{A}, \vec{a})_p) \neq \mathrm{rank}(\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\Phi, \mathbf{B}, \vec{b})_p). \qquad (\dagger)$$

Spoiler now starts the round by declaring s and t and picking up the pebbles with labels $i_1, \ldots, i_s$, and $j_1, \ldots, j_t$. 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 $(*)$. Then the following claim shows that the partitions proposed by Duplicator must contain a block with tuples that disagree on one of the formulae φ_i.

Claim 2. *There is a block* $P \in \mathbf{P}$ *and tuples* $\vec{c} \in P$ *and* $\vec{d} \in f(P)$ *for which there is some formula* φ_i *in* Φ *such that*

$$\mathbf{A} \models \varphi_i[\vec{a}\tfrac{c_1}{i_1} \ldots \tfrac{c_s}{i_s}\tfrac{c_{s+1}}{j_1} \ldots \tfrac{c_{s+t}}{j_t}]$$

and

$$\mathbf{B} \models \neg\varphi_i[\vec{b}\tfrac{d_1}{i_1} \ldots \tfrac{d_s}{i_s}\tfrac{d_{s+1}}{j_1} \ldots \tfrac{d_{s+t}}{j_t}],$$

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 $\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\Phi, \mathbf{A}, \vec{a})_p$ defined over $\mathbf{A}$. By the assumption, we can find a labelling $\gamma : \mathbf{P} \to [0, p-1]$ such that

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

For instance, γ can be defined by $\gamma(P) := \sum_{i \in \iota(P)} i$ for each $P \in \mathbf{P}$. But

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

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

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

Lemma 6.15. *If* $(\mathbf{A}, \vec{a}) \equiv^{\mathcal{R}^k_{p;m}} (\mathbf{B}, \vec{b})$ *then Duplicator has a winning strategy in the* k*-pebble* m*-ary rank-partition game over* GF_p *starting with positions* $(\mathbf{A}, \vec{a})$ *and* $(\mathbf{B}, \vec{b})$.

The basic idea behind the proof of this lemma is as follows. At every round in the game, the strategy of the Duplicator is to define partitions $\mathbf{P}$ and $\mathbf{Q}$ by grouping together in each block of a partition all the elements realising the same $\equiv^{\mathcal{R}^k_{p;m}}$-type (with respect to the current game positions). The bijection $f : \mathbf{P} \to \mathbf{Q}$ is similarly defined by pairing together blocks $P \in \mathbf{P}$ and $Q \in \mathbf{Q}$ whose elements all realise the same $\equiv^{\mathcal{R}^k_{p;m}}$-type. If Duplicator can play in this manner, she can ensure that any choices made by Spoiler are restricted to blocks which do not distinguish the two structures.

Proof. Assume $(\mathbf{A}, \vec{a}) \equiv^{\mathcal{R}^k_{p;m}} (\mathbf{B}, \vec{b})$. We show that Duplicator has a strategy to maintain $\equiv^{\mathcal{R}^k_{p;m}}$-equivalence of game positions. In other words, we show that no matter which pebbles Spoiler chooses in the next round, Duplicator can respond with partitions that satisfy the requirements of the game and which ensure that the resulting game positions will be $\equiv^{\mathcal{R}^k_{p;m}}$-equivalent. Throughout, we write $\vec{x} = (x_1, \ldots, x_k)$ to denote a k-tuple of distinct variables.

Now suppose that Spoiler starts a round by choosing a pair of integers s and t with $s + t = m$ and picking up pebbles labelled $i_1, \ldots, i_s, j_1, \ldots, j_t$, in that sequence. Write $\vec{i} = (i_1, \ldots, i_s)$, $\vec{j} = (j_1, \ldots, j_t)$ and $\vec{l} = \vec{i}\vec{j}$ for short. For each $\alpha \in \mathrm{Tp}(\mathcal{R}^k_{p;m}; \tau, k)$, let $\varphi_\alpha(\vec{x})$ be the formula of

$\mathcal{R}^k_{p;m}$ that isolates α over finite τ-structures (by Lemma 6.10). That is, $\varphi_\alpha(\vec{x})$ will be realised in $\mathbf{A}$ by a tuple $\vec{t}$ if and only if $\mathrm{tp}(\mathcal{R}^k_{p;m};\mathbf{A},\vec{t}) = \alpha$, and similarly for tuples over $\mathbf{B}$. Now define

$$P_\alpha := \varphi_\alpha[\vec{a}]^{\mathbf{A}} \restriction \vec{l} = \{(c_1,\dots,c_m) \in U(\mathbf{A})^m \mid \mathrm{tp}(\mathcal{R}^k_{p;m};\mathbf{A},\vec{a}\tfrac{c_1}{l_1}\cdots\tfrac{c_m}{l_m}) = \alpha\} \subseteq U(\mathbf{A})^m,$$
$$Q_\alpha := \varphi_\alpha[\vec{b}]^{\mathbf{B}} \restriction \vec{l} = \{(d_1,\dots,d_m) \in U(\mathbf{B})^m \mid \mathrm{tp}(\mathcal{R}^k_{p;m};\mathbf{B},\vec{b}\tfrac{d_1}{l_1}\cdots\tfrac{d_m}{l_m}) = \alpha\} \subseteq U(\mathbf{B})^m.$$

That is, each P_α consists of all m-tuples that, when used to replace elements of $\vec{a}$ according to the index pattern $\vec{l}$, 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 partitions

$$\mathbf{P} := \{P_\alpha \mid \alpha \in \mathrm{Tp}(\mathcal{R}^k_{p;m};\tau,k) \text{ and } P_\alpha \neq \varnothing\},$$
$$\mathbf{Q} := \{Q_\alpha \mid \alpha \in \mathrm{Tp}(\mathcal{R}^k_{p;m};\tau,k) \text{ and } Q_\alpha \neq \varnothing\},$$

and a mapping $f : \mathbf{P} \to \mathbf{Q}$ 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 $U(\mathbf{A})^s \times U(\mathbf{A})^t$ and $U(\mathbf{B})^s \times U(\mathbf{B})^t$, respectively (just observe that each tuple of elements realises only one type). It remains to be shown that $\mathbf{P}$, $\mathbf{Q}$ and f satisfy the requirements $(*)$ of the game.

Claim 3. *The mapping f is a bijection.*

Proof of claim. For all types α, it holds that

$$P_\alpha = \varnothing \Leftrightarrow \mathrm{rk}^{=0}_p((x_{i_1},\dots,x_{i_s}),(x_{j_1},\dots,x_{j_t})).(\varphi_\alpha) \in \mathrm{tp}(\mathcal{R}^k_{p;m};\mathbf{A},\vec{a}),$$

where φ_α is defined as the conjunction of all formulae in α, as before. Here the formula $\mathrm{rk}^{=0}_p((x_{i_1},\dots,x_{i_s}),(x_{j_1},\dots,x_{j_t})).(\varphi_\alpha)$ asserts that the number of distinct tuples $(x_{l_1},\dots,x_{l_m})$ that realise φ_α over $(\mathbf{A},\vec{a})$ is nil. As $\mathrm{tp}(\mathcal{R}^k_{p;m};\mathbf{A},\vec{a}) = \mathrm{tp}(\mathcal{R}^k_{p;m};\mathbf{B},\vec{b})$, it follows that the two partitions $\mathbf{P}$ and $\mathbf{Q}$ have the same cardinality, and the claim follows. □

Claim 4. *For all labellings $\gamma : \mathbf{P} \to [0,p-1]$ it holds that* $\mathrm{rank}(M^{\mathbf{P}}_\gamma) = \mathrm{rank}(M^{\mathbf{Q}}_{\gamma\circ f^{-1}})$.

Proof of claim. 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 $i \in [0,p-1]$ by γ corresponds to a class of types $\Omega_i \subseteq \mathrm{tp}(\mathcal{R}^k_{p;m};\mathbf{A},\vec{a})$, with each type $\alpha \in \Omega_i$ isolated by a formula $\varphi_\alpha \in \mathcal{R}^k_{p;m}$, as before. That is, for each type α it holds that

$$\alpha \in \Omega_i \Leftrightarrow \gamma(P_\alpha) = \gamma(\varphi_\alpha[\vec{a}]^{\mathbf{A}} \restriction \vec{l}) = i.$$

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

$$\begin{aligned} M^{\mathbf{P}}_\gamma &:= \sum_{i=1}^{p-1} i \times \Big(\sum_{\alpha\in\Omega_i} \mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\varphi_\alpha,\mathbf{A},\vec{a})_p\Big) \\ &= \sum_{i=1}^{p-1} i \times \mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\psi_i,\mathbf{A},\vec{a})_p \\ &= \mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\psi_1,\dots,\psi_{p-1},\mathbf{A},\vec{a})_p, \end{aligned}$$

and

$$\begin{aligned} M^{\mathbf{Q}}_{\gamma\circ f^{-1}} &:= \sum_{i=1}^{p-1} i \times \Big(\sum_{\alpha\in\Omega_i} \mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\varphi_\alpha, \mathbf{B}, \vec{b})_p \Big) \\ &= \sum_{i=1}^{p-1} i \times \mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\psi_i, \mathbf{B}, \vec{b})_p \\ &= \mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\psi_1, \ldots, \psi_{p-1}, \mathbf{B}, \vec{b})_p. \end{aligned}$$

By Lemma 6.13 we know that

$$\mathrm{rank}\big(\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\psi_1, \ldots, \psi_{p-1}, \mathbf{A}, \vec{a})_p\big) = \mathrm{rank}\big(\mathrm{fmat}_{\vec{x},\vec{i},\vec{j}}(\psi_1, \ldots, \psi_{p-1}, \mathbf{B}, \vec{b})_p\big).$$

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

By these claims it can be seen that for any block $P \in \mathbf{P}$, any choice of elements $(c_1, \ldots, c_m) \in P$ and $(d_1, \ldots, d_m) \in f(P)$ that Spoiler can make will result in tuples

$$\vec{a}\tfrac{c_1}{i_1}\ldots\tfrac{c_1}{i_s}\tfrac{c_{s+1}}{j_1}\ldots\tfrac{c_{s+t}}{j_t} \text{ and } \vec{b}\tfrac{d_1}{i_1}\ldots\tfrac{d_1}{i_s}\tfrac{d_{s+1}}{j_1}\ldots\tfrac{d_{s+t}}{j_t}$$

that realise the same $\mathcal{R}^k_{p;m}$-type. Hence, $\equiv^{\mathcal{R}^k_{p;m}}$-equivalence of game positions is maintained. □

Compared with the Immerman-Lander game or the Hella bijection game, it is clearly much harder for Duplicator to maintain a winning strategy in the rank-partition game. This of course corresponds with the fact that rank logics are strictly more expressive than counting logics. Generally speaking, in order to construct a winning strategy for Duplicator in the rank-partition game, the size of the blocks in the set partitions **P** and **Q** becomes a crucial measure. With larger blocks, Spoiler has more freedom in placing down pebbles at the end of the round, but the rank condition is more easily satisfied. Conversely, with smaller block size, Spoiler is given fewer options for placing down pebbles, but it becomes harder to ensure that the rank condition is fulfilled.

6.3 Pebble games for generalised quantifiers

It was shown by Dawar [15] that if there is a logic that captures PTIME, then there is such a logic obtained by adding one vectorised family of generalised quantifiers to first-order logic. A game-based method that can characterise expressibility in logics with generalised quantifiers would therefore be an important tool for studying the descriptive complexity of PTIME. Previously, there have been some attempts to define a general game template for this purpose. In [37], Hella developed the n-bijective two-player game, and showed that this game characterises exactly the expressive power of finite-variable infinitary logic extended with *all* generalised quantifiers of arity up to n. An attempt to develop a more fine-grained game argument was made by Kolaitis and Väänänen [48], who studied fixed-point and infinitary logics extended by arbitrary sets of generalised quantifiers. Their game crucially relies on Spoiler choosing only *definable* sets or relations in one of the game structures. Since the aim of the game method is to provide an alternative combinatorial view of definability, this approach is not entirely satisfactory.

In this section we address to this topic by giving an alternative game characterisation of finite-variable infinitary logic equipped with *any* set of generalised quantifiers. These games are based on the idea of using set partitions to restrict the possible moves in a pebble game, just like the rank-partition game we discussed earlier.

Definition 6.16 (k-pebble $\mathbf{Q}$-partition game)**.** Each round in the *k-pebble $\mathbf{Q}$-partition game* starts by Spoiler choosing a quantifier $Q \in \mathbf{Q}$. Write $(n_1, \ldots, n_m)$ to denote the type of Q, where each n_i is a positive integer, and let $n := \max\{n_1, \ldots, n_m\}$. The rest of the round then proceeds as follows.

1. Spoiler picks up n pebbles from $\mathbf{A}$ and the n corresponding pebbles in the same order from $\mathbf{B}$.

2. Duplicator has to respond by choosing a triple $(\mathbf{P}, \mathbf{Q}, f)$ where:

   (a) $\mathbf{P}$ is a partition of $U(\mathbf{A})^n$,

   (b) $\mathbf{Q}$ is a partition of $U(\mathbf{B})^n$ with the same number of blocks as $\mathbf{P}$, and

   (c) $f : \mathbf{P} \to \mathbf{Q}$ is a bijection.

   Together, these objects have to satisfy the condition that for all collections of blocks $S_1 \subseteq \mathbf{P}, \ldots, S_m \subseteq \mathbf{P}$, it holds that

   $$(U(\mathbf{A}); X_1, \ldots, X_m) \in Q \Leftrightarrow (U(\mathbf{B}); Y_1, \ldots, Y_m) \in Q, \qquad (\ddagger)$$

   where for each i, $X_i := \mathrm{proj}(\bigcup_{P \in S_i} P, n_i)$ and $Y_i := \mathrm{proj}(\bigcup_{P \in S_i} f(P), n_i)$ are relations of arity n_i over $U(\mathbf{A})$ and $U(\mathbf{B})$, respectively (obtained by taking the projection of a relation of arity n onto its first n_i coordinates)

3. Spoiler next picks a block $P \in \mathbf{P}$ and places the n chosen pebbles from $\mathbf{A}$ on the elements of some tuple in P (in the order they were chosen earlier) and places the corresponding n pebbles from $\mathbf{B}$ on the elements of some tuple in $f(P)$ (in the same order).

That completes one round in the k-pebble $\mathbf{Q}$-partition 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 Duplicator is unable to produce the required partitions, then Spoiler has won the game; otherwise it can continue for another round. ■

As before, we also consider the game where some of the pebbles are initially placed on the game board. The following theorem relates definability in $\mathcal{L}^k(\mathbf{Q})$ with a winning strategy in the game.

Theorem 6.17. *Duplicator has a winning strategy in the k-pebble $\mathbf{Q}$-partition game on $(\mathbf{A}, \vec{a})$ and $(\mathbf{B}, \vec{b})$ if and only if the positions $(\mathbf{A}, \vec{a})$ and $(\mathbf{B}, \vec{b})$ cannot be distinguished in $\mathcal{L}^k(\mathbf{Q})$.* □

The proof of this theorem resembles the proof of Theorem 6.7, but is more technical. We omit the details here as they are somewhat outside the scope of this thesis. Instead, we illustrate the power of this design by giving alternative game characterisations of the relations $\equiv^{\mathcal{L}^k}$ and $\equiv^{\mathcal{C}^k}$, as follows.

At each round in the *k-pebble cardinality-partition game* on **A** and **B**, Spoiler picks up a pebble from **A** and the corresponding pebble from **B**. Duplicator has to respond by choosing (a) a partition **P** of $U(\mathbf{A})$, (b) a partition **Q** of $U(\mathbf{B})$, with the same number of blocks as **P**, and (c) a bijection $f : \mathbf{P} \to \mathbf{Q}$, for which it holds that $\|P\| = \|f(P)\|$, for all blocks $P \in \mathbf{P}$. Spoiler then picks a block $P \in \mathbf{P}$, and places the chosen pebble in **A** on an element in $P \subseteq A$ and places the corresponding pebble in **B** on an element in $f(P) \subseteq B$. This completes one round in the game. If Duplicator fails to produce the required partitions or the partial map defined by the pebbled elements is not a partial isomorphism, then Spoiler wins the game. Otherwise it can continue for another round. It can be shown that Duplicator has a strategy to play this game forever if, and only if, $\mathbf{A} \equiv^{\mathcal{C}^k} \mathbf{B}$. Similarly, the rules of the *k-pebble partition game* are defined in exactly the same way as above, except we drop the requirement that any two corresponding blocks have to have the same cardinality, i.e. Duplicator does not have to show that $\|P\| = \|f(P)\|$ for all $P \in \mathbf{P}$. It can be shown that Duplicator has a strategy to play this game forever if and only if $\mathbf{A} \equiv^{\mathcal{L}^k} \mathbf{B}$.

These two games can be seen as special cases of the rank-partition game, which of course reflects the fact that the corresponding infinitary logics are both certain restrictions of infinitary rank logic.

Finally, we note that Luosto [56] has independently given a back-and-forth characterisation of equality in first-order logic with any set of generalised quantifiers[3]. The game characterisation given by Luosto is not based on a partitioning method like the one we described above. Instead, in Luosto's game, Duplicator can respond to Spoiler choosing a relation in one structure by either accepting the choice (and giving a matching relation in the opposite structure) or by *challenging* Spoiler's choice. The latter ensures that Spoiler will only choose definable relations at any point in the game, without making 'definability' an explicit requirement of the game rules.

[3]This was kindly pointed out to us by one of the examiners of this thesis.

Chapter 7

Non-definability results for fixed-point logic with rank

Throughout this dissertation, we have given a number of examples illustrating the expressive power of rank logics. For instance, we have shown that many of the problems known to separate IFPC from PTIME, such as deciding the parity of CFI graphs and deciding isomorphism of multipedes, are already expressible in first-order logic with rank, so also in IFPR. However, it can be seen that for all the expressive results we have obtained, the underlying construction has been based on matrices or linear equations over a *fixed* finite field. For instance, the two examples mentioned earlier rely on solving linear equations over the two-element field. This raises an important question, which is to what extent does the characteristic of the underlying field affect the expressive power of the corresponding rank logic?

In this chapter we give a partial answer to this question, by using the rank-partition game to delimit the expressive power of rank logics restricted to a fixed arity and a fixed prime field. Recall that for a prime p and $m \geq 1$, we write $\mathrm{FOR}_{p;m}$ to denote FOR_p restricted to rank operators of arity at most m. Similarly, we write $\mathcal{R}^{\omega}_{p;m}$ to denote finite-variable infinitary logic with rank quantifiers of arity at most m over GF_p. With this notation, our main result can be stated as follows.

Theorem 7.1. *For all distinct primes p and q, there is a property of finite structures which is definable in* $\mathrm{FOR}_{q;2}$ *but not in* $\mathcal{R}^{\omega}_{p;2}$.

As a direct corollary we get a partial separation of fixed-point rank logics over different prime fields.

Corollary 7.2. *For all distinct primes p and q,* $\mathrm{IFPR}_{p;2} \not\equiv \mathrm{IFPR}_{q;2}$ *over finite structures.*

For the proof of Theorem 7.1, we define for each pair of distinct primes p and q a sequence of polynomial-time decidable and pairwise disjoint classes of finite structures, $\mathbf{C}_0, \ldots, \mathbf{C}_{q-1}$. Here the signature of the structures depends only on q. We show that for each $i \in [0, q-1]$, there is a first-order definable reduction from the problem of deciding membership in $\mathbf{C}_i$, given a structure in $\mathbf{C} := \bigcup_i \mathbf{C}_i$, to the problem of deciding solvability of a system of linear equations over GF_q. Coupled with the fact that the class $\mathbf{C}$ can be defined in first-order logic with counting, this shows that each $\mathbf{C}_i$ can be defined by a sentence of FOR_q. We then establish that for each integer $k \geq 2$ and all distinct $i, j \in [0, q-1]$, there are structures $\mathbf{A} \in \mathbf{C}_i$

and $\mathbf{B} \in \mathbf{C}_j$ for which it holds that Duplicator has a winning strategy in the k-pebble 2-ary rank-partition game over GF_p on $\mathbf{A}$ and $\mathbf{B}$. This illustrates that none of the classes $\mathbf{C}_i$ can be defined in $\mathcal{R}^{\omega}_{p;2}$, which completes the proof.

The remainder of this chapter is split into three main sections. In §7.1 we define a family of structures, called *$\mathcal{C}$-structures*, that will be used to define the main separating query for the proof of Theorem 7.1. This construction is quite generic, and is not restricted to the application we describe in this chapter. Broadly speaking, the structures we define are obtained by fixing an Abelian group H and encoding a small circuit, with values from H, into a larger graph with some auxiliary relations. This method of encoding a circuit G over H into a structure $\mathcal{C}(G, H)$ can be seen as a variant of a construction by Torán [64], who considered arithmetic circuits with designated input and output nodes at each gate. Our "H-circuits", on the other hand, are designed to model a closed network where each node is assigned a charge, which in this case is an element of the group H. This intuition allows us to show that there is a direct correspondence between the automorphisms of a structure $\mathcal{C}(G, H)$ and redistributions of charge on an H-circuit G.

In §7.2 we describe families of matrices obtained by uniformly partitioning finite sets with certain properties. To simplify our notation, we describe this construction in quite generic terms, even though the partitions we obtain will ultimately be applied to the vertex sets of $\mathcal{C}$-structures. Having defined these partitions, we then explicitly construct invertible linear transformations that relate matrices obtained by one kind of partition to matrices obtained by a slightly different partition.

Finally, in §7.3 we will, for each pair of distinct primes p and q, give a winning strategy for Duplicator in the rank-partition game over GF_p played on a pair of $\mathcal{C}$-structures $\mathbf{A}$ and $\mathbf{B}$ over the same H-circuit G. Here the group H is taken to be the additive group of integers modulo q. The technical argument for showing that Duplicator's winning strategy satisfies the algebraic condition of the rank-partition game is obtained by appropriately applying the matrix partitions developed in §7.2 to $\mathcal{C}$-structures.

7.1 Building blocks

In this section we define the structures that will be used in §7.3 to describe a winning strategy for Duplicator in the rank-partition game. These structures are obtained by first fixing a finite group H and a graph G where the vertices have been labelled with values from H. It will become convenient to view a graph of that form as a circuit over the group H, as we will discuss further in §7.1.1. Then, given G and H as above, we expand the circuit G into a larger structure by combining a series of graph gadgets along with some auxiliary relations on the edges to encode the group operation on H. This construction will be described in further detail in §7.1.2. The main feature of these structures is that they are very rich in symmetries, as we will discuss in §7.1.3. Due to these symmetries, it will be possible for Duplicator to hide the difference between a pair of similar but non-isomorphic structures of this form by continuously moving around the small area of difference, as we will see later.

7.1.1 Circuits over Abelian groups

Consider an Abelian group H, written additively and with zero 0. For any function $f : A \to H$, where A is a finite set, we write $f(A) := \sum_{a \in A} f(a)$. A circuit over H, or an *H-circuit*, is a

pair (G, γ), where $G = (V, E)$ is a graph and $\gamma : V \to H$ a function that assigns a *charge* $\gamma(v)$ to each vertex $v \in V$ in the circuit. An *H-redistribution* on G is a function $t : V \times V \to H$ which satisfies:

1. $t(v, w) = -t(w, v)$ for all $vw \in E$, and
2. $t(v, w) = t(w, v) = 0$ for all $vw \notin E$.

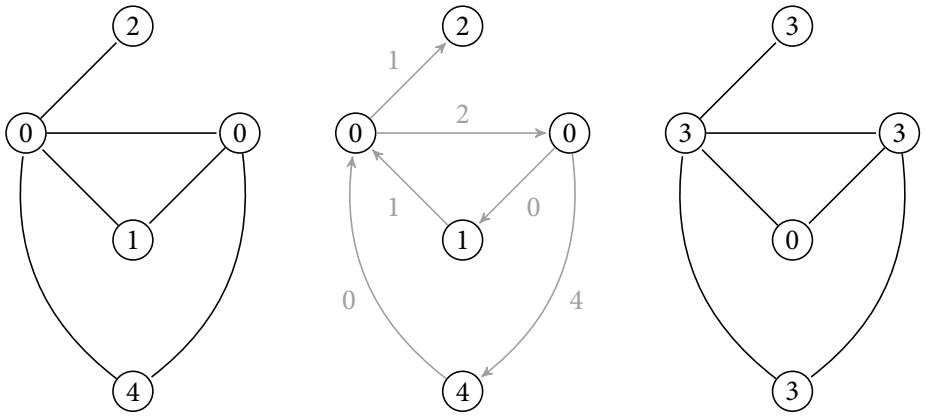
The result of performing an H-redistribution t on a circuit (G, γ) is a new circuit (G, γ^t), where

$$\gamma^t(v) := \gamma(v) + \sum_{w \in N(v)} t(w, v),$$

for all $v \in V$. That is, for each edge $vw \in E$, exactly $t(w, v)$ units of charge flow from w to v. Equivalently, exactly $t(v, w) = -t(w, v)$ units of charge flow from v to w. Note that this process preserves the total charge on the circuit, for

$$\begin{aligned}
\gamma^t(V) = \sum_{v \in V} \gamma^t(v) &= \sum_{v \in V} \Big(\gamma(v) + \sum_{w \in N(v)} t(w, v) \Big) \\
&= \sum_{v \in V} \gamma(v) + \sum_{vw \in E} \big(t(v, w) + t(w, v) \big) \\
&= \sum_{v \in V} \gamma(v) + \sum_{vw \in E} \big(t(v, w) - t(v, w) \big) \\
&= \sum_{v \in V} \gamma(v) = \gamma(V).
\end{aligned}$$

Example 7.3. We illustrate an H-redistribution on a graph $G = (V, E)$, where H is the additive group $\mathbb{Z}/(5\mathbb{Z})$.



(a) Before applying t. **(b)** H-redistribution t. **(c)** After applying t.

The first figure (a) above shows the initial circuit, where the charge on each vertex is indicated by its label. The redistribution $t : V \times V \to H$ is illustrated in figure (b). Here, the edge labels, together with the orientation of the edges, determine t. That is, if there is a directed edge from v to w with label c, then $t(v, w) = c$ and $t(w, v) = -c$. Figure (c) shows the result after applying the redistribution t on the original circuit. Here, all the arithmetic is modulo 5; for instance, the charge at the bottommost vertex, after redistribution, is $4 + 4 - 0 = 8 \equiv 3$

mod 5. Note that the total charge before redistribution ($7 \equiv 2 \mod 5$) and the total charge after redistribution (c) ($12 \equiv 2 \mod 5$) are the same, as expected. ■

Later in this section, we will need the following basic result on H-circuits.

Lemma 7.4 (Charge preservation lemma)**.** *Let (G, γ) and (G, σ) be H-circuits, where $G = (V, E)$ is a connected graph and H a finite Abelian group, written additively. Then $\gamma(V) = \sigma(V)$ if and only if there is an H-redistribution t such that $\gamma^t = \sigma$.*

Proof. The "if" direction is clear, for an H-redistribution preserves the total charge on a circuit, as noted above. For the other direction, suppose (G, γ) and (G, σ) are H-circuits as described and $\gamma(V) = \sigma(V)$. Let $T = (V, F, r)$ be a directed spanning tree of G with root $r \in V$ (that is, the edges are directed from the root). Suppose r has at least one child in T; otherwise the claim holds trivially.

We construct an H-redistribution $t : V \times V \to H$ as follows. Firstly, we set $t(v, w) = 0$ for all $v, w \in V$ with $(v, w), (w, v) \notin F$. Secondly, we define $t(v, w)$ for all (v, w) with either $(v, w) \in F$ or $(w, v) \in F$ by induction, starting with the leaves of the tree T and moving upwards towards the root.

- *Base case.* If v is a leaf of T, with parent w, then set $t(v, w) := \gamma(v) - \sigma(v)$ and $t(w, v) := \sigma(v) - \gamma(v)$. Then $\gamma^t(v) = \gamma(v) + t(w, v) = \sigma(v)$, as required.
- *Inductive step.* Consider a vertex v with children $u_1, \dots, u_k$, and suppose that $t(v, u_i)$ and $t(u_i, v)$ are already defined, for each $i \in [k]$. If v is the root then we are already done; if not, suppose v has parent w. Then we set

$$t(w, v) := \sigma(v) - \gamma(v) - \sum_{i=1}^{k} t(u_i, v),$$

and $t(v, w) := -t(w, v)$. By this definition,

$$\gamma^t(v) = \gamma(v) + \sum_{x \in N_G(v)} t(x, v) = \gamma(v) + t(w, v) + \sum_{i=1}^{k} t(u_i, v) = \sigma(v),$$

as required.

By the induction, this procedure constructs a function t such that $\gamma^t(v) = \sigma(v)$ for all vertices $v \neq r$. Hence, $\gamma^t(V \setminus \{r\}) = \sigma(V \setminus \{r\})$. We claim that it also holds that $\gamma^t(r) = \sigma(r)$, which then completes the proof. To show this, we use the fact that t is a redistribution, so the total charge must be preserved. Hence, $\gamma^t(V) = \sigma(V)$ and

$$\gamma^t(V) = \gamma^t(r) + \gamma^t(V \setminus \{r\}) = \gamma^t(r) + \sigma(V \setminus \{r\}),$$

which shows that $\gamma^t(r) = \sigma(r)$.

□

7.1.2 $\mathcal{C}$-structures

We now describe a scheme for encoding H-circuits (G, γ) into finite relational structures $\mathcal{C}_H(G, \gamma)$ with specific properties. The structure $\mathcal{C}_H(G, \gamma)$ consists of a highly symmetric graph along with auxiliary relations that encode the group operation on H. The role of these auxiliary relations is to ensure that each automorphism of $\mathcal{C}_H(G, \gamma)$ corresponds to a redistribution of charge on the H-circuit (G, γ), as we will see later. Note that from now on, we assume that all graphs have at least two vertices.

Before describing the encoding scheme, we establish some common notation. For $f, g : A \to H$, where A is a finite set, we write $f - g$ and $f + g$ to denote the functions $x \mapsto f(x) - g(x)$ and $x \mapsto f(x) + g(x)$, respectively, for $x \in A$. For $i \in H$, write $(A \to H)_i$ to denote the set of functions $f : A \to H$ with $f(A) = i$. Observe that for any $i, j \in H$, $f \in (A \to H)_i$ and $g \in (A \to H)_j$, we have $f + g \in (A \to H)_{i+j}$ and $f - g \in (A \to H)_{i-j}$.

Now let $G = (V, E)$ be a graph and define sets

$$\begin{aligned} B(v, w) &:= \{(v, w, i) \mid i \in H\} \subseteq V \times V \times H && \forall v, w \in V \text{ with } vw \in E, \\ O(v) &:= \bigcup_{w \in N(v)} B(v, w) \subseteq V \times V \times H && \forall v \in V, \text{ and} \\ I(v, k) &:= (E(v) \to H)_k && \forall v \in V \ \forall k \in H. \end{aligned}$$

Definition 7.5 (Graph gadgets)**.** Let $G = (V, E)$ be a connected graph and let H be a finite Abelian group. For $v \in V$ and $k \in H$, write $\mathcal{X}_H(v, k)$ to denote the graph on vertices $O(v) \dot\cup I(v, k)$ with edge relation

$$E(\mathcal{X}_H(v, k)) := \{\{(v, w, i), f\} \mid (v, w, i) \in O(v), f \in I(v, k) \text{ and } f(vw) = i\}.$$

We collectively refer to graphs of the form $\mathcal{X}_H(v, k)$ as *graph gadgets*. ■

We refer to the two collections of vertices $O(v)$ and $I(v, k)$ as the *outer vertices* and *inner vertices* of $\mathcal{X}_H(v, k)$, respectively. Note that each $\mathcal{X}_H(v, k)$ is a bipartite graph, as the only edges are those between outer vertices and inner vertices. An example of a graph gadget is shown in Figure 7.1.

Definition 7.6. Consider an H-circuit (G, γ), where $G = (V, E)$ is a connected graph and H a finite Abelian group. Let $C^*_H(G, \gamma)$ be the graph obtained from the disjoint union of graph gadgets $\dot\bigcup_{v \in V} \mathcal{X}_H(v, \gamma(v))$ by adding an edge between all vertices (v, w, i) and (w, v, j), for $vw \in E$ and $i, j \in H$. ■

Observe that for each $vw \in E$, the subgraph of $C^*_H(G, \gamma)$ induced by $B(v, w) \cup B(w, v)$ is a complete bipartite graph, with the two parts given by $B(v, w)$ and $B(w, v)$. We now introduce the main building blocks that are used in this chapter, each one a finite relational structure encoding an H-circuit (G, γ), where G is a linearly ordered graph. Such an encoding is obtained from $C^*_H(G, \gamma)$ by including a linear preorder on the vertex set and adding auxiliary relations for describing the group operation on H.

Definition 7.7 ($\mathcal{C}$-structures)**.** Consider an H-circuit (G, γ), where $G = (V, E, \leqslant)$ is an ordered connected graph and H a finite Abelian group, written additively. The ordering $\leqslant$

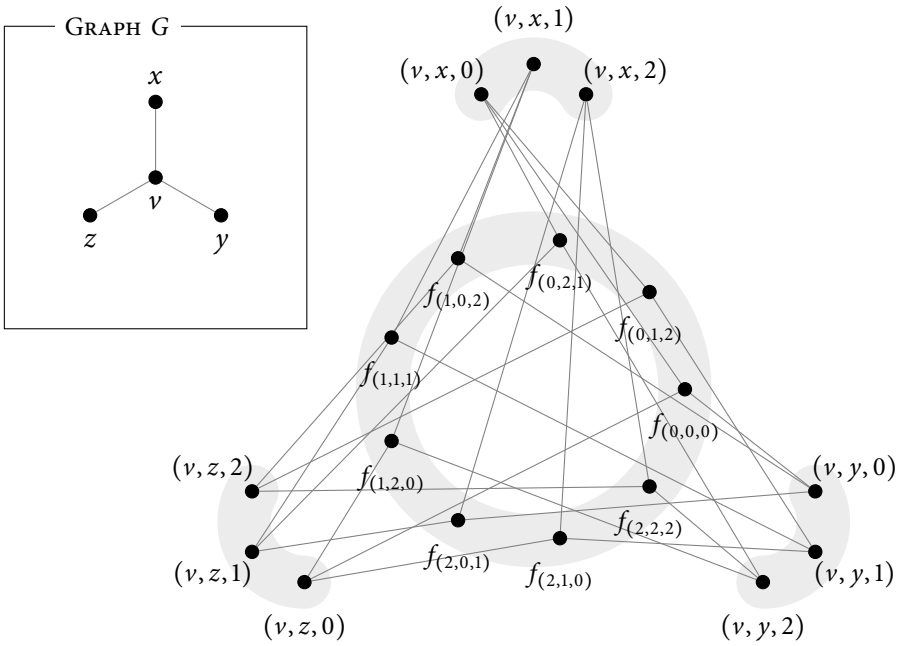


Figure 7.1: A graph gadget $\mathcal{X}_H(v,0)$ for the additive group $H = \mathbb{Z}/(3\mathbb{Z})$, constructed from a vertex v in G of degree three. The vertex v is shown in the inset with its three neighbours x, y and z labelled. Each inner vertex in $(E(v) \to H)_0$ is labelled by the values it takes on each of the three edges vx, vy and vz, in that order. That is, we write $f_{(i,j,k)}$ to denote the function defined by: $vx \mapsto i$, $vy \mapsto j$ and $vz \mapsto k$. Observe that $\mathcal{X}_H(v,0)$ is bipartite: the only edges are those between outer vertices and inner vertices.

induces a lexicographic ordering on $V \times V$ which we call $\leqslant_{\text{lex}}$. Let

$$\mathcal{C}_H(G,\gamma) \coloneqq (C^*_H(G,\gamma), \preccurlyeq, (A_k)_{k\in H})$$

where the linear preorder $\preccurlyeq$ is defined by

- $f \preccurlyeq g$ if and only if $v \leqslant w$, for all $f \in I(v,\gamma(v))$, $g \in I(w,\gamma(w))$ and $v, w \in V$;
- $f \preccurlyeq (v,w,i)$, for all inner vertices f and outer vertices (v,w,i); and
- $(v_1,w_1,i_1) \preccurlyeq (v_2,w_2,i_2)$ if and only if $(v_1,w_1) \leqslant_{\text{lex}} (v_2,w_2)$, for all outer vertices (v_1,w_1,i_1) and (v_2,w_2,i_2);

and

$$A_k \coloneqq \{\{(v,w,i),(w,v,j)\} \mid vw \in E \text{ and } i+j \in k\} \subseteq E(C^*_H(G,\gamma))$$

is a collection of edges, for each $k \in H$. ∎

We collectively refer to structures of the form $\mathcal{C}_H(G,\gamma)$ as *$\mathcal{C}$-structures* (for lack of a better term) and denote by τ_H the signature of $\mathcal{C}$-structures over a group H. Note that the preorder $\preccurlyeq$ serves mainly to restrict the automorphisms of each $\mathcal{C}_H(G,\gamma)$ (as well as isomorphisms between different $\mathcal{C}$-structures) to maps that preserve each set of inner vertices $I(v,\gamma(v))$ and each set of outer vertices $B(v,w)$, for $v \in V$ and $w \in N(v)$. For each $vw \in E$, the

relations A_k, $k \in H$, can be seen as a colouring of the edges in the complete bipartite graph induced by $B(v,w) \cup B(w,v)$. In this sense, an edge $(v,w,i)(w,v,j)$ is given the colour A_k if and only if $i + j = k$.

7.1.3 Isomorphisms of $\mathcal{C}$-structures

We complete our study of $\mathcal{C}$-structures by showing that automorphisms of $\mathcal{C}_H(G,\gamma)$ correspond with certain redistributions of charge on the H-circuit (G,γ). This supports our previous claim that $\mathcal{C}$-structures are rich in symmetries. In particular, the connection between symmetries and H-redistributions allows us to show that two structures $\mathcal{C}_H(G,\gamma)$ and $\mathcal{C}_H(G,\gamma')$ are isomorphic, if and only if the two circuits (G,γ) and (G,γ') have the same amount of H-charge.

The first step in this analysis is to consider maps between different graph gadgets, as in the following lemma.

Lemma 7.8 (Graph gadget isomorphisms)**.** *Let $G = (V,E)$ be a connected graph and let H be a finite Abelian group, written additively. Let $k \in H$ and $v \in V$. Then for any function $h : E(v) \to H$, there is a unique isomorphism*

$$\varphi_h : \mathcal{X}_H(v,k) \to \mathcal{X}_H(v, k + h(E(v)))$$

for which it holds that $\varphi_h : (v,w,i) \mapsto (v,w,i+h(vw))$ for each $vw \in E(v)$ and $i \in H$.

Proof. Let $l := k + h(E(v))$ and write $\mathcal{X}_H(v,k) = (V_k, E_k)$ and $\mathcal{X}_H(v,l) = (V_l, E_l)$. We define $\varphi_h : V_k \to V_l$ by

$$\begin{aligned}&\varphi_h : (v,w,i) \mapsto (v,w,i+h(vw)) \text{ for all } (v,w,i) \in O(v),\\ &\varphi_h : f \mapsto f + h \text{ for all } f \in I(v,k).\end{aligned}$$

We claim that φ_h is an isomorphism. First of all, note that if $f \in I(v,k)$ then $f(E(v)) = k$, and hence

$$(f+h)(E(v)) = f(E(v)) + h(E(v)) = k + h(E(v)) = l.$$

Therefore, $f + h \in I(v,l)$. All that remains is to show that for any two vertices $x, y \in V_k$, $xy \in E_k$ if and only if $\varphi_h(x)\varphi_h(y) \in E_l$. In each of the two graphs, there are only edges between outer vertices and inner vertices. Consider an outer vertex (v,w,i) and an inner vertex f in $\mathcal{X}_H(v,k)$. By definition of the edge relations E_k and E_l,

$$\begin{aligned}(v,w,i)f \in E_k &\Leftrightarrow f(vw) = i\\ &\Leftrightarrow f(vw) + h(vw) = i + h(vw)\\ &\Leftrightarrow (f+h)(vw) = i + h(vw)\\ &\Leftrightarrow (v,w,i+h(vw))(f+h) \in E_l\\ &\Leftrightarrow \varphi_h((v,w,i))\varphi_h(f) \in E_l,\end{aligned}$$

as required. To show uniqueness, suppose that there is an isomorphism $\psi : \mathcal{X}_H(v,k) \to \mathcal{X}_H(v,l)$, different from φ_h, with the property that $\psi((v,w,i)) = (v,w,i+h(vw))$ for all $(v,w,i) \in O(v)$. By assumption, there must then be at least one vertex $f \in I(v,k)$ such that $\psi(f) \neq \varphi_h(f)$.

Consider $(v, w, i) \in O(v)$ so that $(v, w, i)f \in E_k$, and hence $f(vw) = i$. Since ψ is an isomorphism, we must have $\psi((v, w, i))\psi(f) \in E_l$, or equivalently, $\psi(f)(vw) = i + h(vw) = f(vw) + h(vw)$. This implies that $\psi(f)(vw) = (f + h)(vw)$ for all $vw \in E(v)$; that is, $\psi(f) = f + h = \varphi_h(f)$ — a contradiction. □

Lemma 7.9. *Let $G = (V, E, \leqslant)$ be an ordered connected graph and let H be a finite Abelian group, written additively. Let $\gamma, \sigma : V \to H$ be charge functions on G. Then $\mathcal{C}_H(G, \gamma) \cong \mathcal{C}_H(G, \sigma)$ if and only if there is an H-redistribution t on G such that $\sigma = \gamma^t$.*

Proof. Consider two H-circuits (G, γ) and (G, σ), as in the statement of the lemma. For $k \in H$, we write $A_k(\mathcal{C}_H(G, \gamma))$ and $A_k(\mathcal{C}_H(G, \sigma))$ to denote the A_k-relation in $\mathcal{C}_H(G, \gamma)$ and $\mathcal{C}_H(G, \sigma)$, respectively.

For the "if" direction, suppose $t : V \times V \to H$ is an H-redistribution on G, such that $\sigma = \gamma^t$. For each $v \in V$, write $h_{t,v} : E(v) \to H$ for the function defined by $vw \mapsto t(w, v)$. That is, for each edge $vw \in E(v)$, $h_{t,v}(vw)$ describes the amount of charge being redistributed *from w to v*. By definition of H-redistribution,

$$\gamma^t(v) := \gamma(v) + \sum_{w \in N(v)} t(w, v) = \gamma(v) + h_{t,v}(E(v)),$$

so by Lemma 7.8, $\varphi_{h_{t,v}}$ is an isomorphism from $\mathcal{X}_H(v, \gamma(v))$ to $\mathcal{X}_H(v, \gamma(v) + h_{t,v}(E(v))) = \mathcal{X}_H(v, \gamma^t(v))$. By combining all the maps $(\varphi_{h_{t,v}})_{v \in V}$, write

$$\pi_t : \mathcal{C}_H(G, \gamma) \to \mathcal{C}_H(G, \gamma^t)$$

to denote the map which is defined for all $v \in V$ and $x \in \mathcal{X}_H(v, \gamma(v))$ by

$$\pi_t(x) := \varphi_{h_{t,v}}(x).$$

We claim that π_t is an isomorphism. Since each $\varphi_{h_{t,v}}$ is an isomorphism from $\mathcal{X}_H(v, \gamma(v))$ to $\mathcal{X}_H(v, \gamma^t(v))$, it follows that π_t preserves the preorder $\leqslant$ and all edges between inner and outer vertices. In particular, it maps the set of inner vertices $I(v, \gamma(v))$ induced by a vertex $v \in V$ to the corresponding set of inner vertices $I(v, \gamma^t(v))$ and maps the set of outer vertices $B(v, w)$ induced by an edge vw to the corresponding set of outer vertices $B(v, w)$ in $\mathcal{X}_H(v, \gamma^t(v))$. Since $(B(v, w), B(w, v))$ forms a complete bipartite graph, it follows also that π_t preserves the edge relation between elements of $B(v, w)$ and elements of $B(w, v)$, for $vw \in E$. All that remains to show is that π_t preserves the edge colour relations A_k, for $k \in H$. Recall that the H-redistribution t satisfies the condition $t(v, w) = -t(w, v)$, for all $vw \in E$. Then by the definition of π_t, it holds that

$$\begin{aligned}
&\pi_t : (v, w, i) \mapsto (v, w, i + t(w, v)) \text{ and} \\
&\pi_t : (w, v, j) \mapsto (w, v, j + t(v, w)) = (w, v, j - t(w, v)),
\end{aligned}$$

for all $vw \in E$ and $i, j \in H$. Hence,

$$\begin{aligned}
\pi_t(v, w, i)\pi_t(w, v, j) \in A_k(\mathcal{C}_H(G, \gamma^t)) &\\
&\Leftrightarrow (i + t(w, v)) + (j - t(w, v)) = k \\
&\Leftrightarrow i + j = k \\
&\Leftrightarrow (v, w, i)(w, v, j) \in A_k(\mathcal{C}_H(G, \gamma)),
\end{aligned}$$

as required.

For the "only if" direction, suppose there is an isomorphism $\pi : \mathcal{C}_H(G,\gamma) \to \mathcal{C}_H(G,\sigma)$. As π must preserve the preorder $\preccurlyeq$, it follows that each set of outer vertices $B(v,w)$ in $\mathcal{C}_H(G,\gamma)$, induced by an edge vw, is mapped to the corresponding set of outer vertices $B(v,w)$ in $\mathcal{C}_H(G,\sigma)$. The following claim shows that the mapping $B(v,w) \mapsto B(v,w)$ induced by π will not be arbitrary.

Claim 5. *For every $v,w \in V$ with $vw \in E$, there is $\Delta_{v,w} \in H$ for which it holds that $\pi_t : (v,w,i) \mapsto (v,w,i+\Delta_{v,w})$, for all $i \in H$.*

Proof of claim. Suppose, towards a contradiction, that there are $v,w \in V$, with $vw \in E$, and $i,j \in H$, $i \neq j$, such that

$$\pi_t : (v,w,i) \mapsto (v,w,i+\Delta_i) \text{ and}$$
$$\pi_t : (v,w,j) \mapsto (v,w,j+\Delta_j),$$

but $\Delta_i \neq \Delta_j$. Since π_t is an isomorphism, it must preserve the relations A_m, $m \in H$. Hence it must hold that

$$\pi_t : (v,w,-i) \mapsto (v,w,-(i+\Delta_i)) \text{ and}$$
$$\pi_t : (v,w,-j) \mapsto (v,w,-(j+\Delta_j)).$$

Write $i-j=m$, where $m \neq 0$ by assumption. Then $(v,w,i)(w,v,-j) \in A_m(\mathcal{C}_H(G,\gamma))$ and therefore $\pi_t(v,w,i)\pi_t(w,v,-j) = (v,w,i+\Delta_i)\pi_t(w,v,-(j+\Delta_j)) \in A_m(\mathcal{C}_H(G,\sigma))$. But by definition of the relation A_m,

$$\begin{aligned}(v,w,i+\Delta_i)\pi_t(w,v,-(j+\Delta_j)) \in A_m(\mathcal{C}_H(G,\sigma)) \\ \Leftrightarrow (i+\Delta_i)+(-j-\Delta_j) = m \\ \Rightarrow (i-j)+(\Delta_i-\Delta_j) = m+(\Delta_i-\Delta_j) = m,\end{aligned}$$

which is a contradiction. □

Now let $t : V \times V \to H$ be the function defined by

$$t(v,w) := \begin{cases} \Delta_{v,w} & \text{if } vw \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Since π_t preserves the relations A_m, it must hold that $\Delta_{v,w} = -\Delta_{w,v}$ for all $v,w \in V$ with $vw \in E$. Hence, t is an H-redistribution, with $\gamma^t = \sigma$. □

By combining Lemma 7.9 and Lemma 7.4, we get the following characterisation of $\mathcal{C}$-structures, up to isomorphism.

Theorem 7.10. *Let $G = (V,E,\preccurlyeq)$ be an ordered connected graph and let H be a finite Abelian group, written additively. Let $\gamma,\sigma : V \to H$ be charge functions on G. Then $\mathcal{C}_H(G,\gamma) \cong \mathcal{C}_H(G,\sigma)$ if and only if $\gamma(V) = \sigma(V)$.* □

It follows that for each G and H, there are exactly $\|H\|$ distinct structures $\mathcal{C}_H(G,\cdot)$, up to isomorphism. In the following, we write $\mathcal{C}_H^k(G)$, for $k \in H$, to denote the structure $\mathcal{C}_H(G,\delta^k_{v_{\min}})$,

where $v_{\min}$ is the least element of V with respect to $\leqslant$ and $\delta^k_{v_{\min}} : V \to H$ is the k-delta function on V, defined for all $v \in V$ by

$$\delta^k_{v_{\min}} : v \mapsto \begin{cases} k & \text{if } v = v_{\min}, \\ 0 & \text{otherwise.} \end{cases}$$

For each H and $i \in H$, we also write $\mathbf{C}^i_H$ for the class of all $\mathcal{C}$-structures $\mathcal{C}_H(G, \gamma)$, with G an ordered connected graph and $\gamma : V \to H$ a charge function on G with $\gamma(V) = i$.

7.2 Similar matrices defined by set partitions

The main challenge for Duplicator in the rank-partition game is to come up with partitions that satisfy the matrix rank condition: that is, given a pair of partitions, the corresponding matrices must have the same rank for *all* possible labellings. In this section we indicate how this can be done when the game is played on a pair of non-isomorphic $\mathcal{C}$-structures induced by the same H-circuit. The results obtained here will play a key part in our description of the winning strategy in §7.3.

More specifically, we develop a generic matrix construction, based on partitions, and show that non-isomorphic matrices arising from distinct partitions of the same set have equal rank. In fact, we prove a stronger statement and show that for any pair of such partitions, the two families of matrices obtained by running over all labelling functions are uniformly similar, which implies that any two matrices with the same labelling function have the same rank. To simplify our notation we keep this construction quite generic, and describe partitions and matrices over arbitrary sets (with certain properties). It should be kept in mind though that these partitions will ultimately be applied to the vertex sets of $\mathcal{C}$-structures, as noted above.

Our discussion is split into three sections. In §7.2.1 we define the basic partitions as well as maps between a pair of partitions of the same set. In §7.2.2 we develop technical tools to establish similarity of matrices obtained by labelling the partitions with elements of a finite field. Finally in §7.2.3, we briefly sketch how these tools can be used for matrices with slightly expanded index sets, more suitable for the situation in §7.3 when we consider partitions of $\mathcal{C}$-structures.

Throughout this section we work with finite Abelian groups. If H is such a group, then we normally write $\oplus$ to denote the group operation, instead of writing the group additively as before. The reason is that we frequently have to consider expressions that involve both arithmetic over H as well as arithmetic over a finite field, with standard field operations $+$ and $\cdot$. In this case, we write $i \ominus j := i \oplus -j$, where i and j are elements of H and $-j$ is the inverse of j. Finally, we write $\bigoplus$ (a slightly larger version of $\oplus$) to denote a "summation" operator, representing the cumulative application of $\oplus$ over a set of terms from H. For instance, if $X = \{h_1, \ldots, h_m\} \subseteq H$, then we write $\bigoplus_{i=1}^m h_i := h_1 \oplus \cdots \oplus h_m$.

7.2.1 Basic partitions

Let H be a finite Abelian group of cardinality $q > 0$ with group operation $\oplus$. Let p be a prime number with $(p, q) = 1$ and let X be a finite set of cardinality $d > 2$, where d is chosen such

that

$$q^{d-2} \equiv 1 \pmod{p}. \qquad (*)$$

For instance, by taking $d = p+1$ we get $q^{d-2} = q^{p-1} \equiv 1 \pmod{p}$, by Fermat's little theorem. For each $k \in H$, let $F_k := (X \to H)_k = \{f \in (X \to H) \mid f(A) = k\}$, and write $N_k := F_k \dot\cup X$. Observe that for any $k \in H$, $\|F_k\| = q^{d-1}$. For $f, g : X \to H$, we write $f \ominus g$ and $f \oplus g$, to denote the functions $x \mapsto f(x) \ominus g(x)$ and $x \mapsto f(x) \oplus g(x)$, respectively, like we did in §7.1.

In this section we describe ways to partition the set $N_k \times N_k$. Recall that a partition $\mathbf{P}$ of a set A is a collection of mutually disjoint and non-empty subsets of A (called *blocks*) whose union is all of A. If $\mathbf{P}$ is such a partition and $a \in A$, then we write $[\![a]\!]_{\mathbf{P}}$ to denote the block containing a.

Definition 7.11 (Partition blocks)**.** For $k \in H$, define

(i) $\Gamma^k_h := \{(f,g) \in F_k \times F_k \mid f \ominus g = h\} \subset N_k \times N_k$, for each $h \in F_0$; and

(ii) $\Omega^k_{x,i} := \{(f,x) \in F_k \times X \mid f(x) = i\} \subset N_k \times N_k$, for each $x \in X$ and $i \in H$.

We also write $(\Omega^k_{x,i})^{\mathrm{t}} := \{(x,f) \mid (f,x) \in \Omega^k_{x,i}\}$ to denote the "transpose" of $\Omega^k_{x,i}$. ■

It can be seen that for any $h_1, h_2 \in F_0$ it holds that $\Gamma^k_{h_1} \cap \Gamma^k_{h_2} = \varnothing$ whenever $h_1 \neq h_2$. Also, it can be seen that $\bigcup_h \Gamma^k_h = F_k \times F_k$. Similarly, for all $x_1, x_2 \in X$ and $i_1, i_2 \in H$, it holds that $\Omega^k_{x_1,i_1} \cap \Omega^k_{x_2,i_2} = \varnothing$ whenever $(x_1,i_1) \neq (x_2,i_2)$, and $\bigcup_{x,i} \Omega^k_{x,i} = F_k \times X$. By putting all the blocks Γ^k_h, $\Omega^k_{x,i}$ and $(\Omega^k_{x,i})^{\mathrm{t}}$ together, and adding a trivial partition of the set $X \times X$, we obtain a partition of the space $N_k \times N_k$ as follows.

Definition 7.12 (Set partitions)**.** If $k \in H$ then we write $\mathbf{P}_k$ to denote the partition of $N_k \times N_k$ defined by

$$\mathbf{P}_k := \{\Gamma^k_h \mid h \in F_0\} \cup \{\Omega^k_{x,i}, (\Omega^k_{x,i})^{\mathrm{t}} \mid x \in X, i \in H\} \cup \{\{(x,y)\} \mid x, y \in X)\}.$$

■

We also consider maps between the partitions $\mathbf{P}_0$ and $\mathbf{P}_k$, as defined here.

Definition 7.13 (Maps between partitions)**.** For $k \in H$ we write $\varphi_k : \mathbf{P}_0 \to \mathbf{P}_k$ to denote the bijection defined by

$$\begin{aligned}
\varphi_k : \Gamma^0_h &\mapsto \Gamma^k_h && \forall h \in F_0,\\
\varphi_k : \Omega^0_{x,i} &\mapsto \Omega^k_{x,i} && \forall x \in X\ \forall i \in H,\\
\varphi_k : (\Omega^0_{x,i})^{\mathrm{t}} &\mapsto (\Omega^k_{x,i})^{\mathrm{t}} && \forall x \in X\ \forall i \in H,\\
\varphi_k : \{(x,y)\} &\mapsto \{(x,y)\} && \forall x, y \in X.
\end{aligned}$$

■

7.2.2 Matrices defined over partitions

We now consider matrices over the prime field GF_p obtained by labelling the blocks of the partition $\mathbf{P}_k$ by elements of $[0, p-1]$. The idea of defining matrices in this way was discussed in Chapter 6, in relation to the rank-partition game, and our notation here is the same.

Let $k \in H$ and consider a labelling $\gamma : \mathbf{P}_k \to [0, p-1]$ of the blocks in $\mathbf{P}_k$ with elements of GF_p. Then we write $M_k(\gamma)$ to denote the $N_k \times N_k$ matrix over GF_p defined by applying the labelling γ to $\mathbf{P}_k$; that is,

$$M_k(\gamma) : (m, n) \mapsto \gamma([\![(m, n)]\!]_{\mathbf{P}_k}),$$

for all $m, n \in N_k$. Our aim in this section is to prove the following.

Theorem 7.14 (Partition matrices have the same rank)**.** *Let $k \in H$. Then for any labelling $\gamma : \mathbf{P}_0 \to [0, p-1]$, the matrices $M_0(\gamma)$ and $M_k(\gamma \circ \varphi_k^{-1})$ have the same rank over GF_p.*

In proving this theorem, we actually establish a stronger statement. More specifically, for every $k \in H$ and $z \in X$ we construct a non-singular $N_k \times N_0$ matrix $S_{k,z}$ over GF_p such that

$$S_{k,z}\, M_0(\gamma)\, S_{k,z}^{-1} = M_k(\gamma \circ \varphi_k^{-1}),$$

for any labelling γ of the partition $\mathbf{P}_k$. This shows that the two matrices $M_0(\gamma)$ and $M_0(\gamma)$ are similar, which in turn implies that they have the same rank. Note that each matrix $S_{k,z}$ does not depend on the labelling γ. Our construction therefore shows that the two collections of matrices defined by $\mathbf{P}_0$ and $\mathbf{P}_k$ are pairwise *uniformly* similar, when indexed by functions in $(\mathbf{P}_0 \to [0, p-1])$.

The matrices $S_{k,z}$ will be explicitly constructed as a combination of simpler "A" and "B" matrices, which we now describe.

Definition 7.15 (B-matrices)**.** For $z \in X$ and $k \in H$, write $B_{k,z}$ to denote the $F_0 \times F_0$ matrix over GF_p defined by

$$B_{k,z}(f, g) := \begin{cases} 1 & \text{if } f(z) \oplus k = g(z) \\ 0 & \text{otherwise,} \end{cases}$$

for all $f, g \in F_0$. ■

We establish some basic properties of the B-matrices.

Lemma 7.16 (Products of B-matrices)**.** *For all $z \in X$ and $i, j \in H$, $B_{i,z}B_{j,z} = B_{i\oplus j,z}$.*

Proof. Let $f, g \in F_0$ and write

$$\begin{aligned}
(B_{i,z}B_{j,z})(f, g) &= \sum_{h \in F_0} B_{i,z}(f, h)B_{j,z}(h, g) \\
&= \|\{h \in F_0 \mid f(z) \oplus i = h(z) \text{ and } h(z) \oplus j = g(z)\}\| \\
&\qquad\qquad \text{(by definition of } B_{k,z}) \\
&= \|\{h \in F_0 \mid h(z) = f(z) \oplus i \text{ and } f(z) \oplus (i \oplus j) = g(z)\}\| \\
&= \begin{cases} q^{d-2} \equiv 1 \pmod p & \text{if } f(z) \oplus (i \oplus j) = g(z), \\ 0 & \text{otherwise} \end{cases} \\
&= B_{i\oplus j,z}(f, g). \qquad (B_{i\oplus j,z} \text{ a } \mathsf{GF}_p \text{ matrix})
\end{aligned}$$

Here we have used the fact that $q^{d-2} \equiv 1 \pmod{p}$ by $(*)$. □

Lemma 7.17 (B-matrix transpose)**.** *For all $z \in X$ and $k \in H$, $(B_{k,z})^t = B_{-k,z}$.*

Proof. Let $f, g \in F_0$. Then

$$\begin{aligned} B_{k,z}(f,g) = 1 &\Leftrightarrow f(z) \oplus k = g(z) \\ &\Leftrightarrow g(z) \oplus (-k) = f(z) \\ &\Leftrightarrow B_{-k,z}(g,f) = 1. \end{aligned}$$

Hence $(B_{k,z})^{\mathrm{t}}(f,g) := B_{k,z}(g,f) = B_{-k,z}(f,g)$. □

Definition 7.18 (A-matrices)**.** For $z \in X$ and $k \in H$, let $A_{k,z} := B_{k,z} - B_{0,z} + I$, where I is the $F_0 \times F_0$ identity matrix. ■

The following lemma shows that each matrix $A_{k,z}$ is orthogonal, with inverse explicitly given by $(A_{k,z})^{-1} = A_{k,z}^{\mathrm{t}} = A_{-k,z}$.

Lemma 7.19 (Orthogonality of A-matrices)**.** $A_{k,z}\, A_{k,z}^t = A_{k,z}\, A_{-k,z} = I$.

Proof. The transpose operation respects addition, so by Lemma 7.17,

$$A_{k,z}^{\mathrm{t}} = B_{k,z}^{\mathrm{t}} - B_{0,z}^{\mathrm{t}} + I^{\mathrm{t}} = B_{-k,z} - B_{0,z} + I = A_{-k,z}.$$

Hence

$$\begin{aligned} A_{k,z}A_{-k,z} &= (B_{k,z} - B_{0,z} + I)(B_{-k,z} - B_{0,z} + I) \\ &= (B_{k,z}B_{-k,z} - B_{k,z}B_{0,z} + B_{k,z}) - (B_{0,z}B_{-k,z} - B_{0,z}B_{0,z} + B_{0,z}) + (B_{-k,z} - B_{0,z} + I) \\ &= (B_{0,z} - B_{k,z} + B_{k,z}) - (B_{-k,z} - B_{0,z} + B_{0,z}) + (B_{-k,z} - B_{0,z} + I) \quad \text{(Lemma 7.16)} \\ &= I. \end{aligned}$$

□

From now on, fix $k \in H$ and $z \in X$ and let $\gamma : \mathbf{P}_0 \to [0, p-1]$ be a labelling of the partition $\mathbf{P}_0$. Consider the two matrices $M_0(\gamma)$ and $M_k(\gamma \circ \varphi_k^{-1})$, which are indexed by $N_0 \times N_0$ and $N_k \times N_k$, respectively. Our aim is to map $M_0(\gamma)$ to $M_k(\gamma \circ \varphi_k^{-1})$ by applying the A- and B-transformations, which are indexed by $F_0 \times F_0$. To ensure that all matrices have the same row and column index sets, we will map $M_0(\gamma)$ to a matrix obtained by first applying a suitable $N_0 \times N_k$ invertible linear transformation to $M_k(\gamma \circ \varphi_k^{-1})$, as we describe next.

Let $\pi_{z,k} : F_0 \to F_k$ be the bijection defined for all $f \in F_0$, $g \in F_k$ by

$$\pi_{z,k}(f) = g :\Leftrightarrow f(z) \oplus k = g(z) \text{ and } f(x) = g(x) \text{ for all } x \neq z.$$

Write $P_{k,z}$ for the permutation matrix representation of $\pi_{k,z}$. That is, $P_{k,z}$ is the $F_0 \times F_k$ permutation matrix defined for all $f \in F_0$ and $g \in F_k$ by $P_{k,z}(f,g) = 1$ if and only if $\pi_{z,k}(f) = g$. Let $Q_{k,z}$ denote the direct sum of $P_{k,z}$ and I_X, where I_X is the $X \times X$ identity matrix. That is,

$$Q_{k,z} := \begin{array}{c} \\ F_0 \\ X \end{array} \begin{array}{c} \begin{array}{cc} F_k & X \end{array} \\ \left(\begin{array}{c|c} P_{k,z} & 0 \\ \hline 0 & I_X \end{array}\right) \end{array}.$$

Observe that $Q_{k,z}$ is indexed by $N_0 \times N_k$ and its inverse $Q_{k,z}^{-1}$ is indexed by $N_k \times N_0$. We now consider the matrices $M_0 := M_0(\gamma)$ and $M_k := Q_{k,z}\, M_k(\gamma \circ \varphi_k^{-1})\, Q_{k,z}^{-1}$, which are both indexed by $N_0 \times N_0$. We can write these as

$$M_0 = \begin{matrix} F_0 \\ X \end{matrix} \overset{\begin{matrix} F_0 & X \end{matrix}}{\left(\begin{array}{c|c} U_0 & R_0 \\ \hline S_0 & W_0 \end{array}\right)} \quad \text{and} \quad M_k = \begin{matrix} F_0 \\ X \end{matrix} \overset{\begin{matrix} F_0 & X \end{matrix}}{\left(\begin{array}{c|c} U_k & R_k \\ \hline S_k & W_k \end{array}\right)}.$$

We observe that the two matrices M_0 and M_k are identical in the two submatrices indexed by $F_0 \times F_0$ and $X \times X$, respectively. That is, $W_0 = W_k$ and $U_0 = U_k$ (and we write $W := W_0 = W_k$ and $U := U_0 = U_k$). The first equality is clear, since both φ_k and $\pi_{k,z}$ act as identity on $X \times X$. For the second equality, consider a pair of functions $f, g \in F_0$. From the definition of $\pi_{k,z}$ it holds for any $h \in F_0$ that $(f,g) \in \Gamma_h^0$ if and only if $(\pi_{k,z}(f), \pi_{k,z}(g)) \in \Gamma_h^k$. Hence,

$$\begin{aligned} U_k(f,g) &= \gamma\big(\varphi_{k,z}^{-1}([\![(\pi_{k,z}(f), \pi_{k,z}(g))]\!]_{\mathbf{P}_k})\big) \\ &= \gamma\big([\![(f,g)]\!]_{\mathbf{P}_0}\big) \\ &= U_0(f,g). \end{aligned}$$

Similarly, it can also be seen that the two matrices S_0 and S_k are equivalent in all rows indexed by $x \in X$ with $x \neq z$. The same holds for the two matrices R_0 and R_k, column-wise.

In summary, we see that the two matrices M_0 and M_k are everywhere equal apart from (potentially) the rows indexed by $z \in X$ in each matrix (the $\{z\} \times F_0$ submatrices) or the columns indexed by $z \in X$ in each matrix (the $F_0 \times \{z\}$ submatrices).

It remains to show that we can apply the A- and B-transformations to map M_0 to M_k. To do that, we consider a series of lemmas, starting with the following.

Lemma 7.20. $A_{k,z} R_0 = R_k$.

Proof. For $x \in X$, let $C_{0,x}$ and $C_{k,x}$ denote the columns indexed by x in R_0 and R_k, respectively. We will show that for each $x \in X$,

$$A_{k,z} C_{0,x} = C_{k,x},$$

which will conclude the proof. Observe that for all $x \neq z$, $C_{0,x} = C_{k,x}$, as discussed earlier. It is only at the columns $C_{0,z}$ and $C_{k,z}$ that the two matrices potentially differ. In order to prove the lemma, we now consider two cases: columns indexed by x when $x \neq z$ and columns indexed by z.

For the first case, let $C = C_{0,x} = C_{k,x}$ be a column indexed by an element $x \neq z$. The column C can be written as a linear combination

$$C = \sum_{i \in H} \sigma_i D_i,$$

where each D_i is a $(0,1)$-column and $\sigma_i \in [0, p-1]$. More specifically, each D_i is the column vector that corresponds to the partition block $\Omega_{x,i}^0 = \{(f,x) \in F_0 \times X \mid f(x) = i\}$ and $\sigma_i := \gamma(\Omega_{x,i}^0)$ is the value assigned to $\Omega_{x,i}^0$ by the labelling γ. By linearity, it will be sufficient to consider each column vector D_i; that is, to show that $A_{k,z} D_i = D_i$ for any $i \in H$.

Claim 6. *For any $i, m, n \in H$, $B_{m,z}D_i = B_{n,z}D_i$.*

Proof of claim. Consider $i, m \in H$ and $f \in F_0$ and write

$$\begin{aligned}(B_{m,z}D_i)(f) &= \sum_{g\in F_0} B_{m,z}(f,g)D_i(g)\\ &= \|\{g \in F_0 \mid f(z) \oplus m = g(z) \wedge g(x) = i\}\|\\ &= q^{d-3}.\end{aligned}$$

This shows that the value of $(B_{m,z}D_i)(f)$ does not depend on either m or f, so in particular for any $n \in H$, $(B_{m,z}D_i - B_{n,z}D_i)(f) = q^{d-3} - q^{d-3} = 0$. □

From the claim, it now follows that

$$A_{k,z}D_i = (B_{k,z} - B_{0,z} + I)D_i = (B_{k,z}D_i - B_{0,z}D_i) + D_i = D_i,$$

as required.

Now for the second case, consider the columns $C_0 := C_{0,z}$ and $C_k := C_{k,z}$ indexed by z in R_0 and R_k, respectively. For $i \in H$ we define a pair of $(0,1)$-vectors

$$D_i^0(f) = \begin{cases}1 & \text{if } f(z) = i,\\ 0 & \text{otherwise,}\end{cases}$$

and

$$D_i^k(f) = \begin{cases}1 & \text{if } f(z) \oplus k = i,\\ 0 & \text{otherwise,}\end{cases}$$

for all $f \in F_0$. As before, we can express each column C_j, $j \in \{0, k\}$, as a linear combination of D_i^j-vectors. Hence, it will suffice by linearity to show for each $i \in H$ that $A_{k,z}D_i^0 = D_i^k$. To do that, fix an $i \in H$ and write

$$\begin{aligned}(B_{k,z}D_i^0)(f) &= \sum_{g\in F_0} B_{k,z}(f,g)D_i^0(g)\\ &= \|\{g \in F_0 \mid f(z) \oplus k = g(z) \wedge g(z) = i\}\|\\ &= \|\{g \in F_0 \mid f(z) \oplus k = g(z) = i\}\|\\ &= \begin{cases}q^{d-2} & \text{if } f(z) \oplus k = i,\\ 0 & \text{otherwise.}\end{cases}\end{aligned}$$

Hence,

$$((B_{k,z} - B_{0,z} + I)D_i^0))(f) = \begin{cases}q^{d-2} & \text{if } f(z) \oplus k = i,\\ -q^{d-2}+1 & \text{if } f(z) = i,\\ 0 & \text{otherwise.}\end{cases}$$

Now considering that $q^{d-2} \equiv 1 \pmod{p}$, we see that

$$(A_{k,z}D_i^0)(f) = D_i^k(f) = \begin{cases} 1 & \text{if } f(z) \oplus k = i, \\ 0 & \text{otherwise.} \end{cases}$$

□

The proof of the following lemma is entirely analogous to the proof of Lemma 7.20.

Lemma 7.21. $S_0\, A_{k,z}^{-1} = S_k$. □

We now consider a transformation of the square matrix U.

Lemma 7.22. $A_{k,z}\, U\, A_{k,z}^{-1} = U$.

Proof. Write $U = \sum_{h \in F_0} \sigma_h U_h$, where each U_h is the $(0,1)$-matrix that corresponds with the partition block $\Gamma_h = \{(f,g) \in F_0 \times F_0 \mid f \ominus g = h\}$ and $\sigma_h := \gamma(\Gamma_h)$ is the value assigned to Γ_h by the labelling γ. By linearity, it will be sufficient to consider the matrices U_h; that is, to show that

$$A_{k,z}\, U_h\, A_{k,z}^{-1} = U_h,$$

for each $h \in F_0$.

Claim 7. *For all $m \in H$, $B_{m,z}U_h = U_hB_{m,z}$.*

Proof of claim. Consider $f, g \in F_0$ and check:

$$\begin{aligned}(B_{m,z}U_h)(f,g) &= \sum_{e \in F_0} B_{m,z}(f,e)U_h(e,g) \\ &= \|\{e \in F_0 \mid f(z) \oplus m = e(z) \wedge e \ominus g = h\}\| \\ &= \begin{cases} 1 & \text{if } f(z) \oplus m = g(z) \oplus h(z), \\ 0 & \text{otherwise,} \end{cases}\end{aligned}$$

and

$$\begin{aligned}(U_hB_{m,z})(f,g) &= \sum_{e \in F_0} U_h(f,e)B_{m,z}(e,g) \\ &= \|\{e \in F_0 \mid f \ominus e = h \wedge e(z) \oplus m = g(z)\}\| \\ &= \begin{cases} 1 & \text{if } f(z) \oplus m = g(z) \oplus h(z), \\ 0 & \text{otherwise.} \end{cases}\end{aligned}$$

□

By this claim, it follows that $B_{m,z}U_h = U_hB_{m,z}$. Hence for all $m, n \in H$,

$$B_{m,z}\, U_h\, B_{n,z} = B_{m,z}(B_{n,z}U_h) = B_{m \oplus n,z}\, U_h,$$

using Lemma 7.16. Expanding $A_{k,z}$ into a sum of B-matrices, we can finally conclude:

$$\begin{aligned}A_{k,z}\, U_h\, A_{k,z}^{-1} &= (B_{k,z} - B_{0,z} + I)U_h(B_{-k,z} - B_{0,z} + I) \\ &= (B_{k \ominus k,z} - B_{k \oplus 0,z} + B_{k,z} - B_{0 \ominus k,z} + B_{0 \oplus 0,z} - B_{0,z} + B_{-k,z} - B_{0,z} + I)U_h \\ &= U_h.\end{aligned}$$

□

We are now finally ready to prove the main theorem of this section.

Proof of Theorem 7.14. We construct an invertible $N_k \times N_0$ linear transformation acting on $M_0(\gamma)$. Let

$$T_{k,z} := \begin{array}{c} \\ F_0 \\ X \end{array} \begin{array}{c} \begin{array}{cc} F_0 & X \end{array} \\ \left(\begin{array}{c|c} A_{k,z} & 0 \\ \hline 0 & I_X \end{array}\right) \end{array} = \begin{array}{c} \begin{array}{cc} F_0 & X \end{array} \\ \left(\begin{array}{c|c} B_{k,z} - B_{0,z} + I_{F_0} & 0 \\ \hline 0 & I_X \end{array}\right), \end{array}$$

where I_{F_0} and I_X denote the $F_0 \times F_0$ and $X \times X$ identity matrices over GF_p, respectively. Observe that

$$T_{k,z}^{-1} = \begin{array}{c} \\ F_0 \\ X \end{array} \begin{array}{c} \begin{array}{cc} F_0 & X \end{array} \\ \left(\begin{array}{c|c} A_{k,z}^{-1} & 0 \\ \hline 0 & I_X \end{array}\right) \end{array} = T_{-k,z}.$$

Multiplying together $T_{k,z}\, M_0\, T_{k,z}^{-1}$ in block form, we get

$$T_{k,z}\, M_0\, T_{k,z}^{-1} = \left(\begin{array}{c|c} A_{k,z}\, U\, A_{k,z}^{-1} & A_{k,z}\, R_0 \\ \hline S_0\, A_{k,z}^{-1} & W \end{array}\right)$$

$$= \left(\begin{array}{c|c} U & R_k \\ \hline S_k & W \end{array}\right) = M_k,$$

where the second equality follows from lemmas 7.20, 7.21 and 7.22. Finally, this allows us to conclude that the matrices $M_0(\gamma)$ and $M_k(\gamma \circ \varphi_k^{-1})$ are similar for any $\gamma : \mathbf{P}_0 \to [0, p-1]$, with the similarity transformation given by the $F_k \times F_0$ matrix

$$S_{k,z} := Q_{k,z}^{-1} T_{k,z}.$$

As similar matrices have the same rank, the theorem now follows. □

7.2.3 Extended partitions

The analysis of the previous sections can also be applied to partitions of $N_k \times N_k$, where we now define $N_k := F_k \dot\cup (X \times H)$. In this case, we simply extend the action of each $f \in F_k$ to elements of $X \times H$, by setting $f((x,i)) := f(x)$, for all $(x,i) \in X \times H$. All partitions and bijections of partitions are then defined with respect to this extended action of f on $X \times H$. In particular, note that for each $x \in X$ and $i \in H$, there will be $\|H\|$ blocks in the partition of the form

$$\Omega^k_{(x,j),i} := \{(f,(x,j)) \in F_k \times X \mid f(x) = i\} \subset N_k \times N_k,$$

for $j \in H$. All the transformation matrices can be defined as before, except that we replace all occurrences of the $X \times X$ identity matrix with the $(X \times H) \times (X \times H)$ identity matrix.

Similarly, we can consider partitions of $Z_k \times Z_k$, where $Z_k := N_k \dot\cup Y$ and Y is a non-empty finite set. Let $\mathbf{D}_k(X,H,Y)$ denote the partition of $Z_k \times Z_k$ obtained by

- partitioning $N_k \times N_k$ according to $\mathbf{P}_k$ from before;
- partitioning $Y \times F_k$ into blocks $\Pi_y := \{(y,f) \mid f \in F_k\}$ for each $y \in Y$;

- partitioning $F_k \times Y$ into blocks $\Pi^{\mathrm{t}}_y := \{(f, y) \mid f \in F_k\}$ for each $y \in Y$; and
- placing all other elements into singleton blocks.

We can define a bijection $\varphi_k : \mathbf{D}_0(X, H, Y) \to \mathbf{D}_k(X, H, Y)$ just like before, except that we map $\Pi_y \mapsto \Pi_y$ and $\Pi^{\mathrm{t}}_y \mapsto \Pi^{\mathrm{t}}_y$ for each $y \in Y$, and map each singleton block to the corresponding singleton block. Overloading our notation, we write $M_k(\gamma)$ to denote the $Z_k \times Z_k$ matrix over GF_p defined by applying a labelling $\gamma : \mathbf{D}_0(X, H, Y) \to [0, p-1]$ to the partition. By adapting the proof of Theorem 7.14 to account for the new blocks, we get the following corollary.

Corollary 7.23 (Rank of extended partition matrices)**.** *Let $k \in H$. Then for all labellings $\gamma : \mathbf{D}_0(X, H, Y) \to [0, p-1]$, the matrices $M_0(\gamma)$ and $M_k(\gamma \circ \varphi_k^{-1})$ have the same rank over GF_p.* □

This corollary will play a key role in the next section, where we apply the **D**-partition to the vertex set of a $\mathcal{C}$-structure.

7.3 Application of the game method

In this section we apply the game method to show that for each prime p, there is a finite Abelian group H for which it holds that for any $i \in H$, there is no fixed sentence of $\mathcal{R}^{\omega}_{p;2}$ that defines the class $\mathbf{C}^i_H$. This is stated more formally by the following theorem.

Theorem 7.24 (Game winning strategy)**.** *Let p and q be distinct primes and write $H = \mathbb{Z}/(q\mathbb{Z})$. Then for each $c \in H$ and all $k \geq 2$, there is a graph G for which it holds that Duplicator has a winning strategy in the k-pebble 2-ary rank-partition game on $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$ over GF_p.*

The winning strategy that we describe is obtained by combining the set-partition scheme developed in §7.2 with a procedure for maintaining at every round in the game a mapping π from $\mathcal{C}^0_H(G)$ to $\mathcal{C}^c_H(G)$ which agrees with all currently pebbled positions and is *almost* an isomorphism. That is, at each round in the game there will only be a small subset $X \subset \mathcal{C}^0_H(G)$ where the substructures induced by X and $\pi(X)$ are not isomorphic. To ensure that Spoiler is unable to pinpoint the actual difference between the two structures, Duplicator keeps moving around the violating regions X and $\pi(X)$, to avoid the pebbled elements in each of the two structures. By playing in this way, Duplicator can at each round satisfy the algebraic condition of the rank-partition game and ensure that Spoiler is unable to find the difference between the two structures, no matter where he chooses to place his pebbles.

To explain this game strategy in a little more detail, recall that the structures $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$ are constructed from H-circuits $(G, \delta^0_{v_{\min}})$ and $(G, \delta^c_{v_{\min}})$, respectively, where $v_{\min}$ denotes the $\leqslant$-least vertex in V. For $c \neq 0$, these two circuits are exactly the same, except that the first one has nil charge on every vertex while the second one has charge c on the initial vertex $v_{\min}$ and nil charge everywhere else. At every round in the rank-partition game on $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$, the almost-isomorphism π kept by the Duplicator will be chosen so that $\pi(\mathcal{C}^0_H(G))$ and $\mathcal{C}^c_H(G)$ disagree only on the elements of the graph gadget induced by some vertex $u \in V$. It can be shown that such a mapping π corresponds to a redistribution of charge on $(G, \delta^c_{v_{\min}})$, by moving c units of charge from $v_{\min}$ to u. In these terms, Duplicator's winning strategy is to continually move the c-charge around the graph G, in order to hide the

difference between $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$ from the Spoiler. To ensure that the charge can always be moved around without violating the partial isomorphism defined by the current pebble positions, Duplicator will simultaneously play a separate graph-searching game on G, called the "cops-and-robber game". By choosing the graph G in a certain way (that is, such that it has large enough tree-width), it can be ensured that in the cops-and-robber game on G the robber always has a strategy to evade capture. By simulating the movement of the cops in the game on G according to the placement of pebbles in the rank-partition game on $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$, Duplicator can use the winning strategy of the robber to decide where to shift the charge around the H-circuit on G. This game strategy can now be summarised as follows:

1. At the beginning of each round, Duplicator has an H-redistribution $t : V \times V \to H$ on G, for which there is a vertex u such that $(\delta^c_{\nu_{\min}})^t = \delta^c_u$.

2. The redistribution t will be used to construct a bijection $\pi : \mathcal{C}^0_H(G) \to \mathcal{C}^c_H(G)$, which is an isomorphism everywhere apart from the elements of the graph gadget induced by u.

3. The mapping π is then used to construct set partitions of the two structures, using the techniques developed in §7.2. By results shown there, these partitions will satisfy the algebraic requirements of the game.

4. At the end of the round, Duplicator updates her H-redistribution t, so that it moves the charge c from u to u', where u' is a safe point for robber in the cops-and-robber game on G, starting with robber on vertex u.

The method of constructing a winning strategy in a pebble game based on a game strategy in the cops-and-robber game was originally described by Dawar and Richerby [20] (see also Atserias et al. [4] for further applications of this idea). Our contribution here is to extend this method to work in a highly *uniform* setting, with respect to possible moves in the cops-and-robber game, as we will explain in further detail later.

Throughout this section, we let p and q be distinct primes and write $H = \mathbb{Z}/(q\mathbb{Z})$ to denote the additive group of integers modulo q, with group operation $\oplus$. If $G = (V, E, \leqslant)$ is an ordered and connected graph, then we write $\nu_{\min}$ to denote the $\leqslant$-least element in V, as before. Furthermore, for each $c \in H$, we write $\gamma_c := \delta^c_{\nu_{\min}}$ to denote the charge function that assigns c to $\nu_{\min}$ and is zero everywhere else.

Our presentation is organised into four main parts. In §7.3.1 we recall the definition of tree-width and the cops-and-robber game. We then prove the existence of graphs that have arbitrarily large tree-width in addition to certain uniformity and regularity conditions, which are required for the proof of Theorem 7.24. In §7.3.2 we define H-redistributions that result in the shifting of charge from one vertex to another on a circuit, and show that such functions induce bijections between a pair of $\mathcal{C}$-structures. Then in §7.3.3 we apply the set partitions developed abstractly in §7.2 to the $\mathcal{C}$-structures $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$. By the results of §7.2, it follows that the resulting partitions will satisfy the algebraic condition of the rank-partition game. All the above will then be tied together in §7.3.4, where we describe the actual winning strategy of the Duplicator in full detail. Finally, in §7.3.5, we show that the classes $\mathbf{C}^i_H$ can be axiomatised in FOR_q for each $i \in H$. Combined with Theorem 7.24, this finally gives us a proof of Theorem 7.1, which is the main result of this chapter.

7.3.1 Tree-width and the cops-and-robber game

Tree-width is a graph parameter that, broadly speaking, measures how closely a graph resembles a tree. For instance, the tree-width of a tree, of a cycle and of the $n \times n$ grid graph is one, two and n, respectively. We will write $\mathrm{tw}(G)$ to denote the tree-width of a graph G. We will not need a formal definition of tree-width here (for details, see for instance Diestel [22, Chapter 12]). Instead, we rely on the following game characterisation of tree-width, due to Seymour and Thomas [62].

The *k-cops-and-robber game* is played by two players, one of whom controls a set of k *cops* that are attempting to catch a *robber*, which is controlled by the other player. At the beginning of each round in the game, the robber is sitting on some vertex v on the graph. The cops player then moves some or all of the cops from their current position (either on or off the game board G) and places them on vertices of the graph. The cops that are not being moved in that round are said to be *stationary*. While the chosen cops are moving to their new positions, the robber can simultaneously move along any path in the graph, starting with his current position, provided that there are no stationary cops on that path. The cops player wins the game if at some round, the robber is unable to flee without running into a cop. It is shown by Seymour and Thomas [62] that the cops player has a strategy to win the game on a graph G using $k+1$ cops if and only if G has tree-width at most k.

Definition 7.25 (Uniform winning strategies)**.** Let $d, k \geq 1$ and let G be a graph. Suppose that at some round in the k-cops-and-robber game on G, the robber is sitting on a vertex v and the cops player prepares to move $l \leq k$ cops to positions $h_1, \ldots, h_l \in V$, with stationary cops remaining at positions $s_{l+1}, \ldots, s_k \in V$. Then we say that robber has a *d-uniform escape route* from v to w, with respect to the current placement of cops, if $\deg(v) \geq d$ and there are $d+1$ simple paths

$$\begin{aligned} P_1 &: w_{11}, \ldots, w_{1m_1}, \\ &\vdots \\ P_d &: w_{1d}, \ldots, w_{dm_d}, \\ Q &: v_1, \ldots, v_n = w, \end{aligned}$$

where $n, m_1, \ldots, m_d \geq 1$, such that:

- $w_{11}, \ldots, w_{1d} \in N(v)$ are d distinct neighbours of v;
- $P_i Q$ is a simple path, for each $i \in [d]$;
- each path $P_i Q$ avoids the stationary cops.

We say that the robber player has a *d-uniform winning strategy* in the k-cops-and-robber-game on G if he can play forever in such a way that at every round in the game, the robber has a d-uniform escape route from the current position. ■

Note that, according to this definition, while the paths $P_1, \ldots P_d$ have to be distinct, they are not necessarily disjoint. That is, the only assumption is that any two distinct paths *start* with different edges — after that, they might possibly overlap.

Lemma 7.26. *For every $k \geq 1$ and $d \geq 4$, there is a d-regular connected graph G for which it holds that robber has a d-uniform winning strategy in the k-cops-and-robber game on G.*

Proof. Consider $k \geq 1$ and write $X = (V, E)$ to denote the $k \times k$ toroidal graph with vertex set $V = \{(i, j) \mid i, j \in [0, k-1]\}$ and an edge relation defined for all $(x_1, y_1), (x_2, y_2) \in V$ by

$$(x_1, y_1)(x_2, y_2) \in E :\Leftrightarrow \big((x_1 = x_2) \wedge (y_1 - y_2 \equiv \pm 1 \pmod{k})\big) \vee \big((y_1 = y_2) \wedge (x_1 - x_2 \equiv \pm 1 \pmod{k})\big).$$

For $v \in V$, write v_a, v_b, v_c, v_d for the four neighbours of v in X. For $d \geq 4$ and $v \in V$, let $G_d(v)$ denote the graph obtained from the complete graph on

$$\{(v, i) \mid i \in [d-3]\} \dot{\cup} \{(v, w) \mid w \in N_X(v)\}$$

by removing the two edges $(v, v_a)(v, v_b)$ and $(v, v_c)(v, v_d)$. We refer to vertices of the form (v, i), for $i \in [d-3]$, as *internal nodes* and vertices of the form (v, w), for $w \in N_X(v)$, as *external nodes*. It is clear by this construction that each internal node has degree d while each external node has degree $d-1$. Finally, let G be the graph obtained from the union of all the $G_d(v)$, for $v \in V$, by adding an edge between (v, w) and (w, v), whenever $vw \in E$. By this construction it follows that G is regular of degree d.

We claim that the robber player has a d-uniform winning strategy in the k-cops-and-robber game on G. First of all, we note that the toroidal graph X has tree-width k (see e.g. Bodlaender [10, §13.2]), which implies that robber has a winning strategy in the game on X with k cops. A winning strategy for the robber player in the game on G is obtained by moving the robber at each turn according to the winning strategy on X, finally coming to a rest at some internal node. More specifically, if in the game on G, the robber is sitting on some vertex in component $G_d(v)$ and the cops are located in components $G_d(v_1), \ldots, G_d(v_m)$, $m \leq k$, then the robber player consults her winning strategy in the game on X, with the X-robber on vertex v and the X-cops on vertices $v_1, \ldots, v_m$ (possibly with more than one cop on a single vertex). If, according to the winning strategy on X, the X-robber would move from v to $w \in V$, then the G-robber is moved accordingly from its current position to one of the internal nodes of the component $G_d(w)$. By playing in this way, it can be ensured that the robber can evade the k cops on G forever.

Moreover, it can be seen that this game strategy is d-uniform. To show this, suppose at some point the robber is sitting on some internal node (v, i) in component $G_d(v)$, $v \in V$. Suppose, furthermore, that in response to an advancement of the cops, the X-robber would move from v to w along a path that starts with an edge $vu \in E$. In the game on G, the robber would therefore be able to flee the cops by moving along a path that goes through the component $G_d(u)$. The get from $G_d(v)$ to $G_d(u)$ this path has to go through the edge $(v, u)(u, v) \in E(G)$. To reach the vertex (v, u), it is clear that the G-robber can leave its current position (v, i) via any of its neighbours, either by moving directly to (v, u) or by passing through some of the other vertices in $G_d(v)$, in a path that has length no more than two. This shows that this winning strategy is d-uniform, as claimed. □

7.3.2 Some properties of H-redistributions

If $t : V \times V \to H$ is an H-redistribution on an ordered graph $G = (V, E, \leqslant)$ and $v \in V$, then we write $\Delta_{t,v} : E(v) \to H$ for the function defined by $\Delta_{t,v}(vw) := t(v, w)$, for all $vw \in E(v)$. We say that t *moves* c *to* u if $\gamma_c^t := (\delta^c_{v_{\min}})^t = \delta^c_u$.

Definition 7.27 (Maps induced by redistributions)**.** Let $t : V \times V \to H$ be a redistribution on G that moves c to u, for $c \in H$ and $u \in V$. Then for any $w \in N(u)$, we write $\pi_{t,u,w} : \mathcal{C}^0_H(G) \to \mathcal{C}^c_H(G)$ for the function defined as follows.

- *Outer vertices.* For all $v, x \in V$ with $vx \in E$:

$$\pi_{t,u,w} : (v, x, i) \mapsto (v, x, i \oplus t(v, x)) = (v, x, i \oplus \Delta_{t,v}(vx)).$$

- *Inner vertices not induced by u.* For all $v \in V \setminus \{u\}$ and $f \in I(v, 0)$:

$$\pi_{t,u,w} : f \mapsto f \oplus \Delta_{t,v}.$$

- *Inner vertices induced by u.* Finally, for all $f \in I(u, 0)$:

$$\pi_{t,u,w} : f \mapsto f \oplus \Delta_{t,u} \oplus \sigma^c_{uw},$$

  where $\sigma^c_{uw} : E(v) \to H$ is the c-delta function on $E(v)$, defined for all $e \in E(v)$ by

$$\delta^c_{uw} : e \mapsto \begin{cases} c & \text{if } e = uw, \\ 0 & \text{otherwise.} \end{cases}$$

■

It is clear by this definition that $\pi_{t,u,w}$ is a bijection that preserves the preorder $\preccurlyeq$. Further properties of this mapping are summarised by the following lemma, whose proof follows directly from the above definition.

Lemma 7.28. *Let $t : V \times V \to H$ be a redistribution on G that moves c to u. Then for any $w \in N(u)$ it holds that $\pi_{t,u,w}$ is an isomorphism between $\mathcal{C}^0_H(G) \setminus I(u, \gamma_0(u))$ and $\mathcal{C}^c_H(G) \setminus I(u, \gamma_c(u))$.* □

7.3.3 Set partitions on $\mathcal{C}$-structures

We now apply the abstract partition scheme defined in §7.2 to the structures $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$ and study some properties of the resulting partitions. Here we let $G = (V, E, \leqslant)$ be an ordered and connected d-regular graph, where the integer $d \geq 4$ is chosen so that $q^{d-2} \equiv 1 \pmod{p}$.

Consider a vertex $u \in V$ and let uw be an edge incident at u. Let $t : V \times V \to H$ on G be a redistribution which moves c to u, and let $\pi_{t,u,w}$ (as in Definition 7.27) denote the corresponding partial isomorphism with respect to t, u and w. Write $F_0 = I(u, \gamma_0(u))$ and $F_c = I(u, \gamma_c(u))$ to denote the sets of inner vertices induced by u in $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$, respectively. By letting $X := E(u)$ and $Y := V(\mathcal{C}^0_H(G)) \setminus F_0$, we obtain

$$\begin{aligned} \mathbf{P}_{t,u,w} &:= \mathbf{D}_0(X, H, Y), \\ \mathbf{Q}_{t,u,w} &:= \mathbf{D}_c(\pi_{t,u,w}(X), H, \pi_{t,u,w}(Y)), \text{ and} \\ f_{t,u,w} &:= \pi_{t,u,w} \circ \varphi_{c,uw} : \mathbf{P}_{t,u,w} \to \mathbf{Q}_{t,u,w}, \end{aligned}$$

where $\mathbf{D}_0(X, H, Y)$, $\mathbf{D}_c(\pi_{t,u,w}(X), H, \pi_{t,u,w}(Y))$ and $\varphi_{c,uw}$ are defined as in §7.2. Here, the action of $\pi_{t,u,w}$ is extended to sets in the obvious way. Observing that $\|X\| = d$, we can apply Corollary 7.23 and get the following lemma.

Lemma 7.29. *For all labellings* $\gamma : \mathbf{P}_{t,u,w} \to [0, p-1]$ *it holds that* $\operatorname{rank}(M_{\gamma}^{\mathbf{P}_{t,u,w}}) = \operatorname{rank}(M_{\gamma \circ f_{t,u,w}^{-1}}^{\mathbf{Q}_{t,u,w}})$. □

At this point it will be useful to further analyse the structure of the partitions $\mathbf{P}_{t,u,w}$ and $\mathbf{Q}_{t,u,w}$. First of all, observe that if x and y are elements of $\mathcal{C}_H^0(G)$ and neither is an inner vertex from F_0, then $[\![(x,y)]\!]_{\mathbf{P}_{t,u,w}} = \{(x,y)\}$ and $f_{t,u,w}$ simply agrees with the partial isomorphism $\pi_{t,u,w}$. More formally, it holds that

$$f_{t,u,w}([\![(x,y)]\!]_{\mathbf{P}_{t,u,w}}) = [\![(\pi_{t,u,w}(x), \pi_{t,u,w}(y))]\!]_{\mathbf{Q}_{t,u,w}} = \{(\pi_{t,u,w}(x), \pi_{t,u,w}(y))\},$$

for all $x, y \in V(\mathcal{C}_H^0(G)) \setminus F_0$. We refer to singleton blocks of this form as *trivial blocks*. On the other hand, when either or both of x and y are in F_0, then the corresponding block $[\![(x,y)]\!]_{\mathbf{P}_{t,u,w}}$ will always contain more than one element. Such blocks are called *non-trivial blocks*, and they come in three different forms:

(T1) For all $(u,w,i) \in B(u,w)$ and $f \in F_0$, we have

$$[\![((u,w,i),f)]\!]_{\mathbf{P}_{t,u,w}} = \{((u,w,i),g) \mid g(uw) = f(uw)\} \text{ and}$$
$$[\![(f,(u,w,i))]\!]_{\mathbf{P}_{t,u,w}} = \{(g,(u,w,i)) \mid g(uw) = f(uw)\}.$$

(T2) For $x \in Y$ and $f \in F_0$, we have

$$[\![(x,f)]\!]_{\mathbf{P}_{t,u,w}} = \{x\} \times F_0 \text{ and } [\![(f,x)]\!]_{\mathbf{P}_{t,u,w}} = F_0 \times \{x\}.$$

(T3) Finally, for $f, g \in F_0$, with $f \ominus g = h$, we have

$$[\![(f,g)]\!]_{\mathbf{P}_{t,u,w}} = \{(f',g') \mid f', g' \in F_0 \text{ and } f' \ominus g' = f \ominus g = h\}.$$

From these definitions, it can be seen that each block of the two partitions consists of pairs of elements that all realise the same atomic type in the respective structure $\mathcal{C}_H^0(G)$ or $\mathcal{C}_H^c(G)$. Similarly, it can be seen from the definition that the mapping $f_{t,u,w}$ respects atomic types. More formally, for each $P \in \mathbf{P}_{t,u,w}$ it holds that:

- $\operatorname{atp}(\vec{a}, \mathcal{C}_H^0(G)) = \operatorname{atp}(\vec{a}', \mathcal{C}_H^0(G))$, for all $\vec{a}, \vec{a}' \in P$;
- $\operatorname{atp}(\vec{b}, \mathcal{C}_H^c(G)) = \operatorname{atp}(\vec{b}', \mathcal{C}_H^c(G))$, for all $\vec{b}, \vec{b}' \in f_{t,u,w}(P)$; and
- $\operatorname{atp}(\vec{a}, \mathcal{C}_H^0(G)) = \operatorname{atp}(\vec{b}, \mathcal{C}_H^c(G))$, for all $\vec{a} \in P$ and $\vec{b} \in f_{t,u,w}(P)$.

All these observations can be summarised as follows.

Lemma 7.30 (Properties of the partitions)**.** *For all* $P \in \mathbf{P}_{t,u,w}$ *and all* $(x_1, x_2) \in P$, $(y_1, y_2) \in f_{t,u,w}(P)$*, the mapping defined by* $x_1 \mapsto y_1$ *and* $x_2 \mapsto y_2$ *is a partial isomorphism from* $\mathcal{C}_H^0(G)$ *to* $\mathcal{C}_H^c(G)$*. Furthermore, for all* $x, y \in V(\mathcal{C}_H^0(G)) \setminus F_0$*, it holds that* $[\![(x,y)]\!]_{\mathbf{P}_{t,u,w}} = \{(x,y)\}$ *and*

$$f_{t,u,w}(\{(x,y)\}) = \{(\pi_{t,uw}(x), \pi_{t,uw}(y))\} \in \mathbf{Q}_{t,u,w}.$$

□

Lemma 7.30 shows that if, at any round in the rank-partition game on $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$, Duplicator responds to a challenge by Spoiler with partitions $\mathbf{P}_{t,u,w}$ and $\mathbf{Q}_{t,u,w}$, then any placement of pebbles that can be made over those partitions will result in a partial isomorphism, with respect to the two pairs of pebbles. It remains to be shown, however, that the Duplicator can subsequently move the charge c on G from u to another vertex u' in a way that the resulting π-bijection respects the pebbled positions on $\mathbf{P}_{t,u,w}$ and $\mathbf{Q}_{t,u,w}$. This, essentially, is what the next lemma claims.

Lemma 7.31. *For any non-trivial block $P \in \mathbf{P}_{t,u,w}$ and all $(x_1, x_2) \in P$, $(y_1, y_2) \in f_{t,u,w}(P)$, there is a function $\rho : E(u) \to H$ for which it holds that:*

(i) $\rho(E(u)) = c$;

(ii) if $s : V \times V \to H$ is a redistribution on G that moves c to some vertex v and satisfies $s(u,x) = t(u,x) \oplus \rho(ux)$, for all $ux \in E(u)$, then for any $v' \in N(v)$ it holds that

$$\pi_{s,v,v'}(x_1) = y_1 \text{ and } \pi_{s,v,v'}(x_2) = y_2.$$

Here the intuition is that $\rho(uw)$ should specify the amount of charge that should be pushed out of the u-gadget via uw, for each $uw \in E(u)$.

Proof. Consider a function $\rho : E(u) \to H$ and let $s : V \times V \to H$ be a redistribution on G that moves c to some vertex v and satisfies $s(u,x) = t(u,x) \oplus \rho(ux)$, for all $ux \in E(u)$. Observe that by definition, $\Delta_{s,u}(ux) := s(u,x) = t(u,x) \oplus \rho(ux) = \Delta_{t,u}(u,x) \oplus \rho(ux)$. Hence, $\Delta_{s,u} = \Delta_{t,u} \oplus \rho$. Also observe that for any $v' \in N(v)$, the map $\pi_{s,v,v'}$ acts on the outer vertices in $E(u) \times H$ by

$$(u,x,i) \mapsto (u,x,i \oplus \Delta_{s,u}(ux)) = (u,x,i \oplus \Delta_{t,u}(ux) \oplus \rho(ux)),$$

and acts on the inner vertices in $F_0 = I(u, \gamma_0(u))$ by

$$f \mapsto f \oplus \Delta_{s,u} = f \oplus \Delta_{t,u} \oplus \rho.$$

Let $P \in \mathbf{P}_{t,u,w}$ be a non-trivial block. Consider the following cases, according to the type of P.

(T1) Suppose $P = [\![((u,x,i_1),f)]\!]_{\mathbf{P}_{t,u,w}}$. Consider $((u,x,i_1),g_1) \in P$ and $((u,x,i_2),h_2) \in f_{t,u,w}(P)$, where $h_2 = g_2 \oplus \Delta_{t,u}$, $i_2 = i_1 \oplus \Delta_{t,u}(ux)$ and $g_1(ux) = g_2(ux)$. In this case, taking $\rho := g_2 \ominus g_1$ will be sufficient, as $\pi_{s,v,v'} : g_1 \mapsto g_1 \oplus \Delta_{t,u} \oplus \rho = g_2 \oplus \Delta_{t,u} = h_2$, and

$$\begin{aligned}\pi_{s,v,v'} : (u,x,i_1) &\mapsto (u,x,i_1 \oplus \Delta_{t,u}(ux) \oplus \rho(ux)) \\ &= (u,x,i_2 \oplus (g_2(ux) \ominus g_1(ux))) \\ &= (u,x,i_2),\end{aligned}$$

for any $v' \in N(v)$, as required.

(T2) Suppose $P = [\![(x,f)]\!]_{\mathbf{P}_{t,u,w}}$, where $x \in Y$ and $f \in F_0$. Consider $(x,g_1) \in P$ and $(\pi_{t,u}(x), g_2 \oplus \Delta_{t,u}) \in f_{t,u,w}(P)$. In this case, taking $\rho := g_2 \ominus g_1$ will be sufficient, with an argument similar to above.

(T3) Finally, suppose $P = [\![(f,g)]\!]_{\mathbf{P}_{t,u,w}}$, where $f, g \in F_0$ and $f \ominus g = h$. Consider $(f_1, g_1) \in P$ and $(f_2, g_2) \in f_{t,u,w}(P)$, with $f_1 \ominus g_1 = f_2 \ominus g_2 = h$. It follows that $f_2 \ominus f_1 = g_2 \ominus g_1$. Hence, it suffices to consider $\rho := (f_2 \ominus f_1) \ominus \Delta_{t,u} = (g_2 \ominus g_1) \ominus \Delta_{t,u}$.

□

7.3.4 Game strategy

Let p and q be distinct primes and write $H = \mathbb{Z}/(q\mathbb{Z})$, as before. Consider $k \geq 2$ and let $d \geq 4$ be an integer such that $q^{d-2} \equiv 1 \pmod{p}$. Finally, let $G = (V, E, \leqslant)$ be an ordered and connected d-regular graph that satisfies the conditions of Lemma 7.26. That is, in the k-cops-and-robber game on G, the robber will always have a d-uniform strategy to evade capture by the cops. By that same lemma, such a graph is guaranteed to exist.

In the following we put together the various technical results established previously and describe a winning strategy for Duplicator in the k-pebble 2-ary rank partition game on $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$ over GF_p, with $c \in H$. This is trivial when $c = 0$ so assume $c \neq 0$. To describe the game strategy, we will use an induction hypothesis which is stronger than the necessary partial isomorphism claim. That is, we claim that Duplicator can play in such a way that after each round in the game, (a) the map defined by the currently pebbled elements on the two structures is a partial isomorphism *and* (b) there is a vertex $u \in V$ and an H-redistribution $t : V \times V \to H$ on G for which it holds that:

(i) $\gamma^t_c := (\delta^c_{v_{\min}})^t = \delta^c_u$;

(ii) the bijection $\pi_t : \mathcal{C}^0_H(G) \to \mathcal{C}^c_H(G)$ induced by t respects the currently pebbled elements on the two structures; and

(iii) the robber player has a winning strategy in the k-cops-and-robber game on G, starting with cops at positions $v_1, \ldots, v_m$ and robber at position u, where $v_1, \ldots, v_m$ denote the vertices of G that correspond to those graph gadgets in $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$ that contain currently pebbled elements, $m \leq l$.

This induction hypothesis not only implies partial isomorphism but also comes with enough conditions to enable us to describe an inductive winning strategy, as we will show.

Now suppose that at some round in the game, Spoiler picks up a pair of pebbles from $\mathcal{C}^0_H(G)$ and the corresponding pair of pebbles from $\mathcal{C}^c_H(G)$. Let $u \in V$ and $t : V \times V \to H$ be objects satisfying the conditions of the induction hypothesis. If this is the first round of the game, then let $u = v_{\min}$ and let t be the constant-zero function on $V \times V$. Consider an arbitrary neighbour w of u and write $\pi_{t,u,w}$ to denote the bijection associated with t, u and w. Then Duplicator responds to the challenge of Spoiler with partitions $\mathbf{P}_{t,u,w}$ and $\mathbf{Q}_{t,u,w}$, and bijection $f_{t,u,w} : \mathbf{P}_{t,u} \to \mathbf{Q}_{t,u}$, as defined in §7.3.3. By Lemma 7.29, the triple $(\mathbf{P}_{t,u,w}, \mathbf{Q}_{t,u,w}, f_{t,u,w})$ satisfies the rank condition of the partition game, as required.

Suppose then that Spoiler next chooses a block $P \in \mathbf{P}_{t,u,w}$ and places the two chosen pebbles in $\mathcal{C}^0_H(G)$ on a pair in P and places the corresponding two pebbles in $\mathcal{C}^c_H(G)$ on a pair in $f_{t,u,w}(P)$. By Lemma 7.30, this placement of pebbles by the Spoiler will result in positions that preserve the partial isomorphism. This satisfies condition (a) of the induction hypothesis.

All that remains then is to show that, based on the resulting game positions, Duplicator can construct a new transition function t which will satisfy condition (b) of the induction hypothesis. To do that, Duplicator initiates a k-cops-and-robber game on G, initially with the robber on u and cops on vertices $v_1, \ldots, v_m \in V$, corresponding to the pebbled positions over $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$ at the beginning of the round. That is, the vertices $v_1, \ldots, v_m$ denote that there are pebbles on $\mathcal{C}^0_H(G)$ in each of graph gadgets $\mathcal{X}_H(v_1, 0), \ldots, \mathcal{X}_H(v_m, 0)$ and nowhere else. The same holds for $\mathcal{C}^c_H(G)$, as the pebbled positions respect the preorder $\leqslant$.

Duplicator then moves the two cops corresponding to the pebbles chosen earlier by Spoiler to the vertices on G that match the placement of pebbles on $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$. By assumption on G, this move by the cops yields d distinct paths $P_1, \dots, P_d$, all starting at u, for the robber to move along to a vertex w. Here, each path P_i goes from u via $w_i \in N(u)$, where we write $N(u) = \{w_1, \dots, w_d\}$. Let $\rho : E(u) \to H$ be a function as specified by Lemma 7.31. Intuitively, the function ρ describes for each edge $uw_i \in E(u)$ the amount of charge that should be moved out of u via uw_i, as noted earlier. This gives us a recipe for constructing a new redistribution s, moving the c units of charge from u to v, as follows:

- Firstly, for each path P_i, $i \in [d]$, define a redistribution $s_i : V \times V \to H$ by $s_i(x, y) := \rho(uw_i)$, if $(x, y) \in P_i$, $s_i(x, y) := -\rho(uw_i)$, if $(y, x) \in P_i$, and $s(x, y) := 0$ everywhere else. Note that s is well-defined in this way, as P_i is a simple path. It can be seen that s_i moves $\rho(uw_i)$ units of charge from v to w, by following the path P_i in G.
- By combining all the functions s_i, we now obtain a redistribution $s : V \times V \to H$ on G, defined by
$$s(x, y) := \bigoplus_{i=1}^{d} s_i(x, y).$$
That is, s is obtained at each $(x, y) \in V \times V$ by accumulating the charge moved from x to y over all the auxiliary functions s_i.

Since each P_i, by assumption, does not go through any of the cop positions on G, it follows that the H-redistribution s respects all the pebble positions on $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$. In particular, s respects any pebble placement over non-trivial blocks, by Lemma 7.31. Furthermore, we can see that $\gamma^s_c = \delta^c_w$, by design, and the robber player will have a winning strategy in the cops-and-robber game starting with the cops in their current position and the robber at w. This shows that Duplicator has a strategy to play in such a way that the strong induction hypothesis is satisfied at the end of each round, which concludes the proof of Theorem 7.24.

7.3.5 Axiomatisation of $\mathcal{C}$-structures in FOR

We conclude this chapter by showing that for any graph G and prime q, the structures $\mathcal{C}^0_H(G)$ and $\mathcal{C}^c_H(G)$ can be distinguished in first-order logic with rank operators over GF_q, where we write $H = \mathbb{Z}/(q\mathbb{Z})$. More precisely, we show that for every $i \in H$ there is a sentence of $\mathrm{FOR}_{q;3}$ which defines the class $\mathbf{C}^i_H$ over finite τ_H-structures. Of course, this result by itself does not conclude the proof of the main theorem of this chapter, which states there is a property of finite structures which is definable in $\mathrm{FOR}_{q;2}$ but not in $\mathcal{R}^\omega_{p;2}$ for any prime $p \neq q$. However, we show that by slightly modifying our construction of $\mathcal{C}$-structures (that is, by adding one extra vertex), we get a class of structures that can be defined in $\mathrm{FOR}_{q;2}$ but not in $\mathcal{R}^\omega_{p;2}$ for any prime $p \neq q$.

Now let q be a prime and write $H := \mathbb{Z}/(q\mathbb{Z})$ for the group of integers modulo q. As there will be no need to distinguish between addition in H and addition in GF_q, below, we will write H additively with operation $+$. Let $G = (V, E, \leqslant)$ be an ordered connected graph, where every vertex has degree at least two, and let $\gamma : V \to H$ be a charge function on G. For $v \in V$, we write $I(v) := I(v, \gamma(v))$ for the set of inner vertices associated with v, and set $I(V) := \bigcup_{v \in V} I(v)$. For each $c \in H$, let $\mathcal{S}^c_{H,G,\gamma}$ be a system of linear equations over GF_q with

variables $x_{(v,w,i)}$ for all outer vertices (v, w, i) in $\mathcal{C}_H(G, \gamma)$ and x_f for all inner vertices f, in $\mathcal{C}_H(G, \gamma)$, and the following equations.

- *Outer vertex equations.* For each $vw \in E$ and all $i, j \in H$ we add the equation:

$$x_{(v,w,i)} + x_{(w,v,j)} = i + j.$$

- *Inner vertex equations.* For each $f \in I(v)$ we add the equation:

$$\sum_{vw \in E(v)} x_{(v,w,f(vw))} = \sum_{g \in I(v)} x_g.$$

- *Total charge equation.* Finally, we add the following equation:

$$\sum_{f \in I(V)} x_f = c.$$

This construction resembles the system of linear equations we described in §5.1.1, for defining the class of even Cai-Fürer-Immerman graphs. In fact, it can be seen that the system we described there is just a special case of the more general construction above, obtained by taking $q = 2$. A similar argument to the one we gave in §5.1.1 can be given to show that, firstly, the system $\mathcal{S}^c_{H,G,\gamma}$ is definable in $\mathrm{FOR}_{q;3}$ over $\mathcal{C}_H(G, \gamma)$ and, secondly, that $\mathcal{S}^c_{H,G,\gamma}$ has a solution over GF_q if and only if $\gamma(V) = c$. Furthermore, it can be shown that the class of structures $\mathbf{C}_H := \mathbf{C}^i_H$ can be defined in first-order logic with counting, over the signature τ_H. Together, this gives us the following result.

Theorem 7.32 (Definability in FOR_q)**.** *Let q be a prime and write $H = \mathbb{Z}/(q\mathbb{Z})$. Then for every $i \in H$, there is a sentence $\varphi_i \in \mathrm{FOR}_{q;3}$ which defines the class $\mathbf{C}^i_H$ over finite τ_H-structures.* □

In the statement of Theorem 7.32, it seems that "arity three" is really a lower bound for definability in FOR_q, since we need at least two variables to index the set of equations in $\mathcal{S}^c_{H,G,\gamma}$. To see this, note that the number of equations is one more than the number of vertices in $\mathcal{C}_H(G, \gamma)$. Therefore, in order to prove the main theorem of this chapter (Theorem 7.1), it becomes necessary to modify the construction of $\mathcal{C}$-structures slightly, so that they become definable in FOR_q using only rank operators of arity two. This is shown in the proof below of the main theorem, which we restate for convenience.

Theorem 7.1 (Main theorem). For all distinct primes p and q, there is a property of finite structures which is definable in $\mathrm{FOR}_{q;2}$ but not in $\mathcal{R}^\omega_{p;2}$.

Proof. Consider a prime q and write $H := \mathbb{Z}/(q\mathbb{Z})$. Let $G = (V, E, \leqslant)$ be an ordered connected graph, where every vertex has degree at least two, and let $\gamma : V \to H$ be a charge function on G. Consider the "augmented" structure $\mathcal{C}^+_H(G, \gamma)$ obtained by adding a single vertex to $\mathcal{C}_H(G, \gamma)$, disjoint from all edge and colour relations on $\mathcal{C}_H(G, \gamma)$. We refer to this additional (constant) vertex as 'a'.

For $c \in H$, write $A\mathbf{x} = \mathbf{b}$ for the system of linear equations $\mathcal{S}^c_{H,G,\gamma}$ defined as above. With the help of the additional vertex a, it can be seen that the matrix A can be defined in first-order logic over $\mathcal{C}^+_H(G, \gamma)$ by using only two variables. That is, the sets of outer vertex and inner vertex equations can be indexed by the sets of outer and inner vertices, respectively, and the total charge equation can be indexed by the vertex a. Furthermore, the matrix $(A|\mathbf{b})$ can

also be defined over $\mathcal{C}_H^+(G, y)$ in first-order logic using only two variables; here, the vertex a is used to index the column vector $\mathbf{b}$ in the augmented matrix. This shows that there is a sentence of $\mathrm{FOR}_{q;2}$ that is satisfied in $\mathcal{C}_H^+(G, y)$ if and only if the system $\mathcal{S}_{H,G,y}^c$ has a solution.

Furthermore, it can be seen that the addition of a single disjoint vertex does not affect the isomorphism properties of $\mathcal{C}$-structures and does not change Duplicator's winning strategy in the rank-partition game. Therefore, the statement of Theorem 7.24 also holds for "augmented" $\mathcal{C}$-structures and the main theorem follows. □

Chapter 8

Conclusions and further research

In this thesis we have studied the descriptive complexity of various natural problems in linear algebra. We conclude our discussion by recalling the major results established and discussing possible areas for future study.

8.1 Summary of results

In the study of descriptive complexity there have been a number of examples [12, 36, 9] showing that fixed-point logic with counting (IFPC) falls short of defining all polynomial-time properties of finite structures. Most recently, it was shown that there is no sentence of IFPC that can define the solvability of affine equations over any fixed finite Abelian group [4], which is a natural problem in PTIME. By elementary linear algebra, this in turn shows that IFPC is not able to define the rank of a matrix over a finite field.

To address this shortcoming of the logic, we defined inflationary fixed-point logic with rank (IFPR), an extension of IFP with operators for expressing the rank of definable unordered matrix relations over a finite field of prime cardinality. These operators have a simple and natural formalisation in the well-studied framework of two-sorted numerical structures that is used to formalise the counting operators in IFPC. Among our results on the logic IFPR, we showed that it can define the solvability of systems of linear equations over any finite field. Together with the fact that rank operators can simulate counting, this implies that IFPR is strictly more expressive than IFPC. Furthermore, we showed that an even weaker logic, the extension of first-order logic with rank operators (FOR), can already define two of the other problems that were constructed to separate IFPC from PTIME, which are the problem of computing the parity of Cai-Fürer-Immerman graphs and the problem of deciding isomorphism of multipedes. These results illustrate that all known examples of polynomial-time properties that are not definable in IFPC relate to the inability of the logic to express basic properties in linear algebra.

We also studied the descriptive complexity of first-order rank logics over ordered structures. Specifically, we proved that for each prime p, FOR_p captures $\mathrm{MOD}_p\mathrm{L}$ and that $\mathrm{FOR}_{\mathbb{Q}}$ captures $\mathrm{L}^{\mathrm{C}=\mathrm{L}}$, which are natural complexity classes that characterise different levels of logarithmic space complexity. Here FOR_p is the fragment of FOR that only has rank operators over the prime field GF_p and $\mathrm{FOR}_{\mathbb{Q}}$ is the extension of first-order logic by rank operators for expressing the rank of rational-valued matrices.

While all these results demonstrate the expressiveness of logics extended by operators for defining matrix rank, it is of course possible that some other linear-algebraic property could give rise to operators with the same, or even greater, expressive power. For instance, we could alternatively have considered the extension of fixed-point logic with an operator for expressing the *determinant* of definable matrix relations, which is a natural matrix property that is well-defined for square unordered matrices. However, one of our results is that this property is already definable in IFPC for matrices over all finite fields, as well as the field of rationals and the ring of integers. More generally, we showed that IFPC can define the characteristic polynomial of any square matrix over these same domains. By similar techniques, we proved that even the rank and the minimal polynomial of rational-valued matrices are expressible in IFPC. It is therefore seen that the additional expressive power of the logic IFPR comes specifically from the ability to define matrix rank over *finite fields*.

In order to delimit the expressive power of rank logics over finite structures, we developed game-based methods for proving non-definability results. The underlying games are based on variations of Ehrenfeucht-Fraïssé-style pebble games, which form an essential tool for analysing expressiveness of other logics, such as IFP and IFPC. The game protocol that we introduced is based on partitioning the game board into a number of disjoint regions, according to some linear-algebraic criteria, which then limits the possible placement of pebbles on the board. This method of partitioning the game board turned out to be quite flexible and we showed that it can be used to give a game description of logics equipped with *any* set of generalised quantifiers.

In designing these pebble games, we had to take into consideration one important structural property that distinguishes IFPR from IFPC. It is well known that in the presence of fixed-points, unary counting operators are sufficient to count tuples of any arity [23]. On the other hand, we showed that rank logics have a strict *arity hierarchy* with respect to rank operators, where the arity of a rank operator is the number of distinct variables that it binds. More formally, writing $\mathrm{IFPR}_{p;m}$ and $\mathrm{FOR}_{p;m}$ to denote the fragment of IFPR and FOR, respectively, restricted to rank operators of arity at most m over GF_p (with p prime), we showed that the arity hierarchies $\mathrm{FOR}_{p;2} \leqq \mathrm{FOR}_{p;3} \leqq \ldots$ and $\mathrm{IFPR}_{p;2} \leqq \mathrm{IFPR}_{p;3} \leqq \ldots$ are strict for each prime p. One consequence of this is that the pebble game for IFPR that we defined had to take into account the arity of the individual rank operators, in addition to other parameters such as the number of variables.

Finally, we studied the extent to which the expressive power of rank operators depends on the characteristic of the underlying prime field. As a part of that study, we proved that for all distinct primes p and q, $\mathrm{IFPR}_{p;2} \not\equiv \mathrm{IFPR}_{q;2}$ over finite structures. The proof of this result combines linear algebra with an application of the partition-based game method developed earlier, played on a pair of highly symmetric combinatorial structures.

8.2 Future work

Our results in this thesis show that fixed-point rank logic IFPR, and more generally $\mathrm{IFPR}_{\mathrm{var}}$, is strictly more expressive than IFPC while still having polynomial-time data complexity. In symbols, $\mathrm{IFPC} \lneqq \mathrm{IFPR} \leqq \mathrm{IFPR}_{\mathrm{var}} \leqq \mathrm{PTIME}$. Despite these results, we do not have any reason to believe that either IFPR or $\mathrm{IFPR}_{\mathrm{var}}$ captures polynomial time on all finite structures. However, we do believe that in order to answer the question whether there is a logic for PTIME, it is crucial to understand in a logical context many of the natural problems in linear algebra

in general and matrix rank in particular. This really amounts to understanding the logical complexity of Gaussian elimination, a fundamental polynomial-time algorithm which plays a key role in a number of important applications.

A key step in understanding the expressibility of rank operators is to characterise the relationship between first-order rank logics and fixed-point rank logics. While we do believe that FOR $\lneqq$ IFPR, currently this is an open problem. Another open problem is to prove the separation $\mathrm{IFPR}_{p;m} \not\equiv \mathrm{IFPR}_{q;m}$ for all arities $m \geq 2$ and all distinct primes p and q, which would imply that $\mathrm{IFPR}_p \not\equiv \mathrm{IFPR}_q$ over finite structures. Already we proved this for $m = 2$ (Corollary 7.2). To extend that proof for all $m \geq 2$, it remains to "lift" the set partitions and the associated transformation matrices to all arities. But as seen from our proof in Chapter 7, the direct construction for arity two is already quite involved and so it seems that a more abstract algebraic argument is needed for the lifting to higher arities.

Another possible way to prove that $\mathrm{IFPR}_p \not\equiv \mathrm{IFPR}_q$ over finite structures, without going through the messy business of constructing higher-arity matrices, is to show that the arity hierarchy for rank logics collapses over graphs, say. Recall that our proof showing the strictness of the arity hierarchy is based on a construction of Hella. This construction shows that for each $n \geq 1$, there is a vocabulary τ_{n+1} and a class of finite τ_{n+1}-structures which is decidable in polynomial time but not definable by any sentence of $\mathcal{L}^{\omega}(\mathcal{Q}_n)$, finite-variable infinitary logic extended by all generalised quantifiers of arity n. Crucially, it can be seen that the vocabulary τ_{n+1} depends on the integer n and, in particular, contains relation symbols of arity $n + 1$. It is therefore possible that over a fixed signature, such as the language of graphs, the rank arity hierarchy collapses to a fixed level. In particular, if it can be shown that the arity hierarchy over $\mathcal{C}$-structures collapses to its second level, then the separation $\mathrm{IFPR}_p \not\equiv \mathrm{IFPR}_q$ over finite structures will follow by Theorem 7.1.

There are also further unanswered questions in relation to the partition-based pebble games that we defined. In particular, to what extent can we simplify the rules of the rank-partition game? As discussed above, the winning strategy for Duplicator we describe in Chapter 7 is rather complicated and yet it only considers the simplest case, when all matrices are defined by formulae of arity two. Even so, it can be seen from the description of that strategy that it actually takes a very particular form. That is, to show that at every round in the game Duplicator can respond to each challenge of Spoiler with valid set partitions, we explicitly construct a *single* invertible linear map, and show that this map takes each matrix obtained by labelling one partition to the corresponding matrix over the other partition. In other words, we demonstrate that the two families of matrices defined over the pair of partitions (indexed by the class of all suitable labellings over GF_p) are *simultaneously similar*. Clearly, the existence of such an explicit map is always sufficient for Duplicator to win the rank partition game, but can it be shown that this condition is also necessary? If that was the case, then the resulting game would bear a strong resemblance to the bijection game of Hella for infinitary counting logics, where instead of bijections we would have invertible linear maps.

Another possible game-related study is to consider the general partition game, which we defined to characterise definability in logics equipped with any set of generalised quantifiers. Clearly, such games in general will be quite complicated to play. One possible direction of future study is therefore to identify well-behaved families of generalised quantifiers, leading to tractable cases of the partition game.

Yet another direction of research is to study classes of structures possessing natural polynomial-time properties that are not known to be in either IFPC or IFPR. One example that has been extensively studied is the problem of determining whether a given graph has a perfect matching. It is known [9] that there is a sentence of IFPC that defines this property on bipartite graphs, but it is not known whether or not it can be defined in either IFPC or IFPR on general graphs. Recently, there have been some results relating questions about graph matching to linear algebra. For instance, Hoang, Mahajan and Thierauf [40] considered the complexity of the unique perfect matching problem on bipartite graphs, where the problem is to determine whether there is precisely one perfect matching in a given graph G. Hoang et al. show that on bipartite graphs, this problem can be reduced to questions about the characteristic polynomial of certain matrices. It can be seen, using our results in this thesis, that this construction can be defined by a formula of IFPC over any bipartite graph G. It follows that the unique perfect matching problem on bipartite graphs is definable in IFPC while for general graphs definability is not known.

It would also be interesting to investigate the relationship of IFPR with other logics which extend IFPC while remaining inside polynomial time. Here the main candidate is the logic *choiceless polynomial time* (CPT), which was defined by Blass, Gurevich and Shelah [8] in an attempt to characterise how much one can express in a logic which explicitly avoids arbitrary *choice*. This logic is formally defined by a programming language, interpreted within a time-restricted, high-level machine model which forbids unrestricted choice. While the logic CPT is strictly more expressive than fixed-point logic [8], there are still quite simple polynomial-time queries which it cannot express. To overcome this limitation, Blass et al. [9] introduced CPTC, an extension of CPT with a counting operator, which subsumes IFPC. It was shown by Dawar et al. [21] that CPT, and hence CPTC, can define the parity of Cai-Fürer-Immerman graphs. To date, it is not known whether CPTC or any other variant of CPT captures all of PTIME. In particular, it remains an open question whether the rank of a matrix can be computed or the solvability of systems of linear equations determined in CPTC. Indeed, an inclusion either way between IFPR and CPTC is unknown.

Finally, it remains to investigate how solvability of linear equations over a finite field fits more generally with solvability of linear equations over a finite Abelian group. In fact, we don't even know whether IFPR can define solvability of linear equations over a *finite ring*. Here the basic question seems to be this: for a prime p and integer $m \geq 1$, is solvability of linear equations modulo p^m definable in IFPR_p? By "linear equations modulo p^m", we mean a matrix equation $A\mathbf{x} = \mathbf{b} \mod p^m$, where the elements of the matrix A and the column vector $\mathbf{b}$ are integers. In other words, we are interested in solvability of linear equations over the ring $\mathbb{Z}_{p^m} := \mathbb{Z}/(p^m\mathbb{Z})$. When $m = 1$ then $\mathbb{Z}_p \cong \mathsf{GF}_p$ and we can define solvability of linear equations over $\mathbb{Z}_p$ in FOR_p, by Theorem 4.12. However, when $m > 1$ then $\mathbb{Z}_{p^m} \ncong \mathsf{GF}_{p^m}$ and it is not known whether IFPR_p can define solvability over such domains. It can be seen that this question can be further reduced to questions concerning feasibility of linear Diophantine equations. That is, a linear system $A\mathbf{x} = \mathbf{b}$ has a solution in $\mathbb{Z}_{p^m}$ if and only if $A\mathbf{x} + p^m\mathbf{y} = \mathbf{b}$ has a solution in $\mathbb{Z}$, where $\mathbf{y}$ is a column vector of the same dimension as $\mathbf{x}$. Can it be shown that IFPR has the expressive power to define feasibility of Diophantine equations of this form?

Bibliography

[1] E. Allender, R. Beals, and M. Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8(2):99–126, 1999.

[2] E. Allender and M. Ogihara. Relationships among PL, # L, and the determinant. *Informatique théorique et applications*, 30(1):1–21, 1996.

[3] E. Allender, K. Reinhardt, and S. Zhou. Isolation, matching, and counting uniform and nonuniform upper bounds. *Journal of Computer and System Sciences*, 59:164–181, 1999.

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

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

[6] J. Barwise and S. Feferman. *Model-theoretic logics*. Springer-Verlag, 1985.

[7] A. Blass and Y. Gurevich. A quick update on open problems in Blass-Gurevich-Shelah's article 'On polynomial time computations over unordered structures'. Online at `http://research.microsoft.com/~gurevich/annotated.html`, 2005. [Accessed July 19, 2010].

[8] A. Blass, Y. Gurevich, and S. Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100:141–187, 1999.

[9] A. Blass, Y. Gurevich, and S. Shelah. On polynomial time computation over unordered structures. *Journal of Symbolic Logic*, 67(3):1093–1125, 2002.

[10] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.

[11] G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel. Structure and importance of logspace-MOD classes. *Mathematical Systems Theory*, 25:223–237, 1992.

[12] 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.

[13] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

[14] L. Csanky. Fast parallel matrix inversion algorithms. *SIAM Journal on Computing*, 5:618–623, 1976.

[15] A. Dawar. Generalized quantifiers and logical reducibilities. *Journal of Logic and Computation*, 5(2):213, 1995.

[16] A. Dawar. On the descriptive complexity of linear algebra. In *WoLLIC '08*, volume 5110 of *Lecture Notes in Computer Science*, pages 17–25. Springer-Verlag, 2008.

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

[18] A. Dawar and B. Holm. Pebble Games for Logics with Counting and Rank. In *Proceedings of Logical Approaches to Barriers in Computing and Complexity*, 2010.

[19] 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.

[20] A. Dawar and D. Richerby. The power of counting logics on restricted classes of finite structures. In *CSL 2007: Computer Science Logic*, volume 4646 of *Lecture Notes in Computer Science*, pages 84–98. Springer-Verlag, 2007.

[21] A. Dawar, D. Richerby, and B. Rossman. Choiceless polynomial time, counting and the Cai-Fürer-Immerman graphs. *Annals of Pure and Applied Logic*, 152(1-3):31–50, 2008.

[22] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, fourth edition, 2010.

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

[24] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49(129-141):13, 1961.

[25] D. K. Faddeev and V. N. Faddeeva. *Computational Methods of Linear Algebra (Translated by RC Williams)*. Freeman, 1963.

[26] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings, Vol 7*, pages 43–73, 1974.

[27] R. Fraïssé. Sur quelques classifications des systèmes de relations. *Publications Scientifiques de l'Université d'Algerie, Séries A*, 1:35–182, 1954.

[28] M. Goldmann and A. Russell. The complexity of solving equations over finite groups. *Information and Computation*, 178(1):253–262, 2002.

[29] E. Grädel and M. Otto. Inductive definability with counting on finite structures. In *Selected Papers from the Workshop on Computer Science Logic*, pages 231–247. Springer-Verlag, 1993.

[30] M. Grohe. Fixed-point logics on planar graphs. In *Proceedings of the 13th IEEE Symposium on Logic in Computer Science*, pages 6—15, 1998.

[31] M. Grohe. Definable Tree Decompositions. In *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science*, pages 406–417, 2008.

[32] M. Grohe. The quest for a logic capturing PTIME. In *Proceedings of the 22nd IEEE Symposium on Logic in Computer Science*, pages 267–271, 2008.

[33] M. Grohe. Fixed-point definability and polynomial time on graph with excluded minors. In *Proceedings of the 25th IEEE Symposium on Logic in Computer Science*, pages 179 – 188, 2010.

[34] M. Grohe and J. Mariño. Definability and descriptive complexity on databases of bounded tree-width. In *Proceedings of the 7th International Conference on Database Theory*, volume 1540 of *Lecture Notes in Computer Science*, pages 70–82. Springer-Verlag, 1999.

[35] 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.

[36] Y. Gurevich and S. Shelah. On finite rigid structures. *Journal of Symbolic Logic*, 61:61–549, 1996.

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

[38] L. Hella, P. G. Kolaitis, and K. Luosto. Almost everywhere equivalence of logics in finite model theory. *Bulletin of Symbolic Logic*, 2:422–443, 1996.

[39] U. Hertrampf, S. Reith, and H. Vollmer. A note on closure properties of logspace MOD classes. *Information Processing Letters*, 75:91–93, 2000.

[40] T. M. Hoang, M. Mahajan, and T. Thierauf. On the Bipartite Unique Perfect Matching Problem. *Lecture Notes in Computer Science*, pages 453–464, 2006.

[41] T. M. Hoang and T. Thierauf. The complexity of the characteristic and the minimal polynomial. *Theoretical Computer Science*, 295(1-3):205–222, 2003.

[42] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1990.

[43] N. Immerman. Languages which capture complexity classes. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 347–354. ACM, 1983.

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

[45] N. Immerman. Expressibility as a complexity measure: Results and directions. In *Second Structure in Complexity Theory Conference*, pages 194–202, 1987.

[46] 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.

[47] Etessami K. Counting quantifiers, successor relations, and logarithmic space. *Journal of Computer and System Sciences*, 54:400–411, 1997.

[48] P. G. Kolaitis and J. A. Väänänen. Generalized quantifiers and pebble games on finite structures. In *Proceedings of the 7th Annual IEEE Symposium on Logic in Computer Science*, pages 348–359, 1992.

[49] P. G. Kolaitis and M. Y. Vardi. Fixpoint logic vs. infinitary logic in finite-model theory. In *Logic in Computer Science, 1992. LICS'92., Proceedings of the Seventh Annual IEEE Symposium on*, pages 46–57, 1992.

[50] D. C. Kozen. *The Design and Analysis of Algorithms*. Springer-Verlag, 1992.

[51] S. Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics*. Springer-Verlag, third edition, 2002.

[52] L. Libkin. *Elements of finite model theory*. Springer-Verlag, 2004.

[53] R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications*. Cambridge University Press, 1994.

[54] P. Lindström. First order predicate logic with generalized quantifiers. *Theoria*, 32:186–195, 1966.

[55] P. Lindström. On extensions of elementary logic. *Theoria*, 35(1):1–11, 1969.

[56] K. Luosto. Classifying unary quantifiers. Online at `http://www.helsinki.fi/~kluosto/mate/cuqa.pdf`, 2009. [Accessed April 12, 2011].

[57] A. Nash, J. B. Remmel, and V. Vianu. PTIME queries revisited. In *Proceedings of the International Conference on Database Theory 2005*, volume 3363 of *Lecture Notes in Computer Science*, pages 274–288, 2005.

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

[59] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

[60] B. Poizat. Deux ou trois choses que je sais de L_n. *Journal of Symbolic Logic*, 47(3):641–658, 1982.

[61] W. L. Ruzzo. Space-bounded hierarchies and probabilistic computations. *Journal of Computer and System Sciences*, 28(2):216–230, 1984.

[62] P. D. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58(1):22–33, 1993.

[63] S. Toda. Counting problems computationally equivalent to computing the determinant. Technical report, Department of Computer Science, University of Electro-Communications, Tokyo, Japan, 1991.

[64] J. Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004.

[65] M. Y. Vardi. The complexity of relational query languages. In *STOC '82: Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 137–146. ACM Press, 1982.

[66] W. P. Wardlaw. Matrix representation of finite fields. *Mathematics Magazine*, 67(4):289–293, October 1994.