# Linear Equation Systems and the Search for a Logical Characterisation of Polynomial Time

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Gymnasiallehrer
**Wied Pakusa**
aus Siegburg

Berichter: Universitätsprofessor Dr. Erich Grädel
Universitätsprofessor Dr. Martin Otto
Professor Dr. Anuj Dawar

Tag der mündlichen Prüfung: 7. Dezember 2015

## Abstract

The search for a logic which captures polynomial time is one of the most important challenges in finite model theory. During the last years, significant new insights were obtained by the systematic study of the descriptive complexity of queries from (linear) algebra. These investigations were initiated in 2007 by a striking result of Atserias, Bulatov, and Dawar, who showed that fixed-point logic with counting (FPC) cannot define the solvability of linear equation systems over finite Abelian groups. Their result triggered the development of new candidates for capturing polynomial time, for instance of rank logic (FPR), which was introduced by Dawar, Grohe, Holm, and Laubner in 2009. Before that, only few other candidates had been proposed, of which certainly the most important one is Choiceless Polynomial Time (CPT), developed by Blass, Gurevich, and Shelah in 1999. This thesis continues the search for a logic capturing polynomial time in the light of the following leading questions.

(I) How can the algorithmic principles for solving linear equation systems be captured by logical mechanisms (such as operators or quantifiers)?

(II) Are there interesting classes of structures on which the solvability of linear equations systems can be used to capture polynomial time?

Ad (I), we study in Chapter 3 the inter-definability of linear equation systems over finite Abelian groups, rings, and modules. Our aim is to transform linear equation systems over these algebraic domains into equivalent linear equation systems over *simpler* domains, such as fields, or cyclic groups, via a reduction which is definable in fixed-point logic. For linear equation systems over *ordered* Abelian groups, rings, and modules, and also for certain interesting classes of *unordered* commutative rings, we obtain a reduction to cyclic groups. Moreover, we establish a reduction to commutative rings for the general case. In Chapter 4, we study rank logic (FPR), which extends FPC by operators to compute the rank of definable matrices over *finite fields*. Our main result validates a conjecture of Dawar and Holm: rank operators over different prime fields have incomparable expressive power. An important consequence is that rank logic, in the original definition with a distinct rank operator for every prime, fails to capture polynomial time, and should be replaced by a more powerful version with a *uniform* rank operator. We further show that, in the absence of counting, solvability quantifiers are weaker than rank operators.

Ad (II), we introduce in Chapter 5 a class of linear equation systems, so called *cyclic linear equation systems*, which are structurally simple, but general enough to describe the Cai-Fürer-Immerman query, and thus separate FPC from polynomial time. Our main result is that CPT can express the solvability of cyclic linear equation systems. In Chapter 6, we use this definability result to show that CPT captures polynomial time on structures *with Abelian colours*, a class containing many of the known queries which separate FPC from polynomial time. Our result further solves an open question of Blass, Gurevich, and Shelah: the isomorphism problem for multipedes is definable in CPT.

## Zusammenfassung

Die Suche nach einer Logik für Polynomialzeit ist eines der wichtigsten offenen Probleme im Gebiet der Endlichen Modelltheorie. In den letzten Jahren wurden neue Erkenntnisse erzielt durch die Analyse der deskriptiven Komplexität von Problemen aus der (Linearen) Algebra. Gestartet wurden diese Untersuchungen 2007 nach einem Resultat von Atserias, Bulatov und Dawar welches zeigt, dass Fixpunktlogik mit Zählen (FPC) die Lösbarkeit linearer Gleichungssysteme über endlichen Abelschen Gruppen nicht ausdrücken kann. Dieses Ergebnis führte nicht zuletzt zur Definition neuer Kandidaten von Logiken für Polynomialzeit, zum Beispiel von Ranglogik (FPR), welche 2009 von Dawar, Grohe, Holm und Laubner eingeführt wurde. Ein weiterer wichtiger Kandidat ist Choiceless Polynomial Time (CPT), eine Logik die bereits 1999 von Blass, Gurevich und Shelah vorgeschlagen wurde. Diese Arbeit setzt die Suche nach einer Logik für Polynomialzeit fort, geleitet durch die folgenden Fragen.

(I) Wie repräsentiert man algorithmische Techniken zum Lösen linearer Gleichungssysteme durch logische Mechanismen (Quantoren, Operatoren)?

(II) Auf welchen Strukturklassen kann das Lösbarkeitsproblem für lineare Gleichungssysteme genutzt werden, um Polynomialzeit einzufangen?

Zu (I) betrachten wir in Kapitel 3 das Lösbarkeitsproblem für lineare Gleichungssysteme über endlichen Abelschen Gruppen, Ringen und Moduln. Unser Ziel ist die Reduktion auf *einfache* Bereiche, z.B. auf Körper oder zyklische Gruppen, wobei die Transformationen in Fixpunktlogik definierbar sein soll. Wir zeigen, dass eine Reduktion auf zyklische Gruppen möglich ist für Gleichungssysteme über *geordneten* Gruppen, Ringen und Moduln, und auch für Systeme über gewissen Klassen kommutativer Ringe. In Kapitel 4 betrachten wir Ranglogik, das heißt die Erweiterung von FPC um Operatoren, die den Rang von Matrizen über endlichen Körper definieren. Unser Hauptergebnis bestätigt eine Vermutung von Dawar und Holm: Rangoperatoren über verschiedenen Primkörpern haben unterschiedliche Ausdrucksstärke. Eine wichtige Folgerung ist, dass Ranglogik, in der ursprünglichen Definition mit einem separaten Rangoperator für jede Primzahl, nicht Polynomialzeit einfängt, und durch die stärkere Logik mit uniformem Rangoperator ersetzt werden sollte. Weiter zeigen wir, dass, ohne Hinzunahme von Zähloperatoren, Matrizenrang nicht durch entsprechende Lösbarkeitsquantoren ausgedrückt werden kann.

Zu (II) führen wir in Kapitel 5 *zyklische* lineare Gleichungssysteme ein. Solche System sind strukturell einfach, aber dennoch stark genung, um das Cai, Fürer, Immerman Problem zu kodieren, und damit um FPC von Polynomialzeit zu trennen. Unser Hauptergebnis ist, dass CPT die Lösbarkeit zyklischer Gleichungssysteme ausdrücken kann. In Kapitel 6 benutzen wir dieses Resultat um zu zeigen, dass CPT Polynomialzeit einfängt auf Strukturen mit *Abelschen Farben*. Dieses Ergebnis löst auch ein offenes Problem von Blass, Gurevich und Shelah: das Isomorphieproblem von *multipedes* ist definierbar in CPT.

# Contents

# Chapter 1

# Introduction

Finite model theory is the study of model-theoretic questions on classes of finite structures, such as the analysis of the expressive power of logics, or the search for relations between properties of axiom systems and their models. The motivation to focus on *finite* structures primarily stems from the manifold applications of mathematical logic in computer science, for example in database theory, in complexity theory, or in artificial intelligence, where most objects of interest are finite, see [33, 34, 49, 65, 72, 79] for details. From the mathematical perspective, restricting to finite structures opens the door to study novel model-theoretic questions, and, at the same time, it creates the demand for new proof techniques which are tailored to study these questions over finite structures, as many classical model-theoretic tools apply to infinite structures only.

This thesis focuses on *descriptive complexity theory*, a particular research area of finite model theory which studies connections between *definability* and *algorithmic complexity*. More precisely, the aim is to find relations between the definability of a structural property $\mathcal{P}$ in a *logical formalism* and the *computational complexity* of deciding the property $\mathcal{P}$ algorithmically, measured in terms of the required algorithmic resources (such as time, space or the number of processors used in a parallel computation). Ideally, one tries to establish a precise match, that is, given an algorithmic complexity class $\mathcal{C}$ (for example, $\mathcal{C}$ can be NP, PTIME or LOGSPACE), we want to find a logic $\mathcal{L}$ which can define *precisely* those properties of finite structures that can be decided with resource bounds according to $\mathcal{C}$.

This central idea of *capturing* (algorithmic) complexity classes by formal logics is exemplified by a seminal result of Ronald Fagin, which is commonly identified as the birth of descriptive complexity theory. In 1974, Fagin proved that the properties of structures that can be decided in non-deterministic polynomial time (NP) are exactly those properties that can be expressed in existential second-order logic ($\Sigma^1_1$) [35]. His result gives a completely new and machine-independent characterisation of the NP-properties, and it initiated the search for such insightful correspondences also for other complexity classes.

In fact, having a characterisation of a complexity class $\mathcal{C}$ by a logic $\mathcal{L}$

is desirable for many reasons and we want to mention some of them. Most importantly, it may be possible to transfer techniques and results between the areas of mathematical logic and complexity theory, in order to obtain new tools and ideas to solve (longstanding open) problems from both areas. Probably, the most significant example would be the separation of complexity classes (such as PTIME and NP) by model-theoretic methods (or, vice versa, the separation of complexity classes might yield new insights about the relation of the corresponding logics). Furthermore, we can learn something about the structure of algorithms, specifically about their basic building blocks: what are the logical elements in $\mathcal{L}$ (quantifiers, operators, induction principles) which are needed to cover all algorithms with certain resource bounds? Of course, depending on the logic $\mathcal{L}$, we can also get new handles on the structure of classes from $\mathcal{C}$. For instance, if $\mathcal{L}$ allows the stratification along natural parameters (e.g. the number of variables, the quantifier depth or the alternation depth), then we know that, up to a certain threshold, every problem in $\mathcal{C}$ is invariant with respect to these fragments of $\mathcal{L}$. In other words, we could retrieve knowledge about structural properties of classes of complexity $\mathcal{C}$ by analysing syntactic fragments of $\mathcal{L}$ by model-theoretic techniques.

From Fagin's result, it readily follows that every level of the polynomial-time hierarchy can be captured by the corresponding fragment of second-order logic. In particular, second-order logic captures the full polynomial-time hierarchy. Also for other important classes above NP, such as PSPACE or EXPTIME, one soon found characterisations in terms of natural logics [64]. Unfortunately, the situation for complexity classes below NP is entirely different. Until today, we have no logical characterisation of any of the important complexity classes below NP such as PTIME, NLOGSPACE or LOGSPACE. In fact, all capturing results we have for such classes only hold when we impose auxiliary structural assumptions on the input structures. A particular important case arises when we restrict to *ordered* structures, that is to structures which have a built-in linear order on their universe. Interestingly, for the case of ordered structures, we have logical characterisations of PTIME (see below), of NLOGSPACE, and of LOGSPACE by means of very natural languages. For instance, NLOGSPACE is captured by the extension of first-order logic by a transitive closure operator and we obtain LOGSPACE if we replace this operator by an operator for deterministic transitive closures, see also [50, 61, 63, 64, 87].

However, that an abstract structure (like a graph) contains a complete linear order on its universe is a very specific and rather atypical structural property (we discuss the significant role of orders in the following section more precisely). Thus, the search for logics capturing complexity classes below NP over general finite structures, or at least over more natural classes, remains the central challenge in descriptive complexity theory.

In this thesis, we continue the search for a logic for polynomial time. Specifically, we take up recent insights about the relevance of linear algebra in this quest, most importantly, of the solvability problem for linear equation systems over finite algebraic domains.

## 1.1 A logic for polynomial time

The widely accepted theoretical model for "efficient computability" is the complexity class polynomial time (PTIME). Unfortunately, as such, this model does not provide much insight neither into the *nature* of efficiently solvable problems, nor into the *structure* of efficient algorithms. Hence, it would be valuable to have an alternative, specifically a *logical*, characterisation of PTIME, to obtain a clearer view onto these interesting aspects of efficient computability.

The first ones who raised the question for a logical characterisation of polynomial time were Chandra and Harel in the context of database theory. More precisely, they asked for a *natural* database query language which can express exactly those queries which are decidable in polynomial time [23]. The benefit of such a language is immediate: it would be as powerful as possible with the guarantee that each expressible query can be evaluated efficiently.

Of course, if we want to answer the question of Chandra and Harel, then we first have to specify what is considered to be a "natural query language", or, in our terms, what is consider to be a *logic*. Moreover, it strongly matters whether, and also how, formulas of this logic can be translated *effectively* into equivalent polynomial-time algorithms (at least, for the application as a database query language, the mere existence of an equivalent polynomial-time algorithm is not sufficient, but we require an effective way to construct an equivalent algorithm from a given formula). In [53], Gurevich made these requirements precise and formulated the following question which became the central challenge in the field of descriptive complexity theory: Is there a logic which captures polynomial time? We discuss the precise definition of this question in Section 2.1. For further details and for a nice review of different effectivity conditions, we recommend the survey of Martin Grohe [47].

Gurevich himself conjectured that there is no logic capturing polynomial time. However, proving his conjecture is extremely difficult, since it would separate PTIME from NP (recall that NP *is captured* by the logic $\Sigma^1_1$). In fact, the PTIME vs. NP-question is often considered as a central motivation for the search of a logic capturing polynomial time. The hope is that, even if Gurevich's conjecture turns out to be false and if we find a logic $\mathcal{L}$ which captures PTIME, then we can adapt tools from finite model theory for $\mathcal{L}$ (like Ehrenfeucht-Fraïssé games) to separate the logic $\mathcal{L}$ from the logic $\Sigma^1_1$ (which would separate PTIME and NP).

**Order invariance** Finding a logical characterisation of polynomial time turns out be extremely challenging which is due to the following mismatch between logics and algorithms. Although many algorithms are designed to decide properties of abstract structures, like graphs or relational databases, their inputs are not abstract structures themselves, but *encodings* of such structures as strings (for example, the adjacency matrix of a graph). On the other hand, the formulas of a logic are evaluated directly on the structure and not on (one of) its string encoding(s). Hence, algorithms and logics both

decide (or define) structural properties of the same objects, that is of relational structures, but they operate on very different levels of representation.

At first glance, the differences between abstract structures and their string encodings might not seem too significant. The crucial point is, however, that by each of the known and efficient ways, to represent a structure by a string, we always have to specify (at least, implicitly) a linear order on the elements of the encoded structure. Consider, for example, a string $\mathrm{enc}(\mathcal{G})$ which specifies the adjacency matrix of some graph $\mathcal{G}$. Then, indeed, the order in which the vertices of $\mathcal{G}$ appear in the string $\mathrm{enc}(\mathcal{G})$, that is the order of rows and columns of the adjacency matrix, determines a linear order $<$ on the vertices of the encoded graph. Hence, the string $\mathrm{enc}(\mathcal{G})$ does not only encode the graph $\mathcal{G}$ itself, but it encodes an *ordered* version $(\mathcal{G}, <)$ of the graph $\mathcal{G}$.

Of course, if we actually want to represent an *ordered* graph $(\mathcal{G}, <)$ by a string, then we can just take the built-in linear order $<$ of $(\mathcal{G}, <)$ to fix the order of vertices for the string encoding. In other words, we can encode *ordered* graphs in such a way that the implicit linear order, which is determined by the string encoding, coincides with the linear order which was already part of the original (ordered) graph $(\mathcal{G}, <)$ (hence, we have not added any *additional* structure by the encoding). In general, however, a relational structure $\mathfrak{A}$, such as a graph, does *not* contain a built-in linear order. Still, if we want to represent $\mathfrak{A}$ by a string, then this string always encodes a structure which, in contrast to $\mathfrak{A}$, has a linear order on its universe. This, in turn, leads to the following fundamental question: how can we encode $\mathfrak{A}$, which has *no* built-in *linear order*, by a string $\mathrm{enc}(\mathfrak{A})$, which represents an *ordered* structure $(\mathfrak{A}, <)$, or, in other words, how can we reasonably identify the *unordered* structure $\mathfrak{A}$ with an *ordered* structure $(\mathfrak{A}, <)$?

Unfortunately, at least as long as we are interested in *efficient* encodings, the answer is: we don't know. Of course, this is not really satisfactory, since we obviously want to give abstract structures as inputs to algorithms. The common solution thus simply looks as follows. Given an abstract structure $\mathfrak{A}$, we first extend $\mathfrak{A}$ by *some* linear order $<$ on its universe, and then we take the string representation $\mathrm{enc}(\mathfrak{A}, <)$ of the *ordered* structure $(\mathfrak{A}, <)$ as *one of the possible* string representations of $\mathfrak{A}$. The obvious shortcoming of this approach is that the same structure $\mathfrak{A}$ is encoded in many *different* ways (the concrete encoding depends on the order that we choose on the universe). In particular, algorithms will get *different* string encodings of isomorphic structures $\mathfrak{A} \cong \mathfrak{B}$ as inputs, although isomorphic structures certainly share the *same* structural properties. Still, since an algorithm can just ignore the implicit linear order, which only comes from the encoding and which is not part of the original structure, this is a practicable way to represent structures by strings.

For completeness, let us remark that, in principle, there is a way to obtain a *unique* string encoding $\mathrm{enc}(\mathfrak{A})$ also for every (unordered) structure $\mathfrak{A}$ (for example, one could just take the lexicographically least encoding $\mathrm{enc}(\mathfrak{A}, <)$ among all possible extensions $(\mathfrak{A}, <)$ of $\mathfrak{A}$ by a linear order $<$). This also means that, basically, there is a way to resolve the above problem that isomorphic

structures are mapped to different string encodings. Unfortunately, it is far open whether such a *canonical* encoding $\text{enc}(\mathfrak{A})$ of $\mathfrak{A}$ can be computed efficiently (this problem is known as the *structure canonisation problem*, which is at least as hard as the *structure isomorphism problem*, see [68] for a survey and for references to background). Hence, if we want to encode structures as inputs for polynomial-time algorithms (and, of course, the same holds for all complexity classes below), then the approach of encoding structures by a *canonical* string is not feasible, since there might be no efficient way of precomputing a canonical encoding $\text{enc}(\mathfrak{A})$ of $\mathfrak{A}$ in the first place.

As a consequence, we cannot escape the fact that isomorphic structures are encoded by different strings. This, however, also means that not all algorithms really decide properties of structures. For example, an algorithm which accepts the encoding $\text{enc}(\mathfrak{A}, <_1)$ of $\mathfrak{A}$ with respect to some linear order $<_1$, but which rejects an encoding $\text{enc}(\mathfrak{A}, <_2)$ of $\mathfrak{A}$ with respect to a different order $<_2$, does not decide a property of the structure $\mathfrak{A}$, but rather a property depending on the specific encoding of $\mathfrak{A}$. This means that we have to restrict to such algorithms which *do not distinguish between different encodings of isomorphic structures*. This *semantical* requirement is known as *order invariance*. To put it differently, an order-invariant algorithm is allowed to use the linear order on the universe of the encoded input structure, which is only available due to the encoding, but which is *not* part of the original structure, but the semantical condition for the algorithm is that for *all* possible reorderings of this input structure the algorithm has to produce the same output.

Unfortunately, the notion of order invariance is, as a semantical property of algorithms, undecidable. This means that the set of polynomial-time algorithms which decide properties of structures is not recursive. On the other hand, the set of formulas of a logic is decidable (this is one of the natural and basic requirements in the definition of Gurevich [53] for logics). This points to a different incarnation of this crucial dissimilarity between algorithms and logics. To put it differently, the question for a logic for polynomial time really asks whether there is a *recursive* way to capture the order-invariant polynomial-time algorithms: more precisely, is there a recursive way to construct a family of polynomial-time algorithms which are order-invariant and which capture all polynomial-time properties (or, in other words, is there a *recursive enumeration* of all polynomial-time properties [23, 47])?

**Fixed-point logics** First-order logic plays the central role in classical model theory which is due to the perfect balance between its expressive power and the strong metalogical properties (most importantly, compactness and completeness). On finite structures, however, first-order logic is often too weak to define interesting classes of structures. In particular, as witnessed by the theorems of Hanf and Gaifman [33], first-order logic can only speak about *local* properties of finite structures and, in particular, it fails to express properties based on recursive definitions (for example, reachability in graphs).

This is the reason why in finite model theory, and especially in descriptive complexity theory, there is a strong focus on *extensions* of first-order logic (or of certain of its fragments, like propositional modal logic, or conjunctive queries) by different kinds of *fixed-point operators* which are meant to remedy the lack of expressing recursive definitions. The common feature of such operators is that they can determine fixed points of definable functions which transform relations. More precisely, consider a formula $\varphi(X, \bar{x})$ with a free second-order variable $X$ of arity $k$ and with a tuple of free variables $\bar{x}$ of length $k$. Then on every structure $\mathfrak{A}$, the formula $\varphi$ defines an operator $F_\varphi$, which maps a relation $R \subseteq A^k$ to the relation $F_\varphi(R) = \{\bar{a} : \mathfrak{A} \vDash \varphi(R, \bar{a})\}$, and a fixed point of the operator $F_\varphi$ is just a relation $R \subseteq A^k$ which satisfies $F_\varphi(R) = R$.

Of course, in general, the function $F_\varphi$ does not need to have a fixed point, and even if it has one, then, in most cases, the fixed point will not be unique. Still, there are several ways to guarantee the existence of a fixed point of $F_\varphi$ with distinguished properties. Probably the most important approach is to enforce that $F_\varphi$ is monotone, that is $F_\varphi(R) \subseteq F_\varphi(S)$ holds for all $R \subseteq S \subseteq A^k$. In this case, it is well-known that, by the Knaster-Tarski Theorem, $F_\varphi$ has a *least* and a *greatest* fixed point. Moreover, there is a simple syntactical criterion for $\varphi$ which guarantees monotonocity of $F_\varphi$: if the relation variable $X$ only occurs positively in $\varphi(X, \bar{x})$, then $F_\varphi$ is monotone. This allows us to define new fixed-point operators lfp and gfp (for least and greatest fixed-points) which take a formula $\varphi(X, \bar{x})$ (with positive occurrence of $X$) as input and which express, in each structure $\mathfrak{A}$, the least and greatest fixed-points of $F_\varphi$, respectively. The extension of first-order logic by the operators lfp (and gfp) is known as *least fixed-point logic* (LFP), see [33, 40]. Since many mathematical and structural properties can be specified in terms of least and greatest fixed-points (e.g. reachability, bisimulation, or winning regions in games), LFP is a powerful logic which extends FO by the possibility to express recursive definitions in a natural, and mathematically quite elegant, way.

Another approach to associate a fixed point with $F_\varphi$ is via the corresponding *inflationary* operator $G_\varphi$, which is defined as $G_\varphi(R) = R \cup F_\varphi(R)$. Indeed, this inflationary operator $G_\varphi$ always has an *inductive* fixed point which is constructed by setting $X^0 := G_\varphi(\varnothing)$ and $X^\alpha := G_\varphi(\bigcup_{\beta<\alpha} X^\beta)$ for ordinals $\alpha > 0$. By definition, the stages $X^\alpha$ form an increasing chain $X^0 \subseteq X^1 \subseteq \cdots \subseteq X^\alpha \subseteq \cdots$ which reaches a fixed point $X^\infty = X^\alpha = X^{\alpha+1}$ for some ordinal $\alpha$. This inductive fixed point $X^\infty$ of $G_\varphi$ is called the *inflationary fixed point* of $F_\varphi$. Observe that the inflationary fixed point of $F_\varphi$ always exists, no matter what kind of properties the formula $\varphi$ has (in particular, the relation variable $X$ can also occur negatively in $\varphi$). Analogously to least fixed-point logic, we can extend first-order logic by a new fixed-point operator ifp which takes an arbitrary formula $\varphi(X, \bar{x})$ as input and which defines the inflationary fixed point of $F_\varphi$. The resulting logic is known as *inflationary fixed-point logic* (IFP) [33, 40].

Obviously, the fixed-point mechanisms of LFP and IFP are inherently different. While least fixed points specify minimal objects which are stable with respect to certain conditions, inflationary fixed points capture inductive

definitions and the algorithmic idea of iteration. Surprisingly, it turns out that LFP and IFP still have the same expressive power, that is LFP = IFP. One direction of this equivalence is easy to obtain: It follows from a result of Knaster and Tarski that for monotone operators, the inflationary and the least fixed-point of $F_\varphi$ coincide, thus LFP $\subseteq$ IFP. The other direction is far more difficult to prove (in particular, for inductive definitions which reach their fixed points only after infinitely many steps), see [54, 69, 70]. Still, because of this equivalence it is justified to use LFP and IFP interchangeably. However, to avoid problems with monotonocity, whenever we speak of fixed-point logic in this thesis, then we mean *inflationary* fixed-point logic, if not stated otherwise. For more background on the numerous incarnations of fixed-point logics, on their role in finite model theory, on their various applications in verification, database theory, complexity theory, and so on, and, in particular, on their strong connections to game theory, we recommend [40, 41].

Let us come back to the question for a logic capturing polynomial time and to the role of ordered structures. The relevance of fixed-point logics in this context is strongly highlighted by a seminal result of Immerman and Vardi. In [61] and [87] they independently proved that on the class of ordered structures (least) fixed-point logic captures polynomial time. In other words, the *Immerman-Vardi Theorem* solves the main open question of descriptive complexity theory on the class of ordered structures.

The proof consists of two important steps. First, it shows that ordered structures can be transformed into a canonical string encoding by using a first-order interpretation. This implies that, over ordered structures, the differences between logics and algorithms, which we discussed earlier, disappear. Secondly, it shows that, over strings, it is possible to simulate algorithms (say in the form of a Turing machine) in LFP. The main argument is that configurations can be represented as relations (over the string) in such a way that the transition function is first-order definable and such that the fixed-point mechanism of LFP can be used to express the reachability of an accepting configuration.

Based on similar arguments one also found natural logics which capture other important complexity classes, like Logspace or NLogspace (on the class of ordered structures). However, despite of these nice capturing results, the assumption of having a complete linear order on the universe imposes a very strong, and rather atypical, auxiliary structure on the inputs. Thus, this situation leads to the question of how far these capturing results can be transferred to other, more natural, classes of finite structures.

**Fixed-point logic with counting** Unfortunately it turns out that over general, that means unordered, structures, the fragment of polynomial time which is captured by LFP is not very robust. On the one hand, LFP can express many Ptime-complete problems also in the absence of an order, for instance alternating reachability in game graphs, but on the other hand, LFP fails to express simple queries which involve counting, for example saying that a

structure has an even number of elements. This situation motivated Immerman to propose an extension of LFP by a *counting mechanism*. He asked whether the resulting logic, known as *fixed-point logic with counting* (FPC), suffices to capture polynomial time on general structures [61, 62]. In other words, is the order only needed to express counting properties?

It was only a few years later that Immerman himself together with Cai and Fürer proved that counting is not sufficient to compensate for a linear order on the universe [21]. By using a clever construction they came up with a query of graphs which can be decided in polynomial time, but which cannot be defined in fixed-point logic with counting. Since then, this *C(ai)F(ürer)I(mmerman)-construction* has been applied frequently in many different areas and it became one of the important standard tools in descriptive complexity theory. In this thesis, the CFI-construction plays an important role as well (see Chapter 4).

Although FPC does not capture polynomial time, this logic still turns out to be surprisingly powerful and, in particular, it captures a natural and robust fragment of polynomial time. To illustrate this, we want to mention a few of the many nice expressivity results and equivalent characterisations which have been established for FPC during the last two decades. For details, we refer to the nice survey of Anuj Dawar [25] and to the monograph of Martin Otto [79].

One quite successful approach has been to identify natural classes of structures on which FPC captures PTIME. Recall that, by the Immerman-Vardi Theorem, fixed-point logic can express every polynomial-time property in the presence of a linear order on the input structure. Hence, a promising approach to express all polynomial-time properties on a certain class of structures in FPC would be as follows. First, we try to *define* (in FPC) a linear order on a given input structure. Then we express, in a second step, every polynomial-time property on the resulting *ordered* structure by using the fixed-point mechanism of FPC. Unfortunately, for almost all interesting classes of structures, this idea does not take us very far: due to symmetries, we provably cannot define, in any reasonable logic, a total order on the input structures.

Still, there is a clever way to circumvent this problem. Instead of defining the linear order directly on the input structure, we can try to define an *isomorphic copy* of the input structure over an *ordered universe*. More precisely, given a structure $\mathfrak{A}$ we aim to define (again in FPC, say) a new structure $(\mathfrak{B}, <)$ over an *ordered* universe $B$ such that $\mathfrak{A} \cong \mathfrak{B}$. If we can define such a *canonical ordered copy* of $\mathfrak{A}$, then we can use the Immerman-Vardi Theorem to express every polynomial-time property of $\mathfrak{A}$ on this ordered copy $(\mathfrak{B}, <)$. This fundamental technique is known as the method of *definable canonisation* and it has been applied successfully to show that FPC captures polynomial time on many important and natural classes of structures, for instance, on trees [66], on all classes of bounded treewidth [51], on planar graphs [45], on graphs which exclude the complete graph on five vertices as a minor [46], and on interval graphs [71]. Recently, Martin Grohe was able to strongly generalise almost all of the known capturing results for FPC by proving that FPC captures polynomial time on *every* class of graphs which excludes

some graph as a minor [48, 49]. These results certainly belong to the most significant achievements of descriptive complexity theory, as they show that on numerous important classes of structures, the notion of efficient computability can be captured purely in terms of an extremely natural logical system. One central contribution of this thesis is to extend the remarkable list of classes of structures and corresponding logics which capture polynomial time by a new pair: Choiceless Polynomial Time captures polynomial time on classes of structures with Abelian colours (see Section 1.4 and Chapter 6 for details).

Additionally to the aforementioned capturing results, it is further known that FPC can express important and highly non-trivial algorithmic techniques, like tractable approximations of the graph isomorphism problem [21, 79] or, by a recent result of Anderson et. al., also the ellipsoid method for solving linear programs [5]. To finally justify that FPC also captures a very *natural* level of polynomial-time complexity, we want to point to some of the various equivalent characterisations for its expressive power. Besides the powerful game-theoretic techniques [21] which have already been applied by Cai, Fürer, and Immerman, we now have many, partly quite surprising, characterisations, for instance, by relaxations of linear programs [11, 52] or by families of symmetric circuits [4]. Altogether, because of its expressive power and its robustness, FPC became a well-established reference for the search of a logic capturing polynomial time.

## 1.2 Linear equation systems over finite domains

Although fixed-point logic with counting is much more powerful than least fixed-point logic, the situation we face resembles the one that Immerman discovered in 1982: We found a logic (in this case FPC) which captures polynomial-time on certain classes of finite structures, but there is a query (in this case, the isomorphism problem for CFI-structures) that separates FPC from polynomial time in the general case. Hence, we could continue the quest in the same way as Immerman proposed FPC as an extension of LFP: we could add new logical operators to FPC which express the CFI-query.

However, the catch with this approach is that the isomorphism problem of Cai, Fürer, Immerman graphs can hardly be called a natural problem of polynomial-time complexity, since the corresponding graphs arise from a very specific technical construction. Hence, adding operators for the CFI-query to fixed-point logic with counting would result in an artificial extension and not in an interesting new candidate for capturing polynomial time. After all, the main question is: are there any "important" polynomial-time problems and general algorithmic techniques which cannot be expressed in fixed-point logic with counting (or is the CFI-query just a singular phenomenon)?

Unfortunately, it turns out that the answer is positive. In fact, the CFI-query is an instance of a fairly important and general problem of polynomial-time complexity. In [10], Atserias, Bulatov, and Dawar proved that the *solvability problem* for *linear equation systems* over *any finite Abelian group*

is not definable in fixed-point logic with counting. On the other hand, it is well-known that the solvability of linear equation systems over Abelian groups can be decided efficiently, for example, by using (a variant of) the method of *Gaussian elimination*. More strikingly, it turned out that the CFI-query reduces to a linear equation system over the finite field with two elements $\mathbb{F}_2$ [16, 28]. This shows that fixed-point logic with counting fails to express a significant part of the polynomial-time computable problems, and that the CFI-query is only one specific example from this class.

The undefinability result of Atserias, Bulatov, and Dawar initiated the study of the descriptive complexity of problems from linear algebra, in particular of the solvability problem for linear equation systems over finite algebraic domains, see [16, 24, 27, 28, 29, 59, 71, 80], which we continue in this thesis.

**Succinct encodings of algebraically structured objects** Despite the strong undefinability result of Atserias, Bulatov, and Dawar for linear equation systems over finite Abelian groups, it turns out that fixed-point logic with counting can express many important queries from the field of linear algebra. For example, FPC can define the determinant of matrices, or the inverse of a non-singular matrix, see [59] for a detailed account of the known FPC-definability results for queries from linear algebra.

On the other hand, of course, many problems which resemble, or which generalise, the solvability problem for linear equation systems over finite Abelian groups are undefinable in fixed-point logic with counting as well. Besides the various examples from linear algebra [27, 59, 71, 80], this also holds for the (NP-complete) satisfiability problem for Boolean formulas [10], but also for many efficiently solvable problems, like for tractable cases of the constraint satisfaction problem [10, 19], or for problems from computational group theory [36, 60].

The common approach to solve these algebraic problems algorithmically is to compute *succinct representations* of the *large algebraic objects*. For example, Gaussian elimination computes a basis for the linear space of entailed equations in row echelon form, the algorithm of Bulatov and Dalmau for solving constraint satisfaction problems with a Malt'sev polymorphism constructs compact representations of solutions sets, and the Schreier-Sims algorithm for deciding membership in a permutation group constructs a strong generating set for the given permutation group.

More importantly, such algorithmic ideas often appear as subroutines in more complex algorithms, most importantly, in procedures to efficiently decide the (graph) isomorphism problem on certain classes of structures, like on graphs with bounded degree, bounded colour class size, and so on, see [68]. Indeed, the two main sources of problems which are known to separate FPC from PTIME are tractable cases of the graph isomorphism problem and computational queries from the field of algebra like the ones mentioned above. For instance, the CFI-construction does not only show that FPC fails to define the solvability

of linear equation systems, but also that FPC cannot define the isomorphism problem on graphs with bounded degree *and* bounded colour class size, whereas the isomorphism problem is known to be tractable on both classes.

These observations emphasise the shortcoming of FPC to express properties of large, algebraically structured objects which are given in a compact way. Thus, if we want to make progress towards a logic capturing polynomial time, then we have to understand how these algorithmic principles to compute normal form representations of algebraic objects, which crucially rely on the order on the input structure, can be captured by natural logical operators and methods over unordered structures. The solvability problem for linear equation systems over finite domains is a good starting point to explore this question and, in this thesis, we pursue investigations in this direction.

## 1.3 Candidates for capturing polynomial time

At the moment, there are basically three different (types of) logics that have been proposed as candidates to capture polynomial time. Notably, the underlying ideas for these languages differ significantly, and the relationship between almost all of the candidates is unsettled. Still, they all strictly extend fixed-point logic with counting and thus capture a highly non-trivial fragment of polynomial time. In this thesis, we study the first two (families of) candidates in the light of the question of whether the solvability problem for linear equation systems over finite domains is definable, and how this can be used to capture polynomial time on new classes of structures by natural logics.

**Operators for algebraic problems** The first family of extensions is directly motivated by the result of Atserias, Bulatov, and Dawar showing that the solvability of linear equation systems over finite Abelian groups cannot be defined in fixed-point logic with counting. This leads to the natural idea of extending FPC by new operators which are capable of expressing this solvability problem. In particular, since the CFI-query reduces to a linear equation system over the finite field with two elements, such extensions of FPC can define the CFI-query (and with it, most queries which are known to separate FPC from PTIME).

Certainly, there are numerous ways to add such operators to FPC and we will discuss some approaches and related aspects in this thesis in detail (see Chapter 3 and Chapter 4). One quite interesting concept, proposed by Dawar, Grohe, Holm, and Laubner in [28], is the notion of *rank operators*. These logical operators can compute the rank of a matrix over a finite field. Since over any field the solvability problem for linear equation systems can be reduced to the matrix rank problem, the extension of FPC by rank operators can express the solvability of linear equation systems over *all* finite fields (note, however, that this does not capture linear equation systems over all finite Abelian groups). Another possibility, which has been studied in [27], is to

directly add quantifiers for the solvability problem to FPC in the form of so-called Lindström quantifiers.

Of course, one could also study extensions by more general operators from computational group theory. By what we know today, a quite universal approach would be to add operators to FPC which can compute the size of a definable permutation group (which is succinctly specified by a definable set of generators). However, we are just starting to explore this huge landscape of new possible operators from algebra and so, at the moment, it does not make sense to propose more and more powerful logics without having a reasonable understanding of the basic ones (obviously, the whole approach of capturing larger and larger fragments of PTIME by logics only makes sense as long as we identify natural levels of expressiveness for which we obtain a good understanding and insightful characterisations).

**Choiceless Polynomial Time** The reason why it is difficult to capture algorithms which operate on *string encodings* of *ordered* structures by formulas which are evaluated on abstract, and in particular *unordered*, structures is that algorithms can make arbitrary *choices*. Indeed, an algorithm can fix an element from any arbitrary set, for example it can choose the element which is minimal with respect to the given linear order (which is available due to the encoding of the structure as a string). In particular, the selected element does not need to have any unique structural property, which distinguishes it from the other elements in the set. The only semantical requirement is that the algorithm is order invariant, which means that for the final output the concrete choices of elements are not relevant.

The ability to make arbitrary choices is used by many algorithms. For instance, consider the method of Gaussian elimination, which transforms a given matrix into row echelon form. This procedure heavily relies on making non-canonical choices, since it has to fix, in every step, a pivot element (and this element is, in general, not unique). Still, in the end, we obtain, no matter what the specific choices were, a basis for the space of entailed linear equations in row echelon form, from which we can easily read off whether or not the given linear equation system is solvable (and, again, this output does not depend on the specific choices for the pivot elements during the transformation).

On the other hand, such non-canonical choices are impossible in every logical formalism which operates directly on *unordered* relational input structures in an isomorphism-invariant way. In particular, a logical formula cannot distinguish between two elements which are related via an automorphism, and thus, it cannot fix only one of the two elements. This leads to the intuition that a logic (for polynomial time) is an abstract computation model that operates directly on relational structures (rather than on string encodings) in such a way that all computations are *choiceless*, or, in other words a computation model which captures the "*choiceless* fragment" of polynomial time.

In [15], Blass, Gurevich, and Shelah formalised this intuition by introducing

the model of *abstract state machines*, today known as *BGS-machines*. These machines directly operate on relational input structures, and not on string encodings. In particular, computations of BGS-machines are *choiceless* and they preserve the symmetries of their inputs. To compensate for the lack of making non-canonical choices, these machines can manipulate higher-order states which model *parallel executions*. The rough idea is as follows: although we cannot fix an arbitrary element to continue our computation, we could proceed by considering *all* possible choices in parallel.

Indeed, with unlimited resources, BGS-machines can define every decidable class of structures, that is BGS-machines are Turing-complete. On the other hand, if we restrict BGS-machines to polynomial resources, then we obtain the logic *Choiceless Polynomial Time* (CPT), and the question of whether BGS-machines with polynomial resources can express all polynomial-time properties is much more challenging and, until today, far open. Also the relation between Choiceless Polynomial Time and the extensions of fixed-point logic with counting by linear-algebraic operators (see above) is far open, basically because it is not known whether CPT can define the solvability of linear equation systems over finite Abelian groups.

What is known is that Choiceless Polynomial Time strictly extends fixed-point logic with counting [15, 16] and that it can express, due to a very nice construction of Dawar, Richerby, and Rossman, the Cai, Fürer, Immerman query [32]. Moreover, today we have various equivalent characterisations of Choiceless Polynomial Time [17, 42, 84]. This together indicates that CPT is a quite natural and powerful candidate for capturing polynomial time. We give the formal definition of Choiceless Polynomial Time in Section 2.4, and we establish new definability results for CPT in Chapter 5 and in Chapter 6.

**Symmetric choices** For completeness, we also want to mention *fixed-point logic with specified symmetric choice* (SSCFP), a third candidate of a logic which might capture polynomial time (although we do not study SSCFP in this thesis). Similarly to Choiceless Polynomial Time, the basic motivation of SSCFP is to bridge the gap between algorithms, which can make arbitrary, non-canonical choices, and logics, which can only define symmetric objects.

Surprisingly, the basic ingredient of SSCFP is an inductive mechanism which explicitly allows choices. More precisely, in every step of the induction it allows to non-deterministically pick an element from a definable set (the so-called *choice set*) and then to continue the induction with the so chosen element. However, in order to turn SSCFP into a logic (which after all means that it can *not* distinguish between symmetric objects), we require that all pairs of elements in the choice set are related by automorphisms of the current state. Since it is not clear whether this condition can be decided efficiently, for SSCFP it is additionally required that a *witnessing* formula explicitly specifies a set of automorphisms which relate all pairs of elements from the choice set, see [30, 38, 82] for details. It is known that SSCFP strictly extends

fixed-point logic with counting and it is open whether it suffices to express all polynomial-time properties. Also the relation to the other candidates, mentioned above, is unsettled so far.

## 1.4 Contributions

We continue to explore the descriptive complexity of (linear) algebra in the light of its relevance for the search of a logical characterisation of polynomial time. In particular, we study the (inter-)definability of solvability problems (for linear equation systems) over different classes of finite algebraic domains. One of our main results is that over fields with different prime characteristic, these solvability problems cannot be reduced to each other (within fixed-point logic with counting). This solves an open question of Dawar and Holm and it also separates rank logic, in its original version, from polynomial time. On the more positive side, we show that the solvability problem for certain families of linear equation systems can be defined in Choiceless Polynomial Time. We further apply this definability result to establish a CPT-definable canonisation procedure for structures *with Abelian colours*. It turns out that such classes appear frequently in finite model theory, and, in particular, they generalise most queries which are based on constructions which resemble the Cai, Fürer, and Immerman construction. Our capturing result further solves an open question of Blass, Gurevich, and Shelah: the isomorphism problem for multipedes is definable in Choiceless Polynomial Time.

**Solvability problems over finite algebraic domains** In Chapter 3, we study the (inter-)definability of linear equation systems over (finite) Abelian groups, rings, and modules. Over these domains, fixed-point logic with counting fails to express the solvability of linear equation systems, although linear systems can be solved in polynomial time by using (variants of) Gaussian elimination. Our aim in Chapter 3 is to identify *simple* classes of algebraic domains to which one can reduce, in fixed-point logic, *all* linear equation systems over Abelian groups, rings, and modules. The canonical candidates are the class of all (finite) fields and the classes of all cyclic groups or rings.

Such reductions to simple classes of algebraic domains are desirable, since they isolate more clearly the essential linear-algebraic properties which cannot be defined in fixed-point logic with counting and, in particular, allow us to disregard many sophisticated algebraic structures. Besides that, another central motivation is that most of the extensions of fixed-point logic with counting by new linear-algebraic operators, such as rank logic [28] or solvability logic [27, 43], are defined with operators over such *restricted* classes of domains. As a consequence, our investigations are intimately linked to several open questions about these logics. For instance, we know that rank logic can solve linear equation system over all *finite fields* [59], but it is still open whether rank logic can express the solvability of linear systems also over *all* Abelian groups,

or at least over *all* cyclic groups. A similar question arises for solvability logic which can express the solvability of linear equation systems over all cyclic groups, but for which it is open whether it can also solve linear equation systems over *all* Abelian groups.

Our results are summarised in Figure 3.3 on page 54. First of all, by using the structure theory of Abelian groups, we show that linear equation systems over all *ordered* Abelian groups, rings, and modules, can be reduced to equivalent systems over cyclic groups of prime power order (we remark that only the algebraic domains and *not* the sets of equations or variables are assumed to be ordered). Secondly, for the case of linear systems over unordered Abelian groups, rings, and modules, we obtain a transformation into equivalent linear systems over local commutative rings (recall that a local ring is a ring with a *unique* maximal ideal). In both cases, the reductions are shown to be definable in fixed-point logic. The immediate consequence is that every extension of fixed-point logic which can express the solvability of linear equation systems over local commutative rings can already express the solvability of linear systems over all Abelian groups, rings, and modules. Moreover, if the specific local rings allow for a definable order, as it is, for instance, the case for cyclic rings or finite fields, it already suffices to express the solvability of linear systems over cyclic groups of prime power order.

Motivated by these insights, we proceed to identify classes of local rings on which we can define a linear order in fixed-point logic. By using the algebraic structure theory of local rings, we define a strict hierarchy of local rings and show that the local rings in every fixed level can be ordered in fixed-point logic. More precisely, the $k$-th level in this hierarchy consist of all local rings for which the unique maximal ideal is generated by at most $k$ elements. Together with our previous results this shows that, for every fixed $k$, linear equation systems over *$k$-generated local rings* can be reduced to equivalent linear systems over cyclic groups in fixed-point logic.

The results from Chapter 3 appeared in [27].

**Rank operators and solvability quantifiers** In Chapter 4, we study *solvability quantifiers* and *rank operators* over finite fields. Both notions have been proposed to extend the expressive power of fixed-point logic with counting by mechanisms to define the solvability of linear equation systems over finite fields [27, 28, 59, 71, 80]. Although solvability quantifiers and rank operators answer the same purpose, there are subtle differences between both approaches. Most importantly, solvability quantifiers are *Boolean-valued* operators which express the solvability of linear equation systems over finite fields. In contrast, rank operators are *numerical* operators which compute the numerical invariant of ranks of definable matrices. Specifically, we are interested in the expressive power of extensions of fixed-point logic (with counting) by solvability quantifiers (*solvability logic*) and rank operators (*rank logic*).

In the first part we study the relationship between solvability logic and

rank logic. While it is easy to see that rank operators can simulate solvability quantifiers (the solvability of a linear equation systems reduces to the matrix rank problem), it is an open question whether a reduction in the other direction is possible as well (in FPC, say), see [27]. In Section 4.3 we give a first partial answer to this question: in the absence of counting, solvability quantifiers are strictly weaker than rank operators (Theorem 4.17). The intuitive explanation is that rank operators, as numerical operators, can simulate the counting mechanism, while, in contrast, we have to add counting for the Boolean-valued solvability quantifiers explicitly. On the other hand, we also obtain evidence for the equivalence of both kinds of operators in the presence of counting: over interesting classes of structures, for instance structures of *bounded colour class size*, we can show that the extension of FPC by solvability quantifiers over finite fields has the same expressive power as rank logic.

The second part focuses on the question of *uniformity* of rank operators and solvability quantifiers. In the original version of rank logic, the approach was to take a *separate* rank operator $\mathsf{rk}_p$ for every prime $p \in \mathbb{P}$ which computes the rank of matrices over prime fields $\mathbb{F}_p$. Later, it was observed that this might be problematic, since with this setting every formula can only access rank operators for a constant number of different primes. Hence, it was proposed to rather use a *uniform* rank operator $\mathsf{rk}_*$ which takes the prime $p$ as part of its input and which can uniformly simulate *all* rank operators $\mathsf{rk}_p$ [59, 71, 80]. However, it remained open whether a uniform rank operator really leads to more expressive power. More importantly, a much more basic question of Dawar and Holm remained open as well: can rank operators over *different* prime fields simulate each other? [29, 59]. We are finally able to solve both questions and show that rank operators *and* solvability quantifiers over different prime fields have incomparable expressive power (Theorem 4.19). Furthermore, the original version of rank logic with separate operators for all primes fails to capture polynomial time and should be replaced by the new version with a uniform rank operator (Theorem 4.20).

Besides these new insights into the expressive power of solvability quantifiers and rank operators, we think that a central contribution of Chapter 4 is our new proof technique which is based on a combination of symmetry arguments, basic ideas from group theory and tools from finite model theory (such as a generalisation of the well-known Cai, Fürer, Immerman construction).

The results from Chapter 4 appeared in [27, 43, 44].

**Cyclic linear equation systems** In Chapter 5, we introduce cyclic linear equation systems. The main feature of these linear equation systems is that they bring a strong auxiliary structure on their set of variables and equations. Specifically, their set of variables is *almost* linearly ordered up to classes of incomparable variables $V_i$ with the property that for all pairs of variables $v, w \in V_i$ the linear system contains an equation of the form $v = w + c$ for a *constant* $c \in \mathbb{Z}_d$ which depends on $v$ and $w$. Thus every class $V_i$ could be

replaced by a *single* variable $v_i \in V_i$ (however, not in a canonical way).

At first glance, it might seem that this strong auxiliary structure makes the solvability problem for cyclic linear equation systems rather simple. In fact, cyclic equation systems come close to be *ordered* linear systems, and we know that the solvability of ordered linear systems can be expressed in fixed-point logic (even without counting) by the Immerman-Vardi Theorem. However, in general, cyclic linear equation systems have exponential-sized symmetry groups and they are structurally much more complicated than ordered linear equation systems. In fact, it turns out that cyclic linear equation systems can express highly non-trivial structural properties. For instance, the isomorphism problem for Cai, Fürer, Immerman graphs (also in a generalised version for Abelian groups) can be rephrased as the solvability problem for cyclic linear equation systems. The immediate consequence is that fixed-point logic with counting fails to define the solvability of cyclic linear equation systems. Hence, we have identified a structurally tame, but still powerful, class of linear systems which cannot be solved in fixed-point logic with counting.

Our main result is that Choiceless Polynomial Time can define the solvability of cyclic linear equation systems (Theorem 5.12). This yields a new family of queries which witness that Choiceless Polynomial Time can express general algorithmic techniques which cannot be defined in fixed-point logic with counting. The strategy of our solvability procedure is to express a variant of Gaussian elimination which operates on *classes* of equivalent linear equations (instead of single equations). We have to use this kind of higher-order version of Gaussian elimination as it is impossible to canonically choose single representatives from the equation classes without breaking symmetries. However, since these classes of equivalent terms are of exponential size, the crucial step is to find *succinct* representations which further allow us to define elementary operations in Choiceless Polynomial Time. To this end, we introduce the notion of *hyperterms*, a generalisation of a very elegant technique of Dawar, Richerby, and Rossman [32], to succinctly encode the isomorphism types of Cai, Fürer, Immerman graphs as highly-nested objects in the universe of hereditarily finite sets. Thus, it comes at no surprise that the procedure from [32] to decide the Cai, Fürer, Immerman query appears as a special case of our procedure for deciding the solvability of cyclic linear equation systems.

The results from Chapter 5 appeared in [1].

**Canonisation of structures with Abelian colours** In Chapter 6, we use our solvability procedure for cyclic equation systems to show that Choiceless Polynomial Time captures polynomial time on structures *with Abelian colours*. More precisely, we establish a canonisation procedure for structures with Abelian colours which maps a given input structure $\mathfrak{A}$ to an isomorphic structure $\mathfrak{B}$ over an *ordered* universe, and we show that this mapping can be expressed in Choiceless Polynomial Time (Theorem 6.13). It then follows by the Immerman-Vardi Theorem that every polynomial-time property of

structures with Abelian colours can be defined in Choiceless Polynomial Time.

A structure with Abelian colours consists of a relational structure $\mathfrak{A}$ and a linear preorder $\preceq$ on its elements. This preorder $\preceq$ orders the universe $A$ of $\mathfrak{A}$ up to sets $A_i \subseteq A$ of $\preceq$-equivalent elements, which we call the *colour classes* of $\mathfrak{A}$. Moreover, for every such colour class $A_i$ the structure provides an *ordered* and *Abelian* permutation group $\Gamma_i$ which acts *transitively* on $A_i$. In other words, a structure with Abelian colours is an almost linearly ordered structure which additionally provides for all classes of incomparable elements the action of an Abelian permutation group which relates all pairs of elements.

Implicitly, structures with Abelian colours have been considered quite frequently in finite model theory. In particular, most of the lower bounds for fixed-point logic with counting were obtained via constructions based on structures with Abelian colours, for instance Cai, Fürer, and Immerman graphs, multipedes [55], the structures in Hella's proof of the arity hierarchy of generalised quantifiers [57], and also Cai, Fürer, Immerman structures over general Abelian groups [10, 59]. In fact, for the particular case of multipedes, our canonisation procedure solves an open problem posed by Blass, Gurevich, and Shelah in [16, Question (5.12)]: the isomorphism problem of multipedes can be defined in Choiceless Polynomial Time.

The basic idea of our canonisation procedure is to decompose the input structure into an ordered family of induced substructures which can be canonised easily, and then to inductively construct a complete canonisation by combining the ordered copies for the small parts. The difficulty is to ensure that the choices that we make during this construction remain consistent (different substructures may have common vertices). To guarantee this, we maintain a set of valid isomorphisms which map the processed part of the input structure to the partial canonised version which we obtained so far. The challenge is to represent these exponential-sized sets of witnessing isomorphisms succinctly, in a way that allows us to express certain basic operations for the encoded sets in Choiceless Polynomial Time (such as intersections and a non-emptiness test).

At this point cyclic linear equation systems come into play. We show that the algebraic structure on sets of isomorphisms between two structures with Abelian colours allows us to succinctly represent such sets as solution spaces of families of cyclic linear equation systems. In particular, in this representation the required operations turn out to be definable in Choiceless Polynomial Time; for instance, the non-emptiness test corresponds to the solvability procedure for cyclic linear equation systems from Chapter 5.

Moreover, we establish connections between the notion of Abelian colours and the well-studied notion of structures with bounded colour class size. For example, an interesting consequence of our results is that Choiceless Polynomial Time captures polynomial time on classes of structures with colour class size two (see Theorem 6.8 and Corollary 6.14).

The results from Chapter 6 appeared in [1].

## 1.5 Acknowledgements

I am thankful to all people and institutions that have contributed to the success of this thesis. In particular, many results have been achieved in collaboration and I sincerely thank my colleagues and co-authors for the exciting journey.

Most importantly, my contributions have only been possible thanks to the profound guidance and support of Erich Grädel. Especially, I am thankful for the freedom to follow my own ideas. I am very grateful to Martin Otto and Anuj Dawar for acting as co-examiners of this thesis.

The results in Chapter 3 and also the normal form theorems for the logic $\mathrm{FOS}_p$ in Chapter 4 appeared in [27], and they were obtained in collaboration with Anuj Dawar, Bjarki Holm, Erich Grädel, and Eryk Kopczyński. My grateful thanks goes to Anuj Dawar who kindly supported me during my research stay in his group. I am also very grateful to Bjarki Holm from whom I learned a lot about science in general, and about rank logic in particular.

The separation results for linear-algebraic operators over finite fields in Chapter 4 are joint work with Erich Grädel and they appeared in [43, 44].

The results from Chapter 5 and Chapter 6 appeared in [1], and they were obtained together with Faried Abu Zaid, Erich Grädel, and Martin Grohe. Especially, I want to thank Martin Grohe who was a driving, inspiring, and specifically motivating force in our project. Specifically, he came up with the idea of cyclic linear equation systems and he established the connection to canonisation. Also, I like to thank Pascal Schweitzer for numerous insightful discussions about (computational) group theory, which also lead to the notion of Abelian colours. Special thanks goes to cafe "Lammerskötter", where Faried and I often worked on representations of algebraic objects by hereditarily finite sets, and also on many other ideas, while we enjoyed a delicious breakfast.

Last, but not least, I strongly enjoyed the research in all projects which did not become part of this thesis. Thus, I want to thank Faried Abu Zaid, Erich Grädel, and Łukasz Kaiser for our project on automatic structures [2], Erich Grädel, Łukasz Kaiser, and Svenja Schalthöfer for our work on an equivalent characterisation of Choiceless Polynomial Time [42], and Felix Canavoi, Erich Grädel, and Simon Leßenich for our study on the definability of positional winning strategies in infinite games [22].

I further like to thank my colleagues and friends at the research groups *Mathematische Grundlagen der Informatik* and *Informatik 7* for the nice time that we had, and I am grateful to *Deutsche Forschungsgemeinschaft (DFG)* for funding my research.

Moreover, my deep gratitude goes to my parents, my whole family, and my friends for their constant support during the last years.

Finally, my love and deepest appreciation goes to Petra. Without her patience, trust, and support, all of this would not have been possible. I am also deeply thankful for our wonderful daughter Luisa who brings so much joy to our lives. Let me remark: *Ich liebe Euch kleine Familie!*

# Chapter 2

# Preliminaries

In this chapter we fix our notation and we recall some basic definitions and facts. We assume that the reader is familiar with the standard notions, ideas, and concepts from finite model theory and descriptive complexity theory. In particular, we only give detailed expositions of the (few) concepts for which we think that they are less common in the literature, or of which we use slightly adapted variants in this thesis. For most parts of this background material, however, we give no details but refer to the many excellent handbooks on finite model theory and descriptive complexity theory [33, 34, 49, 65, 72, 79]. We further assume that the reader is familiar with the important (algorithmic) complexity classes like PTIME, NP, LOGSPACE, and so on, see for example [81].

We denote by $\mathbb{N} = \{0, 1, \dots\}$ the set of natural numbers and by $\mathbb{Z}$ the set of integers. For $n \in \mathbb{N}$ we denote by $[n] \subseteq \mathbb{N}$ the initial segment $[n] = \{0, \dots, n-1\}$ of $\mathbb{N}$ of length $n$. Set inclusion is denoted by $\subseteq$ and strict inclusion by $\subset$. The power set of $A$ is denoted by $\mathcal{P}(A)$. For a set $A$ and for $k \geq 1$, $A^k$ denotes the set of $A$-tuples of length $k$, $A^{\leq k} = \bigcup_{\ell \leq k} A^\ell$ denotes the set of $A$-tuples of length at most $k$, and $A^{<\omega} = \bigcup_{\ell \geq 1} A^\ell$ denotes the the set of $A$-tuples of arbitrary (finite) length. For an equivalence relation $\approx$ on $A$ we write $[a]_\approx$ to denote the equivalence class of an element $a \in A$ and $A/\approx \, = \{[a] : a \in A\}$ for the set of all equivalence classes. For a binary relation $E \subseteq A^2$ we denote by $\mathrm{TC}(E) \subseteq A^2$ its transitive closure. A *(linear) preorder* $\preceq$ on a set $A$ is a reflexive, transitive and total binary relation. It induces a linear order on the classes of the associated equivalence relation $x \sim y := (x \preceq y \wedge y \preceq x)$. We write $A = C_0 \preceq \cdots \preceq C_{n-1}$ for the decomposition of $A$ into $\sim$-classes $C_i$ which are ordered by $\preceq$ as indicated.

In Section 2.1, we fix our notation for relational structures, formulas, and queries, and we recall the notion of logics which capture complexity classes. We then define in Section 2.2 different kinds of logical reductions which are based on the concept of interpretations. In Section 2.3, we recall the extensions of first-order logic and of fixed-point logic with counting. We further give a compact definition of the logic Choiceless Polynomial Time in Section 2.4. Finally, we summarise the basic algebraic notions which we require throughout this thesis in Section 2.5.

## 2.1 Descriptive complexity theory

A *vocabulary* (or *signature*) is a finite set $\tau = \{R_1, \dots, R_k\}$ of relation symbols $R_i$ with a given arity $\mathrm{ar}(R_i) = r_i$, $1 \leq i \leq k$. A $\tau$-structure $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, \dots, R_k^{\mathfrak{A}})$ consists of a non-empty set $A$, the *universe* of $\mathfrak{A}$, together with interpretations $R_i^{\mathfrak{A}} \subseteq A^{r_i}$ for the relation symbols $R_i$ as $r_i$-ary relations over $A$. If not stated otherwise, *structures are assumed to be finite*. This also holds for algebraic structures which we consider in this thesis like groups, rings, and so on.

We only consider classes of structures which are closed under isomorphisms. Let $\mathcal{K}$ be a class of structures and let $\tau$ be a vocabulary. Then we denote by $\mathcal{K}(\tau) \subseteq \mathcal{K}$ the subclass of all $\tau$-structures which belong to $\mathcal{K}$. Moreover, we denote by $\mathcal{S}$ the class of all (finite) structures and by $\mathcal{S}(\tau)$ the class of all (finite) $\tau$-structures. Often we consider pairs $(\mathfrak{A}, \bar{x} \mapsto \bar{a})$ consisting of a $\tau$-structure $\mathfrak{A} \in \mathcal{S}(\tau)$ and a variable assignment $\bar{x} \mapsto \bar{a}$ where $\bar{a} \subseteq A$, and we let $\mathcal{S}(\tau, \bar{x})$ denote the class of all such pairs.

We assume that the reader is familiar with *first-order logic* FO, *monadic second order logic* MSO, *deterministic transitive closure logic* DTC, and *inflationary fixed-point logic* FP. For a logic $\mathcal{L}$ we denote by $\mathcal{L}(\tau)$ the set of all $\mathcal{L}$-formulas over the signature $\tau$. For a formula $\varphi \in \mathcal{L}$ we write $\varphi(x_1, \dots, x_k)$ to indicate that every variable which occurs free in $\varphi$ is among $x_1, \dots, x_k$.

A *$k$-ary query of $\tau$-structures*, for $k \geq 0$, is a function $Q$ which maps $\mathfrak{A} \in \mathcal{S}(\tau)$ to a $k$-ary relation $Q(\mathfrak{A}) \subseteq A^k$ in an isomorphism-invariant way. For $m = 0$ we say that $Q$ is a *Boolean query*. Each formula $\varphi(x_1, \dots, x_k) \in \mathcal{L}(\tau)$ defines a *$k$-ary query* which maps a $\tau$-structure $\mathfrak{A} \in \mathcal{S}(\tau)$ to the $k$-ary relation $\varphi^{\mathfrak{A}}$ on $A$ which is given as $\varphi^{\mathfrak{A}} = \{(a_1, \dots, a_k) \in A^k : \mathfrak{A} \models \varphi(\bar{a})\}$. For a class $\mathcal{K} \subseteq \mathcal{S}(\tau)$ we write that $\mathcal{K} \in \mathcal{L}$ if the Boolean query $\mathcal{K}$ can be defined by a sentence of $\mathcal{L}$. Given two logics $\mathcal{L}_1$ and $\mathcal{L}_2$ we write $\mathcal{L}_1 \leq \mathcal{L}_2$ if every $k$-ary query which can be expressed in $\mathcal{L}_1$ can also be expressed in $\mathcal{L}_2$.

Let $\mathcal{C}$ be an (algorithmic) complexity class, such as PTIME, LOGSPACE, NP, and so on. Then we say that a logic $\mathcal{L}$ *captures* $\mathcal{C}$ on a class of structures $\mathcal{K}$ if for every Boolean query $\mathcal{K}' \subseteq \mathcal{K}$ it holds that $\mathcal{K}'$ can be decided in $\mathcal{C}$ if, and only if, $\mathcal{K}'$ can be defined in $\mathcal{L}$ on $\mathcal{K}$. We remark that this definition is not completely precise and we again refer to the standard handbooks on finite model theory for more details. However, since it is the central motivation of this thesis, we want to give a precise definition for the main open question of descriptive complexity theory, that is of whether there exists a logic $\mathcal{L}$ which captures polynomial time.

**A logic capturing polynomial time** We follow the exposition of Martin Grohe in [47]. There are four basic requirements that a logic $\mathcal{L}$ for PTIME has to satisfy. First of all, we require that, for every vocabulary $\tau$, the set $\mathcal{L}(\tau)$ of $\tau$-sentences is decidable (L1). Moreover, we want that the satisfaction relation $\models$ between $\mathcal{L}(\tau)$-sentences and $\tau$-structures is closed under isomorphisms (L2). Certainly, we would expect that any reasonable logic satisfies (L1) and (L2). Note, however, that these requirements are not trivial. For instance, the set of

all polynomial-time algorithms (say in the form of clocked Turing-machines) which are order-invariant (this corresponds to property (L2)) is not recursive, that is it satisfies (L2), but it violates (L1).

Of course, the main feature of the logic $\mathcal{L}$ is the precise match between the definability (in $\mathcal{L}$) and the algorithmic tractability in polynomial time. This means that we have to consider two directions. First of all, for every property $\mathcal{K} \subseteq \mathcal{S}(\tau)$ of $\tau$-structures which is decidable in polynomial time, we can find a $\tau$-sentence $\varphi \in \mathcal{L}(\tau)$ which defines the class $\mathcal{K}$, that is every polynomial-time property is $\mathcal{L}$-definable (C1).

For the other direction, we want that every $\tau$-sentence $\varphi \in \mathcal{L}(\tau)$ defines a class $\mathcal{K} = \{\mathfrak{A} : \mathfrak{A} \models \varphi\} \subseteq \mathcal{S}(\tau)$ of $\tau$-structures which can be decided in polynomial time. However, instead of requiring the existence of an equivalent polynomial-time algorithm only, we want that such an equivalent algorithm can be constructed effectively (maybe together with a polynomial which witnesses its running time). This assumption of "effective constructability" is justified by the application of such a logic $\mathcal{L}$ as a database query language (recall the original question of Chandra and Harel that we discussed in Section 1.1). Moreover, if we only require the mere existence (but not the effective constructability) of an equivalent polynomial-time algorithm for each formula, then there actually are certain artificial "solutions", that is logics which capture polynomial time, but from which we can not gain any insights. Hence, we require that there exists an algorithm which associates with every sentence $\varphi \in \mathcal{L}(\tau)$ a polynomial-time algorithm $\mathcal{A}_\varphi$ (and a polynomial $p_\varphi(n)$ which witnesses the running time of $\mathcal{A}_\varphi$) which decides the class $\mathcal{K} = \{\mathfrak{A} : \mathfrak{A} \models \varphi\}$ of $\tau$-structures defined by $\varphi$, that is all $\mathcal{L}$-definable properties can be decided in polynomial-time (C2).

We remark that these four natural requirements (L1), (L2), (C1), and (C2) are satisfied by any of the logics which we consider in this thesis and which we mentioned as possible candidates in Section 1.3.

**Definable canonisation** Recall that on the class of ordered structures we know, by the Immerman-Vardi Theorem [61, 87], that (least) fixed-point logic captures polynomial time. Often, this theorem can be applied to obtain such partial capturing results also for classes of *unordered* structures.

To see this, let us assume that $\mathcal{L}$ is a logic which is at least as powerful as fixed-point logic and which has polynomial-time data complexity (for instance, $\mathcal{L}$ might be one of the logics $\mathrm{FPC}, \mathrm{FPR}$, or $\mathrm{CPT}$). Moreover, let $\mathcal{K}$ be a class of structures. Assume further that there is system of $\mathcal{L}$-formulae $\Phi$ which defines for every structure $\mathfrak{A} \in \mathcal{K}$ an *isomorphic copy* $\Phi(\mathfrak{A}) = (\mathfrak{B}, \leq)$ of $\mathfrak{A}$ over an *ordered* domain (the precise form of $\Phi$ and the specific ordered domain may depend on the logic $\mathcal{L}$). In other words, $\mathfrak{B} \cong \mathfrak{A}$ and $\leq$ is a linear order on the universe of $\mathfrak{B}$. If this is the case, then we can conclude that $\mathcal{L}$ captures PTIME on $\mathcal{K}$, since given any input structure $\mathfrak{A}$ we can first use $\Phi$ to define the isomorphic copy $\Phi(\mathfrak{A}) = (\mathfrak{B}, \leq)$ of $\mathfrak{A}$ over an ordered universe, and then we can apply the Immerman-Vardi Theorem to express every polynomial-time

property of $\mathfrak{A}$ by expressing this property on $(\mathfrak{B}, \leq)$.

This method, known as *definable canonisation*, has been used very successfully to identify classes of structures (trees, planar graphs, and so on) on which (natural) logics, most importantly FPC, capture polynomial time. This canonisation technique is very well explained in [40], and also in the forthcoming book of Martin Grohe in which he proves that every class of graphs which excludes some graph as a minor allows definable canonisation in fixed-point logic with counting (the recent version of this book can be accessed at [49]). In Chapter 6 we use the technique of definable canonisation to show that Choiceless Polynomial Time (see Section 2.4) captures PTIME on classes of structures with Abelian colours.

## 2.2 Interpretations and logical reductions

Algorithmic reductions are arguably one of the most successful and fundamental tools in (algorithmic) complexity theory. In descriptive complexity theory, their counterpart is given by the notion of *logical interpretations*, also called *transductions* in [49]. Such interpretations can be used to logically reduce one class of relational structures to another (recall that such classes correspond to Boolean queries). Depending on the syntactic properties, on the way in which interpretations are nested and iterated, and on the specific logics which we choose to define our interpretations, we obtain various kinds of logical reductions with different expressive power.

**Interpretations** We start to introduce the important notion of logical interpretations. The idea is to define one structure $\mathfrak{A}$ inside another structure $\mathfrak{B}$. More specifically, the universe of $\mathfrak{A}$ is defined by equivalence classes of tuples of elements from $\mathfrak{B}$ and the relations of $\mathfrak{A}$ are defined by logical formulas on these classes.

Formally, let $\sigma$, $\tau$ be vocabularies with $\tau = \{S_1, \dots, S_\ell\}$ and $s_i = \mathrm{ar}(S_i)$, and let $\mathcal{L}$ be a logic. An *$\mathcal{L}$-interpretation of $\mathcal{S}(\tau)$ in $\mathcal{S}(\sigma)$* is a tuple

$$\mathcal{I}(\bar{z}) = \big(\varphi_\delta(\bar{x}, \bar{z}), \varphi_\approx(\bar{x}_1, \bar{x}_2, \bar{z}), \varphi_1(\bar{x}_1, \dots, \bar{x}_{s_1}, \bar{z}), \dots, \varphi_\ell(\bar{x}_1, \dots, \bar{x}_{s_\ell}, \bar{z})\big),$$

where $\varphi_\delta, \varphi_\approx, \varphi_1, \dots, \varphi_\ell \in \mathcal{L}(\sigma)$, and where $\bar{x}, \bar{x}_1, \dots$ are tuples of pairwise distinct variables of the same length, say $d$, and where $\bar{z}$ is a tuple of variables (again, pairwise distinct from all the aforementioned variables) which subsumes the remaining free variables of the formulas $\varphi_\delta, \varphi_\approx, \varphi_1, \dots, \varphi_\ell$. We say that $\mathcal{I} = \mathcal{I}(\bar{z})$ is a *$d$-dimensional* interpretation with *parameters* $\bar{z}$. Moreover, we say that $\mathcal{I}$ *does not use a congruence relation* (or that $\mathcal{I}$ is an interpretation *without congruence*) if $\varphi_\approx$ is trivial, that is if $\varphi_\approx = (\bar{x} = \bar{y})$.

Let $\mathcal{L}(\sigma \to \tau, \bar{z})$ denote the set of all $\mathcal{L}$-interpretations $\mathcal{I}(\bar{z})$ of $\mathcal{S}(\tau)$ in $\mathcal{S}(\sigma)$ with parameters $\bar{z}$. We write $\mathcal{L}(\sigma \to \tau)$ to denote the set $\mathcal{L}(\sigma \to \tau, \varnothing)$, that is the set of $\mathcal{L}$-interpretations $\mathcal{I}$ of $\mathcal{S}(\tau)$ in $\mathcal{S}(\sigma)$ without parameters.

To every $k$-dimensional interpretation $\mathcal{I}(\bar{z}) \in \mathcal{L}(\sigma \to \tau, \bar{z})$ (given as above) we associate a partial mapping $\mathcal{I} : \mathcal{S}(\sigma, \bar{z}) \to \mathcal{S}(\tau)$ as follows. For $(\mathfrak{A}, \bar{z} \mapsto \bar{a}) \in \mathcal{S}(\sigma, \bar{z})$ we obtain a $\tau$-structure $\mathfrak{B}$ over the universe $B = \{\bar{b} \in A^k : \mathfrak{A} \models \varphi_\delta(\bar{b}, \bar{a})\}$ by setting

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

for each $S_i \in \tau$. Moreover we let $\mathcal{E} = \{(\bar{b}_1, \bar{b}_2) \in A^k \times A^k : \mathfrak{A} \models \varphi_\approx(\bar{b}_1, \bar{b}_2, \bar{a})\}$. Now, if $\mathcal{E}$ is a congruence relation on $\mathfrak{B}$ then we let $\mathcal{I}(\mathfrak{A}, \bar{z} \mapsto \bar{a})$ be the structure $\mathfrak{B}/\mathcal{E}$, and otherwise, the image $\mathcal{I}(\mathfrak{A}, \bar{z} \mapsto \bar{a})$ is undefined.

**Logical reductions and Lindström quantifiers** The notion of interpretation provides a method to compare the descriptive complexity of classes of structures by means of logical reductions. Given two classes $\mathcal{K}_A \subseteq \mathcal{S}(\sigma)$ and $\mathcal{K}_B \subseteq \mathcal{S}(\tau)$ and a logic $\mathcal{L}$, the idea is to say that, with respect to $\mathcal{L}$-definability, the class $\mathcal{K}_A$ is "not harder" than the class $\mathcal{K}_B$ if we can translate $\tau$-structures into $\sigma$-structures by using $\mathcal{L}$-interpretations in such a way that the membership problem of $\mathcal{K}_B$ can be used to define (in $\mathcal{L}$) the membership problem for $\mathcal{K}_A$.

Similar to the case of algorithmic reductions there are various ways in which these membership queries can be combined, nested, or iterated, and of course, the allowed operations should be linked to the expressive power of the logic $\mathcal{L}$. In order to formalise the different kinds of logical reductions we first need to introduce the notion of *Lindström quantifiers*.

In [73] Lindström introduced a general technique to extend the expressive power of a logic $\mathcal{L}$ in a minimal way such that a certain class $\mathcal{K}$ of structures becomes definable. More precisely, for a class $\mathcal{K}$ of $\tau$-structures he defined a set of associated *Lindström quantifiers* $\mathcal{Q}_\mathcal{K}$ which are capable of defining membership in $\mathcal{K}$. His approach strongly resembles the concept of oracles used, for example, in computability theory and algorithmic complexity theory.

Following [26, 27] we introduce these extensions using logical interpretations. Let $\mathcal{L}$ be a logic and let $\mathcal{K} \subseteq \mathcal{S}(\tau)$ be a class of $\tau$-structures with $\tau = \{S_1, \ldots, S_\ell\}$. The *Lindström extension* $\mathcal{L}(\mathcal{Q}_\mathcal{K})$ of $\mathcal{L}$ by *Lindström quantifiers for the class* $\mathcal{K}$ results by extending the syntax of $\mathcal{L}$ by the following formula creation rule.

($\mathcal{Q}_\mathcal{K}$) Let $\varphi_\delta, \varphi_\approx, \varphi_1, \ldots, \varphi_\ell$ be formulas of $\mathcal{L}(\mathcal{Q}_\mathcal{K})$ which form an interpretation $\mathcal{I}(\bar{z}) = (\varphi_\delta, \varphi_\approx, \varphi_1, \ldots, \varphi_\ell)$ of $\mathcal{S}(\tau)$ in $\mathcal{S}(\sigma)$ with parameters $\bar{z}$. Then $\psi(\bar{z}) = \mathcal{Q}_\mathcal{K}\mathcal{I}(\bar{z})$ is a formula of $\mathcal{L}(\mathcal{Q}_\mathcal{K})$ over the signature $\sigma$ where the free variables are given by $\bar{z}$.

The semantics for the new formulas $\psi(\bar{z}) = \mathcal{Q}_\mathcal{K}\mathcal{I}(\bar{z})$ is defined as follows. A pair $(\mathfrak{A}, \bar{z} \mapsto \bar{a}) \in \mathcal{S}(\sigma, \bar{z})$ is a model of $\psi(\bar{z})$ if $\mathfrak{B} := \mathcal{I}(\mathfrak{A}, \bar{z} \mapsto \bar{a})$ is defined and if $\mathfrak{B} \in \mathcal{K}$.

**Definition 2.1** (Logical reductions)**.** Let $\mathcal{L}$ be a logic and let $\mathcal{K}_A \subseteq \mathcal{S}(\sigma)$ and $\mathcal{K}_B \subseteq \mathcal{S}(\tau)$ be two classes of structures.

(a) $\mathcal{K}_A$ is $\mathcal{L}$-*reducible* to $\mathcal{K}_B$ ($\mathcal{K}_A \leq_{\mathcal{L}} \mathcal{K}_B$) if $\mathcal{K}_A$ is definable in $\mathcal{L}(\mathcal{Q}_{\mathcal{K}_B})$.

(b) $\mathcal{K}_A$ is $\mathcal{L}$-*truth-table reducible* to $\mathcal{K}_B$ ($\mathcal{K}_A \leq^{\mathrm{tt}}_{\mathcal{L}} \mathcal{K}_B$) if $\mathcal{K}_A$ is definable in $\mathcal{L}(\mathcal{Q}_{\mathcal{K}_B})$ without using nested applications of the $\mathcal{Q}_{\mathcal{K}_B}$-quantifier.

(c) $\mathcal{K}_A$ is $\mathcal{L}$-*many-one reducible* to $\mathcal{K}_B$ ($\mathcal{K}_A \leq^{\mathrm{m}}_{\mathcal{L}} \mathcal{K}_B$) if $\mathcal{K}_A$ is definable in $\mathcal{L}(\mathcal{Q}_{\mathcal{K}_B})$ by a sentence of the form $\psi = \mathcal{Q}_{\mathcal{K}_B} \mathcal{I}$ where $\mathcal{I} \in \mathcal{L}(\sigma \to \tau)$.

**Remark 2.2.** *It holds that* $\mathcal{K}_A \leq^{\mathrm{m}}_{\mathcal{L}} \mathcal{K}_B \Rightarrow \mathcal{K}_A \leq^{\mathrm{tt}}_{\mathcal{L}} \mathcal{K}_B \Rightarrow \mathcal{K}_A \leq_{\mathcal{L}} \mathcal{K}_B$.

We remark that for many important logics $\mathcal{L}$, the relations $\leq^{\mathrm{m}}_{\mathcal{L}}, \leq^{\mathrm{tt}}_{\mathcal{L}}$ and $\leq_{\mathcal{L}}$ are transitive. In particular this holds for $\mathcal{L} \in \{\mathrm{FO}, \mathrm{FP}\}$, see for example [79]. However, this is not always true and does not hold, for example, for the logic DTC [40].

To indicate that the reduction relation between two classes $\mathcal{K}_A$ and $\mathcal{K}_B$ holds in both directions, we introduce a special notation.

**Definition 2.3.** We write $\mathcal{K}_A \equiv_{\mathcal{L}} \mathcal{K}_B$ (or $\mathcal{K}_A \equiv^{\mathrm{m}}_{\mathcal{L}} \mathcal{K}_B$ or $\mathcal{K}_A \equiv^{\mathrm{tt}}_{\mathcal{L}} \mathcal{K}_B$) if $\mathcal{K}_A \leq_{\mathcal{L}} \mathcal{K}_B$ and $\mathcal{K}_B \leq_{\mathcal{L}} \mathcal{K}_A$ (or if $\mathcal{K}_A \leq^{\mathrm{m}}_{\mathcal{L}} \mathcal{K}_B$ and $\mathcal{K}_B \leq^{\mathrm{m}}_{\mathcal{L}} \mathcal{K}_A$, or if $\mathcal{K}_A \leq^{\mathrm{tt}}_{\mathcal{L}} \mathcal{K}_B$ and $\mathcal{K}_B \leq^{\mathrm{tt}}_{\mathcal{L}} \mathcal{K}_A$, respectively).

We use these reduction concepts to compare the descriptive complexity of different classes of structures. This is justified by the fact that many important logics are *closed under logical reductions*.

**Definition 2.4.** Let $\mathcal{L}_1, \mathcal{L}_2$ be two logics. We say that $\mathcal{L}_1$ is *closed under* $\mathcal{L}_2$*-(many-one, truth-table) reductions* if for all classes $\mathcal{K}_A \subseteq \mathcal{S}(\sigma)$ and $\mathcal{K}_B \subseteq \mathcal{S}(\tau)$ such that $\mathcal{K}_B \in \mathcal{L}_1$ and

- $\mathcal{K}_A \leq_{\mathcal{L}_2} \mathcal{K}_B$ (or $\mathcal{K}_A \leq^{\mathrm{m}}_{\mathcal{L}_2} \mathcal{K}_B$, $\mathcal{K}_A \leq^{\mathrm{tt}}_{\mathcal{L}_2} \mathcal{K}_B$, respectively),

we have that $\mathcal{K}_A \in \mathcal{L}_1$. Moreover, we say that $\mathcal{L}_1$ is *closed under (many-one, truth-table) reductions* if $\mathcal{L}_1$ is closed under $\mathcal{L}_1$-(many-one, truth-table) reductions.

In other words, if $\mathcal{L}$ can define some class $\mathcal{K} \in \mathcal{L}$ and is closed under (many-one, truth-table) reductions, then $\mathcal{L}$ can also define every class reducible to $\mathcal{K}$ via $\mathcal{L}$-(many-one, truth-table) reductions. We summarise some well-known examples of logics with and without this property.

**Example 2.5** ([40, 79])**.** FO *and* FP *are closed under logical reductions, but* MSO *and* DTC *are not even closed under* FO*-many-one reductions.*

In this thesis, we also consider logics which are evaluated over extensions of the input structure by an additional second sort. In particular, we consider counting logics FOC and FPC, which we introduce in the following Section 2.3, and we study rank logic FPR and solvability logic FPS which we introduce in Chapter 4. The formulas of these logics are evaluated over the extension of

the input structure by a copy of the natural numbers. Moreover, for the logic Choiceless Polynomial Time CPT, whose definition we recall in the Section 2.4, we extend input structures by the class of all hereditarily finite sets.

Naturally, for these logics $\mathcal{L}$ (i.e. $\mathcal{L} \in \{\mathrm{FOC}, \mathrm{FPC}, \mathrm{FPR}, \mathrm{FPS}, \mathrm{CPT}\}$) it makes sense to generalise the notion of $\mathcal{L}$-interpretations and to allow the definition of structures which are built from equivalence classes of tuples which also contain these additional elements (this has been formalised for FPC in [49, 79], for FPR in [59, 71], and for CPT in [42, 84]). However, in this thesis we only consider reductions for the logics FO, FP, and DTC which are evaluated directly over the input structure.

## 2.3 Fixed-point logic with counting

We recall the extensions of first-order logic and of fixed-point logic by counting terms. The main feature of these logics is that formulas are evaluated over the *two-sorted extension* of the input structure by a copy of the arithmetic. Following [28], we let $\mathfrak{A}^{\#}$ denote the two-sorted extension of a $\tau$-structure $\mathfrak{A} = (A, R_1, \ldots, R_k)$ by the arithmetic $\mathfrak{N} = (\mathbb{N}, +, \cdot, 0, 1)$, that is the two-sorted structure $\mathfrak{A}^{\#} = (A, R_1, \ldots, R_k, \mathbb{N}, +, \cdot, 0, 1)$ where the universe of the first sort (also referred to as *vertex sort*) is $A$ and the universe of the second sort (also referred to as *number sort* or *counting sort*) is $\mathbb{N}$.

As usual for the two-sorted setting we have, for both, the vertex and the number sort, a collection of typed first-order variables. We agree to use Latin letters $x, y, z, \ldots$ for variables which range over the vertices and Greek letters $\nu, \mu, \ldots$ for variables ranging over the numbers. Similarly, for second-order variables $R$ we allow mixed types, that is a relation symbol $R$ of type $(k, \ell) \in \mathbb{N} \times \mathbb{N}$ stands for a relation $R \subseteq A^k \times \mathbb{N}^\ell$. Of course, already first-order logic over such two-sorted extensions is undecidable. In order to obtain logics whose data complexity is in polynomial time we restrict the quantification over the number sort by a numeric term $t$, that is $Q\nu \leq t.\varphi$ where $Q \in \{\exists, \forall\}$ and where $t$ is a closed term which takes values in the number sort $\mathbb{N}$. Note that, up to this point, the only closed numeric terms $t$ which can be formed result by combining the constants 0 and 1 and the addition and multiplication operation. Similarly, for fixed-point logic FP we bound the numeric components of fixed-point variables $R$ of type $(k, \ell)$ in all fixed-point definitions

$$\left[\mathsf{ifp}\, R\bar{x}\bar{\nu} \leq \bar{t}\,.\,(\varphi(\bar{x}, \bar{\nu}))\right](\bar{x}, \bar{\nu})$$

by a tuple of closed numeric terms $\bar{t} = (t_1, \ldots, t_\ell)$ where each $t_i$ bounds the range of the variable $\nu_i$ in the tuple $\bar{\nu}$.

So far our two-sorted versions of FO and FP are unable to relate the vertex sort and the number sort. More strikingly, the quantification of variables over the number sort is restricted to sets of constant size. As a matter of fact, the two-sorted extensions of FO and FP are expressively equivalent to their one-sorted counterparts.

We aim to change this by introducing the notion of *counting terms*. For a mixed tuple of variables $\bar{x}\bar{\nu}$, and for a tuple $\bar{t}$ of closed numeric terms designated to bound the range of the numeric variables in $\bar{\nu}$ and for a formula $\varphi$ we define the counting term $s = [\#\bar{x}\bar{\nu} \leq \bar{t}\,.\,\varphi]$ which is a term over the second sort and whose value $s^{\mathfrak{A}} \in \mathbb{N}$ in a structure $\mathfrak{A}$ corresponds to the number of tuples $(\bar{a}, \bar{n}) \in A^k \times \mathbb{N}^\ell$ such that $\mathfrak{A} \models \varphi(\bar{a}, \bar{n})$ and $n_i \leq t_i^{\mathfrak{A}}$ where $k = |\bar{x}|$ and $\ell = |\bar{\nu}|$. Note that, in particular, we obtain a closed numeric (counting) term which defines the size of the input structure as $[\#x\,.\,(x = x)]$.

We define *first-order logic with counting* FOC as the extension of (the above described two-sorted variant of) FO with counting terms. Similarly, by adding counting terms to the logic FP we obtain *(inflationary) fixed-point logic with counting* FPC.

**Infinitary logic with counting** We denote by $\mathrm{L}^k_{\infty\omega}$ the $k$-variable fragment of infinitary logic, that is the extension of the $k$-variable fragment of first-order logic by infinitary conjunctions and disjunctions. Moreover, we denote by $\mathrm{L}^\omega_{\infty\omega}$ infinitary logic with finitely many variables, that is $\mathrm{L}^\omega_{\infty\omega} = \bigcup_{k\geq 1} \mathrm{L}^k_{\infty\omega}$. Similarly, the $k$-variable fragment of infinitary counting logic arises by extending the syntax of $\mathrm{L}^k_{\infty\omega}$ under the formation rule for counting quantifiers $\exists^{\geq i} x$ for all $i \geq 1$ (these quantifiers can be simulated by usual first-order quantifiers, but not if we restrict the number of variables). We denote infinitary counting logic with finitely many variables by $\mathrm{C}^\omega_{\infty\omega} = \bigcup_{k\geq 1} \mathrm{C}^k_{\infty\omega}$. For a pair of structures $\mathfrak{A}, \mathfrak{B}$ we write $\mathfrak{A} \equiv^C_k \mathfrak{B}$ if no sentence of $\mathrm{C}^k_{\infty\omega}$ distinguishes between the two structures.

The importance of $\mathrm{C}^k_{\infty\omega}$ stems from the fact that $\mathrm{FPC} \leq \mathrm{C}^\omega_{\infty\omega}$, see for example [79]. Hence, if we want to prove that some class $\mathcal{K}$ is not definable in FPC, it suffices to find for every $k \geq 1$ a pair of structures $\mathfrak{A}_k \in \mathcal{K}$ and $\mathfrak{B}_k \notin \mathcal{K}$ such that $\mathfrak{A}_k \equiv^C_k \mathfrak{B}_k$. Moreover, it is well-known that the equivalence relation $\equiv^C_k$ has a game-theoretic characterisation by the so-called $k$-pebble bijection game, see [21, 79]. We will make use of these facts in Chapter 4 to prove that fixed-point logic with counting cannot distinguish between pairs of generalised Cai, Fürer, Immerman structures.

## 2.4 Choiceless Polynomial Time

Choiceless Polynomial Time (CPT) was introduced by Blass, Gurevich, and Shelah in [15]. Their motivation came from the study of so-called *abstract state machines* (also known as BGS-*machines*), a model of machines which directly compute on abstract mathematical structures (and not, like Turing machines, for example, on encodings of *ordered* structures as strings). In particular, abstract state machines do not break symmetries of the input structure, that is all states during the computation of a BGS-machines respect the symmetries of the input structures.

If we assume that BGS-machines have access to unlimited resources, then they have the same expressive power as Turing machines, that is they can define precisely the decidable classes of structures. Choiceless Polynomial Time is the restriction of BGS-machines to polynomial resources ("time" and "space"). It is open whether Choiceless Polynomial Time captures polynomial time, that is whether BGS-machines restricted to polynomial resources have the same expressive power as classical polynomial time algorithms. In fact, the difficulty of simulating classical polynomial-time algorithms by BGS-machines is that BGS-machines cannot implement *choice functions*, that is statements of the form "pick some element $x$ to continue", which are used in many polynomial-time algorithms (e.g. choosing pivot elements in Gaussian elimination).

The idea of Choiceless Polynomial Time is to remedy the lack for definable choice functions by using higher-order objects which model parallel computations. The reasoning is as follows: although it is impossible to fix *one* particular element $x$ without breaking symmetries, it is possible to consider *all* choices for such elements $x$ in parallel. Of course, in the end, the amount of parallelism has to be restricted in order to maintain polynomial-time data complexity. Technically, this is achieved by allowing BGS-machines to access, to create and to manipulate *hereditarily finite sets* over the input structure and by polynomially bounding the sizes of such sets during the run of a CPT-program. This approach leads to a significant difference between Choiceless Polynomial Time and a classical logic, such as fixed-point logic with counting, since it allows the creation and manipulation of these higher-order objects that cannot be described by tuples of any fixed length (and these are the only objects which can be accessed by a classical logic like FPC, say).

Choiceless Polynomial Time satisfies the requirements of a "logic" which Gurevich formulated in [53]. Nevertheless, besides the fact that CPT-programs are evaluated on structures extended by the (infinite) class of hereditarily finite sets, in its original definition Choiceless Polynomial Time rather reminds of an abstract programming language than of a *natural* logic (for example, it provides control structures, like loops and conditional statements, dynamic functions, and so on).

In what follows, we present an equivalent definition of Choiceless Polynomial Time by Rossman [83] which builds on two more common "logical" elements, first-order logic and iteration, though it maintains the framework of hereditarily finite sets. Interestingly, by a recent result of Schalthöfer et. al. [42, 84], we know that Choiceless Polynomial Time can be characterised *purely* based on iteration and first-order logic and without using hereditarily finite sets. Still, the framework of hereditarily finite sets is very convenient to formulate CPT-procedures. Hence, for this thesis, we decided to stick to the definition of Rossman and to use hereditarily finite sets. We remark, that there also exists a further equivalent definition of Choiceless Polynomial Time in terms of a fixed-point logic extended by a mechanism to create new, set-like, objects [17]. These various different characterisations show that Choiceless Polynomial Time is a very natural logical formalism.

The following presentation is strongly based on our definition of Choiceless Polynomial Time in [1] which, in turn, is an adapted version of the definition of Rossman [83].

We start to define, for a (relational) vocabulary $\tau$, the extension $\tau^{\mathrm{HF}} = \tau \uplus \{\varnothing, \mathsf{A}atoms, \mathsf{P}air, \mathsf{U}nion, \mathsf{U}nique, \mathsf{C}ard\}$ of $\tau$ by the set-theoretic *function* symbols $\varnothing, \mathsf{A}atoms$ (two constant symbols), $\mathsf{U}nion, \mathsf{U}nique, \mathsf{C}ard$ (three unary function symbols) and $\mathsf{P}air$ (a binary function symbol). [1]

For a set $A$ we denote by $\mathrm{HF}(A)$ the class of *hereditarily finite sets* over the *atoms* $A$, that is $\mathrm{HF}(A)$ is the least set such that $A \subseteq \mathrm{HF}(A)$ and such that $x \in \mathrm{HF}(A)$ for every $x \subseteq \mathrm{HF}(A)$. In other words, $\mathrm{HF}(A)$ can be described as

$$\mathrm{HF}(A) = A_0 \cup A_1 \cup A_2 \cup \cdots = \bigcup_{i \geq 0} A_i,$$

where $A_0 := A$ and $A_{i+1} := \mathcal{P}(\bigcup_{0 \leq j \leq i} A_j)$. A set $x \in \mathrm{HF}(A)$ is *transitive* if for all $z \in y \in x$ we have $z \in x$. The *transitive closure* of $x \in \mathrm{HF}(A)$ is the least transitive set $\mathrm{TC}(x)$ with $x \subseteq \mathrm{TC}(x)$.

For a $\tau$-structure $\mathfrak{A}$, its *hereditarily finite expansion* $\mathrm{HF}(\mathfrak{A})$ is the following $\tau^{\mathrm{HF}}$-structure over the universe $\mathrm{HF}(A)$ where relations $R \in \tau$ are interpreted as in $\mathfrak{A}$ and the set theoretic functions in $\tau^{\mathrm{HF}} \setminus \tau$ are interpreted as follows:

- $\varnothing^{\mathrm{HF}(\mathfrak{A})} = \varnothing$, $\mathsf{A}atoms^{\mathrm{HF}(\mathfrak{A})} = A$, and
- $\mathsf{P}air^{\mathrm{HF}(\mathfrak{A})}(x, y) = \{x, y\}$, $\mathsf{U}nion^{\mathrm{HF}(\mathfrak{A})}(x) = \{y \in z : z \in x\}$, and
- $\mathsf{U}nique^{\mathrm{HF}(\mathfrak{A})}(x) = \begin{cases} y, & \text{if } x = \{y\} \\ \varnothing, & \text{else,} \end{cases}$ and $\mathsf{C}ard^{\mathrm{HF}(\mathfrak{A})}(x) = \begin{cases} |x|, & x \notin A \\ \varnothing, & \text{else.} \end{cases}$, where $|x|$ is the cardinality of $x$ encoded as a von Neumann ordinal.

A bijection $\pi : A \to A$ extends to a bijection $\pi' : \mathrm{HF}(A) \to \mathrm{HF}(A)$ in a natural way. If $\pi$ is an automorphism of $\mathfrak{A}$, then $\pi'$ is an automorphism of $\mathrm{HF}(\mathfrak{A})$. BGS-*logic* is evaluated over hereditarily finite expansions $\mathrm{HF}(\mathfrak{A})$ and is defined using three syntactic kinds: *terms*, *formulas* and *programs*.

- *Terms* can be built from variables and function names in $\tau^{\mathrm{HF}}$ using the standard formation rules. For an input structure $\mathfrak{A}$, terms take values in $\mathrm{HF}(A)$. Moreover, it is allowed to build *comprehension terms*: if $s(\bar{x}, y)$ and $t(\bar{x})$ are terms, and $\varphi(\bar{x}, y)$ is a formula then $r(\bar{x}) = \{s(\bar{x}, y) : y \in t(\bar{x}) : \varphi(\bar{x}, y)\}$ is a term (in which the variable $y$ is bound). In a structure $\mathfrak{A}$ the value $r^{\mathfrak{A}}(\bar{a})$ of the term $r(\bar{x})$ under an assignment $\bar{a} \subseteq \mathrm{HF}(A)$ is the set $r^{\mathfrak{A}}(\bar{a}) = \{s^{\mathfrak{A}}(\bar{a}, b) : b \in t^{\mathfrak{A}}(\bar{a}) : \mathrm{HF}(\mathfrak{A}) \models \varphi(\bar{a}, b)\} \in \mathrm{HF}(A)$.
- *Formulas* can be built from terms $t_1, t_2, \ldots, t_k$ as $t_1 = t_2$ and $R(t_1, \ldots, t_k)$ (for $R \in \tau$), and from other formulas using the Boolean connectives $\wedge, \vee, \neg$.

[1] This means that we allow vocabularies which contain function symbols, but *only* for defining CPT-programs. In particular, the input structure of a CPT-program is always relational.

- *Programs* are triples $\Pi = (\Pi_{\text{step}}, \Pi_{\text{halt}}, \Pi_{\text{out}})$ where $\Pi_{\text{step}}(x)$ is a term, and $\Pi_{\text{halt}}(x)$ and $\Pi_{\text{out}}(x)$ are formulas. On an input structure $\mathfrak{A}$ a program $\Pi$ induces a *run* which is the sequence $(x_i)_{i\geq 0}$ of *states* $x_i \in \mathrm{HF}(A)$ defined inductively as $x_0 = \varnothing$ and $x_{i+1} = \Pi_{\text{step}}(x_i)$. Let $\rho = \rho(\mathfrak{A}) \in \mathbb{N}$ be minimal such that $\mathfrak{A} \models \Pi_{\text{halt}}(x_\rho)$ (if no such $\rho$ exists we set $\rho = \rho(\mathfrak{A}) = \infty$). The *output* $\Pi(\mathfrak{A})$ of the run of $\Pi$ on $\mathfrak{A}$ is *undefined* ($\Pi(\mathfrak{A}) = \perp$) if $\rho = \infty$ and is defined as the truth value of $\mathfrak{A} \models \Pi_{\text{out}}(x_\rho)$ otherwise.

To summarise, a BGS-program transforms a state (an object in $\mathrm{HF}(A)$) into a next state until a halting condition is satisfied, and it produces the output from the final state. To obtain CPT-programs, that is BGS-program which only use polynomial resources and which can be simulated by classical polynomial-time algorithms, we have to put polynomial bounds on the complexity of states and on the length of runs. An appropriate measure for the complexity of objects from $\mathrm{HF}(A)$ is the size of their transitive closure (since this is the number of elements required to represent them as a graph).

**Definition 2.6.** A CPT-*program* is a pair $\mathcal{C} = (\Pi, p(n))$ consisting of a BGS-program $\Pi$ and a polynomial $p(n)$. The output $\mathcal{C}(\mathfrak{A})$ of $\mathcal{C}$ on an input structure $\mathfrak{A}$ is defined as $\mathcal{C}(\mathfrak{A}) = \Pi(\mathfrak{A})$ if the following resource bounds are satisfied (and otherwise we agree to set $\mathcal{C}(\mathfrak{A}) = \text{false}$):

- the length $\rho(\mathfrak{A})$ of the run of $\Pi$ on $\mathfrak{A}$ is at most $p(|A|)$ and
- for each state $(x_i)_{i\leq\rho(\mathfrak{A})}$ in the run of $\Pi$ on $\mathfrak{A}$ we have $|\mathrm{TC}(x_i)| \leq p(|A|)$.

## 2.5 Notions from algebra

In this section, we summarise basic notions and results from linear algebra and from group theory. For more background on group theory we refer to [56] and for more background on linear algebra (also over commutative rings) we refer to [18, 77]. Also, to get a good overview about the (non-)definability results of linear-algebraic problems in fixed-point logic with counting, we refer to the theses of Bjarki Holm and Bastian Laubner [59, 71] and to the survey of Anuj Dawar [24]. If not explicitly mentioned otherwise, all algebraic structures are assumed to be finite.

**Group actions** For a set $V$, we denote by $\mathrm{Sym}(V)$ the *symmetric group* of permutations on $V$. As usual, we use the *cycle notation* $(v_1\, v_2 \cdots v_\ell)$ to specify permutations in $\mathrm{Sym}(V)$. A group $\Gamma$ *acts on* $V$ if there is a group homomorphism from $\Gamma$ to $\mathrm{Sym}(V)$ (and we usually identify $\Gamma$ with the image under this homomorphism).

Let $\Gamma$ be a group which acts on $V$. For $v \in V$ we write $\Gamma(v) = \{\gamma(v) : \gamma \in \Gamma\}$ to denote the *orbit* of $v$ under the action of $\Gamma$. The set of $\Gamma$-orbits $\{\Gamma(v) : v \in V\}$ yields a partition of $V$. We say that $\Gamma$ acts *transitively* on $V$ if $\Gamma(v) = V$ for some (or equivalently all) $v \in V$. The action of $\Gamma$ on $V$ is *regular* if for all pairs

of elements $v, w \in V$ there is precisely one group element $\gamma$ such that $\gamma(v) = w$. In particular, if $\Gamma$ is an Abelian group which acts transitively on $V$, then the action of $\Gamma$ on $V$ is regular (we make use of this fact frequently). Moreover, the *stabiliser group* $\mathrm{Stab}(v) = \mathrm{Stab}^{\Gamma}(v)$ of an element $v \in V$ is the subgroup of $\Gamma$ consisting of all elements $\gamma \in \Gamma$ which stabilise $v$, that is $\gamma(v) = v$. We also make use of the *orbit stabiliser theorem* which says that $|\Gamma| = |\mathrm{Stab}(v)| \cdot |\Gamma(v)|$. In particular, the size of the orbit of an element $v$ divides the order of $\Gamma$. In general, we read group operations from *right to left* and we use the notation ${}^{\sigma}\gamma$ as a shorthand for $\sigma\gamma\sigma^{-1}$ whenever this makes sense (hence ${}^{\sigma}({}^{\tau}\gamma) = {}^{\sigma\tau}\gamma$). Likewise, we let $\sigma\Gamma = \{\sigma\gamma : \gamma \in \Gamma\}$ and ${}^{\sigma}\Gamma = \{{}^{\sigma}\gamma : \gamma \in \Gamma\}$. For a set $x \subseteq V$ we denote the *point-wise stabiliser* of $x$ in $\Gamma$ by $\mathrm{Fix}^{\Gamma}(x) = \mathrm{Fix}(x) \leq \Gamma$, that is $\mathrm{Fix}(x) = \{\gamma \in \Gamma : \gamma(a) = a \text{ for all } a \in x\}$.

For a $\tau$-structure $\mathfrak{A}$ we let $\mathrm{Aut}(\mathfrak{A}) \leq \mathrm{Sym}(A)$ denote the *automorphism group* of $\mathfrak{A}$. In this thesis, $\mathrm{Aut}(\mathfrak{A})$ often is Abelian. A structure is *rigid* if $\mathrm{Aut}(\mathfrak{A})$ consists of the identity only. Recall that every (finite) Abelian group is the inner direct sum of cyclic groups of prime power order. Let $\Gamma$ be a group. The *exponent of* $\Gamma$ is the least common multiple of the orders of group elements of $\Gamma$. In particular, if $\Gamma$ is an Abelian group, then the exponent of $\Gamma$ is $\max\{|\gamma| : \gamma \in \Gamma\}$. For a group $\Gamma$ and for an element $\gamma \in \Gamma$ we denote by $\langle\gamma\rangle$ the *cyclic* subgroup of $\Gamma$ which is generated by $\gamma$. Moreover, we denote the *order* of the group element $\gamma \in \Gamma$, that is the size of the group $\langle\gamma\rangle$, by $|\gamma|$.

**Computing a decomposition of an Abelian group** As we already recalled above, every finite Abelian group $\Gamma$ can be decomposed into a direct (inner) sum of cyclic groups, that is $\Gamma = \langle\delta_0\rangle \oplus \cdots \oplus \langle\delta_{k-1}\rangle$ for group elements $\delta_i \in \Gamma$ which have prime power order. In this thesis, we sometimes make use of the fact that such a decomposition $\{\delta_0, \ldots, \delta_{k-1}\}$ can be computed in polynomial time if the group $\Gamma$ is given explicitly by its multiplication table. In particular, if $\Gamma$ is an *ordered* Abelian group, then, by the Immerman-Vardi Theorem, such a group decomposition can also be defined in fixed-point logic.

First of all, we can assume that $\Gamma$ is a $p$-group with exponent $d = p^{\ell}$ for a prime $p \in \mathbb{P}$ and $\ell \geq 1$. Then $\Gamma$ is $\mathbb{Z}_d$-module. Secondly, we can easily find a generating set $E = \{e_0, \ldots, e_{r-1}\}$ for $\Gamma$. However, the elements in $E$ might not be linearly independent, that is it might be the case that for certain $z_0, \ldots, z_{r-1} \in \mathbb{Z}_d$ we have $z_0 \cdot e_0 + \cdots + z_{r-1} \cdot e_{r-1} = 0$ although $z_i \cdot e_i \neq 0$ for at least one $i \in [r]$. In other words, if we let $\varphi : \mathbb{Z}_d^r \to \Gamma$ be the group homomorphism which maps a linear combination $(z_0, \ldots, z_{r-1}) \in \mathbb{Z}_d^r$ to the group element $z_0 \cdot e_0 + \cdots + z_{r-1} \cdot e_{r-1} \in \Gamma$, then it might be the case that for some $(z_0, \ldots, z_{r-1}) \in \ker(\varphi)$ we have $z_i \cdot e_i \neq 0$ for at least one $i \in [r]$. Hence, the set $E$ does not define a decomposition of $\Gamma$ as a direct sum.

Our next step is to compute a matrix $M \in \mathbb{Z}_d^{[r]\times[s]}$, for a certain $s \geq 1$, such that $\mathrm{im}(M) = \ker(\varphi)$. In other words, we want to obtain a generating set for $\ker(\varphi)$ (the set of columns of the matrix $M$). To obtain such a set it suffices to take for every $i \leq r$ and every $z \in \mathbb{Z}_d$ a linear combination of the

form $(\vec{y}, z, \vec{0}) \in \ker(\varphi)$ for $\vec{y} \in \mathbb{Z}_d^i$ (if there exists one). It is easy to see that the number of these combinations is polynomially bounded and, furthermore, if there exists a linear combination of this form, then one can find one in polynomial time as well.

Secondly, we compute invertible matrices $S \in \mathbb{Z}_d^{[r]\times[r]}$ and $T \in \mathbb{Z}_d^{[s]\times[s]}$ such that $N := S \cdot M \cdot T \in \mathbb{Z}_d^{[r]\times[s]}$ is in Smith normal form. This can easily be done in polynomial time since $\mathbb{Z}_d$ is a finite principal ideal ring (a chain ring) and, hence, divisibility is a linear preorder in $\mathbb{Z}_d$. It holds that $\mathrm{im}(N) = \ker(\varphi \circ S^{-1})$ and that $\mathrm{im}(\varphi) = \mathrm{im}(\varphi \circ S^{-1}) = \Gamma$.

Let $f_i \in \mathbb{Z}_d^r$ denote the $i$-th identity vector, that is $f_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 in component $i$. Moreover, let $x_i := \varphi \circ S^{-1}(f_i)$. Then $\{x_0, \ldots, x_{r-1}\}$ is a generating set for $\Gamma$, since $\{f_0, \ldots, f_{r-1}\}$ generates $\mathbb{Z}_d^r$ and $\mathrm{im}(\varphi \circ S^{-1}) = \Gamma$. Finally, we claim that the (non-zero) elements in $X$ yield a decomposition of $\Gamma$ into a direct sum. To see this, assume that for some $(z_0, \ldots, z_{r-1}) \in \mathbb{Z}_d^r$ we have $z_0 \cdot x_0 + \cdots + z_{r-1} \cdot x_{r-1} = 0$. Then $z_0 \cdot f_0 + \cdots + z_{r-1} \cdot f_{r-1} \in \ker(\varphi \circ S^{-1}) = \mathrm{im}(N)$. Since $N$ is in Smith normal form, it follows that $z_i \cdot f_i \in \ker(\varphi \circ S^{-1})$ for all $i \in [r]$. Hence $z_i \cdot x_i = 0$ for all $i \in [r]$ which proves our claim.

**Linear algebra** Let $I, J$ and $X$ be non-empty sets. An *$I \times J$-matrix over $X$* is a matrix $M$ with entries in $X$ whose rows are indexed by elements in $I$ and whose columns are indexed by elements in $J$, that is $M : I \times J \to X$. Similarly, an *$I$-vector $\vec{v}$ over $X$* is a mapping $\vec{v} : I \to X$. A *square matrix* is an $I \times I$-matrix. [2] We denote the set of all $I \times J$-matrices over $X$ by $X^{I\times J}$ and the set of all $I$-vectors over $X$ by $X^I$.

If $\Phi : X \to Y$ is a mapping and $M$ is an $I \times J$-matrix over $X$, then we write $\Phi(M) \in Y^{I\times J}$ to denote the $I \times J$-matrix over $Y$ which results from $M$ by applying $\Phi$ to every entry of $M$, that is $\Phi(M)(i, j) = \Phi(M(i, j))$. Similarly, if $\vec{c} \in X^I$ denotes an $I$-vector over $X$ then we write $\Phi(\vec{c}) \in Y^I$ to denote the $I$-vector over $Y$ defined as $\Phi(\vec{c})(i) = \Phi(\vec{c}(i))$.

Usually, the index sets $I$ and $J$ come without any intrinsic structure. To stress this fact, we sometimes speak of *unordered* matrices. In contrast, we also consider matrices over ordered index sets, that is *ordered $m \times n$-matrices* for the case where $I = [m]$ and $J = [n]$ are ordered sets of size $m$ and $n$ respectively. Addition and multiplication of (unordered) matrices is defined as usual. Moreover, for an $I \times J$-matrix $M$ we write $M^T$ to denote the *transpose* of $M$, that is the $J \times I$-matrix defined as $M^T(i, j) = M(j, i)$.

For $m \geq 1$ we denote by $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$ the ring (or, depending on the context, the Abelian group) of residues modulo $m$. We denote the finite prime field with $p \in \mathbb{P}$ elements by $\mathbb{F}_p$ and the finite field with $p^k$ elements by $\mathbb{F}_{p^k}$ for $k \geq 2$.

We assume that each ring $R = (R, +, \cdot)$ has a multiplicative identity $1 \in R$. A *unit* $r \in R$ is an element which has a multiplicative inverse, that is there exists an element $r^{-1} \in R$ such that $r \cdot r^{-1} = r^{-1} \cdot r = 1$. The set of units in $R$ is denoted by $R^*$ and it forms a multiplicative group. For example, in every

[2] This is different from saying that a square matrix is an $I \times J$-matrix with $|I| = |J|$.

ring $\mathbb{Z}_d$ each generator of the additive group is a unit. We say that two ring elements $r, s \in R$ are *orthogonal* if $r \cdot s = 0$. An element $r \in R$ is *idempotent* if $r^2 = r$. If $r^n = 0$ for some $n \geq 1$, then we say that $r$ is *nilpotent*, and we call the minimal $n \geq 1$ such that $r^n = 0$ the *nilpotency index* of the ring element $r$.

Let $R$ be a commutative ring. We say that $R$ is the *inner direct sum* of ideals $\mathfrak{m}_1, \ldots, \mathfrak{m}_k \trianglelefteq R$, and we write $R = \mathfrak{m}_1 \oplus \cdots \oplus \mathfrak{m}_k$, if every element $r \in R$ has a unique representation as $r = r_1 + r_2 + \cdots + r_k$ for elements $r_i \in \mathfrak{m}_i$ and if all pairs of elements $r_i \in \mathfrak{m}_i$ and $r_j \in \mathfrak{m}_j$, for $i \neq j$, are orthogonal. The *characteristic* of $R$ is the exponent of its additive group $(R, +)$, or equivalently, it is defined as $\min\{n \geq 1 : 0 = 1 + 1 + \cdots + 1 \,(n\text{-times})\}$. A *local* ring is a commutative ring with a unique maximal ideal (see Lemma 3.17 in Section 3.3 for equivalent characterisations). A *chain ring* is a local ring in which every ideal is *principal*, that is generated by a single element. For example, all rings $\mathbb{Z}_d$ where $d$ is a prime-power are chain rings. For a local ring $R$ with maximal ideal $\mathfrak{m}$ we often consider the associated *residue field* $R/\mathfrak{m}$.

For a set $I$ and a finite field $\mathbb{F}$, we denote the *general linear group* of invertible $I \times I$-matrices over $\mathbb{F}$ by $\mathrm{GL}_I(\mathbb{F})$. Let $R$ be a commutative ring. An Abelian group $G$ together with an operation $\cdot : R \times G \to G$ (called *scalar multiplication*) is *an $R$-module* if

(i) $1 \cdot g = g$ for all $g \in G$ and for the multiplicative identity $1 \in R$,

(ii) and if for all $g, h \in G$ and $r, s \in R$ we have

$$(r + s) \cdot g = r \cdot g + s \cdot g \text{ and } r \cdot (g + h) = r \cdot g + r \cdot h,$$

(iii) and if for all $g \in G$ and $r, s \in R$ we have

$$r \cdot (s \cdot g) = (r \cdot s) \cdot g.$$

We remark that modules can also be defined over non-commutative rings $R$. In this case one has to distinguish between the notion of a *left $R$-module* and a *right $R$-module*. In the latter case, the scalar multiplication is a mapping of the form $G \times R \to G$ and the axioms have to be adapted accordingly. In this thesis, we only consider modules over commutative rings $R$ in which case the notions are equivalent.

As an important example, consider an Abelian group $G$ and let $d$ be the maximal order of elements in $G$, that is the exponent of $G$. Then $G$ is an $\mathbb{Z}_d$-module by defining the scalar multiplication for $g \in G$ and $z \in \mathbb{Z}_d$ as

$$z \cdot g := \begin{cases} 0, & \text{if } z = 0, \\ g + (z - 1) \cdot g, & \text{if } z > 0. \end{cases}$$

# Chapter 3

# Linear equation systems over groups, rings, and modules

In this chapter, we study the (inter-)definability of linear equation systems over *finite* Abelian groups, rings, and modules. More precisely, we are interested in the following aspect: how does the algebraic structure, which is given on the set of coefficients and on the set of values for variables, influence the descriptive complexity of the associated *solvability problem*? Our investigations are motivated by the work of Atserias, Bulatov, and Dawar [10] who showed that over *every* (non-trivial) Abelian group the solvability problem (for linear equation systems) cannot be defined in fixed-point logic with counting. On the other hand, it is well-known that the solvability problem over Abelian groups can be decided by a polynomial-time algorithm which is based on Gaussian elimination, see for example [39]. Thus, the solvability problem over finite algebraic domains is a *natural* query which separates FPC from PTIME and which nicely indicates that fundamental algorithmic techniques, such as mechanisms to manipulate succinct representations of large algebraically structured sets, cannot be expressed in fixed-point logic with counting. We remark that similar (though much more sophisticated) algorithmic techniques are frequently used not only to solve linear equation systems, but also to efficiently decide the isomorphism problem on certain graph classes, see for example [12, 37, 75] and Chapter 6.

This situation naturally leads to the development of new logics which extend fixed-point logic with counting by mechanisms to express the solvability of linear equation systems over (certain classes of) Abelian groups. The most important logic in this context is *rank logic* FPR which was introduced by Dawar, Grohe, Holm, and Laubner in [28]. We will give the precise definition of rank logic in Section 4.1, so let us only briefly mention the main ingredients at this point. For every prime $p \in \mathbb{P}$ the logic FPR extends FPC by a new rank operator $\mathsf{rk}_p$ which can be used to determine the matrix rank of a definable matrix over the prime field $\mathbb{F}_p$. Since the solvability problem over $\mathbb{F}_p$ can be reduced to the matrix rank problem over $\mathbb{F}_p$ (recall that $M \cdot \vec{x} = \vec{c}$ is solvable if,

and only if, $\mathsf{rk}(M) = \mathsf{rk}(M \mid \vec{c})$), we can define in FPR the solvability problem over every finite prime field $\mathbb{F}_p$ (and, moreover, it was shown by Holm in [59] that this holds for *every* finite field). Let us briefly mention that we will see, in Section 4.4, that the "right" way to define FPR is by using a *uniform* rank operator $\mathsf{rk}$ which gets the prime $p \in \mathbb{P}$ as an additional part of its input (instead of having a separate rank operator for every prime $p \in \mathbb{P}$). Still, even for the revised version FPR$^*$ of rank logic, it is not clear whether it can express the solvability problem over *all* (finite) Abelian groups. The problem is that not all such groups appear as additive groups of finite fields. For instance, it is not clear how to define the solvability of linear equation systems over $\mathbb{Z}_4$ in rank logic. This raises the question whether the way in which rank logic extends fixed-point logic with counting (namely, by adding operators which compute the numerical invariant of matrix rank over finite fields) is general enough to obtain a *robust* extension of FPC which can express the important algorithmic principles which underlie the efficient procedures, such as Gaussian elimination, to decide the solvability of linear equation systems over Abelian groups. To approach this question systematically we set out to investigate the logical inter-definability of linear equation systems

- over Abelian groups (linear systems where the coefficients belong to the set $\{0, 1\}$ and where the variables range over an Abelian group $G$),
- over (not necessarily commutative) rings (where both, the coefficients and the variables, range over a ring $R$),
- and over modules (where the coefficients belong to some commutative ring $R$, and where the variables range over an Abelian group $G$, and such that we have an $R$-scalar multiplication defined on $G$).

The kind of questions we study are as follows. Is it possible to reduce the solvability problem over rings and modules to Abelian (or even to cyclic) groups? Or, more generally, are there other *simple* classes $\mathcal{C}$ of algebraic domains such that all solvability problems (over groups, rings and modules) can be reduced to solvability problems over domains from $\mathcal{C}$ (for instance in fixed-point logic)? If this is the case, then it makes sense to say that the class $\mathcal{C}$ is *complete* for the solvability problem with respect to FP-reductions.

Throughout this chapter we consider the solvability problem in the *uniform* setting, that is the algebraic domains for coefficients and variable assignments, such as groups, rings, and modules, are given explicitly as part of our input (in contrast to the setting where these domains are implicitly fixed from outside). It turns out that we can answer the above questions completely if we make the further assumption that the algebraic domains (but, of course, neither the set of variables, nor the set of equations) are linearly ordered. More precisely, we show that in this case *all* solvability problems can be reduced (via an FP-reduction) to solvability problems over cyclic groups, see Theorem 3.12. Since cyclic groups appear as basic building blocks inside all algebraic structures we

consider here, this reduction identifies a natural and simple class of algebraic domains which is complete for the solvability problem (under FP-reductions). For the unordered case, however, our reduction to cyclic group fails. Still, we can obtain partial answers to our questions from above. First of all, we can show that, in the unordered setting, the class of commutative rings and the class of modules are both complete in the above described sense with respect to DTC-reductions. Secondly, having the nice picture for ordered domains in mind, we apply the structure theory of finite commutative rings to obtain a stratification of such rings along a strict hierarchy $(L(k))_{k\geq 0}$ with levels $L(k)$ for $k \geq 0$ (more precisely, the $k$-th level in this hierarchy consists of all *$k$-generated local rings*, see Definition 3.30). We then show that, for every fixed level $k \geq 0$ of this hierarchy, we can find an FP-formula, using $k$ parameters, which defines a linear order on all rings from level $L(k)$. Combined with our previous results, this implies that for every fixed $k \geq 0$, we can reduce the solvability problem over commutative rings, whose local components belong to $L(k)$, to the solvability problem over cyclic groups. We remark that the hierarchy $(L(k))_{k\geq 0}$ is quite rich: the first level $L(1)$ already contains all finite fields and all cyclic rings $\mathbb{Z}_d$ where $d$ is a prime-power.

The whole chapter is strongly based on [27]. In Section 3.1, we give precise definitions of the solvability problems over the classes of algebraic domains we consider. In particular, we fix our encoding of these problems as relational structures, see Figure 3.2 on page 42 for an overview of the defined solvability problems. In Section 3.2, we then establish the logical reductions between the solvability problems over Abelian groups, rings, and modules. For the case of *ordered* domains we show that cyclic groups are complete for the solvability problem, and for the *unordered* case we identify modules and commutative rings as complete classes, see Figure 3.3 on page 54 for an overview of the established reductions. Finally, in Section 3.3 we apply the algebraic structure theory of finite commutative rings to show that the solvability problem over commutative rings can be reduced, in first-order logic, to solvability problems over *local* rings. We then define a hierarchy $(L(k))_{k\geq 0}$ on the class of local rings and show that for every fixed $k \geq 0$ the rings in level $L(k)$ can be linearly ordered in fixed-point logic. Combined with our results from Section 3.2, we obtain an FP-reduction for the solvability problem over rings from $L(k)$ to solvability problems over cyclic groups, for every fixed $k \geq 0$.

## 3.1 Solvability problems as relational structures

An instance of the *solvability problem (for linear equation systems)* is specified by an *algebraic domain* (such as a field, a ring, a group, or a module) and a set of *linear equations* over this domain. The question is whether there is a variable assignment which simultaneously satisfies all given linear equations. However, there are some subtle issues with this general definition which have to be discussed in more detail. First, depending on the specific type of domain,

the notion of a "linear equation" has to be described further. This is because, if we want to build linear equations, then we need *two* basic algebraic operations: *scalar multiplication*, between coefficients and variables to define atomic linear terms, and *addition*, to form sums of linear terms. Of course, these two operations are not defined in Abelian groups (where only addition is defined on the group elements). Secondly, since we want to study linear equation systems also over non-commutative rings, we have to take care of the order of multiplication. Thirdly, since we are interested in *logical* reductions, we have to represent solvability problems as classes of relational structures.

We follow the common approach and use matrices and vectors to encode systems of linear equations. Hence, we start to explain how we represent matrices and vectors by relational structures (see also Section 2.5). Let $M \subseteq A^3$ be a ternary relation on the set $A$. We say that $M$ is (a representation of) an $I \times J$-matrix over $X$ (or with entries in $X$) if $M$ is the graph of a mapping $I \times J \to X$ (which in turn is an $I \times J$-matrix over $X$). In this case we identify the relation $M$ with this matrix. Similarly, we say that a binary relation $\vec{c} \subseteq A^2$ is an $I$-vector over $X$ if $\vec{c}$ is the graph of a mapping $I \to X$. Again, we identify $\vec{c}$ with this vector. As mentioned before, we consider linear equation systems over Abelian groups, modules, and rings which we encode as $\tau$-structures over appropriate signatures $\tau$ (see Figure 3.1). In general, we use the notation $\mathrm{Ls}(\tau) \subseteq \mathbb{S}(\tau)$ to denote the class of all $\tau$-structures which represent linear equation systems.

Before we turn to the precise definitions, let us briefly remark that *all* solvability problems which we consider can be decided in polynomial time, see for example [8, 9, 39]. This also explains why we consider linear equation systems only over *Abelian* groups: in fact, it is known that over *every* non-Abelian group, the solvability problem is NP-complete, see [39]. We further remark that from our FP-reductions to cyclic groups (Theorem 3.12), we implicitly obtain an (algorithmic) polynomial-time reduction which transforms solvability problems over all considered domains into solvability problems over cyclic groups of prime-power order. Since we can very easily compute, for instance, the Smith normal form of matrices over such cyclic groups or rings (all pairs of elements are comparable with respect to divisibility), this yields an efficient algorithm to uniformly decide all considered solvability problems. Furthermore, it is even known that linear equation system over the integers can be solved efficiently (and this generalises linear equation systems over *finite* cyclic groups, as we consider them here), see for example [67].

**Solvability problems over Abelian groups** A linear equation system over an Abelian group $G$ is specified by a pair $(M, \vec{c})$ consisting of an $I \times J$-matrix $M$ over $\{0, 1\}$, the *coefficient matrix*, and of an $I$-vector $\vec{c}$ over $G$, the *constants vector*. Let $\vec{x} = (x_j)_{j \in J}$ be a $J$-vector of variables $x_j$ which range over the group $G$ for $j \in J$. The linear equation system, which is defined by the pair $(M, \vec{c})$, is determined via the matrix equation $M \cdot \vec{x} = \vec{c}$. Here, scalar

| Linear equation systems over... | Corresponding vocabulary |
|---|---|
| (Abelian) groups | $\tau_{\text{les-g}} := \{G, +, M, \vec{c}\}$ |
| modules | $\tau_{\text{les-m}} := \{G, R, +, \cdot, M, \vec{c}\}$ |
| ordered modules | $\tau^{\leqslant}_{\text{les-m}} := \tau_{\text{les-m}} \uplus \{\leqslant\}$ |
| rings | $\tau_{\text{les-r}} := \{R, +, \cdot, M_\ell, M_r, \vec{c}\}$ |
| ordered rings | $\tau^{\leqslant}_{\text{les-r}} := \tau_{\text{les-m}} \uplus \{\leqslant\}$ |

Figure 3.1: Vocabularies for encoding linear equation systems

multiplication by constants in $\{0, 1\}$ is defined in the obvious way, and addition is just group addition in $G$. A vector $\vec{b} \in G^J$ which satisfies $M \cdot \vec{b} = \vec{c}$ is called a *solution* of the system, and the system is *solvable* if it has a solution.

We encode linear equation systems over groups as structures of vocabulary $\tau_{\text{les-g}} := \{G, M, \vec{c}\} \uplus \tau_{\text{group}}$, where $\tau_{\text{group}} := \{+\}$ denotes the language of groups, where $G$ is a unary relation symbol (identifying the elements of the group), where $M$ is a ternary relation symbol (representing the coefficient matrix) and where $\vec{c}$ is a binary relation symbol (representing the constants vector).

We let $\text{Ls}(\tau_{\text{les-g}})$ denote the class of all $\tau_{\text{les-g}}$-structures which encode a linear equation system over a group, that is the class of all $\tau_{\text{les-g}}$-structures $\mathfrak{A} = (A, G, +, M, \vec{c})$ such that $(G, +)$ is an (Abelian) group, $M$ an $I \times J$-matrix over $\{0, 1\}$ and $\vec{c}$ is an $I$-vector over $G$.

**Solvability problems over modules** Linear equation systems over modules are given by a pair $(M, \vec{c})$ consisting of an $I \times J$-matrix $M$ over a commutative ring $R$, the *coefficient matrix*, and of an $I$-vector $\vec{c}$ over an $R$-module $G$, the *constants vector*. Let $\vec{x} = (x_j)_{j \in J}$ be a $J$-vector of variables $x_j$ ranging over the module $G$. Then the matrix equation $M \cdot \vec{x} = \vec{c}$ determines the linear equation system represented by $(M, \vec{c})$, where scalar multiplication and addition correspond to the respective operations of the $R$-module $G$. Similarly as before, *solutions* are $J$-vectors $\vec{b} \in G^J$ which satisfy $M \cdot \vec{b} = \vec{c}$.

Linear equation systems over modules are encoded as structures over the vocabulary $\tau_{\text{les-m}} := \{M, \vec{c}\} \uplus \tau_{\text{module}}$ where $\tau_{\text{module}} := \{G, R, +, \cdot\}$ denotes the language of modules. Here, $G$ is a unary relation symbol (to identify the additive group of the module), $R$ is a unary relation symbol (to identify the ring elements), $M$ is a ternary relation symbol (to represent the coefficient matrix) and $\vec{c}$ is a binary relation symbol (to represent the constants vector).

Let $\text{Ls}(\tau_{\text{les-m}})$ denote the class of $\tau_{\text{les-m}}$-structures which encode linear equation systems over a module, that is the class of all $\tau_{\text{les-m}}$-structures $\mathfrak{A} = (A, G, R, +, \cdot, M, \vec{c})$ where $G \cap R = \varnothing$, $(R, +, \cdot)$ is a commutative ring, $(G, R, +, \cdot)$ is an $R$-module, $M$ is an $I \times J$-matrix over $R$, and $\vec{c}$ is an $I$-vector over $G$.

Furthermore, we consider linear equation systems over *ordered modules* in which case the input structure contains a linear order on the ring elements $R$ and on the elements of the module $G$ (but neither on the index set of

equations $I$, nor on the index set of variables $J$). Note that this linear order can be arbitrary and does not need to respect the algebraic operations of the module in any way. Linear equation systems over ordered modules are represented as structures over the extended vocabulary $\tau_{\text{les-m}}^{\leqslant} := \tau_{\text{les-m}} \uplus \{\leqslant\}$ and the class $\mathrm{Ls}(\tau_{\text{les-m}}^{\leqslant})$ contains all $\tau_{\text{les-m}}^{\leqslant}$-structures $\mathfrak{A} = (\mathfrak{A}', \leqslant)$ such that $\mathfrak{A}' = (A, G, R, +, \cdot, M, \vec{c}\,) \in \mathrm{Ls}(\tau_{\text{les-m}})$ and such that $\leqslant$ is a linear order on $G$ and on $R$.

**Solvability problems over rings** Finally, we consider linear equation systems over (not necessarily commutative) rings $R$ given as triples $(M_\ell, M_r, \vec{c})$. Here $M_\ell$ is an $I \times J$-matrix over $R$, the *left coefficient matrix*, $M_r$ is an $J \times I$-matrix over $R$, the *right coefficient matrix*, and $\vec{c}$ is an $I$-vector over $R$, the *constants vector*. Let $\vec{x} = (x_j)_{j \in J}$ be a $J$-vector of variables $x_j$ ranging over the ring $R$. Then the matrix equation $M_\ell \cdot \vec{x} + (\vec{x} \cdot M_r)^T = \vec{c}$ determines the linear equation system represented by $(M_\ell, M_r, \vec{c})$ which is solvable if there exists a solution vector $\vec{b} \in R^J$ satisfying $M_\ell \cdot \vec{b} + (\vec{b} \cdot M_r)^T = \vec{c}$. [1]

A small remark about the case of non-commutative rings is in place. At first glance, it might seem that our definition does not capture linear terms of the form $a \cdot x \cdot b$ for $a, b \in R$ and a variable $x$ ranging over $R$. However, recall that we assume that every ring $R$ contains a multiplicative identity. Hence, we can easily replace each linear term $a \cdot x \cdot b$ by the linear term $y \cdot b$ and additionally add an auxiliary linear term $y = a \cdot x$. Hence, up to a simple (quantifier-free) transformation, every linear equation system over a (non-commutative) ring $R$ can be represented in the above form.

Linear equation systems over rings are encoded as structures of vocabulary $\tau_{\text{les-r}} := \{R, M_\ell, M_r, \vec{c}\,\} \uplus \tau_{\text{ring}}$, where $\tau_{\text{ring}} := \{+, \cdot\}$ denotes the language of rings, $R$ is a unary relation symbol (to identify ring elements), $M_\ell$ and $M_r$ are ternary relation symbols (to represent the left and right coefficient matrix, respectively) and $\vec{c}$ is a binary relation symbol (to represent the constants vector).

We denote by $\mathrm{Ls}(\tau_{\text{les-r}})$ the class of all $\tau_{\text{les-r}}$-structures which encode linear equation systems over a ring, that is the class of all $\tau_{\text{les-r}}$-structures $\mathfrak{A} = (A, R, +, \cdot, M_\ell, M_r, \vec{c}\,)$ such that $(R, +, \cdot)$ is a ring, $M_\ell$ is an $I \times J$-matrix over $R$, $M_r$ is a $J \times I$-matrix over $R$ and $\vec{c}$ is an $I$-vector over $R$.

Similar to the case of modules, we also consider linear equation systems over ordered rings. We represent such systems as $\tau_{\text{les-r}}^{\leqslant}$-structures where $\tau_{\text{les-r}}^{\leqslant} := \tau_{\text{les-r}} \uplus \{\leqslant\}$. Correspondingly, $\mathrm{Ls}(\tau_{\text{les-r}}^{\leqslant})$ contains all $\tau_{\text{les-r}}^{\leqslant}$-structures $\mathfrak{A} = (\mathfrak{A}', \leqslant)$ such that $\mathfrak{A}' = (A, R, +, \cdot, M_\ell, M_r, \vec{c}\,) \in \mathrm{Ls}(\tau_{\text{les-r}})$ and such that $\leqslant$ is a linear order on $R$.

[1] Note that, formally, it would be more accurate to write $\vec{x}^T \cdot M_r$ instead of $\vec{x} \cdot M_r$. However, in this thesis, we do not distinguish between column and row vectors: In fact, we only defined vectors which are indexed over a *single* set. In particular, in all situations it will be clear how multiplication of vectors and matrices is defined, and we continue to stick to this relaxed notation for the sake of better readability.

**Remark 3.1.** *The introduced classes for representing linear equation systems* $\mathrm{Ls}(\tau_{les\text{-}g})$, $\mathrm{Ls}(\tau_{les\text{-}m})$, $\mathrm{Ls}(\tau^{\leqslant}_{les\text{-}m})$, $\mathrm{Ls}(\tau_{les\text{-}r})$ *and* $\mathrm{Ls}(\tau^{\leqslant}_{les\text{-}r})$ *are* FO*-definable.*

We fixed our encoding of linear equation systems over Abelian groups, rings, and modules as relational structures. For $\tau \in \{\tau_{\text{les-g}}, \tau_{\text{les-m}}, \tau^{\leqslant}_{\text{les-m}}, \tau_{\text{les-r}}, \tau^{\leqslant}_{\text{les-r}}\}$ we let $\mathrm{SLs}(\tau) \subseteq \mathrm{Ls}(\tau)$ denote the subclass of (representations of) linear equation systems in $\mathrm{Ls}(\tau)$ which are solvable, that is the *solvability problems* over Abelian groups, (ordered) modules, and (ordered) rings. We further consider variations of these solvability problems which arise by putting additional assumptions on the structure of the algebraic domains. The following definition specifies all solvability problem which we consider in this chapter. We summarise the introduced classes of solvability problems in Figure 3.2.

**Definition 3.2.** We define the following classes of solvability problems.

- Solvability problem over *(Abelian) groups*: $\textsc{SlvG} = \mathrm{SLs}(\tau_{\text{les-g}})$
- Solvability problem over *cyclic groups*:

$$\textsc{SlvCG} = \{(A, G, +, M, \vec{c}\,) \in \mathrm{SLs}(\tau_{\text{les-g}}) : G \text{ is cyclic}\}$$

- Solvability problem over *modules*: $\textsc{SlvM} = \mathrm{SLs}(\tau_{\text{les-m}})$
- Solvability problem over *ordered modules*: $\textsc{SlvM}_{\leqslant} = \mathrm{SLs}(\tau^{\leqslant}_{\text{les-m}})$
- Solvability problem over *rings*: $\textsc{SlvR} = \mathrm{SLs}(\tau_{\text{les-r}})$
- Solvability problem over *ordered rings*: $\textsc{SlvR}_{\leqslant} = \mathrm{SLs}(\tau^{\leqslant}_{\text{les-r}})$
- Solvability problem over *commutative rings*:

$$\textsc{SlvCR} = \{(A, R, +, \cdot, M_\ell, M_r, \vec{c}\,) \in \mathrm{SLs}(\tau_{\text{les-r}}) : R \text{ is commutative}\}$$

- Solvability problem over *ordered commutative rings*:

$$\textsc{SlvCR}_{\leqslant} = \{(A, R, +, \cdot, \leqslant, M_\ell, M_r, \vec{c}\,) \in \mathrm{SLs}(\tau^{\leqslant}_{\text{les-r}}) : R \text{ is commutative}\}$$

- Solvability problem over *local rings*:

$$\textsc{SlvLR} = \{(A, R, +, \cdot, M_\ell, M_r, \vec{c}\,) \in \mathrm{SLs}(\tau_{\text{les-r}}) : R \text{ is local}\}$$

- Solvability problem over *k-generated local rings* (see Definition 3.30).

$$\textsc{SlvLR}_k = \{(A, R, +, \cdot, M_\ell, M_r, \vec{c}\,) \in \textsc{SlvLR} : R \text{ is } k\text{-generated}\}$$

| **Solvability problem over...** | |
|---|---|
| Abelian groups | SLVG |
| cyclic groups | SLVCG |
| (ordered) modules | $\text{SLVM}_{(\leqslant)}$ |
| (ordered) rings | $\text{SLVR}_{(\leqslant)}$ |
| (ordered) commutative rings | $\text{SLVCR}_{(\leqslant)}$ |
| local rings | SLVLR |
| $k$-generated local rings | $\text{SLVLR}_k$ |

Figure 3.2: Solvability problems over groups, rings, and modules

## 3.2 Reductions between groups, rings, and modules

In the previous section we defined solvability problems over different classes of algebraic domains. In this section, we study to what extent the algebraic properties of the underlying domain influence the descriptive complexity of the corresponding solvability problem. More specifically, we ask whether the various solvability problems can be reduced to each other via logical transformations.

Our results can be summarised as follows. Up to DTC-reductions, the solvability problems over modules, over commutative rings and over non-commutative rings are equivalent. Moreover, the solvability problem over Abelian groups reduces, again via a DTC-transformation, to any of these problems. In the case of *ordered* modules and rings we establish a reduction in the other direction. More precisely, we show that solvability problems over *ordered* modules and rings can be reduced, in fixed-point logic, to solvability problems over cyclic groups of prime-power order. Note that cyclic groups are the simplest domains we consider here, and that they appear as basic building blocks inside Abelian groups, rings, and modules. A detailed overview of the reductions can be found in Figure 3.3.

We start to show that, in Abelian groups, various relations based on the order of elements can be defined in DTC. In fact, we often use similar ideas throughout this section, so the proof of the next lemma also serves as an illustration for the following constructions.

**Lemma 3.3.** *The following relations are* DTC*-definable in Abelian groups $G$.*

*(a)* $|x| \leq |y| = \{(x, y) \in G^2 : |x| \leq |y|\}$

*(b)* $\text{MAXORD}(x) = \{x \in G : |x| = \max\{|g| : g \in G\}\}$

*(c)* $x \in \langle y \rangle = \{(x, y) \in G^2 : x \in \langle y \rangle\}$

*(d)* $x \leq^z y = \{(x, y) \in \langle z \rangle^2 : x = k \cdot z, y = \ell \cdot z, 0 \leq k \leq \ell < |z|\}$

*Proof.* Consider the (parameterised) graph $\mathcal{G}(z_1, z_2) = (V, E(z_1, z_2))$ on the vertex set $V = G \times G$ given by the first-order definable edge relation

$$E(z_1, z_2) = \{((g_1, g_2), (g_1 + z_1, g_2 + z_2)) : g_1, g_2 \neq 0\}.$$

Since $E(z_1, z_2)$ is deterministic, we can define $\mathrm{TC}(E(z_1, z_2))$ in DTC. Then the relation $|x| \leq |y|$ can be expressed as

$$|x| \leq |y| = \exists h((x, y), (0, h)) \in \mathrm{TC}(E(x, y)).$$

Moreover, with this preparation we can set $\textsc{maxord}(x) = \forall y(|y| \leq |x|)$ and $x \in \langle y \rangle = ((y, y), (x, x)) \in \mathrm{TC}(E(y, y))$.

Similarly, to define the linear order $x \leq^z y$ on the cyclic group $\langle z \rangle$ generated by $z$ we use reachability in the deterministic graph $\mathcal{G} = (V, E)$ on the vertex set $V = \langle z \rangle$ with the edge relation $E = \{(g, g + z) \in \langle z \rangle^2 : g + z \neq 0\}$. It suffices to set $x \leq^z y = (x = y) \vee (x, y) \in \mathrm{TC}(E)$. □

### 3.2.1 Translations from groups to modules

We start by translating the solvability problem over Abelian groups to the solvability problem over modules. To this end, let us recall the main difference between linear equation systems over Abelian groups and over modules. Over a group $G$, each atomic linear term $t$ is either of the form $t = x$ for a variable $x$, or of the form $t = g$ for a constant $g \in G$. Over an $R$-module $G$, we additionally have atomic linear terms $t$ of the form $t = r \cdot x$ for ring elements $r \in R$ and variables $x$ (recall that we assume that every ring $R$ contains a multiplicative identity 1, which allows us to build the linear terms $t = 1 \cdot x = x$). Hence, to reduce the solvability problem over Abelian groups to the solvability problem over modules, the crucial step is to transform the group $G$ into an $R$-module for an appropriate commutative ring $R$.

**Theorem 3.4.** $\textsc{SlvCG} \leq^{\mathrm{tt}}_{\mathrm{DTC}} \textsc{SlvM}_{\leqslant}$ *and* $\textsc{SlvG} \leq^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SlvM}$.

To establish Theorem 3.4 we make use of the fact that every Abelian group $G$ can be extended to a $\mathbb{Z}_d$-module where $d = \max\{|g| : g \in G\}$ is the exponent of the group $G$. Hence, the only difficulty is to construct the ring $\mathbb{Z}_d$ and the corresponding scalar multiplication on $G$ via a DTC-interpretation. As a preparation, also with regard to our later reductions, we prove something slightly more general here. We show that every Abelian group $G$ can be extended, via a DTC-interpretation, to a certain commutative ring $\mathbb{Z}_d \circledast G$. It will turn out that from this ring $\mathbb{Z}_d \circledast G$ it is very easy to extract the extension of $G$ to a $\mathbb{Z}_d$-module.

For the general construction, let $R$ be a commutative ring and let $G$ be an $R$-module. We define the commutative ring $R \circledast G = (R \times G, +, \cdot)$ on the set $R \times G$ by defining the ring addition as the component-wise addition on $R \times G$, and by defining the ring multiplication as $(r, g) \cdot (s, h) = (r \cdot s, r \cdot h + s \cdot g)$ for $r, s \in R$ and $g, h \in G$. It is straightforward to verify, using the axioms of

$R$-modules, that $R \circledast G$ forms a commutative ring where the identity element of addition is $(0,0)$ and the identity element of multiplication is $(1,0)$. Moreover, $R \times \{0\}$ is a subring of $R \circledast G$, $\{0\} \times G$ is an ideal of $R \circledast G$, and from our construction it follows that there is a one-to-one correspondence between the ring multiplication of elements $(r,0)$ and $(0,g)$ and the scalar multiplication of the $R$-module $G$. In this sense, the ring $R \circledast G$ *contains* the $R$-module $G$. Let us introduce some notation to state these observations more formally.

**Lemma 3.5.** *Let $\Phi : G \to R \circledast G$ be defined as $g \mapsto (0,g)$ for $g \in G$ and let $\Psi : R \to R \circledast G$ be defined as $r \mapsto (r,0)$ for $r \in R$.*

*(a) $\Phi$ is a group embedding, $\Phi(G) \trianglelefteq R \circledast G$ and $\Psi$ is a ring embedding.*

*(b) For all $r \in R$ and $g \in G$ we have $\Phi(r \cdot g) = \Psi(r) \cdot \Phi(g)$. More generally, for an $I \times J$-matrix $M$ over $R$ and an $J$-vector $\vec{b}$ over $G$ we have*

$$\Psi(M) \cdot \Phi(\vec{b}) = \Phi(M \cdot \vec{b}).$$

*(c) For all $g, h \in G$ we have $\Phi(g) \cdot \Phi(h) = 0 \in R \circledast G$.*

*(d) Every element $x \in R \circledast G$ can uniquely be written as $x = x_R + x_G$ for $x_R \in \Psi(R)$ and $x_G \in \Phi(G)$. Indeed, the additive group of $R \circledast G$ trivially decomposes into the additive groups $(R,+)$ and $(G,+)$.*

It will be a central step in many of our reductions to construct a commutative ring of the form $R \circledast G$. For these cases we use $\Phi$ and $\Psi$ to denote the embeddings of the underlying group $G$ and the underlying ring $R$, respectively, as defined in the previous lemma.

In the present case we have $R = \mathbb{Z}_d$ and our task is to construct the commutative ring $\mathbb{Z}_d \circledast G$ via a DTC-interpretation. Of course this requires, in particular, to construct objects to represent the elements of the ring $\mathbb{Z}_d$. Clearly, there is a canonical way to do this: every group element $g \in G$ of maximal order $|g| = d$ defines a cyclic subgroup $\langle g \rangle \leq G$ which is isomorphic to the additive group of $\mathbb{Z}_d$. Thus we can use $\langle g \rangle$ for the domain of $\mathbb{Z}_d$.

However, in general, there is no *unique* element of maximal order. Hence, we have to combine subgroups $\langle g \rangle$ and $\langle h \rangle$ for different elements $g, h \in G$ with $|g| = |h| = d$ to obtain a *single* group which is isomorphic to $\mathbb{Z}_d$. Let $X_G = \{(g,h) \in G^2 : |g| = d, h \in \langle g \rangle\}$, that is $X_G$ is the collection of all cyclic subgroups $\langle g \rangle \leq G$ of maximal order. By Lemma 3.3, the set $X_G$ is DTC-definable in $G$. We consider the following equivalence relation $\approx$ on $X_G$:

$$(g_1,h_1) \approx (g_2,h_2) \ :\Longleftrightarrow\ h_1 = k \cdot g_1 \text{ and } h_2 = k \cdot g_2 \text{ for some } 0 \leq k < d.$$

By using similar arguments as in the proof of Lemma 3.3 it follows that the equivalence relation $\approx$ on $X_G$ is DTC-definable in $G$. We observe that each equivalence class can be identified with an element $0 \leq k < d$. Hence, the idea is to use the set $X_G/\approx$ as the domain for the ring $\mathbb{Z}_d$.

More specifically, to represent the elements of $\mathbb{Z}_d \times G$ we use tuples $(g_1, g_2, h) \in X_G \times G$ and lift the equivalence relation $\approx$ on $X_G$ to $X_G \times G$ by setting $(g_1, g_2, h) \approx (g_1', g_2', h')$ if $(g_1, g_2) \approx (g_1', g_2')$ and $h = h'$. Then the set $(X_G \times G)/\approx$ can be identified with the set $\mathbb{Z}_d \times G$ and it can be constructed via a DTC-interpretation. It remains to define the appropriate ring addition and ring multiplication on this set to turn it into the ring $\mathbb{Z}_d \circledast G$.

The crucial observation is that for a fixed $g \in G$, with $|g| = d$, we obtain a set of representatives for $(X_G \times G)/\approx$ as $\{(g, h_1, h_2) \in X_G \times G\} \sim \langle g \rangle \times G$. Of course, every relation on these sets of representatives uniquely induces a relation on $(X_G \times G)/\approx$. Hence, it suffices to express in DTC the ring addition and ring multiplication on such sets of representatives in a way that the operations induced on $(X_G \times G)/\approx$ do not depend on the specific parameters $g \in G$.

Recall that, by Lemma 3.3 (d), we can define in DTC the linear order $\leq^g = \{(k \cdot g, \ell \cdot g) : 0 \leq k \leq \ell < d\}$ on $\langle g \rangle$. Let $h \in G$ be a different parameter with $|h| = d$ and let $\leq^h$ be the corresponding linear order on $\langle h \rangle$. Then we have $(g, g_1) \approx (h, h_1)$ for $(g, g_1), (h, h_1) \in X_G$ if, and only if, the position of $g_1$ in $\leq^g$ coincides with the position of $h_1$ in $\leq^h$. Hence, these linear orders induce $\approx$-canonical orderings on $X_G/\approx$, and can hence be used to canonically define the ring operations on $(X_G \times G)/\approx$. Since DTC can express every LOGSPACE-computable function on ordered structures, it follows that the ring addition on $\langle g \rangle \times G$ can be defined in DTC. Finally, for the ring multiplication we can use similar arguments as in the proof of Lemma 3.3.

**Lemma 3.6.** *There exists a* DTC*-interpretation $\mathcal{I}$ which translates $\tau_{group}$-structures into $\tau_{ring}$-structures such that for every group $\mathfrak{G} = (G, +)$ the structure $\mathcal{I}(\mathfrak{G})$ is the ring $\mathbb{Z}_d \circledast G$ where $d = \max\{|g| : g \in G\}$ is the exponent of $G$.*

By an inspection of our construction, it further follows that the embedding $\Phi : G \to \mathbb{Z}_d \circledast G$ of the group $G$ into $\mathbb{Z}_d \circledast G$ can be defined in DTC as well. The same is true for the ring $R = \mathbb{Z}_d$ (which we construct via the interpretation) and the embedding $\Psi : \mathbb{Z}_d \to \mathbb{Z}_d \circledast G$ of $R$ into $\mathbb{Z}_d \circledast G$.

Furthermore, our interpretation crucially makes use of the definable congruence relation in order to represent the elements of the ring $\mathbb{Z}_d$ as equivalence classes of cyclic subgroups of maximal order. It is easy to see that a similar translation cannot be achieved without using this congruence relation. On the other hand, if the interpretation can use a parameter $g$ (to fix a group element $g \in G$ of maximal order $d = |g|$), then we can identify the elements of $\mathbb{Z}_d$ with the elements of $\langle g \rangle$, and a trivial congruence relation indeed suffices.

Moreover, if we want to define the commutative ring $R \circledast G$ starting from a given $R$-module $G$ (which means that, in contrast to the previous case, we do not have to construct the ring elements of $R$ in the first place), then this is trivial, since the elements of $R \circledast G$ are just tuples in $R \times G$, and the ring operations of $R \circledast G$ can immediately be be defined from the group addition and the scalar multiplication in the $R$-module $G$.

**Lemma 3.7.** *There exists an* FO*-interpretation $\mathcal{I}$ which translates $\tau_{module}$-structures into $\tau_{ring}$-structures, such that for every module $\mathfrak{M} = (A, G, R, +, \cdot)$ the structure $\mathcal{I}(\mathfrak{M})$ is the ring $R \circledast G$.*

Of course, also for this case, the embedding $\Phi : G \to R \circledast G$ of the group $G$ into the ring $R \circledast G$ and the embedding $\Psi : R \to R \circledast G$ of the ring $R$ into $R \circledast G$ are definable as well.

Let us come back to our proof of Theorem 3.4. Our original aim was to construct in DTC, given an Abelian group $G$ with exponent $d$, the extension of $G$ to a $\mathbb{Z}_d$-module. Since we proved that we can obtain the ring $\mathbb{Z}_d \circledast G$ via a DTC-interpretation, this now follows immediately, since the ring $\mathbb{Z}_d \circledast G$ already "contains" the extension of $G$ to a $\mathbb{Z}_d$-module, as we saw above.

**Lemma 3.8.** *There exists a* DTC*-interpretation $\mathcal{I}$ which translates $\tau_{group}$-structures into $\tau_{module}$-structures such that, for every group $\mathfrak{G} = (G, +)$, the structure $\mathcal{I}(\mathfrak{G})$ is the extension of $\mathfrak{G}$ to a $\mathbb{Z}_d$-module where $d = \max\{|g| : g \in G\}$.*

*Proof of Theorem 3.4.* Let us start with the reduction $\textsc{SlvG} \leq^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SlvM}$. Given a (representation of) a linear equation system $(A, G, +, M, \vec{c}) \in \mathrm{Ls}(\tau_{\text{les-g}})$ over a group $(G, +)$, we first use Lemma 3.8 to extend $(G, +)$ via a DTC-interpretation to a $\mathbb{Z}_d$-module $(G, \mathbb{Z}_d, +, \cdot)$. We then lift the $I \times J$-matrix $M$ over $\{0, 1\}$ to an $I \times J$-matrix $M_*$ over $\mathbb{Z}_d$ where we naturally identify the element 1 with the multiplicative identity of the ring $\mathbb{Z}_d$. Moreover, the $I$-vector $\vec{c}$ over the group $G$ naturally translates into an $I$-vector $\vec{c}_* = \Phi(\vec{c})$ over the $\mathbb{Z}_d$-module $G$. Obviously, the linear equation system $M \cdot \vec{x} = \vec{c}$ over the group $(G, +)$ and the linear equation system $M_* \cdot \vec{x}_* = \vec{c}_*$ over the module $(\mathbb{Z}_d, G, +, \cdot)$, where $\vec{x}$ is a $J$-vector of variables ranging over $G$ and $\vec{x}_*$ is a $J$-vector of variables ranging over $\Phi(G)$, are equivalent.

For the reduction $\textsc{SlvCG} \leq^{\mathrm{tt}}_{\mathrm{DTC}} \textsc{SlvM}_{\leqslant}$ we additionally have to construct a linear order on the interpreted module $(\mathbb{Z}_d, G, +, \cdot)$. By Lemma 3.3 (d), we can define a linear order on $\mathbb{Z}_d$, so it suffices to construct an order on $G$. Since $(G, +)$ is a *cyclic* group, we can again use Lemma 3.3 (d) to obtain a DTC-definable linear order where we use a generator $g \in G$ of $G$, that is $\langle g \rangle = G$, as parameter. Thus, together with the construction from above, we obtain a DTC-interpretation $\mathcal{I}(z)$, with parameter $z$, which translates a linear equation system $\mathfrak{A} = (A, G, +, M, \vec{c}) \in \mathrm{Ls}(\tau_{\text{les-g}})$ with a cyclic group $(G, +)$ for every $g \in G$ with $\langle g \rangle = G$ into an equivalent linear equation system $\mathcal{I}(\mathfrak{A}, z \mapsto g) \in \mathrm{Ls}(\tau^{\leqslant}_{\text{les-m}})$ over an *ordered* module. Hence, if we let $\mathcal{Q}$ denote the Lindström-quantifier associated with the class $\textsc{SlvM}_{\leqslant}$, then

$$\varphi = (\exists z \in G)\,(\forall x.x \in \langle z \rangle \wedge \mathcal{Q}(\mathcal{I}(z)))$$

is a sentence of $\mathrm{DTC}(\mathcal{Q})$ which defines SlvCG. □

### 3.2.2 Translations between modules and rings

We proceed to study the relationship between the solvability problems over modules and rings. To start, let us briefly discuss the case of commutative rings.

It is easy to see that a reduction of the solvability problem from (ordered) commutative rings to (ordered) modules is trivial. This is because every commutative ring $R$ also is an $R$-module. For the other direction, we can apply the FO-definable transformation of an $R$-module $G$ into the commutative ring $R \circledast G$, see Lemma 3.7.

**Theorem 3.9.** $\textsc{SlvM} \equiv^{\mathrm{m}}_{\mathrm{FO}} \textsc{SlvCR}$ *and* $\textsc{SlvM}_{\leqslant} \equiv^{\mathrm{m}}_{\mathrm{FO}} \textsc{SlvCR}_{\leqslant}$.

*Proof.* We first establish the reduction $\textsc{SlvM} \leq^{\mathrm{m}}_{\mathrm{FO}} \textsc{SlvCR}$. To this end, let $(A, G, R, +, \cdot, M, \vec{c}) \in \textsc{Ls}(\tau_{\text{les-m}})$ be a (representation of) a linear equation system over an $R$-module $G$. We use Lemma 3.7 to translate the $R$-module $G$ into the ring $R \circledast G$ and we lift the $I \times J$-matrix over $R$ to an $I \times J$-matrix $M_\Psi = \Psi(M)$ over $\Psi(R)$ via $\Psi$ and the $I$-vector $\vec{c}$ over $G$ to an $I$-vector $\vec{c}_\Phi = \Phi(\vec{c})$ over $\Phi(G)$ via $\Phi$. These transformation can be defined in FO.

The linear equation system $M_\Psi \cdot \vec{x} = \vec{c}_\Phi$ over $R \circledast G$, where $\vec{x}$ is a $J$-vector of variables over $R \circledast G$, is equivalent to the linear equation system $M \cdot \vec{x} = \vec{c}$ over the $R$-module $G$ where the variables of the $J$-vector $\vec{x}$ range over $G$. To see this, assume that the system $M \cdot \vec{x} = \vec{c}$ over the $R$-module $G$ has a solution $\vec{b} \in G^J$. From $M \cdot \vec{b} = \vec{c}$ we conclude that $\Phi(M \cdot \vec{b}) = \Phi(\vec{c})$. By Lemma 3.5, we have $\Phi(M \cdot \vec{b}) = \Psi(M) \cdot \Phi(\vec{b}) = \Phi(\vec{c})$ which means that $\vec{b}_\Phi := \Phi(\vec{b}) \in \Phi(G)^J$ is a solution of the system $M_\Psi \cdot \vec{x} = \vec{c}_\Phi$

For the other direction, let $\vec{b} \in (R \circledast G)^J$ be such that $M_\Psi \cdot \vec{b} = \vec{c}_\Phi$. By Lemma 3.5, we can write the solution vector $\vec{b}$ as $\vec{b} = \vec{b}_\Psi + \vec{b}_\Phi$ for two unique $J$-vectors $\vec{b}_\Psi \in \Psi(R)^J$ and $\vec{b}_\Phi \in \Phi(G)^J$. Then we have

$$M_\Psi \cdot (\vec{b}_\Psi + \vec{b}_\Phi) = M_\Psi \cdot \vec{b}_\Psi + M_\Psi \cdot \vec{b}_\Phi = \vec{c}_\Phi.$$

Since $\vec{c}_\Phi \in \Phi(G)^I$, we conclude that $M_\Psi \cdot \vec{b}_\Psi = \vec{0}$ which means that also $\vec{b}_\Phi$ is a solution of the system $M_\Psi \cdot \vec{x} = \vec{c}_\Phi$.

We claim that $\vec{b}_* := \Phi^{-1}(\vec{b}_\Phi)$ is a solution of the linear system $M \cdot \vec{x} = \vec{c}$ over the $R$-module $G$. Since we have $\Psi(M) \cdot \Phi(\vec{b}_*) = \Phi(\vec{c})$, we can again use Lemma 3.5 to conclude that $\Phi(M \cdot \vec{b}_*) = \Phi(\vec{c})$ which means that $M \cdot \vec{b}_* = \vec{c}$.

Finally, to establish the remaining reduction $\textsc{SlvM}_{\leqslant} \leq^{\mathrm{m}}_{\mathrm{FO}} \textsc{SlvCR}_{\leqslant}$, we only need to observe that if we start with an *ordered* $R$-module $G$, then we obtain a linear order on $R \circledast G$ by using the lexicographical ordering on $R \times G$. □

We turn our attention to the solvability problem over general, that is not necessarily commutative, rings. Of course, for the reduction from modules to rings we can proceed as before. However, for the other direction we need to modify our arguments from above to take care of the non-commutative ring multiplication. For example, the linear terms $r \cdot x$ and $x \cdot r$ for coefficients $r \in R$ and a variable $x$ are, in general, not equivalent.

To understand the main idea of the reduction, it is helpful to think of the solvability problem in the following way. We know that a linear equation system $M \cdot \vec{x} = \vec{c}$ over a commutative ring $R$ is solvable if, and only if, the constants vector $\vec{c}$ can be expressed as a linear combination of the columns of

the coefficient matrix $M$. Now assume that we modify the coefficient matrix $M$ by replacing each column $\vec{d}$ by all of its multiples $r \cdot \vec{d}$ for ring elements $r \in R$. Then the linear equation system is solvable if, and only if, the constants vector $\vec{c}$ can be expressed as the *sum over a subset* of columns of the modified coefficient matrix $M'$. Hence, we have reduced the search for an *arbitrary* linear combination of columns to the search for a linear combination of columns of a simple form, that is to a combination for which all coefficients are zero or one. Using this trick together with our techniques from Lemma 3.6 to transform Abelian groups into modules we can prove the following result.

**Theorem 3.10.** $\textsc{SlvM} \equiv^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SlvR}$ *and* $\textsc{SlvM}_{\leqslant} \equiv^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SlvR}_{\leqslant}$.

*Proof.* The interesting case is the reduction $\textsc{SlvR} \leq^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SlvM}$. Recall that an instance of SlvR is of the form $\mathfrak{A} = (A, R, +, \cdot, M_\ell, M_r, \vec{c}\,) \in \mathrm{Ls}(\tau_{\text{les-r}})$ where $(R, +, \cdot)$ is a (not necessarily commutative) ring, where $M_\ell$ is an $I \times J$-matrix over $R$, where $M_r$ is a $J \times I$ matrix over $R$ and where $\vec{c}$ is an $I$-vector over $R$. The represented linear equation system is $M_\ell \cdot \vec{x} + (\vec{x} \cdot M_r)^T = \vec{c}$ where $\vec{x} = (x_j)_{j \in J}$ is a $J$-vector of variables ranging over $R$.

Since the additive group $(R, +)$ of the ring $R$ is Abelian, we can use Lemma 3.6 to construct in DTC the ring $\mathbb{Z}_d \circledast R$ where $d = \max\{|r| : r \in R\}$ is the exponent of the group $(R, +)$. Recall that $\mathbb{Z}_d \circledast R$ is a *commutative* ring and that every commutative ring is a module over itself.

From Lemma 3.5 we know that the group $(R, +)$ embeds into $\mathbb{Z}_d \circledast R$ via the definable mapping $\Phi : R \to \mathbb{Z}_d \circledast R$. We introduce for each variable $x_j$ a set of new variables $\{x_j^r : r \in R\}$ which range over $\mathbb{Z}_d \circledast R$ and we let $\vec{x}_* = (x_j^r)_{j \in J, r \in R}$ be the $J_* := (J \times R)$-vector consisting of these new variables. Moreover, we let $M$ be the $I \times J_*$-matrix over $\mathbb{Z}_d \circledast R$ which is defined for $i \in I$ and $(j, r) \in J_*$ as

$$M(i, j, r) := \Phi\big(M_\ell(i, j) \cdot r + r \cdot M_r(j, i)\big).$$

Thus, the column with index $(j, r)$ of the matrix $M$ is the sum of the $j$-th column of the matrix $M_\ell$ multiplied by $r$ from the right and the $j$-th row of the matrix $M_r$ multiplied by $r$ from the left lifted to the ring $\mathbb{Z}_d \circledast R$. We claim that the linear system $M \cdot \vec{x}_* = \Phi(\vec{c})$ over the *commutative* ring $\mathbb{Z}_d \circledast R$ is equivalent to the linear equation system $M_\ell \cdot \vec{x} + (\vec{x} \cdot M_r)^T = \vec{c}$ over the ring $R$. Indeed, if we can prove this, then the theorem follows, because Lemma 3.6 guarantees that the system $M \cdot \vec{x}_* = \Phi(\vec{c})$ can be constructed by using a DTC-interpretation.

For the first direction, let $\vec{b} \in R^J$ be a such that $M_\ell \cdot \vec{b} + (\vec{b} \cdot M_r)^T = \vec{c}$. We define the $J_*$-vector $\vec{b}_*$ over $\mathbb{Z}_d$ for $(j, r) \in J_*$ as

$$\vec{b}_*(j, r) = \begin{cases} (1, 0), & \text{if } \vec{b}(j) = r \\ (0, 0), & \text{otherwise.} \end{cases}$$

We claim that $\vec{b}_*$ is a solution of the linear equation system $M \cdot \vec{x}_* = \Phi(\vec{c})$.

Indeed for $i \in I$ we have

$$\begin{aligned}(M \cdot \vec{b}_*)(i) &= \sum_{(j,r)\in J_*} M(i,j,r) \cdot \vec{b}_*(j,r) = \sum_{\vec{b}(j)=r} M(i,j,r) \\ &= \sum_{\vec{b}(j)=r} \Phi\Big(M_\ell(i,j)\cdot r + r \cdot M_r(i,j)\Big) \\ &= \Phi\Big(\sum_{\vec{b}(j)=r} M_\ell(i,j)\cdot r + r \cdot M_r(i,j)\Big) \\ &= \Phi\Big((M_\ell \cdot \vec{b} + (\vec{b}\cdot M_r)^T)(i)\Big) = \Phi\Big(\vec{c}(i)\Big) = \Phi(\vec{c})(i).\end{aligned}$$

For the other direction, let $\vec{b} \in (\mathbb{Z}_d \times R)^{J_*}$ be a solution of the linear system $M \cdot \vec{x}_* = \Phi(\vec{c})$ over $\mathbb{Z}_d \circledast R$. We use Lemma 3.5 to decompose $\vec{b} = \vec{b}_\Psi + \vec{b}_\Phi$ into two $J_*$-vectors $\vec{b}_\Psi \in \Psi(\mathbb{Z}_d)^{J_*}$ and $\vec{b}_\Phi \in \Phi(R)^{J_*}$. Since $M$ is an $I \times J_*$-matrix over $\Phi(R)$, we have $M \cdot \vec{b}_\Phi = \vec{0}$. Hence, also $\vec{b}_\Psi$ is a solution of the linear system.

We use $\vec{b}_\Psi$ to construct a vector $\vec{b}_* \in R^J$ which is a solution of the system $M_\ell \cdot \vec{x} + (\vec{x} \cdot M_r)^T = \vec{c}$. Specifically we set for $j \in J$:

$$\vec{b}_*(j) = \Phi^{-1}\Big(\sum_{r\in R} \vec{b}_\Psi(j,r)\cdot\Phi(r)\Big).$$

**Claim:** For all $r, s \in R$ and $z \in \Psi(\mathbb{Z}_d)$ we have

$$\Phi\Big(r \cdot \Phi^{-1}(\Phi(s)\cdot z)\Big) = \Phi(r\cdot s)\cdot z = z \cdot \Phi(r\cdot s) = \Phi\Big(\Phi^{-1}(\Phi(r)\cdot z)\cdot s\Big).$$

*Proof of claim:* Since $z \in \Psi(\mathbb{Z}_d)$, we can equivalently write the term $t \cdot z$ as $t+t+\cdots+t$ ($z$-times) for every $t \in \Phi(R)$. This immediately yields the claim. ⊣

With this preparation we have for $i \in I$ that $\Phi\big(M_\ell \cdot b_* + (b_* \cdot M_r)^T\big)(i) =$

$$\begin{aligned}&\Phi\Big(\sum_{j\in J} M_\ell(i,j)\cdot\Phi^{-1}\Big(\sum_{r\in R}\vec{b}_\Psi(j,r)\cdot\Phi(r)\Big) + \Phi^{-1}\Big(\sum_{r\in R}\vec{b}_\Psi(j,r)\cdot\Phi(r)\Big)\cdot M_r(j,i)\Big) \\ &= \sum_{(j,r)\in J_*} \Phi\Big(M_\ell(i,j)\cdot\Phi^{-1}\big(\vec{b}_\Psi(j,r)\cdot\Phi(r)\big) + \Phi^{-1}\big(\vec{b}_\Psi(j,r)\cdot\Phi(r)\big)\cdot M_r(j,i)\Big) \\ &= \sum_{(j,r)\in J_*} \Phi\Big(M_\ell(i,j)\cdot r + r\cdot M_r(j,i)\Big)\cdot b_\Psi(j,r) \\ &= \sum_{(j,r)\in J_*} M(i,j,r)\cdot\vec{b}_\Psi(j,r) = \Phi(\vec{c})(i).\end{aligned}$$

This shows that $M_\ell \cdot b_* + (b_* \cdot M_r)^T = \vec{c}$ and thus completes our first reduction.

Finally, we observe that if we start from an ordered ring $R$, then also the ring $\mathbb{Z}_d \circledast R$ can easily be extended by a linear order. Indeed we can take the lexicographical order on $\mathbb{Z}_d \times R$. This shows that $\textsc{SLvR}_\leqslant \leq^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SLvM}_\leqslant$. □

A closer inspection of our reduction $\textsc{SLvR} \leq^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SLvM}$ (see the proof of Theorem 3.10) reveals that we have also established a DTC-reduction which translates from (not necessarily commutative) rings to *commutative* rings.

**Theorem 3.11.** $\textsc{SLvR} \equiv^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SLvCR}$ *and* $\textsc{SLvR}_\leqslant \equiv^{\mathrm{m}}_{\mathrm{DTC}} \textsc{SLvCR}_\leqslant$.

**From ordered modules to cyclic groups** We saw that, up to DTC-reductions, the solvability problems over modules, rings, and commutative rings are equivalent. Moreover, we presented a reduction from the solvability problem over Abelian groups to the solvability problem over modules. To complete the picture, we had to reduce the solvability problem over modules to the solvability problems over Abelian groups. However, we can only give a reduction for the special case of *ordered* modules. Actually, for this particular case, we obtain a much stronger reduction, not only to Abelian groups, but even to cyclic groups.

The reason why we need an ordering on the module is that, for our proof, we have to decompose the additive group of the module into cyclic subgroups. In general, such a decomposition is not unique and, due to symmetries, provably not definable in a logic. Thus, it remains an interesting open question whether a logical reduction, from modules to Abelian groups, can also be achieved by using a different approach (for example, canonisation techniques).

**Theorem 3.12.** $\textsc{SLVM}_{\leqslant} \leq_{\text{FP}} \textsc{SLVCG}$.

The proof of this theorem consists of three parts. First of all, we translate a linear equation system over an $R$-module $G$ into an equivalent linear equation system over the commutative ring $\mathbb{Z}_d \circledast G$ where $d$ is the exponent of $G$. Moreover, we show that the resulting system has a solution over $\mathbb{Z}_d \circledast G$ if, and only if, it has a solution over $\Psi(\mathbb{Z}_d) \leq \mathbb{Z}_d \circledast G$. For this step we use the same idea as in the proof of Theorem 3.10, that is we replace each column $\vec{d}$ of the coefficient matrix by the set of columns $\{\vec{d} \cdot g : g \in G\}$.

As a second step, we use the linear order on the group $G$ to fix a set of generators $g_1, \dots, g_k$. This allows us to identify each group element with an element in $\langle g_1 \rangle \oplus \cdots \oplus \langle g_k \rangle \leq \mathbb{Z}_d \oplus \cdots \oplus \mathbb{Z}_d$.

Finally, we transform the resulting linear equation system over $\mathbb{Z}_d \circledast G$ into an equivalent linear equation over $\mathbb{Z}_d$ where we use the decomposition of $G$ into cyclic subgroups together with the fact that solutions of the modified system can always be found over $\mathbb{Z}_d$.

**Lemma 3.13.** *There is a* DTC*-interpretation $\mathcal{I}$ which translates a linear equation system $\mathfrak{A} = (A, G, R, +, \cdot, \leqslant, M, \vec{c}) \in \text{LS}(\tau^{\leqslant}_{\textit{les-m}})$ over an ordered $R$-module $G$ into an equivalent linear equation system $\mathcal{I}(\mathfrak{A}) \in \text{LS}(\tau^{\leqslant}_{\textit{les-r}})$ over the ordered commutative ring $\mathbb{Z}_d \circledast G$, where $d$ denotes the exponent of the group $G$. Moreover, the coefficient matrix and the constants vector of $\mathcal{I}(\mathfrak{A})$ have entries in $\Phi(G)$, and, in particular, the system $\mathcal{I}(\mathfrak{A})$ is solvable if, and only if, it has a solution over $\Psi(\mathbb{Z}_d)$.*

*Proof.* Let $\mathfrak{A} = (A, G, R, +, \cdot, \leqslant, M, \vec{c}) \in \text{LS}(\tau^{\leqslant}_{\text{les-m}})$ be (a representation of) a linear equation system over the $R$-module $G$ where $M$ is an $I \times J$-matrix over the commutative ring $R$, $\vec{c}$ is an $I$-vector over $G$, and where the encoded linear equation system is $M \cdot \vec{x} = \vec{c}$ for a $J$-vector $\vec{x} = (x_j)_{j \in J}$ of variables $x_j$ that

range over $G$. For $J_* := J \times G$ we let $M_*$ be the following $I \times J_*$-matrix over $\Phi(G) \leq \mathbb{Z}_d \circledast G$:

$$M_*(i,j,g) := \Phi(M(i,j) \cdot g) \text{ for } i \in I, (j,g) \in J_*.$$

We claim that the linear equation system $M_* \cdot \vec{x}_* = \Phi(\vec{c})$ over the ring $\mathbb{Z}_d \circledast G$ is solvable if, and only if, the system $M \cdot \vec{x} = \vec{c}$ has a solution $\vec{b} \in G^J$. Moreover, we claim that whenever the system $M_* \cdot \vec{x}_* = \Phi(\vec{c})$ is solvable, then we can find a solution $\vec{b}_* \in \Psi(\mathbb{Z}_d)^{J_*}$. For the one direction, let $\vec{b} \in G^J$ be such that $M \cdot \vec{b} = \vec{c}$. For

$$\vec{b}_*(j,g) := \begin{cases} (1,0), & \text{if } \vec{b}(j) = g \\ (0,0), & \text{else,} \end{cases}$$

we have $\vec{b}_* \in \Psi(\mathbb{Z}_d)^{J_*}$ and

$$\begin{aligned} M_* \cdot \vec{b}_* &= \Big( \sum_{j \in J} \sum_{g \in G} \Phi(M(i,j) \cdot g) \cdot \vec{b}_*(j,g) \Big)_{i \in I} = \Big( \sum_{j \in J} \Phi(M(i,j) \cdot \vec{b}(j)) \Big)_{i \in I} \\ &= \Phi\Big( \sum_{j \in J} M(i,j) \cdot \vec{b}(j) \Big)_{i \in I} = \Phi(\vec{c}). \end{aligned}$$

For the other direction, let $\vec{b}_* \in (\mathbb{Z}_d \times G)^{J_*}$ be a solution of the linear system $M_* \cdot \vec{x}_* = \Phi(\vec{c})$. By Lemma 3.5 we can write $\vec{b} = \vec{b}_\Psi + \vec{b}_\Phi$ for two unique $J_*$-vectors $\vec{b}_\Psi \in \Psi(\mathbb{Z}_d)^{J_*}$ and $\vec{b}_\Phi \in \Phi(G)^{J_*}$. Since all entries of $M_*$ are in $\Phi(G)$ and since $\Phi(g) \cdot \Phi(h) = 0$ for all $g, h \in G$, we have $M_* \cdot \vec{b}_\Phi = \vec{0}$ which means that also $\vec{b}_\Psi$ is a solution of the linear system. This in turn proves our second claim.

To construct a solution $\vec{b} \in G^J$ of the linear system $M \cdot \vec{x} = \vec{c}$ we set for $j \in J$

$$\vec{b}(j) = \Phi^{-1}\big( \sum_{g \in G} \Phi(g) \cdot \vec{b}_\Psi(j,g) \big).$$

This completes our proof, since for all $i \in I$ we have

$$\begin{aligned} \Phi(M \cdot \vec{b})(i) &= \Phi\Big( \sum_{j \in J} M(i,j) \cdot \Phi^{-1}\big( \sum_{g \in G} \Phi(g) \cdot \vec{b}_\Psi(j,g) \big) \Big) \\ &= \sum_{j \in J} \sum_{g \in G} \Phi\Big( M(i,j) \cdot \Phi^{-1}(\Phi(g) \cdot \vec{b}_\Psi(j,g)) \Big) \\ &= \sum_{(j,g) \in J_*} \Phi\Big( M(i,j) \cdot g \Big) \cdot \vec{b}_\Psi(j,g) = \Phi(\vec{c})(i). \end{aligned}$$

Finally, the linear order on $G$ suffices to construct a linear order on $\mathbb{Z}_d \circledast G$. □

In order to prove Theorem 3.12, we describe an FP-reduction which shows that $\text{SLVM}_\leqslant \leq_{\text{FP}} \text{SLVCG}$. By Lemma 3.13 it suffices to treat the case of a linear equation system $M \cdot \vec{x} = \vec{c}$ over the ring $\mathbb{Z}_d \circledast G$ where $M$ is an $I \times J$-matrix over $\Phi(G)$ and where $\vec{c}$ is an $I$-vector over $\Phi(G)$. Moreover, we know that the system is solvable if, and only if, there exists a solution vector $\vec{b} \in \Psi(\mathbb{Z}_d)^J$. As before, $d$ denotes the exponent of the group $G$.

The next step is to use the linear order on $\Phi(G)$ to fix a generating set $\{g_1, \dots, g_k\} \subseteq \Phi(G)$ of the (Abelian) group $(\Phi(G), +)$ such that the group $(\Phi(G), +)$ decomposes as $(\Phi(G), +) \cong \langle g_1 \rangle \oplus \cdots \oplus \langle g_k \rangle$. Recall from Section 2.5 that this step can be expressed in fixed-point logic. We let $\ell_i$ denote the order of $g_i$ in $(\Phi(G), +)$. Note that $\ell_i \mid d$ for all $1 \leq i \leq k$. For notational convenience we set $e_i := d/\ell_i$. We identify the cyclic group $\langle g_i \rangle \leq \Phi(G)$ with $\mathbb{Z}_{\ell_i}$. More specifically, we use the group isomorphism $\mathbb{Z}_{\ell_i} \to e_i\mathbb{Z}_d, z \mapsto e_i \cdot z$ (for $0 \leq z < \ell_i - 1$) to see that the groups $\langle g_i \rangle \leq \Phi(G)$ can homomorphically be embedded into $\mathbb{Z}_d$. Altogether, we obtain a group isomorphism $\Lambda : \Phi(G) \xrightarrow{\sim} \Lambda(G) := (e_1\mathbb{Z}_d) \oplus \cdots \oplus (e_k\mathbb{Z}_d)$ which represents every $h \in \Phi(G)$ as $\Lambda(h) = (h_1, \dots, h_k)$ where $h_i \in e_i\mathbb{Z}_d$. We write $\Lambda^i(h)$ for the projection onto the $i$-th component, i.e. $\Lambda^i(h) = h_i$. Considering $h_i$ as an element of $\Lambda(G)$, we have $\Lambda(h) = \sum_{i=1}^{k} \Lambda^i(h)$ for $h \in \Phi(G)$.

Clearly, due to the linear order on $\Phi(G)$, the decomposition of $(\Phi(G), +)$ into its cyclic components $\langle g_i \rangle$, the groups $e_i\mathbb{Z}_d$, and the isomorphism $\Lambda$ together with the projections $\Lambda^i$ can easily be constructed via an FP-interpretation. The exponent of the group $\Lambda(G)$ is $d$, so we can consider the natural extension of $\Lambda(G)$ to a (right) $\mathbb{Z}_d$-module. As it turns out, the isomorphism $\Lambda$ respects the multiplication by elements from $\mathbb{Z}_d$ in the following sense.

**Claim:** For $g \in \Phi(G)$ and $z \in \mathbb{Z}_d$ we have $\Lambda(g \cdot \Psi(z)) = \Lambda(g) \cdot z$. Moreover, we have $\Lambda(g) \cdot z = \sum_{i=1}^{k} \Lambda^i(g) \cdot z$.

*Proof of claim:* For the first claim just use that $g \cdot \Psi(z) = g + \cdots + g$ and $\Lambda(g) \cdot z = \Lambda(g) + \cdots + \Lambda(g)$ (where $x + \cdots + x$ abbreviates the $z$-fold sum of the element $x$). The second claim follows analogously by using distributivity. ⊣

By the above claim, the linear equation system $M \cdot \vec{x} = \vec{c}$ has a solution $\vec{b} \in \Psi(\mathbb{Z}_d)^J$ if, and only if, there is a $J$-vector $\vec{b}_* \in \mathbb{Z}_d^J$ such that $\Lambda(M) \cdot \vec{b}_* = \Lambda(\vec{c})$. We write $\Lambda(M) = \sum_{i=1}^{k} \Lambda^i(M)$ and $\Lambda(\vec{c}) = \sum_{i=1}^{k} \Lambda^i(\vec{c})$. Then $\Lambda(M) \cdot \vec{b}_* = \Lambda(\vec{c})$ is equivalent to $\Lambda^i(M) \cdot \vec{b}_* = \Lambda^i(\vec{c})$ for all $1 \leq i \leq k$. Note that $\Lambda^i(M)$ is an $I \times J$-matrix over $\mathbb{Z}_d$ and $\Lambda^i(\vec{c})$ is an $I$-vector over $\mathbb{Z}_d$.

We proceed to combine the linear equation systems $\Lambda^i(M) \cdot \vec{x}_* = \Lambda^i(\vec{c})$ for $1 \leq i \leq k$ over $\mathbb{Z}_d$ into a single linear system over $\mathbb{Z}_d$. To this end, we set $I_* := \{1, \dots, k\} \times I$ and we define an $I_* \times J$-coefficient matrix $M_*$ and an $I_*$-vector $\vec{c}_*$ over $\mathbb{Z}_d$ as follows

$$M_*(\ell, i, j) := \Lambda^\ell(M(i, j)) \text{ and } \vec{c}_*(\ell, i) = \Lambda^\ell(\vec{c}(i)), \text{ for } (\ell, i) \in I_*, j \in J.$$

Then for every $J$-vector $\vec{b}_*$ over $\mathbb{Z}_d$ we have $M_* \cdot \vec{b}_* = \vec{c}_*$ if, and only if, $\Lambda^i(M) \cdot \vec{b}_* = \Lambda^i(\vec{c})$ for all $1 \leq i \leq k$. Again, it is clear that $M_*$ and $\vec{c}_*$ can easily be constructed via an FP-interpretation.

So far, we have translated the given linear equation system over an ordered $R$-module $G$ into an equivalent linear equation system over the ring $\mathbb{Z}_d$, where $d$ is the exponent of $G$. To finally obtain an equivalent linear equation system over the cyclic *group* $\mathbb{Z}_d$, the crucial step is to transform linear terms $z \cdot x$ into equivalent sets of linear terms which only use the element $1 \in \mathbb{Z}_d$ as coefficients. The following lemma shows that this is possible and thus it completes our proof of Theorem 3.12.

**Lemma 3.14.** *There is a* DTC*-interpretation $\mathcal{I}$ which translates a linear equation system $\mathfrak{A} \in \mathrm{Ls}(\tau_{les\text{-}r})$ over a ring $\mathbb{Z}_d$ into an equivalent linear equation system $\mathcal{I}(\mathfrak{A}) \in \mathrm{Ls}(\tau_{les\text{-}g})$ over the cyclic group $\mathbb{Z}_d$.*

*Proof.* Let $M \cdot \vec{x} = \vec{c}$ be a linear equation over the ring $\mathbb{Z}_d$ where $M$ is an $I \times J$-matrix and $\vec{c}$ is an $I$-vector over $\mathbb{Z}_d$. Note that by Lemma 3.3 we can define the natural linear order on $\mathbb{Z}_d$ in DTC.

The idea to obtain an equivalent linear system over the *group* $\mathbb{Z}_d$ is very simple. We just need to rewrite linear terms $z \cdot x$ for constants $z \in \mathbb{Z}_d$ and variables $x$ as $z \cdot x = x + \cdots + x$ ($z$-times). Technically, this requires to introduce $d$ copies $\{x^z : z \in \mathbb{Z}_d\}$ of each variable $x$ together with the constraints $x^{z_1} = x^{z_2}$ for all $z_1, z_2 \in \mathbb{Z}_d$. Then we can substitute each term $z \cdot x$ by the term $\sum_{z_1 < z} x^{z_1}$ and we obtain an equivalent linear equation system over the extended set of variables $\{x_j^z : (j, z) \in J_*\}$ where $J_* := J \times \mathbb{Z}_d$.

However, we still face a problem. When we write the constraints $x^{z_1} = x^{z_2}$ for $z_1, z_2 \in \mathbb{Z}_d$ as linear equations $x^{z_1} - x^{z_2} = 0$ they still contain $-1$ as coefficient. To overcome this obstacle we introduce for every variable $x_j^z$ its dual variable $\overline{x}_j^z$ together with the equations $x_j^z + \overline{x}_j^z = 0$. Then, the linear equations $x^{z_1} - x^{z_2} = 0$ can equivalently be expressed as $x^{z_1} + \overline{x}^{z_2} = 0$.

Formally, we let $I_+ := I \uplus (J \times \mathbb{Z}_d \times \mathbb{Z}_d) \uplus (J \times \mathbb{Z}_d)$ and $J_+ := J_* \times \{1, -1\}$ and we define an $I_+ \times J_+$-coefficient matrix $M_+$ over $\{0, 1\}$ and an $I_+$-constants vector $\vec{c}_+$ over $\mathbb{Z}_d$ which encode three different types of equations. Here, a variable indexed by $(j, z, \ell) \in J_+$ corresponds to $x_j^z$ if $\ell = 1$ and to its dual version $\overline{x}_j^z$ if $\ell = -1$. Equations indexed by elements $i \in I$ correspond to equations of the original linear equation system where we substitute linear terms $z \cdot x$ by $\sum_{z_1 < z} x^{z_1}$. Moreover, equations indexed by elements $(j, z_1, z_2) \in J \times \mathbb{Z}_d \times \mathbb{Z}_d$ correspond to the constraints $x_j^{z_1} + \overline{x}_j^{z_2} = 0$, and finally, equations indexed by elements $(j, z) \in J \times \mathbb{Z}_d$ correspond to the constraints $x_j^z + \overline{x}_j^z = 0$. Specifically, we set for $(j, z, \ell) \in J_+$,

$$M_+(i, (j, z, \ell)) := \begin{cases} 1, & \ell = 1, z < M(i, j) \\ 0, & \text{else,} \end{cases}$$

$$M_+((j, z_1, z_2), (j, z, \ell)) := \begin{cases} 1, & z = z_1, \ell = 1 \\ 1, & z = z_2, \ell = -1 \\ 0, & \text{else,} \end{cases}$$

$$M_+((j, z), (j, z, \ell)) := \begin{cases} 1, & \ell = 1 \\ 1, & \ell = -1 \\ 0, & \text{else.} \end{cases}$$

Accordingly, the constants vector $\vec{c}_+$ over $\mathbb{Z}_d$ is defined as $\vec{c}_+(i) = \vec{c}(i)$ for $i \in I$ and $\vec{c}_+(j, z_1, z_2) = \vec{c}_+(j, z) = 0$ for all other components $(j, z_1, z_2), (j, z) \in I_+$. In this way we obtain an equivalent linear equation system $M_+ \cdot \vec{x} = \vec{c}_+$ over the *group* $\mathbb{Z}_d$. Clearly, the described transformations can easily be defined using a DTC-interpretation, since an order on the ring $\mathbb{Z}_d$ is DTC-definable. □

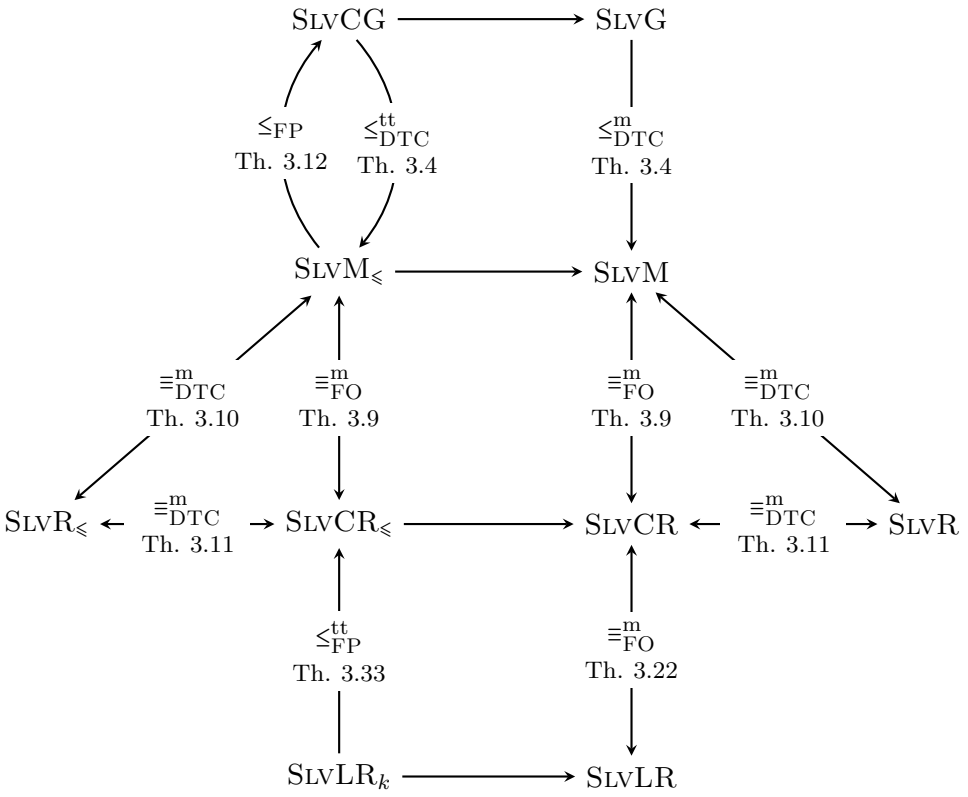


Figure 3.3: Logical reductions between the solvability problems over groups, rings, and modules

## 3.3 Definable structure theory of finite commutative rings

In Section 3.2 we saw that the solvability problems over cyclic groups and *ordered* modules, or rings, are, up to FP-reductions, equivalent. In other words, if we restrict ourselves to the case of solvability problems over *ordered* algebraic domains, then every logic $\mathcal{L}$ which can express SLVCG, and which is closed under FP-reductions, can already express *all* considered solvability problems.

While the step from Abelian groups to modules, or to commutative rings, also works in the absence of a linear order, for the other direction, we crucially made use of an order to fix a generating set for the underlying additive group. However, our reduction can obviously be applied also for such *unordered* commutative rings for which can *define* a linear order on the ring elements. We thus set out to systematically study in this section on which classes of commutative rings we can obtain definable orderings. Let us first formulate this question more precisely.

**Definition 3.15.** Let $\mathcal{K} \subseteq \mathcal{S}(\tau_{\text{ring}})$ be a class of commutative rings. We say that $\mathcal{K}$ *allows* $\mathcal{L}$*-orderings* if there is an $\mathcal{L}$-formula $\varphi(x_1, \dots, x_k, y, z) \in \mathcal{L}(\tau_{\text{ring}})$ such that for every commutative ring $\mathfrak{A} = (R, +, \cdot) \in \mathcal{K}$ the relation $\varphi^{\mathfrak{A}}(r_1, \dots, r_k)$ is a linear order on $R$ for some choice of ring elements $r_1, \dots, r_k \in R$.

Due to symmetries, it is easy to see that the class of all commutative rings does not allow FP-orderings. As a simple example, consider the family of commutative rings $\mathbb{Z}_2^n$ for $n \geq 1$. Then for every possible number $k \geq 1$ of parameters there is an $n \geq k$ such that $(\mathbb{Z}_2^n, r_1, \dots, r_k)$ has non-trivial automorphisms for every choice of $r_1, \dots, r_k \in \mathbb{Z}_2^n$. In other words, the class of commutative rings $\mathcal{K} = \{\mathbb{Z}_2^n : n \geq 1\}$ does not allow $\mathcal{L}$-definable orderings for any reasonable logic $\mathcal{L}$.

What makes this example further interesting is that we can still reduce linear equation systems over rings from $\mathcal{K}$ to equivalent linear equation systems over cyclic groups (actually to systems over $\mathbb{Z}_2$) *without* using any linear order at all. Let us go through this reduction, as we think, that many of the following algebraic ideas, in particular the notion of *local rings* and the first-order definable projection onto local summands (see below), can well be illustrated by considering the rings $\mathbb{Z}_2^n$. Hence, we let $R = \mathbb{Z}_2^n$, and we denote, for $1 \leq i \leq n$, by $e_i = (0, \dots, 0, 1, 0, \dots, 0) \in \mathbb{Z}_2^n$ the ring element which has zeros in every component, except for the $i$-th one. Then the following holds:

- $e_i^2 = e_i$, and $e_i \cdot e_j = 0$ for $i \neq j$, and
- $1 = e_1 + e_2 + \dots + e_n$, and $e_i \cdot R \cong \mathbb{Z}_2$.

Let $M \cdot \vec{x} = \vec{c}$ be a linear equation system over $R$. Then we claim that this system is solvable if, and only if, for all $1 \leq i \leq n$, the linear equation systems $e_i \cdot M \cdot \vec{x} = e_i \cdot \vec{c}$ over $e_i \cdot R \cong \mathbb{Z}_2$ are solvable. To see this, we write $\vec{x}$ as $\vec{x} = \sum e_i \cdot \vec{x} = e_1\vec{x} + \dots + e_n\vec{x}$, and, analogously, we write $M$ as $M = \sum e_i \cdot M$ and $\vec{c}$ as $\vec{c} = \sum e_i \cdot \vec{c}$ (this is possible, since $1 = e_1 + \dots + e_n$). Then, the linear equation system can be written as $(\sum e_i \cdot M) \cdot (\sum e_i \cdot \vec{x}) = \sum e_i\vec{c}$. Moreover, since $e_i \cdot e_j = 0$ for all $i \neq j$, and since, $e_i^2 = e_i$, we can also write this equation as $\sum e_i \cdot M \cdot \vec{x} = \sum e_i\vec{c}$. In particular, by multiplying this equation by $e_i$, we obtain, again by using that $e_i \cdot e_j = 0$ for $i \neq j$, the linear system $e_i \cdot M \cdot \vec{x} = e_i \cdot \vec{c}$ over $\mathbb{Z}_2$ from above. Hence, indeed, the solvability of these linear equation systems over $\mathbb{Z}_2$ is a necessary and sufficient condition for the solvability of the original system $M \cdot \vec{x} = \vec{c}$ over $R$.

What have we learnt from this example? First, we cannot hope to obtain definable orderings on the class of *all* commutative rings. Secondly, for reducing linear equation systems to cyclic groups, this may not be necessary at all, since, in some cases, such linear systems can be reduced to an equivalent family of linear systems over simpler rings which can be ordered very easily. In other words, the hope might be that we can obtain definable orderings on these "basic building" blocks of commutative ring. To pursue this target, we will look at the algebraic structure theory of (finite) commutative rings.

Before, let us also give two examples of classes of commutative rings which do allow FP-orderings. First of all, this obviously holds for the class of all residue rings $\mathbb{Z}_d$, since such rings have a cyclic additive group. The following example might be a bit more surprising, but is based on the same argument.

**Remark 3.16.** *The class of all fields allows* DTC*-orderings.*

*Proof.* It suffices to recall from algebra that the multiplicative group of every (finite) field is cyclic [56]. Then the result follows from Lemma 3.3. □

We now set out to systematically identify classes of commutative rings with FP-orderings. From our previous observations it follows that for each such class of rings $\mathcal{K}$, every logic $\mathcal{L}$ which extends FP and which can express SLvCG can also express the solvability problem over rings from $\mathcal{K}$. In other words, the solvability problem over rings from $\mathcal{K}$ is, up to FP-reductions, at most as difficult as the solvability problem SLvCG over cyclic groups.

Remark 3.16 relies on the fact that fields are simple commutative rings. This raises the question of whether there is some kind of algebraic measure for the complexity of commutative rings which is helpful to identify classes that allow FP-orderings. Fortunately, the algebraic structure of commutative rings is well characterised, and we can apply the available algebraic theory to study our question about the FP-definability of linear orderings. Since many of the following algebraic characterisations only hold for *finite* commutative rings, we remind the reader of our implicit agreement that all structures, including all algebraic structures, such as rings, are finite, if not explicitly stated otherwise.

Let us briefly sketch our next steps. First, we establish a first-order reduction which translates (many) problems from the field of linear algebra over commutative rings (for instance, the solvability problem) to corresponding problems over *local* rings. In particular, this means that, for our applications, it suffices to study the question of FP-orderings on classes of local rings. Secondly, we consider the class of *chain rings*, and prove that they allow FP-orderings. Thirdly, we generalise our techniques further to arrive at our main result in this section: for every fixed $k \geq 1$, the class of local rings for which the maximal ideal is generated by at most $k$ elements (so called *$k$-generated local rings*, see Definition 3.30), allows FP-orderings.

We start to review some basic structure theory of commutative rings. Recall that a commutative ring $R$ is *local* if it contains a unique maximal ideal $\mathfrak{m} \trianglelefteq R$. Let us summarise some equivalent characterisations to obtain a better understanding for this notion.

**Lemma 3.17.** *A commutative ring $R$ is local if, and only if, one of the following equivalent conditions is satisfied.*

*(i)* $R \setminus R^*$ *is an ideal in* $R$.

*(ii) The elements* 0 *and* 1 *are the only non-trivial idempotent elements, that is if* $x^2 = x$ *for* $x \in R$ *then* $x \in \{0, 1\}$.

*Proof.* If $R \setminus R^*$ is an ideal, then it has to be the unique maximal ideal, since each proper ideal $\mathfrak{i}$ is contained in $R \setminus R^*$. For the other direction, assume that there exists a unique maximal ideal $\mathfrak{m} \subset R \setminus R^*$. Then we fix a non-unit $r \in R \setminus (R^* \cup \mathfrak{m})$ and consider a maximal ideal $\mathfrak{i} \supseteq r \cdot R$. Since $\mathfrak{m} \neq \mathfrak{i}$ this contradicts the uniqueness of the ideal $\mathfrak{m}$.

For the second characterisation, assume that the ring $R$ is local but contains a non-trivial idempotent element $x \in R$, i.e. $x \cdot (1-x) = 0$ but $x \notin \{0,1\}$. Then also $(1-x) \notin \{0,1\}$ and $(1-x)^2 = (1-x)$ is another non-trivial idempotent element. Moreover, $x$ and $(1-x)$ are both non-units: if $x$ were a unit, then the equation $x \cdot (1-x) = 0$ would imply $1-x = 0$ which means that $x = 1$, a contradiction. Similarly, if $(1-x)$ were a unit, then we had $x = 0$, again a contradiction. In particular, this means that $x, (1-x) \in \mathfrak{m}$ where $\mathfrak{m} = R \setminus R^*$ denotes the maximal ideal of $R$. But then $x + (1-x) = 1 \in \mathfrak{m}$ which is a contradiction to $\mathfrak{m}$ being a proper ideal. For the other direction, if $R$ only contains trivial idempotents, then we claim that every non-unit in $R$ is nilpotent. To see this, note that, since $R$ is finite, we can find for every element $x \in R$ an integer $\ell \geq 1$ such that $x^\ell \cdot x^\ell = x^\ell$. If $x$ is a non-unit then clearly $x^\ell \neq 1$ which means that $x^\ell = 0$ as claimed. We conclude that $x$ is a non-unit if, and only if, $x$ is nilpotent. Hence, if $x$ and $y$ are non-units, then we can find $\ell \geq 1$ such that $x^\ell = y^\ell = 0$ which also means that $(x+y)^{2\ell} = 0$. Hence, the set $R \setminus R^*$ forms an ideal in $R$. □

Local rings play the central role in the structure theory of (finite) commutative rings. The reason is that one can decompose each commutative ring $R$ into a sum of local rings. Moreover, these local rings can be chosen as ideals generated by certain *minimal* idempotent elements, and this decomposition is, up to a permutation of the summands, unique (cf. [14]). As we show next, this decomposition is also definable in first-order logic.

**Lemma 3.18.** *Let $R$ be a commutative ring and let $E \subseteq R$ be a set of idempotent elements which are pairwise orthogonal and for which $\sum_{e \in E} e = 1$. Then $R = \bigoplus_{e \in E} eR$.*

*Proof.* Since $1 \in \bigoplus_{e \in E} eR$ we have $R \subseteq \bigoplus_{e \in E} eR$. Moreover, let $\sum_{e \in E} er_e = 0$ for some $er_e \in eR$. Since $e^2 = e$ and $ef = 0$ for different $e, f \in E$, it follows that $e\big(\sum_{e \in E} er_e\big) = er_e$, which in turn shows that $er_e = 0$ for all $e \in E$. Hence, all ring elements have a unique representation as elements in $\bigoplus_{e \in E} eR$. Moreover, we have $\big(\sum_{e \in E} r_e e\big) \cdot \big(\sum_{e \in E} s_e e\big) = \sum_{e \in E} (r_e s_e) e$ and $\big(\sum_{e \in E} r_e e\big) + \big(\sum_{e \in E} s_e e\big) = \sum_{e \in E} (r_e + s_e) e$ for all $r_e s_e \in R$ which proves our claim. □

We say that an idempotent element $e \in R$, $e \neq 0$, is *minimal* if it cannot be written as the sum $e = f_0 + f_1$ of two orthogonal idempotent elements $f_0, f_1 \neq 0$. For example, by Lemma 3.17, the element 1 is minimal if, and only if, $R$ is local. As we show next, each idempotent element, which is different from 0, can be expressed as a sum of minimal idempotent elements which

are pairwise orthogonal. Moreover, there is a strong connection between the minimal idempotent elements of the ring $R$ and its local subrings.

**Lemma 3.19.** *Let $e \in R$, $e \neq 0$, be an idempotent element.*

*(a) The ring $eR$ is local if, and only if, $e$ is minimal.*

*(b) The element $e$ can be written as $e = f_1 + \cdots + f_k$ for idempotent elements $f_i$ which are pairwise orthogonal and minimal.*

*Proof.* The identity element of multiplication in $eR$ is $e$. Assume that $eR$ is local and that $e = f_0 + f_1$ for two non-zero idempotent elements $f_0$ and $f_1$ which are orthogonal. Then $ef_i = f_i$ for $i = 0, 1$. Since $eR$ is local, we conclude that $f_i \in \{0, e\}$ for $i = 0, 1$. Hence, $f_i = e$ for both $i = 0, 1$ and thus $e = 0$, a contradiction.

For the other direction, assume that $eR$ is not local. By Lemma 3.17 we can find an idempotent element $er \in eR$ different from 0 and $e$. Then also $(e - er)$ is idempotent and different from 0 which proves our claim as $e = er + (e - er)$.

For the second part, assume that $e$ is not minimal. By definition we can find two non-zero idempotent orthogonal elements $f_0, f_1$ such that $e = f_0 + f_1$. Then $f_i e = f_i$ for $i = 0, 1$ because of the orthogonality of $f_0$ and $f_1$. In particular, $f_i \in eR$ for $i = 0, 1$. We also have $f_i R \subset eR$ since otherwise we had $e = f_i r$ for some $r \in R$ which would imply $f_{1-i} = ef_{1-i} = 0$. If the elements $f_0$ or $f_1$ are not minimal, then we can continue the process with these elements. Note that if $r, s, t \in R$ are idempotent elements such that $rs = 0$ and $(r + s)t = 0$, then the elements $r, s$ and $t$ are pairwise orthogonal. Since the ring $R$ is finite, the condition $f_i R \subset eR$ guarantees that the recursion eventually stops. □

For a commutative ring $R$ we define its *base* as the set $\mathcal{B}(R) \subseteq R$ consisting of the minimal, non-zero, idempotent elements of the ring $R$. By Lemma 3.17 we have $\mathcal{B}(R) = \{1\}$ if, and only if, the ring $R$ is local. From the definition it follows that the base $\mathcal{B}(R)$ is first-order definable. We are prepared to state the central structure theorem for commutative rings.

**Theorem 3.20** (see e.g. [14])**.** *Let $R$ be a (finite) commutative ring. Then $R = \bigoplus_{e \in \mathcal{B}(R)} eR$ is a decomposition of $R$ into local rings.*

*Proof.* Assume that $R$ is not local. For convenience, we set $E := \mathcal{B}(R)$. By Lemma 3.18 and Lemma 3.19 (a) it suffices to show that $\sum_{e \in E} e = 1$ and $e \cdot f = 0$ for all $e, f \in E$ with $e \neq f$. First of all, if $e \cdot f \neq 0$ for $e, f \in E$ with $e \neq f$, then $e = ef + (e - ef)$ were a decomposition of $e$ into orthogonal idempotents. We know that $ef \neq 0$, so by the minimality of $e$ we have $e = ef$. However, we also have the decomposition $f = ef + (f - ef)$ of $f$ into the orthogonal idempotents $ef$ and $f - ef$. Again by minimality of $f$ we conclude that $f = ef$ which implies that $e = f$, a contradiction.

Moreover, Lemma 3.19 (b) shows that for some $E' \subseteq E$ we have $1 = \sum_{e \in E'} e$. Assume that $E' \subset E$. Then we can choose $f \in E \setminus E'$. By orthogonality we have $f = f \cdot 1 = f \cdot (\sum_{e \in E'} e) = 0$, a contradiction. □

Next, we use Theorem 3.20 to reduce a wide range of linear-algebraic problems over commutative rings to local rings. In particular, our reduction can be applied for the solvability problem SLVCR over commutative rings. The crucial step is to observe that not only the ring base $\mathcal{B}(R)$, but also the canonical isomorphism $\pi : R \to \bigoplus_{e \in \mathcal{B}(R)} eR$ can be defined in first-order logic.

**Lemma 3.21.** *There is an* $\mathrm{FO}(\tau_{ring})$*-formula* $\psi(x,y,z)$ *such that for all commutative rings* $R$*, all* $e \in \mathcal{B}(R)$ *and all elements* $r, r_e \in R$*, we have* $R \models \psi(e,r,r_e)$ *if, and only if,* $r_e$ *is the projection of* $r$ *onto the local summand* $eR$*.*

*Proof.* To see this, we observe that $r = \sum_{e \in \mathcal{B}(R)} er$. Hence, we can simply set $\psi(x,y,z) = (z = x \cdot y)$. □

The canonical isomorphism $R \to \bigoplus_{e \in \mathcal{B}(R)} eR$ can be extended component-wise to a mapping $R^{I \times J} \to \bigoplus_{e \in \mathcal{B}(R)} (eR)^{I \times J}$ which decomposes every $I \times J$-matrix $M$ over the ring $R$ into a set $\{eM : e \in \mathcal{B}(R)\}$ of $I \times J$-matrices $eM$ over $eR$ in such a way that $M = \sum_{e \in \mathcal{B}(R)} eM$. Similarly, every $I$-vector $\vec{v}$ over $R$ can be decomposed into a set $\{e\vec{v} : e \in \mathcal{B}(R)\}$ of $I$-vectors over $eR$ with $\vec{v} = \sum_{e \in \mathcal{B}(R)} e\vec{v}$. Moreover, this decomposition can be expressed in first-order logic by Lemma 3.21.

According to the ring decomposition $R = \sum_{e \in \mathcal{B}(R)} eR$ of $R$ into the local summands $eR$, we can identify the addition and multiplication on $R$ with the component-wise operations induced on the summands $eR$. In particular, this shows that arithmetic over $R$ can be reduced to arithmetic over the local summands $eR$. Let us illustrate this by the following two examples.

First, let $M$ be an $I \times J$-matrix over $R$ and let $\vec{c}$ be an $I$-vector over $R$ which together determine the linear equation system $M \cdot \vec{x} = \vec{c}$. We claim that the linear equation system $M \cdot \vec{x} = \vec{c}$ over $R$ is solvable if, and only if, for every $e \in \mathcal{B}(R)$ the linear equation system $(eM) \cdot \vec{x} = e\vec{c}$ over the local ring $eR$ is solvable. Indeed, for $\vec{b} \in R^J$ we have

$$M \cdot \vec{b} = \Big( \sum_{e \in \mathcal{B}(R)} eM \Big) \cdot \Big( \sum_{e \in \mathcal{B}(R)} e\vec{b} \Big) = \sum_{e \in \mathcal{B}(R)} (eM) \cdot (e\vec{b}) = \sum_{e \in \mathcal{B}(R)} e(M \cdot \vec{b}).$$

Hence, we have $M \cdot \vec{b} = \vec{c}$ if, and only if, $(eM) \cdot (e\vec{b}) = e\vec{c}$ for all $e \in \mathcal{B}(R)$, which yields our claim.

Secondly, we discuss the problem of matrix multiplication. To this end, let $M$ be an $I \times J$-matrix over $R$ and let $N$ be a $J \times K$-matrix over $R$. Again, to obtain the $I \times K$-matrix $M \cdot N$ over $R$, it suffices to determine the products of all pairs of matrices $eM$ and $eN$ for $e \in \mathcal{B}(R)$, since

$$M \cdot N = \Big( \sum_{e \in \mathcal{B}(R)} eM \Big) \cdot \Big( \sum_{e \in \mathcal{B}(R)} eN \Big) = \sum_{e \in \mathcal{B}(R)} (eM) \cdot (eN) = \sum_{e \in \mathcal{B}(R)} e(M \cdot N).$$

Hence, for both queries, that is for the solvability of linear equation systems and for matrix multiplication, we obtain first-order truth table reductions which transform instances over commutative rings into an equivalent family of instances over local rings. In particular, we have:

**Theorem 3.22.** $\textsc{SlvCR} \equiv^{\mathrm{tt}}_{\mathrm{FO}} \textsc{SlvLR}$.

By using similar arguments, it follows that many other linear-algebraic problems over commutative rings can be reduced, via first-order truth table reductions, to equivalent problems over local rings. This includes, for example, defining the characteristic polynomial of a square matrix, or the inverse of a non-singular square matrix. As a consequence, it suffices to study the descriptive complexity of these problems over *local* commutative rings. In fact, in [27] we show that the characteristic polynomial of a square matrix over a *Galois ring* (which is a certain kind of local ring) can be defined in FPC, and for the case of ordered local rings, we prove that the inverse of a non-singular square matrix can be defined in FPC. By the above reduction, this can be lifted to all commutative rings whose local summands are Galois rings and ordered local rings, respectively.

Recall that we set out to identify classes of commutative rings which allow FP-orderings (cf. Definition 3.15). Such classes $\mathcal{K}$ of commutative rings are of special interest, since the solvability problem over $\mathcal{K}$ reduces, via FP-reductions, to the solvability problem over cyclic groups. We saw already that the class of all commutative rings does not allow FP-orderings. However, by Theorem 3.22, we can restrict ourselves to classes of *local* rings, so we might hope that the class of all local rings allows FP-orderings. Unfortunately, it turns out that this is not the case.

To see this, we again construct a family of local rings with rich automorphism groups. Specifically, for $n \geq 0$ we consider the infinite commutative ring $T_n = \mathbb{Z}_2[X_1, \ldots, X_n]$ together with the ideal $\mathfrak{i}_n \trianglelefteq T_n$ which is generated by $\{X_i X_j : 1 \leq i, j \leq n\}$. Then the family of finite commutative rings

$$R_n := T_n/\mathfrak{i}_n \tag{E 3.1}$$

has the desired properties.

**Lemma 3.23.** *For all $n \geq 0$, $R_n$ is a local ring and $Aut(R_n) \cong \mathrm{GL}_n(\mathbb{Z}_2)$.*

*Proof.* Let $R = R_n$. The elements of $R$ can be represented as $\alpha + \sum_{i=1}^n \beta_i X_i$ with $\alpha, \beta_i \in \{0,1\}$. To see that $R$ is local, we let $\mathfrak{m} \trianglelefteq R$ be the ideal generated by the elements $\{X_1, \ldots, X_n\} \subseteq R$, i.e. $\mathfrak{m} = \{\beta_1 X_1 + \cdots + \beta_n X_n : \beta_i \in \{0,1\}\}$. We claim that $\mathfrak{m}$ is the unique maximal ideal of the ring $R$. Indeed, every element $s \in R \setminus \mathfrak{m}$ is of the form $s = 1 + \sum_{i=1}^n \beta_i X_i$ for suitable $\beta_i \in \{0,1\}$. But then it is easy to verify that $s^2 = 1$ which means that $s \in R^*$.

Moreover, we can identify the ideal $\mathfrak{m}$ with the group $\mathbb{Z}_2^n$ which means that there is a one-to-one correspondence between $\mathrm{GL}_n(\mathbb{Z}_2)$ and the set of (group) automorphisms of $\mathfrak{m}$. Hence, $\mathrm{Aut}(\mathfrak{m}) \cong \mathrm{GL}_n(\mathbb{Z}_2)$. We extend each of these group automorphisms $\pi \in \mathrm{GL}_n(\mathbb{Z}_2)$ to a ring automorphism of $R$ by setting $\pi(\alpha + \sum_{i=1}^n \beta_i X_i) := \alpha + \pi(\sum_{i=1}^n \beta_i X_i)$. It is straightforward to verify that this gives rise to a ring automorphism of $R$. □

**Theorem 3.24.** *The class of all local rings does not allow* FP*-orderings.*

*Proof.* Assume that $k \geq 1$ is the number of parameters of an FP-formula which defines a linear order on the class of local rings. We consider the local ring $R_{k+2}$. By Lemma 3.23 we know that for every choice of parameters $r_1, \ldots, r_k \in R_{k+2}$ there is a non-trivial automorphism $\pi \in \mathrm{Aut}(R)$ which pointwise fixes the elements $r_i$. This, however, contradicts the fact that linearly ordered (finite) structures are rigid. □

Hence, to identify classes $\mathcal{K}$ of local rings which allow an FP-ordering, we have to bound the complexity of local rings which are contained in $\mathcal{K}$. But what are "complicated" local rings? Ideally, we want that an appropriate complexity measure reflects our observations that the class of all fields and the class of all cyclic rings $\mathbb{Z}_d$ are *simple*. Of course, one can design such a measure in various ways. Here, we choose to define the complexity of a local ring via the size of a minimal set of generators for the maximal ideal of the ring.

To motivate this approach, let us first focus on the class of *chain rings*. Recall that a chain ring is a local ring in which every ideal is principal, that is generated by a single element. Thus, with respect to our complexity measure, chain rings are as simple as possible. Note, however, that the class of chain rings is quite rich as it includes, for example, the class of all fields and the class of all cyclic rings $\mathbb{Z}_d$ (where $d$ is a prime power). As we show next, the class of chain rings indeed allows an FP-ordering. Our argument relies on a particular normal form for the ring elements in chain rings. The crucial ingredient for this normal form is the notion of the *Teichmüller coordinate set*.

**Definition 3.25.** Let $R$ be a local ring with maximal ideal $\mathfrak{m}$ and let $\mathbb{F} = R/\mathfrak{m}$ be its residue field. The *Teichmüller coordinate set* $\Gamma(R)$ is defined as $\Gamma(R) \coloneqq \{r \in R : r^q = r\}$ where $q \coloneqq |\mathbb{F}|$ is the size of the residue field.

It is easy to see that $\Gamma(R) \setminus \{0\}$ forms a multiplicative subgroup of the group of units of $R$. More importantly, $\Gamma(R)$ yields a system of representatives for the residue field $\mathbb{F} = R/\mathfrak{m}$.

**Lemma 3.26.** *Let $R$ be a local ring and let $a, b \in \Gamma(R)$, $a \neq b$. Then $(a-b) \in R^*$.*

*Proof.* Assume that $a - b \notin R^*$. Since $R$ is local, we know that $a - b$ is contained in the maximal ideal $\mathfrak{m} \trianglelefteq R$ which is the set of all non-units of $R$. Let $a - b = x$ for $x \in \mathfrak{m}$. For $q = |\mathbb{F}|$ being the size of the residue field $\mathbb{F} = R/\mathfrak{m}$ we have $a^q = a$ and thus $(b + x)^q = b + x$. Moreover, since $b^q = b$ it holds that

$$b + x = (b + x)^q = b^q + \sum_{i=1}^{q} \binom{q}{i} x^i b^{q-i} = b + x \cdot \Big( \sum_{i=1}^{q} \binom{q}{i} x^{i-1} b^{q-i} \Big).$$

Since $q \in \mathfrak{m}$ and $x \in \mathfrak{m}$ we have that $y \coloneqq \sum_{i=1}^{q} \binom{q}{i} x^{i-1} b^{q-i} \in \mathfrak{m}$. We obtain $x = xy$ for $y \in \mathfrak{m}$. But this means $x(1 - y) = 0$ and since $(1 - y) \in R^*$, as in a local ring the sum of a unit and a non-unit is a unit, this means $x = 0$. □

**Lemma 3.27.** *Let $R$ be a local ring with maximal ideal $\mathfrak{m}$. Then every $r \in R$ can be written as $r = a + s$ for $a \in \Gamma(R)$ and $s \in \mathfrak{m}$.*

*Proof.* Let $u \in R^*$ be such that $u + \mathfrak{m}$ generates the multiplicative group of the residue field $\mathbb{F} = R/\mathfrak{m}$. If we denote by $q = |\mathbb{F}|$ the size of $\mathbb{F}$, then the size of the multiplicative group of $\mathbb{F}$ is $q - 1$. Consequently we have $(u + \mathfrak{m})^{q-1} = (1 + \mathfrak{m})$.

Next, we write $q = p^\ell$ for a prime $p$ and $\ell \geq 1$. Since $R$ is local, the set $1 + \mathfrak{m}$ forms a multiplicative group of order $p^k = |\mathfrak{m}|$ for $k \geq 0$. Let $u^{q-1} = 1 + x$ for $x \in \mathfrak{m}$. Then $(1 + x)^{p^k} = 1$ which means that $v^{q-1} = 1$ where $v := u^{p^k}$. Note that $v \in \Gamma(R)$. Since $p^k$ and $q-1$ are co-prime, we conclude that also $v + \mathfrak{m}$ generates the multiplicative group of $\mathbb{F}$. This finishes our proof, because $\Gamma(R) \setminus \{0\}$ forms a multiplicative group and every $r + \mathfrak{m}$ for $r \in R^*$ can be written as $v^n + \mathfrak{m}$ for appropriate $n \geq 0$. □

The two preceding lemmas allow us to establish the following key property of the Teichmüller coordinate set.

**Lemma 3.28.** *Let $R$ be a local ring with maximal ideal $\mathfrak{m}$ and residue field $\mathbb{F} = R/\mathfrak{m}$. Then the mapping $\Gamma(R) \to \mathbb{F}, a \mapsto a + \mathfrak{m}$ is a bijection.*

*Proof.* The mapping is injective, because otherwise, there exists $a, b \in \Gamma(R)$ with $a \neq b$ and $a + \mathfrak{m} = b + \mathfrak{m}$. Hence, $(a - b) \in \mathfrak{m}$ which contradicts Lemma 3.26. Moreover, the mapping is onto by Lemma 3.27. □

We are prepared to show

**Theorem 3.29.** *The class of all chain rings allows* FP*-orderings.*

*Proof.* Let $R$ be a chain ring with maximal ideal $\mathfrak{m}$. We fix a generator $x \in \mathfrak{m}$ for the ideal $\mathfrak{m}$, that is $xR = \mathfrak{m}$. The generator $x$ will be a parameter of the FP-formula which defines a linear order on the class of chain rings. The Teichmüller coordinate set $\Gamma(R)$, the residue field $\mathbb{F} = R/\mathfrak{m}$ and the bijection $\Gamma(R) \to \mathbb{F}$ are FP-definable. Moreover, by fixing a generator for the multiplicative group of $\mathbb{F}$ we obtain an FP-definable linear order on $\mathbb{F}$ and thus on $\Gamma(R)$.

Let $n \geq 0$ denote the nilpotency index of $x$. We claim that every ring element $r \in R$ can uniquely be expressed as

$$r = \sum_{i=0}^{n-1} a_i \cdot x^i, \text{ for appropriate } a_i \in \Gamma(R). \qquad (\star)$$

To show the existence of such an expression, we define for $\ell = 0, \ldots, n$ the ideal $\mathfrak{i}_\ell = x^\ell R$, i.e. the ideal generated by $x^\ell$. By definition we have

$$R = \mathfrak{i}_0 \trianglerighteq \mathfrak{m} = \mathfrak{i}_1 \trianglerighteq \mathfrak{i}_2 \trianglerighteq \cdots \trianglerighteq \mathfrak{i}_{n-1} \trianglerighteq \mathfrak{i}_n = \{0\}.$$

We show by induction on $\ell$ (starting with $\ell = n$) that every element in $\mathfrak{i}_\ell$ can be expressed in the form

$$r = \sum_{i=\ell}^{n-1} a_i \cdot x^i, \text{ for appropriate } a_i \in \Gamma(R).$$

Let $r = x^\ell \cdot s \in \mathfrak{i}_\ell$. If $s \in \mathfrak{m}$, then $x \mid s$ which means that $r \in \mathfrak{i}_{\ell+1}$ and the claim follows from the induction hypothesis. Otherwise, we have $s \in R^*$. By Lemma 3.27 we can find a unique element $a \in \Gamma(R)$ such that $s = a + t$ for an appropriate non-unit $t \in \mathfrak{m}$. Since $x \mid t$ we know from the induction hypothesis that $x^\ell \cdot t$ can be written as $\sum_{i=\ell+1}^{n-1} b_i \cdot x^i$, hence $r = a \cdot x^\ell + \sum_{i=\ell+1}^{n-1} b_i \cdot x^i$.

It remains to prove the uniqueness of such an expression. Assume, for the sake of contradiction, that $\sum_{i=0}^{n-1} a_i \cdot x^i = \sum_{i=0}^{n-1} b_i \cdot x^i$ for distinct tuples $\bar{a}, \bar{b} \in \Gamma(R)^n$. Let us choose the minimal $j = 0, \ldots, n-1$ such that $a_j \neq b_j$. Then $\sum_{i=j}^{n-1} (a_i - b_i) \cdot x^i = 0$, $a_j - b_j \neq 0$ which means that

$$x^j \cdot \Big( \sum_{i=j}^{n-1} (a_i - b_i) \cdot x^{i-j} \Big) = 0.$$

By Lemma 3.26 we know that $(a_j - b_j) \in R^*$ which in turn implies that $(\sum_{i=j}^{n-1} (a_i - b_i) \cdot x^{i-j}) \in R^*$. This, however, yields $x^j = 0$ which is a contradiction to the choice of $n$.

We can easily turn the inductive argument from above into an FP-definable recursive procedure to translate ring elements $r \in R$ into their normal form $(\star)$. Thus we can associate, in an FP-definable way, to each element $r \in R$ with $r = \sum_{i=0}^{n-1} a_i \cdot x^i$ the tuple of elements $\bar{a} \in \Gamma(R)^n$. Finally, since we have given an FP-definable linear order on $\Gamma(R)$, we obtain an FP-definable linear order on $R$ by using the lexicographical order on $\Gamma(R)^n$. □

Let us discuss the connection between the preceding theorem and our earlier observation that the class of all local rings does not allow FP-orderings. Recall that for the proof of Theorem 3.24 we constructed a family of local rings $R_n$ (cf. (E 3.1)) with sufficiently rich automorphism groups. In contrast, the automorphism groups of chain rings are structurally much simpler. In fact, a necessary condition for a class $\mathcal{K}$ of local rings to allow FP-orderings is that there exists a constant $k \geq 0$, such that for every ring $R \in \mathcal{K}$ we can find at most $k$ parameters $r_1, \ldots, r_k \in R$, such that there is no non-trivial automorphism of $(R, r_1, \ldots, r_k)$. Specifically, for the case of chain rings we have $k = 2$: as our proof of Theorem 3.29 shows, in every chain ring $R$ we can find two elements (namely, a generator for the maximal ideal $\mathfrak{m}$ and a generator for $\Gamma(R)$), such that there is no non-trivial automorphism of $R$ which fixes both elements. Furthermore, we observe that we can *always* order the Teichmüller coordinate set $\Gamma(R)$ by fixing a single generator (this was not a specific property of chain rings). Thus, in general, when we want to bound the structural complexity of $R$, then we rather have to control the complexity of the maximal ideal $\mathfrak{m}$. This now naturally leads to the following definition.

**Definition 3.30.** Let $R$ be a local ring with maximal ideal $\mathfrak{m}$ and let $k \geq 0$. We say that the ring $R$ is *$k$-generated* if there exists elements $r_1, \ldots, r_k \in R$ such that $\mathfrak{m} = r_1 R + \cdots + r_k R$.

**Lemma 3.31.** *Let $R_n$ be the local ring defined in (E 3.1) on page 60. Then $R_n$ is $n$-generated but not $(n-1)$-generated.*

*Proof.* Clearly, the maximal ideal $\mathfrak{m}$ of the ring $R_n$ is generated by $X_1, \dots, X_n$. Assume that $\mathfrak{m}$ were generated by a set of ring elements $r_1, \dots, r_{n-1} \in \mathfrak{m}$. It follows that $r_i R = \{0, r_i\}$ which means that $|r_1 R + \cdots + r_{n-1} R| \leq 2^{n-1}$. This, however, yields a contradiction because $|\mathfrak{m}| = 2^n$. □

We conclude that $(R_n)_{n \geq 0}$ is a family of local rings of *strictly increasing* complexity (with respect to the size of minimal generating sets for the maximal ideal). In particular, if we let $\mathcal{LR}_k$ denote the class of $k$-generated local rings, then we obtain the following stratification of the class of local rings:

$$\mathcal{LR}_0 \subset \mathcal{LR}_1 \subset \cdots \subset \mathcal{LR}_k \subset \cdots.$$

We already saw that $\mathcal{LR}_0$, which is the class of all fields, and $\mathcal{LR}_1$, which is the class of all chain rings, allow FP-orderings. In fact this holds for every fixed level of this hierarchy.

**Theorem 3.32.** *For every $k \geq 0$, the class $\mathcal{LR}_k$ of $k$-generated local rings allows* FP*-orderings.*

*Proof.* We generalise the ideas that we used in the proof of Theorem 3.29. Let $R$ be a $k$-generated local ring with maximal ideal $\mathfrak{m}$. We fix a set $x_1, \dots, x_k \in \mathfrak{m}$ of generators of $\mathfrak{m}$, i.e. $\mathfrak{m} = x_1 R + \cdots + x_k R$. These elements will be used as parameters for the FP-formula which defines the linear order on $R$. Moreover, we again fix a generator for $\Gamma(R) \cong R/\mathfrak{m}$ so that we obtain an FP-definable linear order on $\Gamma(R)$.

The main step is to prove the existence of a certain normal form for the elements of $R$ in terms of the generators $x_1, \dots, x_k$ and $\Gamma(R)$. Let $n_i$ denote the nilpotency index of the element $x_i$. First of all, we claim that every ring element $r \in R$ can be expressed in the form

$$r = \sum_{(i_1, \dots, i_k) \leq (n_1 - 1, \dots, n_k - 1)} a_{i_1 \cdots i_k} x_1^{i_1} \cdots x_k^{i_k}, \quad \text{with } a_{i_1 \cdots i_k} \in \Gamma(R). \qquad (\star)$$

Here, $\leq$ denotes the lexicographical order on $[n_1] \times \cdots \times [n_k]$. To see that every ring element $r \in R$ can be written in this form, let us consider the following recursive procedure:

- If $r \in R^*$, then for a unique $a \in \Gamma(R)$ we have $r \in a + \mathfrak{m}$, so $r = a + (x_1 r_1 + \cdots + x_k r_k)$ for some $r_1, \dots, r_k \in R$ and we continue with $r_1, \dots, r_k$.
- Else $r \in \mathfrak{m}$, and $r = x_1 r_1 + \cdots + x_k r_k$ for some $r_1, \dots, r_k \in R$; we continue the process with $r_1, \dots, r_k$.

Since $x_1^{i_1} \cdots x_k^{i_k} = 0$ if $i_\ell \geq n_\ell$ for some $1 \leq \ell \leq k$, this process is guaranteed to stop. Hence, we can express every element $r \in R$ as a sum of elements of the form $a x_1^{i_1} \cdots x_k^{i_k}$ for $a \in \Gamma(R)$ and $(i_1, \dots, i_k) \in [n_1] \times \cdots \times [n_k]$. Moreover, we observe that for all pairs $a, b \in \Gamma(R)$ it either holds that $a + b \in \Gamma(R)$ or there exist elements $c \in \Gamma(R), r \in \mathfrak{m}, r \neq 0$ such that $a x_1^{i_1} \cdots x_k^{i_k} + b x_1^{i_1} \cdots x_k^{i_k} =$

$cx_1^{i_1}\cdots x_k^{i_k} + rx_1^{i_1}\cdots x_k^{i_k}$. Hence, we can combine $\Gamma(R)$-multiples for the same monomial $x_1^{i_1}\cdots x_k^{i_k}$ and obtain a new $\Gamma(R)$-multiple of $x_1^{i_1}\cdots x_k^{i_k}$ together with a remainder which is a strict multiple of $x_1^{i_1}\cdots x_k^{i_k}$. By repeating these two steps, and by again using that $x_1^{i_1}\cdots x_k^{i_k} = 0$ if $i_\ell \geq n_\ell$ for some $1 \leq \ell \leq k$, we finally obtain an expression of $r \in R$ in the form $(\star)$.

Note, however, that the described procedure neither yields a polynomial-time algorithm nor do we obtain a *unique* expression, as for instance, the choice of elements $r_1, \ldots, r_k \in R$ (in both recursion steps) need not to be unique. Still, knowing only the existence of an expression of this kind, we can proceed as follows. For any sequence of exponents $(\ell_1, \ldots, \ell_k) \leq (n_1 - 1, \ldots, n_k - 1)$ we define the ideal $R[\ell_1, \ldots, \ell_k] \trianglelefteq R$ as the set of all elements having an expression of the form $(\star)$ where $a_{i_1 \cdots i_k} = 0$ for all $(i_1, \ldots, i_k) \leq (\ell_1, \ldots, \ell_k)$. Using the same arguments as above, it is straightforward to verify that $R[\ell_1, \ldots, \ell_k]$ indeed forms an ideal. Note that $\mathfrak{m} = R[0, \ldots, 0]$.

It is clear that we can define the ideal $R[\ell_1, \ldots, \ell_k]$ in FP. Now we can use the following FP-definable procedure to obtain a unique expression of the form $(\star)$ for $r \in R$:

- Choose the minimal $(i_1, \ldots, i_k) \leq (n_1, \ldots, n_k)$ such that $r = ax_1^{i_1}\cdots x_k^{i_k} + s$ for a (minimal) $a \in \Gamma(R)$ and $s \in R[i_1, \ldots, i_k]$. Continue with $s$.

Finally, as in the case of chain rings, the lexicographical ordering induced by the lexicographical ordering on $[n_1] \times \cdots \times [n_k]$ and the ordering on $\Gamma(R)$ yields an FP-definable order on $R$. □

As a direct consequence we obtain the main theorem of this section: for every fixed $k \geq 0$ the solvability problem over $\mathcal{LR}_k$, denoted by $\textsc{SlvLR}_k$, reduces via an FP-reduction to the solvability problem over ordered commutative rings.

**Theorem 3.33.** *For every fixed $k \geq 0$ we have* $\textsc{SlvLR}_k \leq_{\mathrm{FP}}^{\mathrm{tt}} \textsc{SlvCR}_{\leqslant}$.

## 3.4 Discussion

We studied the inter-definability of solvability problems over various classes of (finite) Abelian groups, rings, and modules. Our main result is that whenever the algebraic domains possess a built-in order, then linear equation systems can be reduced to equivalent systems over cyclic groups of prime-power order in fixed-point logic. Moreover, we identified rich classes of commutative rings for which the same reduction can be applied, since we can obtain FP-definable orderings on their local components.

The two immediate open problems are as follows. First of all, it remains unclear whether a reduction to cyclic groups can also be achieved in the absence of a linear order on the algebraic domain. In fact, we strongly made use of the linear order to fix a generating set of the group (which is, in general, a highly non-canonical object with large orbit). One way to circumvent this problem

might be to consider a canonisation of the Abelian group. In fact, it is easy to canonise Abelian groups in fixed-point logic with counting, but of course, this does not help, since we also need some kind of definable correspondence between the canonised group and the coefficients of the given linear equation system. However, if we have a full correspondence between the group and its canonical copy, then we already have a definable linear order on the group and we are back at the beginning. Still, maybe the linear equation system provides enough structure on the group to obtain a "partial" correspondence which suffices to split the system and the group into an equivalent *family* of linear equation systems over cyclic groups (similarly as for the case of commutative rings where we projected the systems onto the local components of the ring).

Secondly, we have not answered our question from the beginning, that is whether rank logic is able to express the solvability problem over *all* Abelian groups. For ordered Abelian groups we saw that this question can be reduced to the case of linear equation systems over cyclic groups of prime-power order. The difficulty is that, although cyclic groups are very simple, they can only be embedded into finite fields if they have prime order. For instance, can rank logic define the solvability of linear equation systems over $\mathbb{Z}_4$? If not, then we had a separation of rank logic (even of the revised version $\mathrm{FPR}^*$ with a uniform rank operator, see Chapter 4) from polynomial time. On the other hand, if $\mathrm{FPR}^*$ can define the solvability problem over *all* Abelian groups, then this would open the door to study the definability of more general problems from computational algebra, such as (Abelian) permutation group membership problems, or the solvability of linear equation systems over the integers. So in each case, an answer would be extremely interesting.

Furthermore, we think that it would be interesting to study the above questions for certain special cases. For example, consider the class of all structures $\mathfrak{A} = (G, +, M)$ such that $(G, +)$ is an Abelian group, such that $M \subseteq G$, and such that the sum over the elements in $M$ is the neutral element of the group $(G, +)$, that is such that $\sum_{m \in M} m = 0$. Of course this class is just a very simple instance of the solvability problem over Abelian groups. However, already for this special case the (non-)definability in fixed-point logic with (counting, rank) is open. We think that understanding this particular case seems to be helpful if we want to make progress for the general case. Recently, we studied this problem in the context a Master's thesis. We obtained no definite answer, but we have some interesting preliminary results. For example, the problem turns out to be definable in solvability logic (where we require solvability operators for linear equation systems over cyclic rings $\mathbb{Z}_d$). However, at the moment we think that this class may already be definable in fixed-point logic with counting.

Finally, we want to study solvability problems over infinite algebraic domains. While, in this thesis, we studied linear equation systems only over *finite* groups, rings, and modules, it would also be interesting to study the descriptive complexity of linear equation systems, for example, over the integers. Surpris-

ingly, it turns out that the solvability problem over the rationals is definable in fixed-point logic with counting [28]. Moreover, there are very tight connections between the solvability of certain families of linear programs and the power of fixed-point logic with counting to distinguish between pairs of non-isomorphic graphs [11, 52]. On the other hand, it immediately follows from the results of Atserias, Bulatov, and Dawar [10] that the solvability of linear equation systems over the integers cannot be defined in FPC, see also [80]. It is also open whether the solvability problem over the integers can be defined in rank logic (this might also depend on the representation of the coefficients). In particular, the solvability problem over the integers uniformly generalises the solvability problems over all finite rings $\mathbb{Z}_n$.

Another approach would be to study the solvability problem over non-Abelian groups. We already mentioned that the solvability problem over *every* non-Abelian group is known to be NP-complete (see [39]), but it would be very interesting to prove a separation from, say rank logic, without using any complexity theoretic assumptions.

# Chapter 4

# Linear-algebraic operators over finite fields

In this chapter, we study *solvability logic* and *rank logic*. The main idea of both logics is to extend fixed-point logic with counting by new mechanisms to express the solvability of linear equation systems over finite fields. Recall that Atserias, Bulatov, and Dawar [10] showed that the solvability of linear equation systems over (finite) Abelian groups cannot be defined in fixed-point logic with counting. Moreover, most of the known examples which separate FPC from Ptime, like (variants of) the CFI-construction or the construction of multipedes, reduce to solving linear equation systems over finite fields. Hence, it is natural to study extensions of FPC by mechanisms which can express such solvability problems, see [28, 59, 71, 80]. Solvability logic and rank logic implement two different (though similar) ways of doing this.

Rank logic FPR was introduced in [28]. Very roughly, the idea is to identify a definable binary relation $\varphi^{\mathfrak{A}} \subseteq A \times A$ with the adjacency matrix

$$M_\varphi^{\mathfrak{A}} : A \times A \to \{0,1\}, (a,b) \mapsto \begin{cases} 1, & \text{if } (a,b) \in \varphi^{\mathfrak{A}} \\ 0, & \text{if } (a,b) \notin \varphi^{\mathfrak{A}}, \end{cases}$$

and to extend FPC by a new *rank operator* $\mathsf{rk}_p$, for every prime $p \in \mathbb{P}$, which can be used to form a *rank term* $[\mathsf{rk}_p\ \varphi]$ whose value is the matrix rank of $M_\varphi^{\mathfrak{A}}$ over the prime field $\mathbb{F}_p$ (for the precise definition we refer to Section 4.1). Rank operators have quite surprising expressive power. For example, they can define the transitive closure of symmetric relations, they can count the number of paths in DAGs modulo $p$, and they can express the solvability of linear equation systems over finite fields (recall that a linear equation system $M \cdot \vec{x} = \vec{c}$ is solvable if, and only if, $\mathsf{rk}(M) = \mathsf{rk}(M \,|\, \vec{c})$) [28]. Furthermore, rank operators can be used to define the isomorphism problem on various classes of structures on which the Weisfeiler-Lehman method (and thus fixed-point logic with counting) fails, for example classes of Cai, Fürer, Immerman graphs [21, 28] and multipedes [55, 59], see also [1].

Solvability logic FPS was proposed in [27, 80] as a different, though conceptually similar, extension of FPC by solvability quantifiers $\mathsf{slv}_p$ which express the solvability of linear equation systems over prime fields $\mathbb{F}_p$ (see Section 4.1 for the precise definition). It is clear that FPS can be embedded into FPR (as rank operators can solve linear equation systems), but it is open whether solvability logic is a strict fragment of rank logic.

We mentioned before that most of the known examples which separate FPC from PTIME can be reduced to linear equation systems over finite fields. This immediately shows that both logics, FPS and FPR, are more powerful than FPC. On the other hand, almost nothing was known about the limitations of their expressive power. For instance, it was open whether rank logic suffices to capture polynomial time, whether rank operators (or solvability quantifiers) can simulate fixed-point inductions [28], and also whether FPR (or even FPS) can define the solvability of linear equation systems over *all* Abelian groups [27] (which was the driving question in Chapter 3).

Another intriguing question was whether rank operators over *different* prime fields can simulate each other. In other words: is it possible to reduce (in FPC, say) the solvability problem (or even the matrix rank problem) over the prime field $\mathbb{F}_p$ to the prime field $\mathbb{F}_q$ (where $p, q$ are distinct)? In order to answer this question, Dawar and Holm [29, 59] developed a powerful toolkit of so-called *partition games* and they proved that one variant (so-called *matrix-equivalence games*) precisely characterises the expressive power of infinitary logic with matrix rank quantifiers. By using these games, Holm [59] gave a partial answer to the above question: if we restrict to operators of arity one, then rank operators over different prime fields have incomparable expressive power. In this chapter, we use a different approach, which is based on symmetry arguments, to extend this result to general rank operators (Theorem 4.19).

An important consequence of our separation result for rank operators is that rank logic, as proposed in [28], fails to capture polynomial time (Theorem 4.20). In the original definition of rank logic (and solvability logic) one considers a *distinct* operator $\mathsf{rk}_p$ (or $\mathsf{slv}_p$) for every prime $p \in \mathbb{P}$. This is problematic, since each formula can consequently access operators for a constant number of different primes only. We exploit this deficiency to prove that the *uniform* version of the matrix rank (and of the solvability) problem over $\mathbb{F}_p$, where $p \in \mathbb{P}$ is part of the input, cannot be expressed in FPR (nor in FPS). More specifically, we construct, for every prime $q \in \mathbb{P}$, a class of structures $\mathcal{K}_q$ on which FPC fails to capture polynomial time, and on which rank operators (and thus solvability quantifiers) over *every* prime field $\mathbb{F}_p$, $p \neq q$, can already be simulated in FPC. On the other hand, rank operators (or solvability quantifiers) over $\mathbb{F}_q$ suffice to canonise structures in $\mathcal{K}_q$.

In particular, this shows that the revised versions of solvability logic $\mathrm{FPS}^*$ and of rank logic $\mathrm{FPR}^*$ with *uniform* solvability quantifiers and rank operators, which take $p \in \mathbb{P}$ as part of their input, are strictly more powerful than the original versions of solvability logic and rank logic.

Our second main result concerns the relationship of FPS and FPR. Recall that it is open whether solvability logic is a strict fragment of rank logic, that is whether solvability quantifiers can be used to compute, within FPC, the rank of definable matrices. We give a partial answer to this question and show that in the absence of counting, rank operators are more powerful than solvability quantifiers (Theorem 4.17). To obtain this result, we use that rank operators (which are *numeric* operators) can simulate counting terms, but that this does not hold for solvability quantifiers (which are *Boolean-valued* operators).

Moreover, we study the extensions $\mathrm{FOS}_p$ and $\mathrm{FOR}_p$ of first-order logic by solvability quantifiers and rank operators over $\mathbb{F}_p$ and we obtain a strong normal form for $\mathrm{FOS}_p$: a single application of a solvability quantifier suffices to obtain the full expressive power of $\mathrm{FOS}_p$ (Theorem 4.9). In particular this shows that a *single* solvability quantifier can simulate *arbitrary blocks* of first-order quantifiers (which comes at the price of increasing the arity of the solvability quantifier).

In this chapter, a common idea in many of our proofs is to exploit symmetries of definable linear equation systems. To illustrate this, let $M \cdot \vec{x} = \vec{c}$ be a linear equation system over $\mathbb{F}_p$ where $M$ is a coefficient matrix and $\vec{c}$ is a vector of constants. Moreover, let $\Gamma$ be a group which acts on the index sets of $M$ and $\vec{c}$ and which stabilises $M$ and $\vec{c}$, that is for all $\Pi \in \Gamma$ (written as a permutation matrix) we have $\Pi \cdot M \cdot \Pi^{-1} = M$ and $\Pi \cdot \vec{c} = \vec{c}$. For our applications, $\Gamma$ will always be the automorphism group of a structure $\mathfrak{A}$, and $M$ and $\vec{c}$ are relations which are defined in $\mathfrak{A}$ by formulas of some logic $\mathcal{L}$. Now assume that the system $M \cdot \vec{x} = \vec{c}$ is solvable, and let $\vec{b}$ denote a solution. Then, for all $\Pi \in \Gamma$, the vector $\Pi \cdot \vec{b}$ is also a solution of the system since

$$M \cdot (\Pi \cdot \vec{b}) = (M \cdot \Pi) \cdot \vec{b} = \Pi \cdot (M \cdot \vec{b}) = \Pi \cdot \vec{c} = \vec{c}.$$

Hence, the solution space of the linear equation system $M \cdot \vec{x} = \vec{c}$ is closed under the action of $\Gamma$. Such observations will allow us to transform a given linear equation system into a considerably simpler linear system which still is equivalent to the original one.

The whole chapter is strongly based on [27, 43, 44]. In Section 4.1, we define the extensions of first-order logic (with counting) and of fixed-point logic (with counting) by solvability quantifiers and rank operators. We also demonstrate that it suffices to consider such operators over prime fields. In Section 4.2, we study the extension $\mathrm{FOS}_p$ of first-order logic by solvability quantifiers over $\mathbb{F}_p$. Our main result is a strong normal form: every formula of $\mathrm{FOS}_p$ is equivalent to a formula with a single solvability quantifier. We then set out to study, in Section 4.3, the relationship between solvability quantifiers and rank operators. We prove that in the absence of counting, rank operators are more expressive than solvability quantifiers. Finally, we prove in Section 4.4 that solvability quantifiers and rank operators over different prime fields are incomparable. An important consequence is that rank logic, as defined in [28], does not capture polynomial time.

## 4.1 Solvability quantifiers and rank operators

In this section, we extend first-order logic (with and without counting) and fixed-point logic with counting by rank operators and solvability quantifiers. To this end, recall from Section 2.3 our setting of *two-sorted* structures $\mathfrak{A}^{\#}$, which we obtained by adding to a usual (one-sorted) structure $\mathfrak{A}$ a disjoint copy of $(\mathbb{N}, +, \cdot)$. This *second, numerical sort* is used as a domain for *counting terms* (within FOC and FPC) and we reuse this setting here to define rank operators (as the matrix rank is a *numerical* invariant as well). In this context, also recall our convention that for all two-sorted logics which are evaluated over structures $\mathfrak{A}^{\#}$ (such as FOC, FPC, and also the versions of rank logic which we define below), quantification over the number sort has to be bounded by numeric terms in order to guarantee that the range of quantifiers is polynomially bounded. This technical condition is necessary in order to obtain logics that have polynomial-time data complexity.

**Rank operators** Let $\Theta(\bar{x}\bar{\nu} \leq \bar{t}, \bar{y}\bar{\mu} \leq \bar{s})$ be a numeric term where $\bar{t}$ and $\bar{s}$ are tuples of closed numeric terms which bound the range of the numeric variables in the tuples $\bar{\nu}$ and $\bar{\mu}$, respectively. Given a structure $\mathfrak{A}$, we define $\mathbb{N}^{\leq \bar{t}} := \{\bar{n} \in \mathbb{N}^{|\bar{\nu}|} : n_i \leq t_i^{\mathfrak{A}}\}$. The set $\mathbb{N}^{\leq \bar{s}} \subset \mathbb{N}^{|\bar{\mu}|}$ is defined analogously. The term $\Theta$ defines in the structure $\mathfrak{A}$ for $I := A^{|\bar{x}|} \times \mathbb{N}^{\leq \bar{t}}$ and $J := A^{|\bar{y}|} \times \mathbb{N}^{\leq \bar{s}}$ the $I \times J$-matrix $M_\Theta$ with values in $\mathbb{N}$ that is given as $M_\Theta(\bar{a}\bar{n}, \bar{b}\bar{m}) := \Theta^{\mathfrak{A}}(\bar{a}\bar{n}, \bar{b}\bar{m})$.

The *matrix rank operators* compute the rank of the matrix $M_\Theta$ over a prime field $\mathbb{F}_p$ for $p \in \mathbb{P}$. First, as in [28], we define, for every prime $p$, a matrix rank operator $\mathsf{rk}_p$ which allows us to construct a new numeric *rank term* $[\mathsf{rk}_p\,(\bar{x}\bar{\nu} \leq \bar{t}, \bar{y}\bar{\mu} \leq \bar{s})\,.\,\Theta]$ whose value in the structure $\mathfrak{A}$ is the rank of the matrix $(M_\Theta \mod p)$ over $\mathbb{F}_p$. Secondly, we consider a *uniform* rank operator $\mathsf{rk}$ which takes the prime $p$ as an additional input. Formally, with this rank operator $\mathsf{rk}$ we can construct a rank term $[\mathsf{rk}\,(\bar{x}\bar{\nu} \leq \bar{t}, \bar{y}\bar{\mu} \leq \bar{s}, \pi \leq r)\,.\,\Theta]$ where $\pi$ is an additional free numeric variable whose range is bounded by some closed numeric term $r$. Given a structure $\mathfrak{A}$ and an assignment $\pi \mapsto p$ for some prime $p \leq r^{\mathfrak{A}}$, the value of this rank term is the matrix rank of $(M_\Theta \mod p)$ considered as a matrix over $\mathbb{F}_p$. The rank operator $\mathsf{rk}$ is a unification for the family of rank operators $(\mathsf{rk}_p)_{p \in \mathbb{P}}$ and was first introduced in [59, 71, 80].

We define, for every set of primes $\Omega \subseteq \mathbb{P}$, the extension $\mathrm{FOR}_\Omega$ of FOC and the extension $\mathrm{FPR}_\Omega$ of FPC by matrix rank operators $\mathsf{rk}_p$ with $p \in \Omega$. For convenience, we set $\mathrm{FOR} = \mathrm{FOR}_{\mathbb{P}}$ and $\mathrm{FPR} = \mathrm{FPR}_{\mathbb{P}}$. Similarly, we denote by $\mathrm{FPR}^*$ the extension of FPC by the uniform rank operator $\mathsf{rk}$. We remark that rank operators can directly simulate counting terms. For example,

$$[\#x\,.\,\varphi(x)] = [\mathsf{rk}_p\,(x, y)\,.\,(x = y \wedge \varphi(x))].$$

Hence, we could equivalently define the rank logics $\mathrm{FOR}_\Omega, \mathrm{FPR}_\Omega$ and $\mathrm{FPR}^*$ as the extensions of (the two-sorted variants of) FO and FP, respectively.

**Solvability quantifier** The expressive power of rank logic significantly goes beyond that of fixed-point logic with counting. However, most of the known examples which separate FPR and FPC are obtained by reducing the respective queries to solvability problems of linear equation systems over finite fields. This is why we find it natural to study analogous extensions of FO, FP, FOC and FPC by quantifiers which can directly express the solvability of linear equation systems. The most important advantage of this approach is that solvability quantifiers are much easier to analyse. This is basically because linear equation systems are compatible with linear-algebraic transformations while matrix rank is not. For example, in general there is no connection between the matrix rank of a family of matrices and the matrix rank of their sum. On the other hand, the solution space of a linear equation system certainly has a nice linear-algebraic structure. This is also the reason why in Section 4.4, where we show that rank operators over different prime fields have incomparable expressive power, we first reduce rank operators to solvability quantifiers and then apply our arguments in the framework of solvability logics. A second important advantage of solvability quantifiers is that they can easily be generalised to other classes of algebraic domains, such as rings, for example, for which no appropriate notion of matrix rank exists, see [27] and Chapter 3 and our discussion in Section 4.5.

Let $\Omega \subseteq \mathbb{P}$ be a set of primes. We want to introduce the extension $\mathrm{FOS}_\Omega$ of first-order logic FO and the extension $\mathrm{FPS}_\Omega$ of FPC by *solvability quantifiers* $\mathsf{slv}_p$ for $p \in \Omega$. Note that, besides of the presence of the fixed-point operators, the tremendous difference between the logics $\mathrm{FOS}_\Omega$ and $\mathrm{FPS}_\Omega$ is that $\mathrm{FOS}_\Omega$ is a *one-sorted* logic while $\mathrm{FPS}_\Omega$ has counting terms and thus a second numerical counting sort. Moreover, this also distinguishes the logic $\mathrm{FOS}_\Omega$ (which, again, is a *one-sorted* logic) from the extension $\mathrm{FOR}_\Omega$ of first-order by matrix rank operators (which can access a *second counting sort*). Indeed, we exploit this significant mismatch between both logics in Section 4.3 to show that in the absence of counting, rank operators are strictly more expressive than solvability quantifiers. Let us start by introducing the logic $\mathrm{FOS}_\Omega$.

**Definition 4.1.** Let $\Omega \subseteq \mathbb{P}$. The logic $\mathrm{FOS}_\Omega$ extends the syntax of first-order logic FO by the following rule. If $\varphi(\bar{x}, \bar{y}, \bar{z}) \in \mathrm{FOS}_\Omega$, then also $\psi(\bar{z}) = (\mathsf{slv}_p\, \bar{x}, \bar{y})\varphi(\bar{x}, \bar{y}, \bar{z})$ is an $\mathrm{FOS}_\Omega$-formula for $p \in \Omega$.

To specify the semantics of the formula $\psi(\bar{z})$, we let $k = |\bar{x}|$ and $\ell = |\bar{y}|$. A pair $(\mathfrak{A}, \bar{z} \mapsto \bar{c})$ with $\bar{c} \in A^{|z|}$ defines an $I \times J$-matrix $M_\varphi$ over $\{0,1\} \subseteq \mathbb{F}_p$ where $I = A^k$ and $J = A^\ell$ and where $M_\varphi(\bar{a}, \bar{b}) = 1$ if, and only if, $\mathfrak{A} \models \varphi(\bar{a}, \bar{b}, \bar{c})$. Moreover, let $\mathbb{1}$ be the $I$-identity vector over $\mathbb{F}_p$, i.e. $\mathbb{1}(\bar{a}) = 1$ for all $\bar{a} \in I$. Then $M_\varphi$ and $\mathbb{1}$ determine the linear equation system $M_\varphi \cdot \vec{x} = \mathbb{1}$ over $\mathbb{F}_p$. Now we let $\mathfrak{A} \models \psi(\bar{c})$ if, and only if, $M_\varphi \cdot \vec{x} = \mathbb{1}$ is solvable.

We continue to introduce the *solvability logic* $\mathrm{FPS}_\Omega$ which similarly extends the syntax of fixed-point logic with counting by a new formula creation rule for all solvability quantifiers $\mathsf{slv}_p$, $p \in \Omega$.

**Definition 4.2.** Let $\Omega \subseteq \mathbb{P}$. The logic $\mathrm{FPS}_\Omega$ extends the syntax of fixed-point logic with counting FPC by the following rule. Let $\varphi(\bar{x}\bar{\nu}, \bar{y}\bar{\mu}, \bar{z}) \in \mathrm{FPS}_\Omega$ and let $\bar{t}$ and $\bar{s}$ be tuples of closed numeric terms with $|\bar{t}\,| = |\bar{\nu}|$ and $|\bar{s}\,| = |\bar{\mu}|$. Then $\psi(\bar{z}) = (\mathsf{slv}_p\, \bar{x}\bar{\nu} \leq \bar{s}, \bar{y}\bar{\mu} \leq \bar{t})\varphi(\bar{x}\bar{\nu}, \bar{y}\bar{\mu}, \bar{z})$ is a formula of $\mathrm{FPS}_\Omega$.

The semantics of the formula $\psi(\bar{z})$ is defined analogously as for $\mathrm{FOS}_\Omega$ with the difference that for $\mathrm{FPS}_\Omega$ we also allow to define coefficient matrices whose index sets range over the number sort (similar as for the case of rank logic). Of course, we again have to (polynomially) bound such index sets to obtain matrices of polynomial size.

Formally, let $k = |\bar{x}|$ and $\ell = |\bar{y}|$. To a pair $(\mathfrak{A}, \bar{z} \mapsto \bar{c}) \in \mathcal{S}(\sigma, \bar{z})$ we associate the $I \times J$-matrix $M_\varphi$ over $\{0,1\} \subseteq \mathbb{F}_p$ where $I = A^k \times \mathbb{N}^{\leq \bar{s}}$ and $J = A^\ell \times \mathbb{N}^{\leq \bar{t}}$ and where for $\bar{a} \in I$ and $\bar{b} \in J$ we have $M_\varphi(\bar{a}, \bar{b}) = 1$ if, and only if, $\mathfrak{A} \models \varphi(\bar{a}, \bar{b}, \bar{c})$. Let $\mathbb{1}$ be the $I$-identity vector over $\mathbb{F}_p$, i.e. $\mathbb{1}(\bar{a}) = 1$ for all $\bar{a} \in I$. Then $M_\varphi$ and $\mathbb{1}$ determine the linear equation system $M_\varphi \cdot \vec{x} = \mathbb{1}$ over $\mathbb{F}_p$ where $\vec{x} = (x_j)_{j \in J}$ is a $J$-vector of variables $x_j$ which range over $\mathbb{F}_p$. Finally, $\mathfrak{A} \models \psi(\bar{c})$ if, and only if, $M_\varphi \cdot \vec{x} = \mathbb{1}$ is solvable.

A comment is in place regarding the definition of $\mathrm{FOS}_\Omega$ and $\mathrm{FPS}_\Omega$. In fact, at first glance, the solvability quantifiers $\mathsf{slv}_p$ seem to impose serious restrictions on the syntactic form of definable linear equation systems. More precisely, they require that *every* equation in the system is of the form $\sum_{j \in J} a_j \cdot x_j = 1$, where all coefficients $a_j$ are from the set $\{0,1\} \subseteq \mathbb{F}_p$. However, we will see in the following section that this is no restriction at all, since every definable linear equation system can be transformed into this kind of syntactic normal form via a quantifier-free first-order reduction (see Theorem 4.3 in Section 4.2).

Analogously to the definition of $\mathrm{FPR}^*$ we also consider a *uniform* solvability quantifier $\mathsf{slv}$ which takes the prime $p$ as an additional input and which can simulate all solvability quantifiers $\mathsf{slv}_p$ for $p \in \mathbb{P}$. Let $\mathrm{FPS}^*$ denote the extension of FPC by this uniform version of a solvability quantifier. Then the following inclusions easily follow from the definitions and the fact that rank operators can be used to define the solvability problem for linear equation systems.

$$\begin{array}{ccccccccc}
\mathrm{FOR}_\Omega & \leq & \mathrm{FPR}_\Omega & \leq & \mathrm{FPR} & \leq & \mathrm{FPR}^* & \leq & \textsc{Ptime} \\
\wedge\!| & & \wedge\!| & & \wedge\!| & & \wedge\!| & & \\
\mathrm{FOS}_\Omega & \leq & \mathrm{FPS}_\Omega & \leq & \mathrm{FPS} & \leq & \mathrm{FPS}^* & & \\
 & & \wedge\!| & & & & & & \\
 & & \mathrm{FPC} & & & & & &
\end{array}$$

Finally, we remark that, analogously to [28], we defined rank operators and solvability quantifiers over prime fields only. Of course, the definition can easily be generalised to cover *all* finite fields, i.e. also finite fields of prime-power order. However, at least in the presence of fixed-points, such more general operators do not lead to more powerful logics (in particular, the same is true in the non-uniform setting where we consider separate operators for all primes). Indeed, Holm proved in [59] that solvability quantifiers over a finite field $\mathbb{F}_q$ of

prime-power order $q = p^k$ can be simulated by solvability quantifiers over $\mathbb{F}_p$. Although the same holds for rank operators, we are not aware of any reference which explicitly establishes such a reduction. In what follows, we thus briefly sketch how one could proceed for the case of rank operators.

To explain how one can simulate rank operators over $\mathbb{F}_q$, where $q = p^k$, by rank operators over $\mathbb{F}_p$, let $M$ be an $I \times J$-matrix over $\mathbb{F}_q$. We know from algebra that $\mathbb{F}_q$ is an $\mathbb{F}_p$-vector space and we fix a basis $B = (\zeta_1, \dots, \zeta_k)$ (this can be done logically, since finite fields can be ordered by fixing a generator of the multiplicative group, see also Remark 3.16). Then every element $f \in \mathbb{F}_q$ can be represented as a $k$-vector over $\mathbb{F}_p$ with respect to the basis $B$, that is $f$ can be written as $f = \sum_{i=1}^{k} a_i^f \cdot \zeta_i$ for a unique $k$-tuple of elements $\bar{a}^f \in \mathbb{F}_p^k$. In this way we obtain an $\mathbb{F}_p$-vector space isomorphism

$$\varphi : \mathbb{F}_q \to \mathbb{F}_p^k, f \mapsto \bar{a}^f.$$

Moreover, $\varphi$ can be extended in the natural way to an isomorphism of the $\mathbb{F}_p$-vector spaces $\mathbb{F}_q^I$ and $(\mathbb{F}_p^k)^I$.

Note that the rank of the matrix $M$ is, by definition, the dimension of the $\mathbb{F}_q$-vector space $\langle V \rangle$ generated by $V = \{\vec{v}_j : j \in J\}$ where $\vec{v}_j$ denotes the $j$-th column of $M$. Let us consider the following $\mathbb{F}_p$-vector space $\langle W \rangle$ that is generated by $W = \{\varphi(\zeta_i \cdot \vec{v}_j) : j \in J, i = 1, \dots, k\} \leq (\mathbb{F}_p^k)^I$. We claim that if $d_V$ is the dimension of $\langle V \rangle$ (which is an $\mathbb{F}_q$-vector space) and if $d_W$ is the dimension of $\langle W \rangle$ (which is an $\mathbb{F}_p$-vector space), then it holds that

$$d_W = k \cdot d_V.$$

If this is true, then it is clear how we can reduce the matrix rank problem over $\mathbb{F}_q$ to $\mathbb{F}_p$, since the isomorphism $\varphi$, and hence the set $W$, obviously are definable in fixed-point logic.

To verify this claim, we let $X \subseteq \{\vec{v}_j : j \in J\}$ be a basis of the $\mathbb{F}_q$-vector space $\langle V \rangle$. Then we show that $Y = \{\varphi(\zeta_i \cdot \vec{v}) : \vec{v} \in X, i = 1, \dots, k\} \leq (\mathbb{F}_p^k)^I$ is a basis of the $\mathbb{F}_p$-vector space $\langle W \rangle$. Note that $|Y| = k \cdot |X|$, so this would imply our claim from above. To see that $Y$ is a generating set, we show that we can write each $\varphi(\zeta_i \cdot \vec{v}_j) \in W$ as a linear combination of elements in $Y$. Since $X$ is a basis of $\langle V \rangle$ we can find coefficients $a_{\vec{v}} \in \mathbb{F}_q$ for $\vec{v} \in X$ such that

$$\zeta_i \cdot \vec{v}_j = \sum_{\vec{v} \in X} a_{\vec{v}} \cdot \vec{v}.$$

Moreover, since $B = (\zeta_1, \dots, \zeta_k)$ is a basis of the $\mathbb{F}_p$-vector space $\mathbb{F}_q$ we can write each such $a_{\vec{v}}$ as $a_{\vec{v}} = \sum_{\ell=1}^{k} a_{\vec{v}}^{\ell} \cdot \zeta_\ell$ for $a_{\vec{v}}^{\ell} \in \mathbb{F}_p$. Hence, we have that

$$\zeta_i \cdot \vec{v}_j = \sum_{\vec{v} \in X} \sum_{\ell=1}^{k} a_{\vec{v}}^{\ell} \cdot \zeta_\ell \cdot \vec{v}, \text{ and thus}$$

$$\varphi(\zeta_i \cdot \vec{v}_j) = \sum_{\vec{v} \in X} \sum_{\ell=1}^{k} a_{\vec{v}}^{\ell} \cdot \varphi(\zeta_\ell \cdot \vec{v}).$$

Similarly one can show that the elements in $Y$ are linearly independent.

## 4.2 First-order extensions by solvability quantifiers

In this section, we study the extensions $\mathrm{FOS}_p$ of first-order logic by solvability quantifiers over prime fields $\mathbb{F}_p$. Recall that the quantifiers $\mathsf{slv}_p$ require that linear equation systems are given in a particular syntactic normal form (see Definition 4.1 and the subsequent definition of their semantics). Our first aim in this section is to justify these technical conditions: indeed, *every* definable linear equation systems can be transformed into an equivalent system which has this particular syntactic form via quantifier-free interpretations. With this preparation we can then prove our second main result of this section: every $\mathrm{FOS}_p$-formula can equivalently be written as a formula which uses only a *single* solvability quantifier which is applied to a quantifier-free definition of a linear equation system (Theorem 4.9). First of all, this normal form theorem shows that solvability quantifiers are very powerful and that they can, in particular, simulate arbitrary first-order formulas within a single quantifier. Secondly, this normal form for $\mathrm{FOS}_p$ will be convenient later in this chapter when we separate the logics $\mathrm{FOS}_p$ from the analogous extensions $\mathrm{FOR}_p$ by matrix rank operators, see Section 4.1.

Recall that for an $\mathrm{FOS}_p$-formula $\psi(\bar{z}) = (\mathsf{slv}_p\, \bar{x}, \bar{y})\varphi(\bar{x}, \bar{y}, \bar{z})$ we have $\mathfrak{A} \models \psi(\bar{c})$ if, and only if, the linear equation system $M_\varphi \cdot \vec{x} = \mathbb{1}$ over $\mathbb{F}_p$ is solvable, where the coefficient matrix $M_\varphi$ over $\{0,1\} \subseteq \mathbb{F}_p$ is determined by setting $M_\varphi(\bar{a}, \bar{b}) = 1$ if, and only if, $\mathfrak{A} \models \varphi(\bar{a}, \bar{b}, \bar{c})$. We already mentioned that the syntactic requirements for linear equation systems seem to be quite restrictive. Specifically, the coefficient matrix has to be a matrix over $\{0,1\}$ and the vector of constants is fixed from outside. Moreover, the solvability quantifier does not provide a mechanism to interpret a linear equation system by merging elements via a definable congruence relation. Our first step is to show that this is not a serious restriction at all.

In fact, using the machinery of Lindström quantifiers (cf. Section 2.2), the direct way to extend FO by operators for the solvability problem would be as follows. For every prime $p$, we define $\tau_{\mathrm{les}}(\mathbb{F}_p) = \{M_1, \dots, M_{p-1}, \vec{c}_1, \dots, \vec{c}_{p-1}\}$ for binary relation symbols $M_i$ and unary relation symbols $\vec{c}_i$. We say that a $\tau_{\mathrm{les}}(\mathbb{F}_p)$-structure $\mathfrak{A} = (A, M_1, \dots, M_{p-1}, \vec{c}_1, \dots, \vec{c}_{p-1})$ encodes a linear equation system over $\mathbb{F}_p$ if $M_1, \dots, M_{p-1}$ represent $I \times J$-matrices over $\{0,1\}$ and $\vec{c}_1, \dots, \vec{c}_{p-1}$ represent $I$-vectors over $\{0,1\}$ (for suitable sets $I, J \subseteq A$). In this case the encoded linear equation system is given as $M \cdot \vec{x} = \vec{c}$ where $M = \sum_{f=1}^{p-1} f \cdot M_f$ and $\vec{c} = \sum_{f=1}^{p-1} f \cdot \vec{c}_f$ and where $\vec{x}$ is a $J$-vector of variables ranging over $\mathbb{F}_p$. Following our convention from Section 3.1, we let $\mathrm{SLS}(\tau_{\mathrm{les}}(\mathbb{F}_p)) \subseteq \mathcal{S}(\tau_{\mathrm{les}}(\mathbb{F}_p))$ denote the class of $\tau_{\mathrm{les}}(\mathbb{F}_p)$-structures which encode *solvable* linear equation systems over the prime field $\mathbb{F}_p$. If we let $\mathcal{Q}_p$ denote the Lindström quantifier (see Section 2.2) associated with the class $\mathrm{SLS}(\tau_{\mathrm{les}}(\mathbb{F}_p))$, then the logic $\mathrm{FO}(\mathcal{Q}_p)$ is the natural candidate for an extension of first-order logic by operators for the solvability problem over $\mathbb{F}_p$. However, it turns out that this logic has precisely the same expressive power as $\mathrm{FOS}_p$.

**Theorem 4.3.** *For all primes $p$ we have* $\mathrm{FOS}_p = \mathrm{FO}(\mathcal{Q}_p)$.

*Proof.* It is clear that $\mathrm{FOS}_p \leq \mathrm{FO}(\mathcal{Q}_p)$. For the other direction, we set $\tau := \tau_{\mathrm{les}}(\mathbb{F}_p)$ and let $\varphi_\delta(\bar{x},\bar{z}), \varphi_\approx(\bar{x},\bar{y},\bar{z}), \varphi_{M_i}(\bar{x},\bar{y},\bar{z}), \varphi_{\vec{c}_i}(\bar{x},\bar{y},\bar{z})$ for $1 \leq i \leq p-1$ be $\mathrm{FO}(\mathcal{Q}_p)$-formulas that define an interpretation $\mathcal{I}(\bar{z}) = (\varphi_\delta, \varphi_\approx, \varphi_{M_1}, \ldots, \varphi_{\vec{c}_{p-1}})$ of $\mathcal{S}(\tau)$ in $\mathcal{S}(\sigma)$ with parameters $\bar{z}$. Then our task is to translate the formula $\psi(\bar{z}) = \mathcal{Q}_p\mathcal{I}(\bar{z})$ of $\mathrm{FO}(\mathcal{Q}_q)$ into an equivalent formula of $\mathrm{FOS}_p$.

First of all we show that, without loss of generality, we can assume that $\varphi_\approx = (\bar{x} = \bar{y})$. The intuitive reason is that the duplication of equations and variables does not influence the solvability of a linear equation system. To turn this idea into a formal argument, let $(\mathfrak{A}, \bar{z} \mapsto \bar{a}) \in \mathcal{S}(\sigma, \bar{z})$ and consider the interpreted linear equation system $\mathfrak{B} := \mathcal{I}(\mathfrak{A}, \bar{z} \mapsto \bar{a})$ represented as a $\tau$-structure. Then $B = A^k/\approx$ where $k = |\bar{x}|$ and where $\approx$ denotes the congruence defined by $\varphi_\approx$ in $(\mathfrak{A}, \bar{z} \mapsto \bar{a})$. Let $M \cdot \vec{x} = \vec{c}$ be the represented linear equation system over $\mathbb{F}_p$ where $M$ is an $I \times J$-matrix and where $\vec{c}$ is an $I$-vector over $\mathbb{F}_p$. We have that $I, J \subseteq B$.

Let $\mathcal{I}^*$ be the interpretation which arises from $\mathcal{I}$ by replacing $\varphi_\approx$ by the equality relation $\bar{x} = \bar{y}$. We claim that $\mathfrak{C} := \mathcal{I}^*(\mathfrak{A}, \bar{z} \mapsto \bar{a})$ encodes a linear equation system over $\mathbb{F}_p$ which is solvable if, and only if, the system $M \cdot \vec{x} = \vec{c}$ is solvable. To see this, let $\tilde{I} = \bigcup I \subseteq A^k$ and $\tilde{J} = \bigcup J \subseteq A^k$. By definition and since $\approx$ is a congruence relation it follows that $\mathfrak{C}$ represents a linear equation system $N \cdot \vec{x}_* = \vec{d}$ over $\mathbb{F}_p$ with an $\tilde{I} \times \tilde{J}$-coefficient matrix $N$ and an $\tilde{I}$-constants vector $\vec{d}$ such that for all $i \in \tilde{I}, j \in \tilde{J}$ we have $N(i,j) = M([i]_\approx, [j]_\approx)$ and $\vec{d}(i) = \vec{c}([i]_\approx)$. Now, solutions of the linear equation systems $M \cdot \vec{x} = \vec{c}$ and $N \cdot \vec{x}_* = \vec{d}$ translate as follows. Let $\vec{b}$ be a $J$-vector over $\mathbb{F}_p$ with $M \cdot \vec{b} = \vec{c}$. For each $[j] \in J$ we fix an element $[j]_* \in [j]$. Then we define $\vec{b}_*$ as the $\tilde{J}$-vector given as

$$\vec{b}_*(j) = \begin{cases} \vec{b}([j]), & \text{if } j = [j]_* \\ 0, & \text{else.} \end{cases}$$

It easily follows that $N \cdot \vec{b}_* = \vec{d}$. For the other direction, let $\vec{b}_*$ be a $\tilde{J}$-vector with $N \cdot \vec{b}_* = \vec{d}$. We define a $J$-vector $\vec{b}$ by setting $\vec{b}([j]) = \sum_{j \in [j]} \vec{b}_*(j)$. Again it is straightforward to verify that $M \cdot \vec{b} = \vec{c}$.

Hence, from now on we assume that the congruence formula $\varphi_\approx$ in the interpretation $\mathcal{I}$ is trivial. To complete our proof we construct a quantifier-free first-order interpretation $\mathcal{J}$ (with trivial domain and congruence formulas) which transforms (an encoding of) a linear equation system over $\mathbb{F}_p$ (as a $\tau_{\mathrm{les}}(\mathbb{F}_p)$-structure) into an equivalent linear equation system over $\mathbb{F}_p$ in the syntactic form required for the $\mathsf{slv}_p$-quantifier of the logic $\mathrm{FOS}_p$. More precisely, we describe $\mathcal{J}$ as the composition of two quantifier-free first-order transformations: the first one maps a linear equation system $M \cdot \vec{x} = \vec{c}$ over $\mathbb{F}_p$ to an equivalent system $M_* \cdot \vec{x} = \mathbb{1}$ over $\mathbb{F}_p$ where $\mathbb{1}$ denotes the identity vector (of the appropriate dimension). The second interpretation then transforms the linear system $M_* \cdot \vec{x} = \mathbb{1}$ into an equivalent system $N \cdot \vec{x} = \mathbb{1}$ over $\mathbb{F}_p$, where $N_*$ is a matrix with entries in $\{0,1\}$.

For the first transformation, suppose that $M$ is an $I \times J$ matrix and $\vec{c}$ is an $I$-vector over $\mathbb{F}_p$. We define a new linear equation system which contains, besides the variables $\vec{x} = (x_j)_{j \in J}$, a fresh variable $v_i$ for every index $i \in I$ and a new variable $w_f$ for every field element $f = 0, \ldots, p-1$. Then for every element $f = 0, \ldots, p-1$, we add the equation $(1-f)w_1 + w_f = 1$. It is easy to see that this subsystem of equations has a unique solution given by $w_f = f$ for all $f = 0, \ldots, p-1$. Then we can simply replace every equation $\sum_{j \in J} M(i,j) \cdot x_j = \vec{c}(i)$ by the following two equations $v_i + \sum_{j \in J} M(i,j) \cdot x_j = 1$ and $v_i + w_{\vec{c}(i)} = 1$.

For the second transformation, we first replace each variable $x$ by $(p-1)$-many copies $x_1, \ldots, x_{p-1}$ and add the equations $x_e = x_f$ for $e, f = 1, \ldots, p-1$. We then replace each atomic linear term $f \cdot x$ by the linear term $\sum_{1 \leq e \leq f} x_e$ to obtain an equivalent linear equation system in which only the field element 1 occurs as a coefficient. However, in order to establish our original claim we also have to express the auxiliary equations $x_e = x_f$ in the correct syntactic form. To achieve this, we introduce a new variable $x_f^-$ for each $x_f$, and we add the equation $x_f + x_f^- + w_1 = 1$. Finally, we rewrite $x_f = x_e$ as $x_f + x_e^- + w_1 = 1$. The resulting system is equivalent and has the desired syntactic form.

Since the field $\mathbb{F}_p$ is fixed, one can see that the described transformations can be formalised by quantifier-free first-order reductions. □

Let us summarise some interesting facts about the logic $\mathrm{FOS}_p$ (see also [80]). First of all, it follows from [28] that for every prime $p$, the logic $\mathrm{FOS}_p$ can express the symmetric transitive closure of definable relations. Hence, $\mathrm{FOS}_p$ subsumes the logic STC and can express every LOGSPACE-computable property of ordered structures. Secondly, it also follows from [28] that $\mathrm{FOS}_2$ can distinguish between the odd and even version of a CFI-graph, which means that $\mathrm{FOS}_2$ cannot be a fragment of FPC. More generally, by adapting the CFI-construction for other prime fields one can show that $\mathrm{FOS}_p \not\leq \mathrm{FPC}$ for arbitrary primes $p$ (see e.g. [59]). In fact, a proof for this can also be extracted from our proof of Theorem 4.21. In particular this shows that $\mathrm{STC} < \mathrm{FOS}_p$ for all primes $p$.

On the domain of ordered structures, the expressive power of $\mathrm{FOS}_p$ can be characterised in terms of a natural complexity class: in [20], Buntrock et. al. introduced the *logarithmic space modulo counting classes* $\mathrm{MOD}_k\mathrm{L}$ for integers $k \geq 2$. Analogously to the case of modulo counting classes for polynomial time, the idea is to say that a problem is in $\mathrm{MOD}_k\mathrm{L}$ if there exists a non-deterministic logspace Turing machine which verifies its inputs by producing a number of accepting paths which is not congruent 0 mod $k$. For the formal definition we refer the reader to [20]. It turns out that, at least for primes $p$, the class $\mathrm{MOD}_p\mathrm{L}$ is closed under many natural operations, including all Boolean operations and even logspace Turing reductions [20, 58]. Furthermore, many problems from linear algebra over $\mathbb{F}_p$ are complete for $\mathrm{MOD}_p\mathrm{L}$. In particular this is true for the solvability problem of linear equation systems over $\mathbb{F}_p$ and for computing the matrix rank over $\mathbb{F}_p$ [20].

Building on these insights, Dawar et. al. [28] were able to show that for all primes $p$, the logic $\mathrm{FOR}_p$ captures $\mathrm{MOD}_p\mathrm{L}$ on the class of ordered structures. As we already noted in [80], their proof shows that the same correspondence holds for the logic $\mathrm{FOS}_p$.

**Theorem 4.4** ([28],[80])**.** *On the class of ordered structures we have*

$$\mathrm{FOS}_p = \mathrm{FOR}_p = \mathrm{MOD}_p\mathrm{L}.$$

Despite this characterisation over the class of ordered structures, so far the situation over general structures remained open. It easily follows that $\mathrm{FOS}_p \leq \mathrm{FOR}_p \leq \mathrm{FPR}$, but, it has been open whether one, or even both, of these inclusions are strict. In the following section we are going to settle one of these questions and prove that for every prime $p$ we have

$$\mathrm{FOS}_p < \mathrm{FOR}_p.$$

More specifically, we will see that the inclusion $\mathrm{FOS}_p < \mathrm{FOR}_p$ holds over the class of sets $\mathcal{S}(\varnothing)$. Our plan for the remainder of this section is to obtain a strong normal form for the logic $\mathrm{FOS}_p$: every $\mathrm{FOS}_p$-formula is equivalent to an $\mathrm{FOS}_p$-formula with at most one application of a $\mathsf{slv}_p$-quantifier. In particular, a single $\mathsf{slv}_p$-quantifier can express arbitrary blocks of first-order quantifiers and all Boolean operations (which comes at the price of increasing the dimension of the $\mathsf{slv}_p$-quantifier). Let us stress the fact that we obtain this normal form over the class of all finite structures. In Section 4.3 we are going to use this normal form to separate the logics $\mathrm{FOS}_p$ and $\mathrm{FOR}_p$ over the class of sets.

To obtain the normal form, we inductively translate $\mathrm{FOS}_p$-formula into equivalent formulas of the desired form. Here, the simple cases are the inductive steps for conjunction and universal quantification.

**Lemma 4.5.** *Let $\alpha, \beta \in \mathrm{FO}$ be two quantifier-free formulas and let $\varphi = (\mathsf{slv}_p \bar{x}_1, \bar{x}_2)\alpha(\bar{x}_1, \bar{x}_2, \bar{z}) \in \mathrm{FOS}_p$ and $\psi = (\mathsf{slv}_p \bar{x}_1, \bar{x}_2)\beta(\bar{x}_1, \bar{x}_2, \bar{z}) \in \mathrm{FOS}_p$. Then there exists a formula $\vartheta(\bar{z}) = (\mathsf{slv}_p \bar{y}_1, \bar{y}_2)\gamma(\bar{y}_1, \bar{y}_2, \bar{z}) \in \mathrm{FOS}_p$ for a quantifier-free formula $\gamma \in \mathrm{FO}$ such that $\vartheta \equiv \varphi \wedge \psi$.*

*Proof.* The idea is to combine the two linear equation systems into one system by using two independent copies of the variable sets. The only technical difficulty is that, as we are not allowed to use congruences, we have to introduce many duplicates of the variables (which we preferably would merge via a congruence). However, this does not influence the solvability of the resulting linear equation system and thus the translation is sound. Let $\bar{y}_1 = v_1 v_2 \bar{x}_1$ and $\bar{y}_2 = w_1 w_2 \bar{x}_2$ for new variables $v_1, v_2, w_1, w_2$. We set

$$\gamma = (v_1 = v_2 \wedge w_1 = w_2 \wedge \alpha(\bar{x}_1, \bar{x}_2, \bar{z})) \vee (v_1 \neq v_2 \wedge w_1 \neq w_2 \wedge \beta(\bar{x}_1, \bar{x}_2, \bar{z})).$$

Then it is easy to verify that $\vartheta := (\mathsf{slv}_p \bar{y}_1, \bar{y}_2)\gamma(\bar{y}_1, \bar{y}_2, \bar{z}) \equiv \varphi \wedge \psi$. □

**Lemma 4.6.** *Let* $\alpha(\bar{x}_1, \bar{x}_2, y, \bar{z}) \in \mathrm{FO}$ *be a quantifier-free formula and let* $\varphi(\bar{z}) = \forall y\, (\mathsf{slv}_p\, \bar{x}_1, \bar{x}_2)\, \alpha(\bar{x}_1, \bar{x}_2, y, \bar{z})$. *Then there exists a formula* $\vartheta(\bar{z}) = (\mathsf{slv}_p\, \bar{y}_1, \bar{y}_2)\gamma(\bar{y}_1, \bar{y}_2, \bar{z}) \in \mathrm{FOS}_p$ *for a quantifier-free* $\gamma \in \mathrm{FO}$ *such that* $\vartheta \equiv \varphi$.

*Proof.* We use the same idea as in the proof of Lemma 4.5 and construct a new linear equation system which contains, for every possible value of the parameter $y$, an independent linear subsystem which corresponds to the system represented by $\alpha(\bar{x}_1, \bar{x}_2, y, \bar{z})$. Let $\bar{y}_1 = y\bar{x}_1$ and $\bar{y}_2 = v\bar{x}_2$ for a new variable $v$. Then it suffices to set $\gamma := (y = v) \wedge \alpha(\bar{x}_1, \bar{x}_2, y, \bar{z})$. □

For the case of negation we make use of the following fact from linear algebra. Consider a linear equation system in the form $M \cdot \vec{x} = \vec{c}$ for an $I \times J$-coefficient matrix $M$ over $\mathbb{F}_p$ and an $I$-constants vector $\vec{c}$ over $\mathbb{F}_p$. Then this linear equation system is *not solvable* if, and only if, the linear equation $(M \mid \vec{c})^T \cdot \vec{x} = (0, \ldots, 0, 1)$ *is solvable*, that is if the row space of the augmented coefficient matrix $(M \mid \vec{c})$ contains the vector $(0, \ldots, 0, 1)$. In this way one can translate the non-solvability of a linear equation system into the solvability of another linear equation system. Moreover, the translation between these two linear equation systems is very simple and can easily be realised via a quantifier-free first-order interpretation. Finally, with our construction from the proof of Theorem 4.3 we can bring the resulting linear system back to the normal form which is required for the $\mathsf{slv}_p$-quantifier. Hence, we obtain:

**Lemma 4.7.** *Let* $\alpha(\bar{x}_1, \bar{x}_2, \bar{z}) \in \mathrm{FO}$ *be a quantifier-free formula. Then there exists a formula* $\vartheta(\bar{z}) = (\mathsf{slv}_p\, \bar{y}_1, \bar{y}_2)\gamma(\bar{y}_1, \bar{y}_2, \bar{z}) \in \mathrm{FOS}_p$ *where* $\gamma \in \mathrm{FO}$ *is quantifier-free such that* $\vartheta \equiv \neg\big[(\mathsf{slv}_p\, \bar{x}_1, \bar{x}_2)\alpha(\bar{x}_1, \bar{x}_2, \bar{z})\big]$.

**Corollary 4.8.** *Every* FO*-formula* $\varphi \in \mathrm{FO}$ *is equivalent to a formula* $\vartheta = (\mathsf{slv}_p\, \bar{y}_1, \bar{y}_2)\gamma(\bar{y}_1, \bar{y}_2) \in \mathrm{FOS}_p$ *where* $\gamma$ *is quantifier-free.*

It remains to treat the case of nested solvability quantifiers. To this end, we fix an $\mathrm{FOS}_p(\tau)$-formula

$$\vartheta(\bar{z}) = (\mathsf{slv}_p\, \bar{x}_1, \bar{x}_2)\big[(\mathsf{slv}_p\, \bar{y}_1, \bar{y}_2)\alpha(\bar{x}_1, \bar{x}_2, \bar{y}_1, \bar{y}_2, \bar{z})\big],$$

where $\alpha$ is quantifier-free and we show how the nested application of the solvability quantifiers can be be reduced to a single solvability operator. To illustrate the semantics of $\vartheta$ let us fix $(\mathfrak{A}, \bar{z} \to \bar{a}) \in \mathcal{S}(\tau, \bar{z})$. Let $k_1 = |\bar{x}_1|$, $k_2 = |\bar{x}_2|$, $\ell_1 = |\bar{y}_1|$ and $\ell_2 = |\bar{y}_2|$. Then $\mathfrak{A} \vDash \vartheta(\bar{a})$ if the *outer* linear equation system $M \cdot \vec{v} = \mathbb{1}$ is solvable where $M$ is the $A^{k_1} \times A^{k_2}$-coefficient matrix over $\{0, 1\}$ whose entries $M(\bar{b}, \bar{c})$ are determined by the solvability of the *inner* linear equation system $N_{\bar{b},\bar{c}} \cdot \vec{w} = \mathbb{1}$. In this context, $N_{\bar{b},\bar{c}}$ denotes the $A^{\ell_1} \times A^{\ell_2}$-coefficient matrix whose entries $N_{\bar{b},\bar{c}}(\bar{d}, \bar{e}) \in \{0, 1\}$ are determined by the truth value of $\mathfrak{A} \vDash \alpha(\bar{b}, \bar{c}, \bar{d}, \bar{e}, \bar{a})$. For convenience, let us write $I = A^{k_1}$ and $J = A^{k_2}$ to denote the index sets of the equations and of the variables of the outer linear equation system, respectively. The situation is illustrated in Figure 4.1.

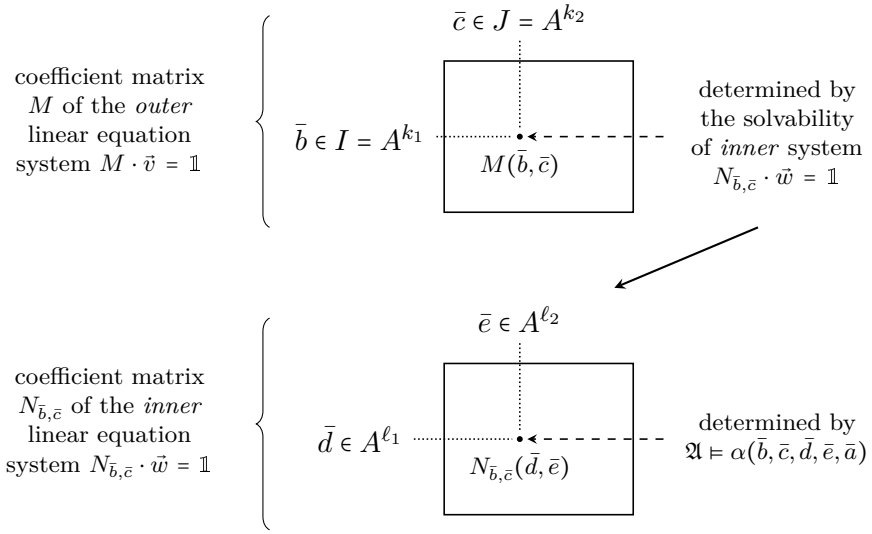


Figure 4.1: Illustration of the nesting of solvability quantifiers in $\vartheta(\bar{z})$

We proceed to explain how we can express the solvability of the "nested" linear equation system $M \cdot \vec{v} = \mathbb{1}$ by the solvability of a "flat" linear equation system $M^* \cdot \vec{v}^* = \mathbb{1}$. The $i$-th equation of $M \cdot \vec{v} = \mathbb{1}$ is given as

$$\sum_{j \in J} M(i,j) \cdot v_j = 1.$$

The interesting part is how to handle the coefficients $M(i,j)$ which are determined by the solvability of the inner linear equation system $N_{i,j} \cdot \vec{w} = \mathbb{1}$. First of all, to construct the new linear equation system $M^* \cdot \vec{v}^* = \mathbb{1}$ we start by introducing, for every pair of indices $(i,j) \in I \times J$, a new variable $v_{i,j}$ and simply replace the $i$-th equation $\sum_{j \in J} M(i,j) \cdot v_j = 1$ by the equation $\sum_{j \in J} v_{i,j} = 1$. The intuition behind that is that the variable $v_{i,j}$ should represent the value of the term $M(i,j) \cdot v_j$. However, the constraint "$v_{i,j} = M(i,j) \cdot v_j$" is a *non-linear* constraint and the crucial step is to extend the new system $M^* \cdot \vec{v}^* = \mathbb{1}$ by a set of *linear* equations which express this condition.

To overcome this problem we construct new linear subsystems which ensure that for all $i, i_* \in I$ and $j \in J$ it holds that:

$$\text{if } v_{i,j} \neq 0, \text{ then } M(i,j) = 1; \text{ and} \tag{E 4.1}$$

$$\text{if } v_{i,j} \neq v_{i_*,j}, \text{ then } \{M(i,j), M(i_*,j)\} = \{0,1\}. \tag{E 4.2}$$

The conditions (E 4.1) and (E 4.2) together ensure that the equations $v_{i,j} = M(i,j) \cdot v_j$ can be used to translate solutions between the nested linear equation system $M \cdot \vec{v} = \mathbb{1}$ and the new linear equation system $M^* \cdot \vec{v}^* = \mathbb{1}$. The question remains how we can express these conditions by subsystems of linear equations.

To express the constraints (E 4.1) we proceed as follows. For each $(i,j) \in I \times J$ we take the inner linear equation system $N_{i,j} \cdot \vec{w} = \mathbb{1}$ and add it as an independent linear subsystem in $M^* \cdot \vec{v}^* = \mathbb{1}$ in which we additionally add to the left-hand side of each equation the term $(v_{i,j} + 1)$. Now, if in a solution of $M^* \cdot \vec{v}^* = \mathbb{1}$ the variable $v_{i,j}$ is evaluated to 0, then the subsystem corresponding to $N_{i,j} \cdot \vec{w} = \mathbb{1}$ has the trivial solution (since the constant term of every equation is 1). However, if a non-zero value is assigned to $v_{i,j}$, then this value is a unit in $\mathbb{F}_p$ and a solution for $M^* \cdot \vec{v}^* = \mathbb{1}$ necessarily contains a solution of the inner linear equation system $N_{i,j} \cdot \vec{w} = \mathbb{1}$; that is, we have $M(i,j) = 1$.

We similarly express the constraints (E 4.2). For indices $i, i_* \in I$ and $j \in J$ the condition on the right-hand side of (E 4.2) is a Boolean combination of solvability (and non-solvability) queries for the inner linear equation systems $N_{i,j} \cdot \vec{w} = \mathbb{1}$ and $N_{i_*,j} \cdot \vec{w} = \mathbb{1}$. By Corollary 4.8 this Boolean combination can be expressed by the solvability of a single linear equation system (which is definable by a quantifier-free formula). Hence, we can embed this linear equation system as an independent subsystem in $M^* \cdot \vec{v}^* = \mathbb{1}$ where we add to each of its equations the term $(1 + v_{i,j} - v_{i_*,j})$. With the same reasoning as above we conclude that this imposes the constraint (E 4.2).

Finally, it is easy to check that the translation from $M \cdot \vec{v} = \mathbb{1}$ to $M^* \cdot \vec{v}^* = \mathbb{1}$ can be realised by a quantifier-free first-order interpretation. This concludes the inductive step for the nesting of solvability quantifiers and we obtain our first main result of this chapter.

**Theorem 4.9.** *Every formula $\vartheta(\bar{z}) \in \mathrm{FOS}_p$ is equivalent to an $\mathrm{FOS}_p$-formula of the form $(\mathsf{slv}_p \bar{x}_1, \bar{x}_2)\alpha(\bar{x}_1, \bar{x}_2, \bar{z})$ where $\alpha(\bar{x}_1, \bar{x}_2, \bar{z})$ is quantifier-free.*

## 4.3 Solvability quantifiers vs. matrix rank operators

In the last section we studied the extensions $\mathrm{FOS}_p$ of first-order logic by solvability quantifiers $\mathsf{slv}_p$ over $\mathbb{F}_p$ and we discussed the strong connections between $\mathrm{FOS}_p$ and the extension $\mathrm{FOR}_p$ of first-order logic by rank operators $\mathsf{rk}_p$ over $\mathbb{F}_p$. In particular, we mentioned that many of the known queries which can be expressed with the help of rank operators can already be expressed using solvability quantifiers. More strikingly, on the class of ordered structures both logics are known to have the same expressive power (see Theorem 4.4). This raises the question whether, in general, it holds that $\mathrm{FOS}_p = \mathrm{FOR}_p$, that is whether rank operators can be simulated by solvability quantifiers via first-order reductions. In this section, we show that this is not the case. Specifically, we prove that over the class of sets there exists a query $\mathcal{K} \subseteq \mathcal{S}(\varnothing)$ which can be expressed in $\mathrm{FOR}_p$ but not in $\mathrm{FOS}_p$, that is

$$\mathrm{FOS}_p < \mathrm{FOR}_p.$$

In fact, this result is not very surprising. In contrast to $\mathrm{FOR}_p$, the logic $\mathrm{FOS}_p$ does not have access to a counting sort and thus has to express properties over $\mathfrak{S}(\varnothing)$ in pure unordered sets (which have a maximal amount of symmetries). However, it is not obvious how one can turn this intuition into a formal argument. In fact, the logic $\mathrm{FOS}_p$ has non-trivial expressive power over sets. For instance, $\mathrm{FOS}_p$ can count the size of sets modulo $p$ [80], and consequently, modulo $p^k$ for every fixed $k$ (observe that $n \equiv 0 \mod p^k$ if, and only if, $n \equiv 0 \mod p$ and $\binom{n}{p} \equiv 0 \mod p^{k-1}$). In contrast, fixed-point logic FP, for example, collapses to first-order logic over sets.

The main idea of our proof is to exploit symmetries of definable linear equation systems to considerably reduce the size of an input linear equation system via an $\mathrm{FOR}_p$-definable transformation. For the remainder of this proof, let us fix a quantifier-free formula $\alpha(x_1, \dots, x_k, y_1, \dots, y_\ell) \in \mathrm{FO}(\varnothing)$ and a prime $p$. According to the semantics of $\mathrm{FOS}_p$, the formula $\alpha$ defines in an input structure $\mathfrak{A} = ([n])$ of size $n$ the $[n]^k \times [n]^\ell$-coefficient matrix $M_n$ whose entries are given, for $\bar{a} \in [n]^k, \bar{b} \in [n]^\ell$, as

$$M_n(\bar{a}, \bar{b}) = \begin{cases} 1, & \text{if } \mathfrak{A} \vDash \alpha(\bar{a}, \bar{b}) \\ 0, & \text{otherwise.} \end{cases}$$

Then $\mathfrak{A} \vDash (\mathsf{slv}_p \bar{x}_1, \bar{x}_2)\alpha(\bar{x}_1, \bar{x}_2)$ if the linear equation system $M_n \cdot \vec{x} = \mathbb{1}$ over $\mathbb{F}_p$ is solvable. For convenience we set $I_n = [n]^k$ and $J_n = [n]^\ell$.

Let $\Gamma = \Gamma_n = \mathrm{Sym}([n])$. Then the group $\Gamma$ acts on $I_n$ and $J_n$ in the natural way. Moreover, $\Gamma$ acts on the set of all $I_n \times J_n$-matrices as follows. To every $\pi \in \Gamma_n$ we can associate the $I_n \times I_n$-permutation matrix $\Pi_I$ which is defined as

$$\Pi_I(\bar{a}, \bar{b}) = \begin{cases} 1, & \pi(\bar{a}) = \bar{b} \\ 0, & \text{otherwise.} \end{cases}$$

Then $\Gamma$ acts on the set of $I_n \times J_n$-matrices by left multiplication with $I_n \times I_n$-permutation matrices. Similarly, we let $\Pi_J$ denote the $J_n \times J_n$-permutation matrix defined as

$$\Pi_J(\bar{a}, \bar{b}) = \begin{cases} 1, & \pi(\bar{a}) = \bar{b} \\ 0, & \text{otherwise.} \end{cases}$$

Then, analogously, $\Gamma$ acts on the set of $I_n \times J_n$-matrices by right multiplication with the associated $J_n \times J_n$-permutation matrices. Specifically, for all $\pi \in \Gamma$ we have $(\Pi_I \cdot M_n)(\bar{a}, \bar{b}) = M_n(\pi(\bar{a}), \bar{b})$ and $(M_n \cdot \Pi_J^{-1})(\bar{a}, \bar{b}) = M_n(\bar{a}, \pi(\bar{b}))$. Since $M_n$ is defined by a first-order formula over the empty signature, we conclude that $(\Pi_I \cdot M_n \cdot \Pi_J^{-1})(\bar{a}, \bar{b}) = M_n(\pi(\bar{a}), \pi(\bar{b})) = M_n(\bar{a}, \bar{b})$ and thus $\Pi_I \cdot M_n \cdot \Pi_J^{-1} = M_n$, which can equivalently be written as

$$\Pi_I \cdot M_n = M_n \cdot \Pi_J.$$

This identity will play a central role in our proof. For what follows, let us fix another prime $q$ which is distinct from $p$ and a subgroup $\Delta \leq \Gamma$ which is

a $q$-group, i.e. $|\Delta| = q^m$ for some $m \geq 0$. The overall strategy is to use the $\Delta$-symmetries of the matrix $M_n$ to strongly reduce the size of the linear equation system $M_n \cdot \vec{x} = \mathbb{1}$. More precisely, we claim that for $M_n^* := \sum_{\pi \in \Delta} \Pi_I \cdot M_n$ the linear equation system $M_n \cdot \vec{x} = \mathbb{1}$ is solvable if, and only if, $M_n^* \cdot \vec{x} = \mathbb{1}$ is solvable. First of all we note that for all $\pi \in \Delta$ we have:

- $\Pi_I \cdot M_n^* = \sum_{\lambda \in \Delta} \Pi_I \cdot \Lambda_I \cdot M_n = \sum_{\pi \in \Delta} \Pi_I \cdot M_n = M_n^*$
- $M_n^* \cdot \Pi_J = \sum_{\lambda \in \Delta} \Lambda_I \cdot M_n \cdot \Pi_J = \sum_{\lambda \in \Delta} \Lambda_I \cdot \Pi_I \cdot M_n = M_n^*$.

To verify our original claim, assume that $M_n^* \cdot \vec{b} = \mathbb{1}$. Then we have

$$\mathbb{1} = M_n^* \cdot \vec{b} = (\sum_{\pi \in \Delta} \Pi_I \cdot M_n) \cdot \vec{b} = (\sum_{\pi \in \Delta} M_n \cdot \Pi_J) \cdot \vec{b} = M_n \cdot \sum_{\pi \in \Delta} (\Pi_J \cdot \vec{b}).$$

For the other direction, let $M_n \cdot \vec{b} = \mathbb{1}$. Then $\sum_{\pi \in \Delta} \Pi_I \cdot M_n \cdot \vec{b} = |\Delta| \cdot \mathbb{1}$, hence $(1/|\Delta|) \cdot \vec{b}$ is a solution of the linear equation system $M_n^* \cdot \vec{x} = \mathbb{1}$. Note that for this direction we require that $q$ and $p$ are co-prime as we have to divide by $|\Delta|$.

Since $M_n^*$ satisfies $\Pi_I \cdot M_n^* = M_n^* \cdot \Pi_J = M_n^*$ for all $\pi \in \Delta$ we have

$$M_n^*(\bar{a}, \bar{b}) = M_n^*(\pi(\bar{a}), \bar{b}) = M_n^*(\bar{a}, \pi(\bar{b}))$$

for all $\bar{a} \in I_n, \bar{b} \in J_n$ and $\pi \in \Delta$. In other words, the entries of the $I_n \times J_n$-matrix $M_n^*$ are constant on the $\Delta$-orbits of $I_n$ and $J_n$ which means that we can *independently* change the indices of rows and columns (within $\Delta$-orbits) without affecting the entry of $M_n^*$. More specifically, if we let $I_n^\Delta$ and $J_n^\Delta$ denote the sets of $\Delta$-orbits on $I_n$ and $J_n$, respectively, then $M_n^*$ can be identified with the matrix $(M_n^*/\Delta)$ which is defined as

$$(M_n^*/\Delta) : I_n^\Delta \times J_n^\Delta \to \mathbb{F}_p, ([\bar{a}], [\bar{b}]) \mapsto M_n^*(\bar{a}, \bar{b}).$$

Note that, depending on the size of the group $\Delta$, the sets $I_n^\Delta$ and $J_n^\Delta$ can be noticeably smaller than the index sets $I_n$ and $J_n$. Hence our strategy is to choose $\Delta$ as large as possible to obtain a more compact linear equation system $M_n^* \cdot \vec{x} = \mathbb{1}$ which is equivalent to the given one.

### 4.3.1 Constructing large groups

Recall that the maximal $q$-subgroups $\Delta \leq \Gamma$ are the *$q$-Sylow groups* of $\Gamma$. It is well-known that for the case where $\Gamma = \mathrm{Sym}([n])$ these groups can be obtained via an inductive construction which we explain here for the special case of $n$ being a power of $q$ (the general case can be handled similarly, see e.g. [56]). Hence from now on, let us assume that $n = q^r$ for some $r \geq 1$.

First of all, we determine the size of $q$-Sylow groups of $\Gamma$. A simple induction shows that the maximal $t \geq 1$ such that $q^t$ divides $n! = (q^r)!$ is given by

$$t = q^{r-1} + q^{r-2} + \cdots + q + 1 = \frac{q^r - 1}{q - 1}.$$

In fact, we can write $(q^r)!$ as $(q^r)! = 1 \cdots (1 \cdot q) \cdots (2 \cdot q) \cdots (q^{r-1} \cdot q)$. Hence $t = t_* + q^{r-1}$ where $t_*$ is maximal such that $q^{t_*}$ divides $(q^{r-1})!$.

In particular, if we denote for $n = q^r$ a $q$-Sylow of $\mathrm{Sym}([n])$ by $\Delta_r$, then our argument from above shows that $|\Delta_1| = q$ and that

$$|\Delta_{r+1}| = |\Delta_r|^q \cdot q.$$

This equation already points to the algebraic structure of $\Delta_r$. In fact, it turns out that $\Delta_{r+1}$ is the *wreath product* of $\Delta_r$ and the cyclic group $\mathbb{Z}_q$. Since $\Delta_1 = \mathbb{Z}_q$ it follows that $\Delta_r$ is the $r$-fold wreath product of the cyclic group $\mathbb{Z}_q$. We omit the formal definition of wreath products and directly illustrate this concept for the case of the $q$-Sylow groups of $\Gamma = \mathrm{Sym}([n]) = \mathrm{Sym}([q^r])$.

More specifically, we proceed to give a construction for the group $\Gamma$ from which we can read off certain algebraic properties which are important for our proofs later in this section. To this end, we inductively construct, for $r \geq 1$, a $q$-Sylow subgroup $\Delta_r \leq \mathrm{Sym}([q^r])$ together with a family of trees $\mathcal{T}_i^x$ for $i = 0, \ldots, r$ and $x \in [q^{r-i}]$ such that the following properties hold.

(I) $\mathcal{T}_i^x$ is a complete $q$-ary tree of height $i$ whose leaves are labelled with elements from $[n]$. More precisely, the labels of the leaves of $\mathcal{T}_i^x$ form the set $\mathcal{P}_i^x = \{x \cdot q^i, x \cdot q^i + 1, \ldots, (x+1) \cdot q^i - 1\}$ (note that $\mathcal{P}_i^x$ is the $x$-th block of the natural partition of $[n]$ into parts of size $q^i$).

(II) For all $i \leq r$, the group $\Delta_r$ transitively acts on the set $\{\mathcal{T}_i^x : x \in [q^{r-i}]\}$ by applying permutations $\delta \in \Delta_r$ to the labels of the leaves of the tress $\mathcal{T}_i^x$. Moreover, for each $i \leq r$, the subgroup of $\Delta_r$ which point-wise stabilises the trees $\mathcal{T}_i^x$ is a normal subgroup of $\Delta_r$.

(III) We have $\Delta_1 \leq \Delta_2 \leq \cdots \leq \Delta_r$ where $\Delta_i$ acts on the set of labels $\mathcal{P}_i^0$ of the tree $\mathcal{T}_i^0$. More generally, for every block $\mathcal{P}_i^x$, the group $\Delta_r$ contains a subgroup $\Delta_r^{i,x} \leq \Delta_r$ which point-wise fixes the elements of all blocks $\mathcal{P}_i^y$ for $y \neq x$ and whose action on $\mathcal{P}_i^x$ corresponds to the action of $\Delta_i$ on $\mathcal{P}_i^0$.

The inductive construction of the trees $\mathcal{T}_i^x$ is depicted in Figure 4.2. To understand this construction, it is quite useful to think of elements $y \in [n]$ as being represented in their $q$-adic encoding, i.e. $y = y_0 + y_1 \cdot q + \cdots + y_{r-1} \cdot q^{r-1}$. Then we have that $y \in \mathcal{P}_r^0 = [n]$ and

- $y \in \mathcal{P}_{r-1}^{y_{r-1}}$
- $y \in \mathcal{P}_{r-2}^{y_{r-2} + y_{r-1} \cdot q}$
- $\ldots$
- $y \in \mathcal{P}_0^{y_0 + \cdots + y_{r-1} \cdot q^{r-1}} = \mathcal{P}_0^y$.

Hence, the $q$-adic encoding of the element $y$ describes the unique path in the tree $\mathcal{T}_r^0$ from its root to the leaf $\mathcal{T}_0^y$. We observe that the trees $\mathcal{T}_i^x$ clearly satisfy the properties which we stated in (I).

For the inductive construction of the $q$-Sylow groups $\Delta_r$, we first fix $\Delta_1$ as the cyclic group generated by the natural cyclic shift $\gamma = (0\,1\cdots q-1)$ on the set $\mathcal{P}_1^0 = \{0,\dots,q-1\}$.

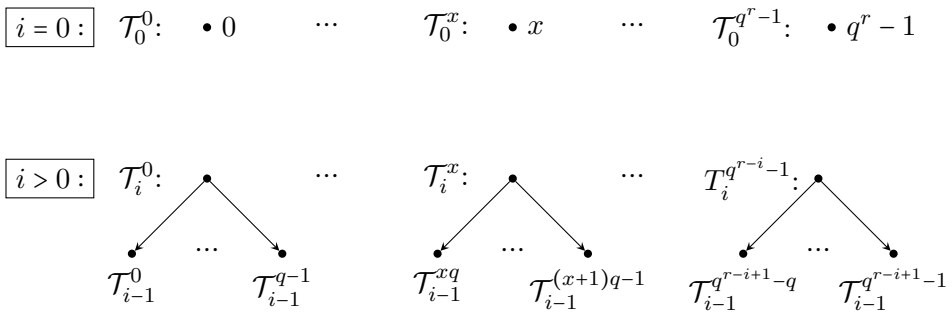


Figure 4.2: Inductive definition of the trees $\mathcal{T}_i^x$

We proceed with the inductive step $r \mapsto r+1$. The set $[q^{r+1}]$ splits into $q$ blocks $\mathcal{P}_r^0,\dots,\mathcal{P}_r^{q-1}$, each of size $q^r$. The group $\Delta_r$ acts on $\mathcal{P}_r^0$ and point-wise fixes the elements from the blocks $\mathcal{P}_r^x$ with $x \neq 0$. Let $\gamma \in \mathrm{Sym}([n])$ for $n = q^{r+1}$ be the following permutation which shifts the segments $\mathcal{P}_r^0,\dots,\mathcal{P}_r^{q-1}$ in a cycle of length $q$ by composing the natural shifts on the sets of residues modulo $q^r$:

$$\gamma = (0\cdots(q-1)q^r)(1\cdots 1+(q-1)q^r)\cdots(q^r-1\cdots q^r-1+(q-1)q^r).$$

Hence for all $a \in [n]$ we have $\gamma(a) = (a+q^r) \bmod q^{r+1}$. We set $\Delta_r^0 = \Delta_r$ and, more generally, $\Delta_r^x = (\gamma^x)\Delta_r(\gamma^x)^{-1}$ for $x = 0,\dots,q-1$ to obtain $q$ copies of $\Delta_r$ which independently act on the segments $\mathcal{P}_r^x$ for $0 \le x \le q-1$. Finally, we define $\Delta_{r+1}$ as the *semi-direct product* of $(\Delta_r^0 \times \cdots \times \Delta_r^{q-1})$ and the cyclic group $\langle\gamma\rangle$ of size $q$. This means that the group elements of $\Delta_{r+1}$ are elements in the set $(\Delta_r^0 \times \cdots \times \Delta_r^{q-1} \times \langle\gamma\rangle)$ and that the group operation is given by

$$(\delta_1,\dots,\delta_{q-1},\alpha)\cdot(\epsilon_1,\dots,\epsilon_{q-1},\beta) = ((\delta_1,\dots,\delta_{q-1})\cdot\alpha(\epsilon_1,\dots,\epsilon_{q-1})\alpha^{-1},\alpha\cdot\beta).$$

Since $|\Delta_{r+1}| = |\Delta_r|^q \cdot q$ we conclude that $\Delta_{r+1}$ indeed is a $q$-Sylow subgroup.

From our construction it immediately follows that $\Delta_{r+1}$ satisfies the properties stated in (III). To see that $\Delta_{r+1}$ also satisfies the properties stated in (II), we start by showing that, for $i \le r$, $\Delta_{r+1}$ transitively acts on $\{\mathcal{T}_i^x : x \in [q^{r+1-i}]\}$. If we split the set $[q^{r+1-i}]$ into $q$ blocks $\mathcal{P}_{r-i}^0,\dots,\mathcal{P}_{r-i}^{q-1}$ of size $q^{r-i}$, then we know from the induction hypothesis that $\Delta_r^0$ transitively acts on the set of trees $\{\mathcal{T}_i^x : x \in \mathcal{P}_{r-i}^0\} = \{\mathcal{T}_i^x : x \in [q^{r-i}]\}$. Moreover, it is easy to verify that for all $x \in [q^{r+1-i}]$ we have $\gamma(\mathcal{T}_i^x) = \mathcal{T}_i^z$ where $z = x + q^{r-i} \bmod q^{r+1-i}$. Hence $(\gamma^y)\{\mathcal{T}_i^x : x \in \mathcal{P}_{r-i}^0\} = \{\mathcal{T}_i^x : x \in \mathcal{P}_{r-i}^y\}$ for all $0 \le y \le q-1$ which means that $\Delta_r^y$ transitively acts on $\{\mathcal{T}_i^x : x \in \mathcal{P}_{r-i}^y\}$ and thus (II) holds.

The crucial step is to understand the action of $\Delta_r$ on the sets $I_n = [n]^k$ and $J_n = [n]^\ell$ (for the case where $n = q^r$). In fact, our next aim is to develop a complete invariant for the $\Delta_r$-orbits on these index sets. Recall that the sets of $\Delta_r$-orbits on $I_n$ and $J_n$ provide index sets for the succinct linear equation system $M_n^* \cdot \vec{x} = \mathbb{1}$. To define this invariant, the main idea is to describe the position of a tuple $\bar{a} \in I_n$ (or $\bar{a} \in J_n$, respectively) in the tree $\mathcal{T} := \mathcal{T}_r^0$.

Let us first define the *signature* $\mathrm{sgn}(a,b)$ of a pair $(a,b) \in [n] \times [n]$ as the tuple $(i,z) \in [r+1] \times [q]$ such that the lowest common ancestor of $a, b$ in $\mathcal{T}$ is the root of a tree $\mathcal{T}_i^x$ and such that $a$ is located in a subtree $\mathcal{T}_{i-1}^{xq+y_a}$ for $y_a \in [q]$ and $b$ is located in the subtree $\mathcal{T}_{i-1}^{xq+y_b}$ where $y_b = y_a + z \bmod q$. For the special case where $i = 0$ we have $a = b$ and agree to set $z = 0$. With this preparation we define the signature $\mathrm{sgn}(\bar{a})$ of a tuple $\bar{a} = (a_1, \dots, a_\ell) \in J_n$ as the list $\sigma \in ([r+1] \times [q])^{\ell(\ell-1)/2}$ consisting of the individual signatures $\mathrm{sgn}(a_i, a_j)$ for all pairs $a_i, a_j$ with $1 \le i < j \le \ell$. The signature of tuples in $I_n$ is defined analogously.

**Lemma 4.10.** *Let $\bar{a} \in J_n$. Then $sgn(\bar{a}) = sgn(\pi\bar{a})$ for all $\pi \in \Delta_r$.*

*Proof.* This easily follows from the constructions of $\Delta_r$ and the trees $\mathcal{T}_i^x$. □

**Lemma 4.11.** *Let $\bar{a}, \bar{b} \in J_n$. If $sgn(\bar{a}) = sgn(\bar{b})$, then $\bar{b} \in \Delta_r(\bar{a})$.*

*Proof.* We proceed by induction on the maximal position $0 \le i \le \ell$ such that $a_j = b_j$ for all $j = 1, \dots, i$. The case $i = \ell$ is clear, so assume that $i < \ell$. Let $\bar{a} = (a_1, \dots, a_i, a_{i+1}, \dots, a_\ell)$ and $\bar{b} = (a_1, \dots, a_i, b_{i+1}, \dots, b_\ell)$. We show that there exists a permutation $\delta \in \Delta_r$ which pointwise fixes $a_1, \dots, a_i$ and such that $\delta(a_{i+1}) = b_{i+1}$. Then the claim follows from Lemma 4.10 together with the induction hypothesis. For $i = 0$ this is easy, because $\Delta_r$ acts transitively on $[n]$. If $i > 0$, we choose $a_w \in \{a_1, \dots, a_i\}$ such that $\mathrm{sgn}(a_w, a_{i+1}) = (c,d)$ and such that $c$ is minimal with this property. Obviously we have $c > 0$. By the choice of $a_w$ the lowest common ancestor of $a_w$ and $a_{i+1}$ is the root of a tree $\mathcal{T}_c^x$. Moreover, $a_w$ is located in a subtree $\mathcal{T}_{c-1}^{xq+y}$ for some $0 \le y \le q-1$ and $a_{i+1}$ is located in the subtree $\mathcal{T}_{c-1}^{xq+z}$ where $z = y + d \mod q$. Since $\mathrm{sgn}(\bar{a}) = \mathrm{sgn}(\bar{b})$, also $b_{i+1}$ occurs as the label of a leaf in the subtree $\mathcal{T}_{c-1}^{xq+z}$. By the minimality assumption on $c$ we know that non of the elements $\{a_1, \dots, a_i\}$ occurs in the tree $\mathcal{T}_{c-1}^{xq+z}$. Hence, by the properties of the group $\Delta_r$ stated in (III), we can find an element $\delta \in \Delta_r$ which point-wise fixes all elements outside the block $\mathcal{P}_{c-1}^{xq+z}$ (in particular, the elements $a_1, \dots, a_i$) and which moves $a_{i+1}$ to $b_{i+1}$. □

### 4.3.2 Defining sizes of orbits in first-order logic with counting

Following our definition from above, the signature $\mathrm{sgn}(\bar{a})$ of an element $\bar{a} \in J_n$ is a tuple of length $\ell(\ell-1)/2$ whose entries are pairs $(i,z) \in [r+1] \times [q]$. We denote the set of all possible sequences of this form by $S_n^\ell = ([r+1] \times [q])^{\ell(\ell-1)/2}$. Of course, not every tuple in $\sigma \in S_n^\ell$ can be realised as the signature $\mathrm{sgn}(\bar{a}) = \sigma$ of an element $\bar{a} \in J_n$. Analogously, we define the set $S_n^k = ([r+1] \times [q])^{k(k-1)/2}$ to capture all possible signatures of elements in $I_n$.

Since the coefficient matrix $M_n^\star$ of the equivalent linear equation system $M_n^\star \cdot \vec{x} = \mathbb{1}$ can be defined as a matrix whose index sets are the collections of $\Delta_r$-orbits on $I_n$ and $J_n$, we can use the notion of signatures to describe $M_n^\star$ as an $(S_n^k \times S_n^\ell)$-matrix. This brings us closer to our goal, since the index sets $S_n^k$ and $S_n^\ell$ of the matrix $M_n^\star$ are much smaller than the index sets $I_n$ and $J_n$ of the coefficient matrix $M_n$ of the original linear equation system. However, it still might be the case that the succinctness of the matrix $M_n^\star$ does not help, because it is not possible to obtain its entries within $\mathrm{FOR}_p$.

We show that this is not the case. More precisely we show that we can define the matrix $M_n^\star$ in FOC in a structure of size $r$ (where we assume that $r \geq q$). Therefore, the main technical step is to show that FOC can count (modulo $p$) the number of realisations of a potential signature $\sigma \in S_n^k$.

First of all, we need some further notation. A *complete equality type in $k + \ell$ variables* is a consistent set $\tau(x_1, \ldots, x_k, x_{k+1}, \ldots, x_{k+\ell})$ of literals $x_i = x_j, x_i \neq x_j$ which contains, for every pair $i < j$, either the atom $x_i = x_j$ or the literal $x_i \neq x_j$. Note that each quantifier-free formula $\alpha \in \mathrm{FO}(\varnothing)$ can be expressed as a Boolean combination of complete equality types.

In the following main technical lemma we show that in the structure $\mathfrak{A} = ([r])$ we can count (modulo $p$) the number of realisations of a (potential) signature $\sigma \in S_n^\ell$ in a subtree $\mathcal{T}_i^x$ in FOC. More generally, this is possible if we additionally fix some entries of the tuples which should realise $\sigma$ in $\mathcal{T}_i^x$. For this we need another prerequisite: as we want to work with elements from the set $[n] = [q^r]$ in a structure of size $r$, we have to agree on some sort of succinct representation. Of course the natural choice is to represent numbers $x \in [n]$ in the structure $\mathfrak{A}$ via their $q$-adic encoding: a binary relation $R \subseteq [r]^2$ which corresponds to a function $R : [r] \to [q]$ represents the number $x(R) \in [n] = \sum_{i=0}^{r-1} R(i) \cdot q^i$. Note that this encoding requires a given linear order on the set $[r]$ (which is *not* available in the structure $\mathfrak{A}$). However, as we are working with FOC, we can just use the number sort on which a linear order is given. Hence in the following, whenever we specify FOC-formulas or FOC-terms with free variables or with free relation symbols which should represent numbers, then we implicitly assume that these variables are *numeric* variables and that the relation symbols are evaluated over the number sort. The same holds for signatures $\sigma \in S_n^\ell$ which we specify in FOC-formulas by a list of pairs $(h_i, d_i)$ of *numeric* variables of length $\binom{\ell}{2}$.

Before we state our main technical lemma it is helpful to recall that our inductive construction of the trees $\mathcal{T}_i^x$ fits very well with the $q$-adic encoding of numbers $x \in [n]$. Again, let $x \in [n]$ be given by its $q$-adic encoding as $x = (x_0, \ldots, x_{r-1}) \in [q]^r$, i.e. $x = \sum_{i=0}^{r-1} x_i \cdot q^i$. Then the $i$-th node on the unique path from the root in the tree $\mathcal{T} = \mathcal{T}_r^0$ to the leaf $\mathcal{T}_0^x$ is the root of the tree $\mathcal{T}_{r-i}^y$ where $y = x_{r-i} + x_{r-i+1} q + \cdots + x_{r-1} q^{i-1}$. In other words, the $q$-adic encoding of $x$ precisely describes the path in the tree $\mathcal{T}$ from the root to the leaf labelled with $x$ where at level $(r - i)$ the $i$ last entries $x_{r-i}, \ldots, x_{r-1}$ in the $q$-adic encoding of $x$ are determined (i.e. $x$ is a member of the block $\mathcal{P}_{r-i}^y$).

**Lemma 4.12.** *For all $\ell \geq 1$ and $0 \leq s \leq \ell$ there exist*

*(a) a term $\Theta(i, h_1, d_1, \ldots, h_t, d_t) \in \mathrm{FOC}(\{R_x, R_1, \ldots, R_s\})$, and*

*(b) formulas $\varphi_e(y, z, i, h_1, d_1, \ldots, h_t, d_t) \in \mathrm{FOC}(\{R_x, R_1, \ldots, R_s\})$ for $e = s+1, \ldots, \ell$,*

*where $t = \binom{\ell}{2}$, such that for all $r \geq q$, all $i \leq r$, all $\sigma = ((h_1, d_1), \ldots, (h_t, d_t)) \in S_n^\ell$ where $n = q^r$, all $x \in [q^{r-i}]$ and all $a_1, \ldots, a_s \in \mathcal{P}_i^x$ the following holds: Let $\mathfrak{A} = ([r])$ and let $R_x, R_1, \ldots, R_s$ be numerical relations such that $R_x$ represents the (q-adic encoding of the) element $x \in [q^{r-i}]$ and such that each $R_i$ represents the (q-adic encoding of the) element $a_i$. Then it holds that*

*(i) the value $\Theta^{\mathfrak{A}}(q, i, h_1, d_1, \ldots, h_t, d_t)$ of the term $\Theta$ in $\mathfrak{A}$ is $|Z| \bmod p$ where*

$$Z = \{(a_{s+1}, \ldots, a_\ell) \in (\mathcal{P}_i^x)^{\ell - s} : sgn(a_1, \ldots, a_s, a_{s+1}, \ldots, a_\ell) = \sigma\}.$$

*(ii) if $Z \neq \varnothing$, then the formulas $(\varphi_e)_{s<e\leq\ell}$ define the q-adic representation of witnessing elements $a_{s+1}, \ldots, a_\ell \in \mathcal{P}_i^x$, i.e. such that $(a_{s+1}, \ldots, a_\ell) \in Z$.*

*Proof.* First of all, by our previous observations it is easy to see that the condition $a_j \in \mathcal{P}_i^x$ for $j = 1, \ldots, s$ can be defined in FOC. More generally, we can use the $q$-adic encoding of the elements $a_j$ to determine $\mathrm{sgn}(a_1, \ldots, a_s)$ in FOC. Hence, for the remainder of the proof we assume that $\mathrm{sgn}(a_1, \ldots, a_s)$ is consistent with $\sigma$ and that $a_j \in \mathcal{P}_i^x$ for $j = 1, \ldots, s$.

We proceed by induction on $\ell$. For $\ell = 1$ it suffices to show that FOC can compute $(n \bmod p)$ where $n = q^r$ in the structure $\mathfrak{A}$. To see this, recall that $p$ and $q$ are co-prime and thus we can use Lagrange's theorem to conclude that $q^r \equiv q^{r'} \bmod p$ if $r' \equiv r \bmod (p-1)$. Since $p$ is a constant, the claim follows.

Let $\ell \geq 2$. We distinguish between the following two cases. If $s = 0$, then we can partition the set of realisations $\bar{a}$ of $\sigma$ according to first entry $a_1$ into $|\mathcal{P}_i^x|$ parts of equal size. It suffices to determine the size of each of these blocks, since we can determine $|\mathcal{P}_i^x| \bmod p$ in FOC similarly as above.

Without loss of generality let us assume that $a_1 = x \cdot q^i$. Since we have access to the $q$-adic encoding of $x$, it is easy to see that we can define the $q$-adic encoding of $xq^i$ in FOC. This gives us the formula $\varphi_1$. Next, we partition the set of indices $\{2, \ldots, \ell\}$ into classes according to the equivalence relation $j_1 \approx j_2$ which is determined by $\sigma(1, j_1) = \sigma(1, j_2)$. Let the resulting classes be $Y_1, \ldots, Y_v$ and let $\sigma(1, y) = (h_w, d_w)$ for all $y \in Y_w$ and $w = 1, \ldots, v$.

We observe that there exists a tuple $\bar{a}$ with $a_1 = x \cdot q^i$ which realises $\sigma$ in the tree $\mathcal{T}_i^x$ (that is $Z \neq \varnothing$) if, and only if, the following conditions are satisfied:

- for all $w = 1, \ldots, v$ we have $h_w \leq i$, and
- for every $Y_w = \{y_1^w, \ldots, y_{\ell_w}^w\}$ there is a tuple $\bar{a}^w$ of length $\ell_w$ which realises $\sigma$ (restricted to the indices from $Y_w$) in the subtree $\mathcal{T}_{h_w - 1}^{xq^{i-h_w+1}+d_w}$, and

- for all pairs $y_1 \in Y_{w_1}$ and $y_2 \in Y_{w_2}$ with $w_1 \neq w_2$ we have that

$$\sigma(y_1, y_2) = \begin{cases} (h_{w_1}, d_{w_2} - d_{w_1} \bmod q), & \text{if } h_{w_1} = h_{w_2} \\ (h_{w_2}, d_{w_2}) & \text{if } h_{w_1} < h_{w_2} \\ (h_{w_1}, d_{w_1}) & \text{if } h_{w_2} < h_{w_1}. \end{cases}$$

Since $\ell$ is a constant, the number of possible partitions of $\{2, \dots, \ell\}$ is bounded by a constant as well. It is easy to see that for every possible such partition we can check the first and third condition in FOC. To verify the second condition in FOC, we use the induction hypothesis. There are two aspects which have to be discussed with more precision. First of all, we have to handle one particular case separately: indeed, if $h_w = 1$ for all $w = 1, \dots, v$, then we cannot use the induction hypothesis since all elements (including $a_1$) have to be chosen in the same subtree of height one. However, in this case there is only one realisation (if the third condition is satisfied) so this does not cause any problems. The other difficulty is that we have to define the $q$-adic encoding of the value $z_w = xq^{i-h_w+1} + d_w$ in FOC. We already noted above that the $q$-adic representation of $xq^{i-h_w+1}$ can be defined in FOC and since $0 \leq d_w < q$ we can also define the $q$-adic encoding of $z$ in FOC.

In fact, the induction hypothesis also provides us with a term which counts modulo $p$ the number of possible realisations of $\sigma$ in the subtrees $\mathcal{T}^{z_w}_{h_w-1}$ restricted to the indices in $Y_w$ together with formulas $\varphi_e$ which define witnessing elements. Finally, since the overall number of possible realisations of $\sigma$ in $\mathcal{T}^x_i$ is the product of the realisations restricted to the components $Y_w$, the claim follows for the case where $s = 0$.

For the general case let $\ell \geq s > 0$ and let $a_1, \dots, a_s \in \mathcal{P}^x_i$ be the components of the tuple $\bar{a}$ that are already fixed. Recall that we can assume without loss of generality that $\mathrm{sgn}(a_1, \dots, a_s)$ is consistent with $\sigma$ and that all elements $a_1, \dots, a_s$ are located in the subtree $\mathcal{T}^x_i$. Since we have fixed the element $a_1$, we can proceed as above except for two small changes. First of all, when applying the induction hypothesis we have to respect the remaining fixed elements $a_2, \dots, a_s$. Moreover, when we form the partitions of $\{2, \dots, \ell\}$ into parts $Y_1, \dots, Y_v$ as above then we have to adapt the position of elements corresponding to the index set $Y_w$ since the element $a_1$ is not necessarily contained in the tree $\mathcal{T}^{xq^{i-h_w+1}}_{h_w+1}$. However, since we can access the $q$-adic representation of $a_1$, we can define in FOC the element $0 \leq d_a < q$ such that $a_1$ is located in the subtree $\mathcal{T}^{xq^{i-h_w+1}+d_a}_{h_w+1}$. The remaining steps can be performed as above. This finishes our proof. □

**Lemma 4.13.** *Let $\tau(x_1, \dots, x_k, y_1, \dots, y_\ell) \in \mathrm{FO}(\varnothing)$ be a complete equality type (in $k + \ell$ variables). Then there is an* FOC*-term $\Theta_\tau(\bar{z}_x, \bar{z}_y)$ such that for all $r \geq q$, all $\sigma_{\bar{a}} \in S^k_n$ and $\sigma_{\bar{b}} \in S^\ell_n$, where $n = q^r$, the value $\Theta^{\mathfrak{A}}_\tau(\sigma_{\bar{a}}, \sigma_{\bar{b}})$ of $\Theta$ in $\mathfrak{A} = ([r])$ is*

$$\Theta^{\mathfrak{A}}_\tau(\sigma_{\bar{a}}, \sigma_{\bar{b}}) = |\{\bar{b} \in J_n : sgn(\bar{b}) = \sigma_{\bar{b}}, ([n]) \vDash \tau(\bar{a}, \bar{b})\}| \bmod p$$

*for some (or, equivalently, all)* $\bar{a} \in I_n$ *with* $sgn(\bar{a}) = \sigma_{\bar{a}}$.

*Proof.* By Lemma 4.12 we can first check in FOC that $\sigma_{\bar{a}}$ and $\sigma_{\bar{b}}$ can be realised (otherwise the answer is trivial). Moreover, if $\tau$ (restricted to $x_1, \dots, x_k$) is not consistent with $\sigma_{\bar{a}}$ or if $\tau$ (restricted to $y_1, \dots, y_\ell$) contradicts $\sigma_{\bar{b}}$, then the answer is trivial as well.

In all other cases, Lemma 4.12 provides FOC-formulas which define in the structure $\mathfrak{A}$ the $q$-adic encoding of elements $a_1, \dots, a_k \in [n]$ such that $\mathrm{sgn}(\bar{a}) = \sigma_{\bar{a}}$. Moreover, if $\tau$ contains a literal $x_i = y_j$, then we can fix the entry $b_j$ as well. Hence, let us assume without loss of generality that $\tau$ contains the literals $x_i \neq y_j$ for all $1 \leq i \leq k$ and $1 \leq j \leq \ell$.

For $Y \subseteq \{1, \dots, \ell\}$ and a partial assignment $\epsilon : \{1, \dots, \ell\} \to \{a_1, \dots, a_k\}$ with $\mathrm{dom}(\epsilon) \cap Y = \varnothing$ we define the set

$$B^\epsilon_Y = \{\bar{b} \in J_n : \mathrm{sgn}(\bar{b}) = \sigma_{\bar{b}},\ \text{for } i \in \mathrm{dom}(\epsilon) : b_i = \epsilon(i),\ \text{for } i \in Y : b_i \neq a_1, \dots, a_k\}.$$

With this notation our overall aim is to determine $(|B^\varnothing_Y| \bmod p)$ for $Y = [\ell]$ in FOC. The first observation is that by Lemma 4.12 we can determine $(|B^\epsilon_\varnothing| \bmod p)$ for all partial assignments $\epsilon$ in FOC. The second observation is that we can construct the values $(|B^\epsilon_Y| \bmod p)$ by induction on $|Y|$ as follows. For $Y \subseteq \{1, \dots, \ell\}$ and a partial assignment $\epsilon$ (with $\mathrm{dom}(\epsilon) \cap Y = \varnothing$) we have for all $j \in Y$ that

$$|B^\epsilon_Y| = |B^\epsilon_{Y \setminus \{j\}}| - \sum_{a \in \{a_1, \dots, a_k\}} |B^{\epsilon \cup \{j \mapsto a\}}_{Y \setminus \{j\}}|.$$

In this way we recursively obtain the value $(|B^\varnothing_Y| \bmod p)$ for $Y = [\ell]$. Since $\ell$ is a constant, the recursion depth is bounded by a constant as well and the procedure can be formalised in FOC. □

**Lemma 4.14.** *There exists an* FOC*-term* $\Theta(\bar{\mu}, \bar{\nu})$ *which defines for all* $r \geq q$ *in the structure* $\mathfrak{A} = ([r])$ *the matrix* $M^\star_n$ *where* $n = q^r$.

*Proof.* Recall that we can view $M^\star_n$ as an $(S^k_n \times S^\ell_n)$-matrix over $\mathbb{F}_p$. To represent the index sets $S^k_n$ and $S^\ell_n$ we let $\bar{\mu}$ and $\bar{\nu}$ be tuples of numeric variables of lengths $|\bar{\mu}| = \binom{k}{2}$ and $|\bar{\nu}| = \binom{\ell}{2}$, respectively.

The entry $M^\star_n(\sigma_{\bar{a}}, \sigma_{\bar{b}})$ of $M^\star_n$ for $\sigma_{\bar{a}} \in S^k_n$ and $\sigma_{\bar{b}} \in S^\ell_n$ is given as

$$M^\star_n(\sigma_{\bar{a}}, \sigma_{\bar{b}}) = |\{\bar{b} \in J_n : \mathrm{sgn}(\bar{b}) = \sigma_{\bar{b}}, M_n(\bar{a}, \bar{b}) = 1\}| \cdot |\mathrm{Stab}(\bar{b})| \mod p,$$

for some (or, equivalently, all) $\bar{a} \in I_n$, $\bar{b} \in J_n$ with $\mathrm{sgn}(\bar{a}) = \sigma_{\bar{a}}$ and $\mathrm{sgn}(\bar{b}) = \sigma_{\bar{b}}$. The entry $M_n(\bar{a}, \bar{b})$, in turn, is determined by the quantifier-free formula $\alpha(\bar{x}_1, \bar{x}_2) \in \mathrm{FO}(\varnothing)$. Lemma 4.13 shows that we can determine the value of the left-hand side of the above equation for the case where $\alpha$ is a complete equality type. For the general case, we write $\alpha$ as the union of complete equality types and combine the constant number of intermediate results. Moreover, we can determine $|\mathrm{Stab}(\bar{b})|$ by Lemma 4.12 (which shows that the size of the orbit of $\bar{b}$ is definable) and by the orbit-stabiliser theorem. □

**Definition 4.15.** Let $\mathcal{K} \subseteq \mathcal{S}(\varnothing)$ be a class of sets. The *q-power* $\mathcal{K}^q \subseteq \mathcal{S}(\varnothing)$ *of* $\mathcal{K}$ consists of all sets $\mathfrak{A} = ([q^r])$ such that $\mathfrak{B} = ([r]) \in \mathcal{K}$.

**Theorem 4.16.** *Let* $\mathcal{K} \subseteq \mathcal{S}(\varnothing)$ *be a class of sets. If* $\mathcal{K}^q$ *is definable in* $\mathrm{FOS}_p$*, then* $\mathcal{K}$ *is definable in* $\mathrm{FOR}_p$*.*

*Proof.* If $\mathcal{K}^q$ is definable in $\mathrm{FOS}_p$, then by Theorem 4.9 we find a formula $\varphi = (\mathsf{slv}_p\, \bar{x}_1, \bar{x}_2)\alpha(\bar{x}_1, \bar{x}_2) \in \mathrm{FOS}_p$ that defines $\mathcal{K}^q$ where $\alpha$ is quantifier-free.

By using the above construction and Lemma 4.14, we conclude that the linear equation system $M_n \cdot \vec{x} = \mathbb{1}$ defined by $\alpha$ in an input structure $\mathfrak{A} = ([n])$ of size $n = q^r$ can be transformed into the equivalent system $M_n^* \cdot \vec{x} = \mathbb{1}$ which is FOC-definable in $\mathfrak{B} = ([r])$. Let $\varphi^* \in \mathrm{FOR}_p$ be a formula which expresses the solvability of the linear system $M_n^* \cdot \vec{x} = \mathbb{1}$ in a structure $\mathcal{B} = ([r])$.

Then $\mathfrak{B} \models \varphi^*$ if, and only if, $\mathfrak{A} \models \varphi$ since the linear equation systems $M_n \cdot \vec{x} = \mathbb{1}$ and $M_n^* \cdot \vec{x} = \mathbb{1}$ are equivalent. □

**Theorem 4.17.** *For all primes* $p$ *we have* $\mathrm{FOS}_p < \mathrm{FOR}_p$ *(even over* $\mathcal{S}(\varnothing)$*).*

*Proof.* Suppose for the sake of a contradiction that $\mathrm{FOS}_p = \mathrm{FOR}_p$. As above we fix some prime $q \neq p$. Let $\mathcal{K} \subseteq \mathcal{S}(\varnothing)$ be a class of sets such that $\mathcal{K} \notin \mathrm{FOR}_p$, but such that $(\mathcal{K}^q)^q \in \mathrm{FOR}_p$. Such a class $\mathcal{K}$ is well-known to exist. In fact, it follows from the space-hierarchy theorem, see e.g. [81], that there exists a language $L \subseteq \{1^n : n \in \mathbb{N}\}$ such that $L \in \textsc{Space}(2^{cn})$ and $L \notin \textsc{Pspace}$. But then for an appropriate prime $q$ we have that $L' = \{q^{q^n} : 1^n \in L\} \in \textsc{Logspace}$. Since, over sets, we have $\textsc{Logspace} \leq \mathrm{FOR}_p \leq \textsc{Ptime} \leq \textsc{Pspace}$, this shows that we can choose $\mathcal{K} = \{([n]) : 1^n \in L\}$.

Now, since we assumed that $\mathrm{FOS}_p = \mathrm{FOR}_p$ we have $(\mathcal{K}^q)^q \in \mathrm{FOS}_p$ and by Theorem 4.16 this means that $\mathcal{K}^q \in \mathrm{FOR}_p$. Again, since $\mathrm{FOR}_p = \mathrm{FOS}_p$, we have $\mathcal{K}^q \in \mathrm{FOS}_p$. A second application of Theorem 4.16 yields $\mathcal{K} \in \mathrm{FOR}_p$, which contradicts our assumptions. □

Let us remark that the same proof also works for the extension of *fixed-point logic* by solvability quantifiers (but still in the absence of counting). The simple reason is that, in the absence of counting, fixed-point operators do not increase the expressive power of first-order logic over the empty signature, since all definable relations consist of constantly many basic building blocks (and thus we can evaluate fixed points already in first-order logic). In other words, if we denote by $\mathrm{FPS}_p^-$ the extension of fixed-point logic by solvability quantifiers $\mathsf{slv}_p$ over $\mathbb{F}_p$ (without counting), then we have $\mathrm{FOS}_p = \mathrm{FPS}_p^-$ over $\mathcal{S}(\varnothing)$.

**Theorem 4.18.** *For all primes* $p$*, we have* $\mathrm{FPS}_p^- < \mathrm{FOR}_p$ *over* $\mathcal{S}(\varnothing)$*.*

Finally, another interesting consequence is that there exists an FPC-definable query over $\mathcal{S}(\varnothing)$ which cannot be defined in $\mathrm{FPS}_p$. This immediately follows from our proof of Theorem 4.16, since the solvability of the linear equation system $M_n^* \cdot \vec{x} = \mathbb{1}$ matrix can also be expressed in FPC (we interpret the coefficient matrix $M_n^*$ over the second *ordered* sort). Note that, in contrast, we have no proof which shows that FPC cannot be embedded into $\mathrm{FOR}_p$.

## 4.4 Separation results over different prime fields

In this section we separate solvability quantifiers and rank operators over different prime fields. This solves an open question by Dawar and Holm who asked whether for distinct primes $p, q \in \mathbb{P}$ we have that $\mathrm{FPR}_p \neq \mathrm{FPR}_q$, [29, 59, 71]. In [59], Holm was able to prove this separation for the special case where rank operators are restricted to dimension one. In this section we settle the general case and show that the expressive power of $\mathrm{FPR}_\Omega$ and $\mathrm{FPR}_{\Omega'}$ is different for all distinct sets of primes $\Omega, \Omega' \subseteq \mathbb{P}$. An important consequence of our result is that rank logic (in the way it was defined in [28]) does not suffice to capture polynomial time. Let us state these two results explicitly.

**Theorem 4.19.** *Let $\Omega$ and $\Omega'$ be two sets of primes such that $\Omega \neq \Omega'$. Then $\mathrm{FPS}_\Omega \neq \mathrm{FPS}_{\Omega'}$ and $\mathrm{FPR}_\Omega \neq \mathrm{FPR}_{\Omega'}$.*

**Theorem 4.20.** *Rank logic fails to capture polynomial time, that is*

$$\mathrm{FPR} < \mathrm{FPR}^* \leq \textsc{Ptime}.$$

Recall from Section 4.1 that $\mathrm{FPR}^*$ is the extension of FPC by a *uniform* rank operator $\mathsf{rk}$ which can express the matrix rank problem uniformly over prime fields $\mathbb{F}_p$ (which means that $p$ is part of the input of the operator). More precisely, we show that this uniform rank operator cannot be expressed in FPR. The intuitive reason is that having separate rank operators $\mathsf{rk}_p$ for every prime $p \in \mathbb{P}$ does not suffice to simultaneously define the matrix rank over *all* prime fields $\mathbb{F}_p$ by a single formula. This idea is made precise in the following main result from which we can directly infer the two theorems from above.

**Theorem 4.21.** *For every prime $q$ there is a class of structures $\mathcal{K}_q$ such that*

*(a)* $\mathrm{FPS}_\Omega = \mathrm{FPC}$ *on $\mathcal{K}_q$ for every set of primes $\Omega$ with $q \notin \Omega$,*

*(b)* $\mathrm{FPR}_\Omega = \mathrm{FPS}_\Omega$ *on $\mathcal{K}_q$ for all sets of primes $\Omega$,*

*(c)* $\mathrm{FPC} < \textsc{Ptime}$ *on $\mathcal{K}_q$, and*

*(d)* $\mathrm{FPS}_q = \textsc{Ptime}$ *on $\mathcal{K}_q$.*

*Proof of Theorem 4.19.* Let $\Omega$ and $\Omega'$ be two distinct sets of primes. Without loss of generality let us assume that there exists a prime $q \in \Omega \setminus \Omega'$. Then by Theorem 4.21 there exists a class $\mathcal{K}_q$ on which $\mathrm{FPS}_\Omega = \mathrm{FPR}_\Omega = \textsc{Ptime}$ and on which $\mathrm{FPS}_{\Omega'} = \mathrm{FPR}_{\Omega'} = \mathrm{FPC} < \textsc{Ptime}$. □

*Proof of Theorem 4.20.* Assume that $\mathrm{FPR} = \textsc{Ptime}$. Then, in particular, $\mathrm{FPR} = \mathrm{FPR}^*$ and there exists a formula $\varphi \in \mathrm{FPR}$ which can uniformly determine the rank of matrices over prime fields, i.e. which can express the uniform rank operator $\mathsf{rk}$. As a matter of fact we have $\varphi \in \mathrm{FPR}_\Omega$ for some *finite* set of primes $\Omega$. By using $\varphi$ we can uniformly express the matrix rank

over each prime field $\mathbb{F}_p$ in $\mathrm{FPR}_\Omega$. In other words, we have $\mathrm{FPS} \leq \mathrm{FPR} \leq \mathrm{FPR}^* \leq \mathrm{FPR}_\Omega$.

Now let $q \in \mathbb{P} \setminus \Omega$. By Theorem 4.21 there exists a class of structures $\mathcal{K}_q$ on which $\mathrm{FPR}_\Omega = \mathrm{FPC} < \textsc{Ptime}$. However, the class $\mathcal{K}_q$ can be chosen such that $\textsc{Ptime} = \mathrm{FPS}_q \leq \mathrm{FPR}_\Omega$ on $\mathcal{K}_q$ by Theorem 4.21 (d) and we obtain the desired contradiction. □

The proof of Theorem 4.20 reveals the already mentioned deficiency in the definition of FPR: each formula can only access $\mathsf{rk}_p$-operators for a *finite* set of primes $\Omega \subseteq \mathbb{P}$. This suggests to generalise the notion of rank operators and to specify the prime $p$ as a part of their input. This uniform version of rank operators has previously been proposed, for example, in [59, 71, 80].

The remainder of this section is devoted to the proof of Theorem 4.21. We fix a prime $q$ and proceed as follows. In a first step, we identify properties of classes of structures $\mathcal{K}_q$ which guarantee that the relations claimed in (a), (b), (c) and (d) hold. In a second step, we proceed to show that we can obtain a class of structures $\mathcal{K}_q$ that satisfies all of these sufficient criteria. This together then proves our theorem.

### 4.4.1 Reducing rank operators to solvability quantifiers

We start by establishing sufficient criteria for the most relevant part of Theorem 4.21, which is the relation claimed in (a). Assume that we have a class of structures $\mathcal{K}_q = \mathcal{K}$ with the following properties.

(I) The groups $\Delta_\mathfrak{A} := \mathrm{Aut}(\mathfrak{A})$ of structures $\mathfrak{A} \in \mathcal{K}$ are Abelian $q$-groups.

(II) The orbits of $\ell$-tuples in structures $\mathfrak{A} \in \mathcal{K}$ can be ordered in FPC.

Formally, for each $\ell \geq 1$ there is a formula $\varphi_\preceq(x_1, \dots, x_\ell, y_1, \dots, y_\ell) \in \mathrm{FPC}$ such that for every structure $\mathfrak{A} \in \mathcal{K}$, the formula $\varphi_\preceq(\bar{x}, \bar{y})$ defines in $\mathfrak{A}$ a linear preorder $\preceq$ on $A^\ell$ with the property that two $\ell$-tuples $\bar{a}, \bar{b} \in A^\ell$ are $\preceq$-equivalent if, and only if, they are in the same $\Delta_\mathfrak{A}$-orbit.

**Lemma 4.22.** *If $\mathcal{K}$ satisfies (I) and (II), then $\mathrm{FPS}_\Omega = \mathrm{FPC}$ on $\mathcal{K}$ for all sets of primes $\Omega$ with $q \notin \Omega$.*

The proof of this lemma is by induction on the structure of $\mathrm{FPS}_\Omega$-formulas. Obviously, the only interesting step is the translation of a solvability formula

$$\psi(\bar{z}) = (\mathsf{slv}_p\, \bar{x}\bar{\nu} \leq \bar{s}, \bar{y}\bar{\mu} \leq \bar{t})\varphi(\bar{x}\bar{\nu}, \bar{y}\bar{\mu}, \bar{z})$$

into an FPC-formula $\vartheta(\bar{z})$ which is equivalent to $\psi(\bar{z})$ on the class $\mathcal{K}$. Let $|\bar{x}| = |\bar{y}| = \ell$, $|\bar{\nu}| = |\bar{\mu}| = \lambda$ and $|\bar{z}| = k$. To explain our main argument, we fix a structure $\mathfrak{A} \in \mathcal{K}$ and a $k$-tuple of parameters $\bar{c} \in (A \uplus \mathbb{N})^k$ which is compatible with the type of the variable tuple $\bar{z}$. According to the semantics of the solvability quantifier, the formula $\varphi$ defines in $(\mathfrak{A}, \bar{z} \mapsto \bar{c})$ an $I \times J$-matrix

$M = M^{\mathfrak{A}}_{\bar{c}}$ with entries in $\{0,1\} \subseteq \mathbb{F}_p$ where $I = I^{\mathfrak{A}}_{\bar{c}} := A^\ell \times \mathbb{N}^{\leq \bar{s}} \subseteq A^\ell \times \mathbb{N}^\lambda$ and $J = J^{\mathfrak{A}}_{\bar{c}} := A^\ell \times \mathbb{N}^{\leq \bar{t}} \subseteq A^\ell \times \mathbb{N}^\lambda$ that is defined for $\bar{a} \in I$ and $\bar{b} \in J$ as

$$M(\bar{a},\bar{b}) = \begin{cases} 1, & \text{if } \mathfrak{A} \models \varphi(\bar{a},\bar{b},\bar{c}) \\ 0, & \text{else.} \end{cases}$$

By definition we have $\mathfrak{A} \models \psi(\bar{c})$ if, and only if, the linear equation system $M \cdot \vec{x} = \mathbb{1}$ over $\mathbb{F}_p$ is solvable. Similar to our approach in Section 4.3, the key idea is to use the symmetries of the structure $\mathfrak{A}$ to translate the linear equation system $M \cdot \vec{x} = \mathbb{1}$ into an equivalent linear system which is *simpler* in the sense that its solvability can be defined in the logic FPC. The reader should observe that each automorphism $\pi \in \Delta_{\mathfrak{A}} = \mathrm{Aut}(\mathfrak{A})$ naturally induces an automorphism of the two-sorted extension $\mathfrak{A}^\#$ which point-wise fixes every number $n \in \mathbb{N}$. In particular we have $\mathrm{Aut}(\mathfrak{A}) = \mathrm{Aut}(\mathfrak{A}^\#)$.

We set $\Gamma = \Gamma^{\mathfrak{A}}_{\bar{c}} := \mathrm{Fix}(c_1,\ldots,c_k) \leq \Delta = \Delta_{\mathfrak{A}}$. In other words, we have that $\Gamma = \mathrm{Aut}(\mathfrak{A},\bar{c})$. The group $\Gamma$ acts on $I$ and $J$ in the natural way. We identify each automorphism $\pi \in \Gamma$ with the corresponding $I \times I$-permutation matrix $\Pi_I$ and the corresponding $J \times J$-permutation matrix $\Pi_J$ in the same way as we did in Section 4.3. Again for every $\pi \in \Gamma$ we have $\Pi_I \cdot M = M \cdot \Pi_J$, which leads to the following important observation.

**Lemma 4.23.** *If $M \cdot \vec{x} = \mathbb{1}$ is solvable, then the system has a $\Gamma$-symmetric solution, that is a solution $\vec{b} \in \mathbb{F}_p^J$ such that $\Pi_J \cdot \vec{b} = \vec{b}$ for all $\pi \in \Gamma$.*

*Proof.* If $M \cdot \vec{b} = \mathbb{1}$, then also $\Pi_I \cdot (M \cdot \vec{b}) = \mathbb{1}$ and thus $M \cdot (\Pi_J \cdot \vec{b}) = \mathbb{1}$ for all $\pi \in \Gamma$. This shows that $\Gamma$ acts on the solution space of the linear equation system. Since $\mathcal{K}$ satisfies property (I) we know that $\Gamma$ is a $q$-group for a prime $q \neq p$. Thus each $\Gamma$-orbit has size $q^r$ for some $r \geq 0$. On the other hand, the number of solutions is a power of $p$. We conclude that there is at least one $\Gamma$-orbit of size one which proves our claim. □

Let $\vec{b} \in \mathbb{F}_p^J$ be a $\Gamma$-symmetric solution. Then the entries of the solution $\vec{b}$ on $\Gamma$-orbits are constant: for $j \in J$ and $\pi \in \Gamma$ we have $\vec{b}(\pi(j)) = (\Pi_J \cdot \vec{b})(j) = \vec{b}(j)$. We proceed to use the property (II) and show that there exists an FPC-formula $\varphi_\preceq(\bar{x},\bar{y})$ which defines for all $\mathfrak{A} \in \mathcal{K}$ and $\bar{c} \in (A \uplus \mathbb{N})^k$ as above a linear preorder $\preceq$ on $A^\ell$ which identifies $\Gamma$-orbits. Note that, in general, $\Gamma = \mathrm{Aut}(\mathfrak{A},\bar{c})$ is a strict subgroup of $\Delta = \mathrm{Aut}(\mathfrak{A})$. Thus we can not directly apply (II). However, the $\Gamma$-orbits on $A^\ell$ correspond to the $\Delta$-orbits on $A^{k'+\ell}$ where the first $k'$ entries are fixed to the elements in $\{c_1,\ldots,c_k\} \cap A$.

The linear preorder $\preceq$ naturally extends to a preorder on the sets $I$ and $J$ with the same properties. Let us write $J = J_0 \preceq J_1 \preceq \cdots \preceq J_{v-1}$ to denote the decomposition of $J$ into $\Gamma$-orbits $J_j$ which are ordered by $\preceq$ as indicated and where $v \geq 1$ denotes the number different $\Gamma$-orbits. Moreover, for $j \in [v]$ we let $e_j$ denote the identity vector on the $j$-th orbit $J_j$, that is the $J$-vector which is

defined for $i \in J$ as

$$e_j(i) := \begin{cases} 1, & \text{if } i \in J_j \\ 0, & \text{else.} \end{cases}$$

Let $E$ denote the $J \times [v]$-matrix whose $j$-th column is the vector $e_j$. It follows that a $\Gamma$-symmetric solution $\vec{b}$ can be written as $E \cdot \vec{b}_* = \vec{b}$ for a unique $[v]$-vector $\vec{b}_*$. Together with Lemma 4.23 this shows the following.

**Lemma 4.24.** *The linear equation system $M \cdot \vec{x} = \mathbb{1}$ is solvable if, and only if, the linear equation system $(M \cdot E) \cdot \vec{x}_* = \mathbb{1}$ is solvable.*

Finally, we observe that the coefficient matrix $M_* := (M \cdot E)$ of the equivalent linear equation system $M_* \cdot \vec{x}_* = \mathbb{1}$ can easily be obtained in FPC and that it is a matrix over the *ordered* set of column indices $[v]$. It is a simple observation that such linear equation systems can be solved in FPC: the linear order on the column set induces (together with some fixed order on $\mathbb{F}_p$) a lexicographical ordering on the set of rows which is, up to duplicates of rows, a linear order on this set. Thus, in general, if we have a linear order on *one* of the index sets of the coefficient matrix, this suffices to obtain an equivalent matrix where *both* index sets are ordered, see also [80]. This finishes our proof of Lemma 4.22.

We proceed to show that the conditions (I) and (II) also guarantee that rank operators can be reduced to solvability quantifiers over the class $\mathcal{K}$. In fact, for this translation we only require the somewhat weaker assumption that we can define in FPC on $\ell$-tuples in structures $\mathfrak{A} \in \mathcal{K}$ a linear preorder in which every class can be totally ordered in FPC by fixing a constant number of parameters. The precise technical requirements will become clear from the proof of the following lemma.

**Lemma 4.25.** *If $\mathcal{K}$ satisfies (I) and (II), then $\mathrm{FPR}_\Omega = \mathrm{FPS}_\Omega$ on $\mathcal{K}$ for all sets of primes $\Omega \subseteq \mathbb{P}$.*

*Proof.* We translate $\mathrm{FPR}_\Omega$-formulas into formulas of $\mathrm{FPS}_\Omega$ which are equivalent on $\mathcal{K}$. The interesting case is the transformation of rank terms

$$\Upsilon(\bar{z}) = [\mathsf{rk}_p\,(\bar{x}\bar{\nu} \le \bar{t}, \bar{y}\bar{\mu} \le \bar{s})\,.\,\Theta(\bar{x}\bar{\nu}, \bar{y}\bar{\mu}, \bar{z})].$$

Let $|\bar{x}| = |\bar{y}| = \ell$, $|\bar{\nu}| = |\bar{\mu}| = \lambda$ and $|\bar{z}| = k$. Let $\mathfrak{A} \in \mathcal{K}$ and let $\bar{c}$ be a $k$-tuple of parameters $\bar{c} \in (A \uplus \mathbb{N})^k$ which is compatible with the type of the variable tuple $\bar{z}$. The term $\Theta$ defines in $(\mathfrak{A}, \bar{z} \mapsto \bar{c})$ for $I^{\mathfrak{A}} = I := A^{|\bar{x}|} \times \mathbb{N}^{\le \bar{t}}$ and $J^{\mathfrak{A}} = J := A^{|\bar{y}|} \times \mathbb{N}^{\le \bar{s}}$ the $I \times J$-matrix $M$ over $\mathbb{Z}_p$ which is defined as

$$M(\bar{a}\bar{n}, \bar{b}\bar{m}) := \Theta^{\mathfrak{A}}(\bar{a}\bar{n}, \bar{b}\bar{m}, \bar{c}) \bmod p.$$

According to the semantics of matrix rank operators, the value $\Upsilon^{\mathfrak{A}}(\bar{c}) \in \mathbb{N}$ is the rank of the matrix $M$. We proceed to show that we can determine the matrix rank of $M$ by a recursive application of solvability queries. To this end we make the following key observation.

**Claim:** There exist FPC-formulas $\varphi_\preceq(\bar{y}_1\bar{\mu}_1, \bar{y}_2\bar{\mu}_2)$, $\psi_\preceq(\bar{v}, \bar{y}_1\bar{\mu}_1, \bar{y}_2\bar{\mu}_2)$ such that for every structure $\mathfrak{A} \in \mathcal{K}$

(a) $\varphi_\preceq^{\mathfrak{A}}$ is a linear preorder $\preceq$ on $J^{\mathfrak{A}}$, and such that

(b) for every $\preceq$-class $[j] \subseteq J^{\mathfrak{A}}$ there exists a parameter tuple $\bar{d} \in A^{|\bar{v}|}$ such that $\psi_\preceq^{\mathfrak{A}}(\bar{d})$ is a linear order $\leq$ on $[j]$.

*Proof of claim:* First of all, we let $\varphi_\preceq$ be an FPC-formula which defines in every structure $\mathfrak{A} \in \mathcal{K}$ a linear preorder $\preceq$ on $J^{\mathfrak{A}}$ such that $\preceq$-classes correspond to $\Delta_{\mathfrak{A}}$-orbits. Such a formula exists by our assumption that $\mathcal{K}$ satisfies property (II). Analogously, we choose an FPC-formula $\vartheta_\preceq$ which defines in every structure $\mathfrak{A} \in \mathcal{K}$ a linear preorder $\preceq^*$ on $J^{\mathfrak{A}} \times J^{\mathfrak{A}}$ that induces a linear order on the $\Delta_{\mathfrak{A}}$-orbits.

Now let $[j] \subseteq J^{\mathfrak{A}}$ be a $\preceq$-class for some $\mathfrak{A} \in \mathcal{K}$. By property (I) we know that $\Delta_{\mathfrak{A}}$ is an Abelian group. Thus, each automorphism $\pi \in \Delta_{\mathfrak{A}}$ which fixes *one* element in the $\Delta_{\mathfrak{A}}$-orbit $[j]$ point-wise fixes *every* element in the class $[j]$. We conclude that the restriction of $\preceq^*$ to elements in $\{j\} \times [j]$ corresponds to a linear order on $[j]$ for each $j \in [j]$. In this way we obtain an FPC-formula $\psi_\preceq$ with the desired properties. ⊣

We are now prepared to describe the recursive procedure which allows us to determine the rank of the matrix $M$ in $\mathrm{FPS}_\Omega$. To this end we fix formulas $\varphi_\preceq$ and $\psi_\preceq$ with the above properties. Moreover, let $\preceq$ denote the linear preorder defined by $\varphi_\preceq$ on $J$ and let $J = J_0 \preceq J_1 \preceq \cdots \preceq J_{r-1}$. We use the formula $\psi_\preceq$ to obtain on each class $J_i$ a family of definable linear orderings (which depend on the choice of different parameters). For $j \in J$ we denote by $\vec{m}_j \in \mathbb{F}_q^I$ the $j$-th column of the matrix $M$. Then the rank of $M$ is the dimension of the $\mathbb{F}_p$-vector space which is generated by the set of columns $\{\vec{m}_j : j \in J\}$ of the matrix $M$.

The important step is to recursively obtain, for $i \in [r]$, the dimension $d_i \in \mathbb{N}$ of the $\mathbb{F}_p$-vector space generated by

$$V_i := \{\vec{m}_j : j \in J_0 \cup J_1 \cup \cdots \cup J_i\}.$$

First, we use $\psi_\preceq$ to fix a linear order on $J_i$ (the following steps are independent of the specific linear order and can thus be performed in parallel for each such order). Using this linear order on $J_i$ we can identify in $\mathrm{FPS}_\Omega$ a maximal set $W \subseteq \{\vec{m}_j : j \in J_i\}$ of linearly independent columns such that $\langle V_{i-1}\rangle \cap \langle W\rangle = \{\vec{0}\}$. Indeed, if $\langle V_{i-1}\rangle \cap \langle W\rangle = \{\vec{0}\}$, then for $\vec{m} \in \{\vec{m}_j : j \in J_i\}$, $\vec{m} \notin \langle W\rangle$ we have that $\langle V_{i-1}\rangle \cap \langle W \cup \{\vec{m}\}\rangle = \{\vec{0}\}$ if, and only if, $\vec{m} \notin \langle V_{i-1} \cup W\rangle$. Observe that the conditions $\vec{m} \notin \langle W\rangle$ and $\vec{m} \notin \langle V_{i-1} \cup W\rangle$ correspond to the solvability of a linear equation system over $\mathbb{F}_p$. We claim that $d_i = d_{i-1} + |W|$. Indeed, by the maximality of $W$ and since $\langle V_i\rangle \cap \langle W\rangle = \{\vec{0}\}$ it follows that $\langle V_i\rangle = \langle V_{i-1}\rangle \oplus \langle W\rangle$. Moreover, $W$ consists of linearly independent columns and is a basis for $\langle W\rangle$.

Since the recursion described above can easily be implemented in $\mathrm{FPS}_\Omega$, we conclude that the rank $d_{r-1}$ of the matrix $M$ can be determined in $\mathrm{FPS}_\Omega$ which completes our proof. □

We now focus on parts (c) and (d) of Theorem 4.21 and establish sufficient criteria which guarantee that FPC fails to capture PTIME on $\mathcal{K}$ while $\mathrm{FPS}_q$ can express every polynomial-time decidable property of $\mathcal{K}$-structures.

(III) There exists an $\mathrm{FPS}_q$-definable canonisation procedure on $\mathcal{K}$.

(IV) For every $k \geq 1$ there exists a pair of structures $\mathfrak{A} \in \mathcal{K}$ and $\mathfrak{B} \in \mathcal{K}$ such that $\mathfrak{A} \not\cong \mathfrak{B}$ and $\mathfrak{A} \equiv_k^C \mathfrak{B}$.

**Lemma 4.26.** *If $\mathcal{K}$ satisfies (III), (IV), then* $\mathrm{FPC} < \mathrm{FPS}_q = \mathrm{PTIME}$ *on $\mathcal{K}$.*

*Proof.* It is clear that by property (III) we have $\mathrm{FPS}_q = \mathrm{PTIME}$ on $\mathcal{K}$. Moreover, if we had $\mathrm{FPC} = \mathrm{PTIME}$ on $\mathcal{K}$ then, by the embedding of FPC into $\mathrm{C}^\omega_{\infty\omega}$ and the fact that $\mathcal{K}$-structures can be canonised in polynomial time, there exists a fixed $k \geq 1$ such that $\mathrm{C}^k_{\infty\omega}$ can identify each structure in $\mathcal{K}$ which, in turn, contradicts property (IV). □

### 4.4.2 A generalised Cai, Fürer, Immerman construction

It remains to construct a class of structures $\mathcal{K}$ which satisfies (I) - (IV). Our approach is a generalisation of the well-known construction of Cai, Fürer, and Immerman [21] for cyclic groups other than $\mathbb{F}_2$. To illustrate the main differences, let us briefly recall the idea of the original construction. Starting with an undirected and connected graph $\mathcal{G} = (V, E)$, we first take two copies $e_0, e_1$ of every edge $e \in E$ for the universe of the associated CFI-graph. For every vertex $v \in V$ we let $vE \subseteq E$ denote the set of edges which are incident with $v$. The crucial idea of the CFI-construction is to consider, for every vertex $v \in V$, one of the following two constraints to restrict the symmetries of the resulting CFI-graph: either the set of all sets $\{e_{\rho(e)} : e \in vE\}$ with $\rho : vE \to \mathbb{F}_2$ and $\sum_{e \in vE} \rho(e) = 0$ is stabilised (an *even* node) or the dual set of all sets $\{e_{\rho(e)} : e \in vE\}$ with $\rho : vE \to \mathbb{F}_2$ and $\sum_{e \in vE} \rho(e) = 1$ is stabilised (an *odd* node). This restricts the symmetries of the resulting CFI-graphs (which are obtained by twisting the atoms $e_0, e_1$ for edges $e \in E$) in a very clever way.

The constraints for even and odd nodes are encoded by simple graph gadgets. Although it seems that for the same undirected graph $\mathcal{G}$ we obtain exponentially many different CFI-graphs (for each $v \in V$ we can choose one out of two possible constraints), there really are, up to isomorphism, only two such graphs which are determined by the parity of the number of odd nodes. The reason is that if we twist two copies $e_0, e_1$ of an edge $e$, then we can move the resulting twist along a path (in the *connected* graph $\mathcal{G}$) to iteratively balance out pairs of odd nodes.

In order to generalise this construction to $\mathbb{F}_q$ we take for every edge $e \in E$ a *directed cycle* of length $q$ over $q$ copies $e_0, e_1, \ldots, e_{q-1}$ of the edge $e$. We then add similar constraints for sets of incident edges as above, but instead of having only two different kinds of such constraints, we have one for each of the possible field elements $0, 1, \ldots, q-1 \in \mathbb{F}_q$. Now, instead of twisting pairs

of edges, we consider *cyclic shifts* on the edge classes $e_0, e_1, \dots, e_{q-1}$. Again, these shifts can be propagated along paths in the original graph $\mathcal{G}$ and, with a reasoning analogous to the original approach, it turns out that there are, up to isomorphism, only $q$ different types of generalised CFI-graphs over $\mathbb{F}_q$. We remark that the same kind of construction has been used, for example, in [59, 86].

Formally, we start with an *(undirected), connected* and *ordered* graph $\mathcal{G} = (V, \leq, E)$. We set $\tau := \{\preceq, C, I, R\}$ for binary relation symbols $C, I$ and $R$. We define, for every $q \in \mathbb{P}$, and for every sequence of *gadget values* $\vec{d} = (d_v)_{v \in V} \in [q]^V$, a $\tau$-structure $\mathrm{CFI}_q(\mathcal{G}, \vec{d})$ which we call a *CFI-structure over* $\mathcal{G}$. For the following construction we implicitly assume that arithmetic is modulo $q$ so that we can drop the operator "mod $q$" in statements of the form $x = y \bmod q$ and $x + y \bmod q$ for the sake of better readability. For what follows, let $E(v) \subseteq E$ denote the set of *directed* edges starting in $v$. Since $\mathcal{G}$ is an undirected graph, this means that for an undirected edge $\{v, w\}$ of $\mathcal{G}$ we have $(v, w) \in E(v)$ and $(w, v) \in E(w)$. The construction is illustrated in Figure 4.3.

- The *universe* of $\mathrm{CFI}_q(\mathcal{G}, \vec{d})$ consists of *edge nodes* and *equation nodes*.
  - The set of *edge nodes* $\hat{E}$ is defined as $\hat{E} := \bigcup_{e \in E} \hat{e}$ where for every *directed* edge $e \in E$ we let the *edge class* $\hat{e} = \{e_0, e_1, \dots, e_{q-1}\}$ consist of $q$ distinct copies of $e$. In particular, for every edge $e = (v, w) \in E$ and its reversed edge $e^{-1} := f = (w, v) \in E$ the sets $\hat{e}$ and $\hat{f}$ are disjoint. We say that the two such edges $e$ and $f$ (or the associated edge classes $\hat{e}$ and $\hat{f}$) are *related*.
  - The set of *equation nodes* $\hat{V}$ is defined as $\hat{V} := \bigcup_{v \in V} \hat{v}^{\vec{d}(v)}$ where for every vertex $v \in V$ and $d \in [q]$ the *equation class* $\hat{v}^d$ consist of all functions $\rho : E(v) \to [q]$ which satisfy $\sum \rho := \sum_{e \in E(v)} \rho(e) = d$.
- The *linear preorder* $\preceq$ orders the edge classes according to the lexicographical order induced by $\leq$ on $E$. More precisely, we let $\hat{e} \preceq \hat{f}$ whenever $e \leq f$. Similarly, $\preceq$ orders the equation classes according to the order of $\leq$ on $V$, that is $\hat{v} \preceq \hat{w}$ if $v \leq w$. Moreover, we let $\hat{e} \preceq \hat{v}$ for edge classes $\hat{e}$ and equation classes $\hat{v}$.
- The *cycle relation* $C$ contains a directed cycle of length $q$ on each of the edge classes $\hat{e}$ for $e \in E$, that is $C = \{(e_i, e_{i+1}) : i \in [q], e \in E\}$.
- The *inverse relation* $I$ connects two related edge classes by pairing additive inverses. More precisely, let $e = (v, w) \in E$ and $f = (w, v) \in E$. Then $I$ contains all edges $(e_x, f_y)$ with $x + y = 0$ for $x, y \in [q]$.
- The *gadget relation* $R$ is defined as $R := \bigcup_{v \in V} R_v^{\vec{d}(v)}$ where for $v \in V$ and $d \in [q]$ the relation $R_v^d$ is given as

$$R_v^d := \{(\rho, e_{\rho(e)}) : \rho \in \hat{v}^d, e \in E(v)\}.$$

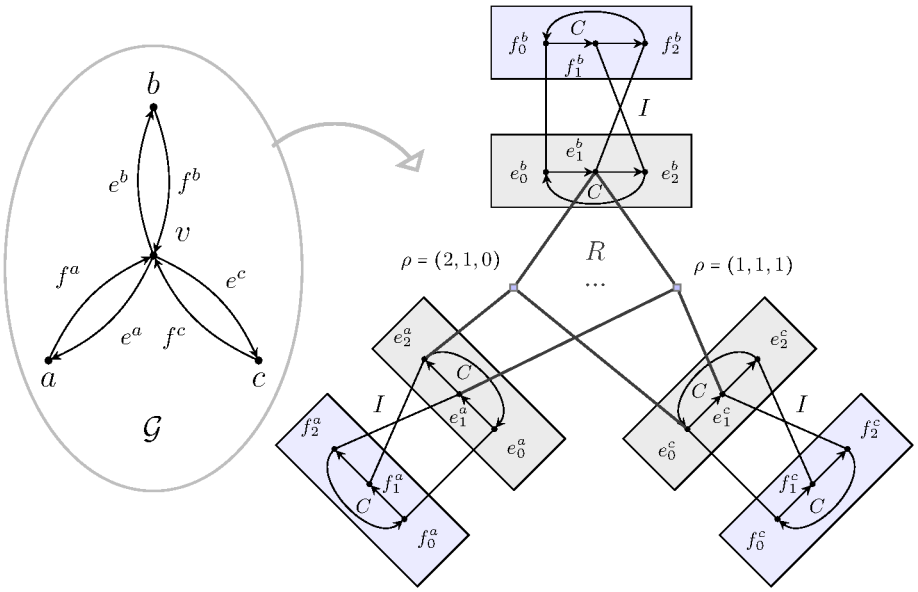


Figure 4.3: CFI-construction for the $v$-gadget where $q = 3$ and $\vec{d}(v) = 0$

At first glance our construction associates to every graph $\mathcal{G}$ (with the above properties) and to each sequence of gadget values $\vec{d} \in [q]^V$ a different structure $\mathrm{CFI}_q(\mathcal{G}, \vec{d})$. However, for each such graph $\mathcal{G}$ there really are, up to isomorphism, only $q$ different CFI-structures $\mathrm{CFI}_q(\mathcal{G}, \vec{d})$. In fact, the value $\sum \vec{d} := \sum_{v \in V} \vec{d}(v)$ completely determines the isomorphism class of a CFI-structure over $\mathcal{G}$.

To obtain this characterisation, we analyse the automorphism groups of CFI-structures and, more generally, the set of isomorphisms between two structures $\mathfrak{A} = \mathrm{CFI}_q(\mathcal{G}, \vec{d}_1)$ and $\mathfrak{B} = \mathrm{CFI}_q(\mathcal{G}, \vec{d}_2)$. For such structures we know that the set $\hat{E}$ of edge nodes, the linear preorder $\preceq$ on $\hat{E}$, the cycle relation $C$ and the inverse relation $I$ do not depend on the sequence of gadget values. This means that each possible isomorphism $\pi$ which maps $\mathfrak{A}$ to $\mathfrak{B}$ induces an automorphism of the common substructure $\mathfrak{C} := (\hat{E}, (\preceq \restriction \hat{E}), C, I)$ which only depends on $\mathcal{G}$ but not on $\vec{d} \in [q]^V$. Thus

$$(\mathrm{Iso}(\mathfrak{A}, \mathfrak{B}) \restriction \hat{E}) \subseteq \Gamma := \mathrm{Aut}(\mathfrak{C}) \leq \mathrm{Sym}(\hat{E}).$$

Let $\pi \in \Gamma$. The linear preorder $\preceq$ on $\hat{E}$ and the cycle relation $C$ enforce that $\pi$ is the composition of cyclic shifts on the individual edge classes $\hat{e}$, that is $\pi \in \prod_{e \in E} \langle (\, e_0 e_1 \cdots e_{q-1} \,) \rangle \leq \mathrm{Sym}(\hat{E})$. It is convenient to identify the group $\prod_{e \in E} \langle (\, e_0 e_1 \cdots e_{q-1} \,) \rangle$ with the vector space $\mathbb{F}_q^E$ in the obvious way.

In addition, the inverse relation $I$ enforces that cyclic shifts for pairs of related edge classes are inverse to each other in the following sense: Let $e = (v, w) \in E$ and $f = (w, v) \in E$ be a pair of related edges. Assume that we have a permutation $\pi \in \mathbb{F}_q^E$ such that $\pi(e) = x$ and $\pi(f) = y$. We have $(e_0, f_0) \in I$. Hence, if $\pi$ is supposed to be an automorphism of $\mathfrak{C}$, then we have $\pi(I) = I$ and thus $(e_x, e_y) \in I$ which means that $x + y = 0$.

In conclusion, it follows that $\Gamma \leq \mathbb{F}_q^E$ is the subgroup of $\mathbb{F}_q^E$ which contains all $E$-vectors $\pi \in \mathbb{F}_q^E$ with the property that $\pi(e) + \pi(f) = 0$ for pairs of related edges $e, f \in E$. Again we remind the reader that $\Gamma$ only depends on $\mathcal{G}$ but not on $\vec{d} \in [q]^V$. If we want to stress this dependence, then we sometimes write $\Gamma(\mathcal{G})$ but usually we omit $\mathcal{G}$ if the graph is clear from the context.

Now, given a CFI-structure $\mathfrak{A} = \mathrm{CFI}_q(\mathcal{G}, \vec{d})$, we define for each vertex $v \in V$ the *$v$-gadget* as the set $\mathrm{gadget}(v) := \hat{v}^{d(v)} \uplus \bigcup_{e \in E(v)} \hat{e}$.

**Lemma 4.27.** *Let $\mathfrak{A} = \mathrm{CFI}_q(\mathcal{G}, \vec{d})$ and let $\pi \in \Gamma$. Then there is precisely one extension $\hat{\pi}$ of $\pi$ to $\hat{E} \uplus \hat{V}$ such that $\hat{\pi}(\mathfrak{A})$ is a CFI-structure over $\mathcal{G}$.*

*Proof.* Let $\rho \in \hat{v} = \hat{v}^{\vec{d}(v)}$ for some $v \in V$. We show that under the assumption that $\hat{\pi}(\mathfrak{A})$ is a CFI-structure over $\mathcal{G}$ the action of $\pi$ on $\hat{E}$ determines $\hat{\pi}(\rho)$.

We have that $(\rho, e_{\rho(e)}) \in R$ for all $e \in E(v)$. Hence for a potential isomorphism $\hat{\pi}$ we must have that $(\hat{\pi}(\rho), \pi(e_{\rho(e)})) \in R'$ (for some gadget relation $R'$ of a CFI-structure over $\mathcal{G}$). Since we have $\pi(e_{\rho(e)}) = e_{\rho(e)+\pi(e)}$, it follows by the definition of CFI-structures that the function $\hat{\pi}(\rho) : E(v) \to [q]$ is determined as $(\hat{\pi}(\rho))(e) = \rho(e) + \pi(e)$ which in turn only depends on the action of $\pi$ on the edge classes $\hat{e}$ for $e \in E(v)$. □

The preceding lemma shows that we can identify the set $\mathrm{Iso}(\mathfrak{A}, \mathfrak{B})$ with a subset of $\Gamma$. More specifically, the set $\mathrm{Aut}(\mathfrak{A})$ turns out to be a subgroup of $\Gamma$ of which $\mathrm{Iso}(\mathfrak{A}, \mathfrak{B})$ is a coset in $\Gamma$. Specifically, we saw that every $\pi \in \Gamma$ can uniquely be identified with an isomorphism of CFI-structures over $\mathcal{G}$ by setting $\pi(\rho) = \rho + \pi$ for $\rho \in \hat{v}^d$ . As a consequence, this means that $\pi(\hat{v}^d) = \hat{v}^{d_*}$ where $d_* = d + \sum_{e \in E(v)} \pi(e)$ and that

$$\pi(R_v^d) = \{(\rho + \pi, e_{\rho(e)+\pi(e)}) : (\rho, e_{\rho(e)}) \in R_v^d\} = R_v^{d_*}.$$

In particular, $\pi$ stabilises the relation $R_v^d$ if, and only if, $\sum_{e \in E(v)} \pi(e) = 0$.

**Lemma 4.28.** *$\Gamma$ acts on $\{\mathrm{CFI}_q(\mathcal{G}, \vec{d}) : \vec{d} \in [q]^V\}$. For $\pi \in \Gamma$ we have*

$$\pi(\mathrm{CFI}_q(\mathcal{G}, \vec{d})) = \mathrm{CFI}_q(\mathcal{G}, \vec{d}_*) \text{ where } \vec{d}_*(v) = (\vec{d}(v) + \sum_{e \in E(v)} \pi(e)).$$

**Lemma 4.29.** *Let $\vec{d}, \vec{d}_* \in [q]^V$ be two sequences of gadget values. Then $\mathrm{CFI}_q(\mathcal{G}, \vec{d}) \cong \mathrm{CFI}_q(\mathcal{G}, \vec{d}_*)$ if, and only if, $\sum \vec{d} = \sum \vec{d}_*$.*

*Proof.* For the one direction, let $\pi \in \Gamma$ such that $\pi(\mathrm{CFI}_q(\mathcal{G}, \vec{d})) = \mathrm{CFI}_q(\mathcal{G}, \vec{d}_*)$. By Lemma 4.28 this means that $\vec{d}_*(v) = (\vec{d}(v) + \sum_{e \in E(v)} \pi(e))$ for $v \in V$. Thus $\sum_{v \in V} \vec{d}_*(v) = \sum_{v \in V} \vec{d}(v) + \sum_{v \in V} \sum_{e \in E(v)} \pi(e) = \sum_{v \in V} \vec{d}(v) + \sum_{e \in E} \pi(e)$. Since for all pairs of related edges $e, f \in E$ we have $\pi(e) + \pi(f) = 0$, the claim follows.

For the other direction we proceed by induction on the number $i$ of vertices $v \in V$ such that $\vec{d}(v) \neq \vec{d}_*(v)$. If no such vertex exists, then the claim is trivial.

Otherwise, because of our assumption, there exist at least two such vertices $v, w \in V$, $v \neq w$. Since $\mathcal{G}$ is connected we find a simple path

$$\bar{p} : v = v_0 \xrightarrow{E} v_1 \xrightarrow{E} v_2 \xrightarrow{E} \cdots \xrightarrow{E} v_m = w$$

from $v$ to $w$ of length $m \geq 1$. Consider the following $E$-vector $\pi \in \mathbb{F}_q^E$ which is defined for $z := \vec{d}_*(v) - \vec{d}(v)$ as

$$\pi(e) := \begin{cases} z, & \text{if } e = (v_i, v_{i+1}), 0 \leq i < m \\ -z, & \text{if } e = (v_{i+1}, v_i), 0 \leq i < m \\ 0, & \text{else.} \end{cases}$$

By the definition of $\pi$ it follows that $\pi \in \Gamma$. Let $\pi(\mathrm{CFI}_q(\mathcal{G}, \vec{d})) = \mathrm{CFI}_q(\mathcal{G}, \vec{d}_+)$. We claim that the number of $v \in V$ such that $\vec{d}_+(v) \neq \vec{d}_*(v)$ is at most $i - 1$. From Lemma 4.28 we know that $\vec{d}_+(v) = \vec{d}(v) + \sum_{e \in E(v)} \pi(e)$. For $v \in V$ it follows that

- if $v \notin \{v_0, \ldots, v_m\}$, then $\vec{d}_+(v) = \vec{d}(v)$, and
- if $v = v_0$, then $\vec{d}_+(v) = \vec{d}(v) + z = \vec{d}_*(v)$, and
- if $v = v_j$ for $1 \leq j < m$, then

$$\vec{d}_+(v) = \vec{d}(v) + \pi(v_j, v_{j-1}) + \pi(v_j, v_{j+1}) = \vec{d}(v) - z + z = \vec{d}(v), \text{ and}$$

- if $v = v_m$, then $\vec{d}_+(v) = \vec{d}(v) - z$.

Thus the claim follows from the induction hypothesis. □

The kind of isomorphism that we constructed in the proof of Lemma 4.29 plays an important role later on. Thus, for a simple path $\bar{p}$ from $v_0$ to $v_m$ $(m \geq 1)$

$$\bar{p} : v = v_0 \xrightarrow{E} v_1 \xrightarrow{E} v_2 \xrightarrow{E} \cdots \xrightarrow{E} v_m = w,$$

and for $z \in \mathbb{F}_q$, we denote this isomorphism by $\pi[\bar{p}, z] \in \Gamma$. In other words, if we let $\sigma^z[e] \in \Gamma$ for $e \in E$ and $z \in \mathbb{F}_q$ denote the $E$-vector which is defined as

$$\sigma^z[e](f) = \begin{cases} z, & \text{if } f = e, \\ -z, & \text{if } f = e^{-1}, \\ 0, & \text{else,} \end{cases}$$

then $\pi[\bar{p}, z] = \sigma^z[(v_0, v_1)] + \sigma^z[(v_1, v_2)] + \cdots + \sigma^z[(v_{m-1}, v_m)]$. Intuitively, the isomorphism $\pi[\bar{p}, z]$ allows us to simultaneously increase the gadget value at $v_0$ by $z$ and to decrease the gadget value at $v_m$ by $z$ while the induced twists are moved along the path $\bar{p}$ through the gadget relations of the vertices $v_j$,

$1 \leq j < m$, whose gadget value does not change. A very important special case arises when $\bar{p}$ is a simple cycle of length $m \geq 3$

$$\bar{p} : v = v_0 \xrightarrow{E} v_1 \xrightarrow{E} v_2 \xrightarrow{E} \cdots \xrightarrow{E} v_m = v.$$

Then for all values $z \in \mathbb{F}_q$ the isomorphism $\pi[\bar{p}, z] \in \Gamma$ is an *automorphism* of CFI-structures over $\mathcal{G}$. We are going to use these automorphisms to show that it is possible to define in FPC an ordering on the orbits of $\ell$-tuples as required by property (II). It turns out that it suffices to ensure that the graph $\mathcal{G}$ is sufficiently connected.

Recall that a graph $\mathcal{G}$ is *$k$-connected*, for $k \geq 1$, if $\mathcal{G}$ contains more than $k$ vertices and if $\mathcal{G}$ stays connected when we remove any set of at most $k$ vertices. The *connectivity* $\mathrm{con}(\mathcal{G})$ of a graph $\mathcal{G}$ is the maximal $k \geq 1$ such that $\mathcal{G}$ is $k$-connected. Moreover, the *connectivity* $\mathrm{con}(\mathfrak{G})$ of a class $\mathfrak{G}$ of graphs is the function $\mathrm{con}(\mathfrak{G}) : \mathbb{N} \to \mathbb{N}$ defined by

$$n \mapsto \min_{\mathcal{G} \in \mathfrak{G}, |\mathcal{G}| = n} \mathrm{con}(\mathcal{G}).$$

We proceed to define the class $\mathcal{K}$: Let $\mathfrak{G}$ be a class of *undirected, ordered* graphs such that $\mathrm{con}(\mathfrak{G}) \in \omega(1)$. Then we set

$$\mathcal{K} = \mathcal{K}_q := \{\mathrm{CFI}_q(\mathcal{G}, \vec{d}) : \mathcal{G} = (V, \leq, E) \in \mathfrak{G}, \vec{d} \in [q]^V\}.$$

### 4.4.3 Orbits in generalised Cai, Fürer, Immerman structures

We proceed to show that $\mathcal{K}$ satisfies the required properties (I) - (IV). First of all, we saw that the automorphism group of each CFI-structure is an $\mathbb{F}_q$-vector space, so property (I) clearly holds for the class $\mathcal{K}$.

The proof that $\mathcal{K}$ satisfies property (II) is more involved. Let us fix the length $\ell \geq 1$ of tuples on which we want to define a linear preorder which identifies $\Delta_{\mathfrak{A}}$-orbits. By the choice of $\mathcal{K}$ it suffices to consider CFI-structures $\mathfrak{A} = \mathrm{CFI}_q(\mathcal{G}, \vec{d})$ over graphs $\mathcal{G} = (V, \leq, E)$ with $\mathrm{con}(\mathcal{G}) > (\ell + 2)$, since almost all structures in $\mathcal{K}$ satisfy this condition. As above let $\Gamma \leq \mathbb{F}_q^E$ denote the group that acts on the set of CFI-structures over $\mathcal{G}$ and let $A := (\hat{V} \uplus \hat{E})$ denote the universe of the CFI-structure $\mathfrak{A}$.

**Definition 4.30.** Let $\lambda \leq \ell$ and let $\bar{a} \in A^\lambda$.

(i) Let $v \in V$. We say that the vertex $v$ is *marked (given the parameters $\bar{a}$)* if for some $x \in \{a_1, \ldots, a_\lambda\}$ we have $x \in \hat{v}$ $(= \hat{v}^{\vec{d}(v)})$.

(ii) Let $e = (v, w) \in E$. We say that the edge $e$ is *marked (given the parameters $\bar{a}$)* if one of the vertices $v$ or $w$ is marked or if for some $x \in \{a_1, \ldots, a_\lambda\}$ we have that $x \in \hat{e} \cup \hat{f}$ where $f = (w, v) \in E$ is the edge related with $e$.

**Lemma 4.31.** *Let $\lambda \leq \ell$ and let $\bar{a} \in A^\lambda$.*

(a) *If $v \in V$ is marked, then the $v$-gadget can be identified in $\mathrm{C}^{\ell+2}_{\infty\omega}$ (using the parameters $\bar{a}$), that is for every $c \in gadget(v)$ there exists a formula $\vartheta(\bar{x}, y) \in \mathrm{C}^{\ell+2}_{\infty\omega}$ such that $\vartheta^{\mathfrak{A}}(\bar{a}) = \{c\}$.*

(b) *If an edge $e \in E$ is marked, then the edge classes $\hat{e}$ and $\hat{f}$ for $f = e^{-1}$ are identified in $\mathrm{C}^{\ell+2}_{\infty\omega}$ (given the parameters $\bar{a}$), that is for every $c \in \hat{e} \uplus \hat{f}$ there exists a formula $\vartheta(\bar{x}, y) \in \mathrm{C}^{\ell+2}_{\infty\omega}$ such that $\vartheta^{\mathfrak{A}}(\bar{a}) = \{c\}$.*

*Proof.* First of all, it is straightforward (even without using the parameters) to fix the $\preceq$-class of any element $c \in A$ in $\mathrm{C}^{\ell+2}_{\infty\omega}$. Secondly, observe that if an element $\rho \in \hat{v}$ is fixed, then we can fix an element in each of the edge classes $\hat{e}$ for $e \in E(v)$ since $\rho$ is $R$-connected to precisely one vertex in each of these classes. Moreover, if we have fixed an element $x \in \hat{e}$ in some edge class $\hat{e}$, then we can simply use the cycle relation $C$ to identify each element $c \in \hat{e}$ via its $C$-distance to $a$ in $\mathrm{C}^{\ell+2}_{\infty\omega}$. Finally, the inverse relation $I$ yields a definable bijection between related edge classes. □

**Lemma 4.32.** *Let $\lambda \leq \ell$, $\bar{a} \in A^{\lambda}$ and let $v \in V$ be a vertex that is not marked. Then for all edges $e, e' \in E(v)$, $e \neq e'$, which are not marked there exists $\pi \in Fix(\bar{a}) := Aut(\mathfrak{A}, \bar{a})$ such that $\pi(e) = -\pi(e') \neq 0$ and such that $\pi(f) = 0$ for all $f \in E(v) \setminus \{e, e'\}$.*

*Proof.* Let $e = (v, w)$ and $e' = (v, w')$ as above. Then the vertices $w$ and $w'$ are not marked.

Consider the graph $\mathcal{G}'$ that results from $\mathcal{G}$ by removing the vertex $v$ and each marked vertex $y \in V$. Let $V' \subseteq V$ denote the vertex set and $E' \subseteq E$ the edge relation of the graph $\mathcal{G}'$. Moreover, let $M := \{a_1, \dots, a_\lambda\} \cap (\bigcup_{e \in E} \hat{e})$. We observe that $|V| - |V'| \leq \lambda - |M| + 1$.

For every $x \in M$ there is an edge $f \in E$ such that $x \in \hat{f}$. For each such edge $f$ that is also contained in the subgraph $\mathcal{G}'$ we delete one of its endpoints but *neither the vertex $w$ nor the vertex $w'$* and denote the resulting subgraph by $\mathcal{G}''$ with vertex set $V'' \subseteq V'$ and edge relation $E'' \subseteq E'$. It still might be the case that there is a parameter $x \in M$ such that $x \in \hat{f}$ for $f \in E''$. However, then we know that $f$ connects $w'$ and $w$. Since we removed at most $(|V| - |V|') + |M| \leq \lambda + 1 \leq (\ell + 1)$ vertices from the graph $\mathcal{G}$ to obtain $\mathcal{G}''$ and since $\mathrm{con}(\mathcal{G}) > (\ell + 2)$, we know that there is a simple path of length $m \geq 2$ (i.e. the path does not consist of a single edge between $w$ and $w'$) which connects $w$ and $w'$ in $\mathcal{G}''$:

$$\bar{p} : w \xrightarrow{E''} v_1 \xrightarrow{E''} v_2 \xrightarrow{E''} \cdots \xrightarrow{E''} v_{m-1} \xrightarrow{E''} w'.$$

We extend the path $\bar{p}$ to a simple cycle $\bar{p}_c$ in $\mathcal{G}$ from $v$ to $v$ by using the edges $(v, w), (v, w') \in E$:

$$\bar{p}_c : v \xrightarrow{E} w \xrightarrow{E} v_1 \xrightarrow{E} v_2 \xrightarrow{E} \cdots \xrightarrow{E} v_{m-1} \xrightarrow{E} w' \xrightarrow{E} v.$$

Let $0 \neq z \in [q]$. We claim that $\pi := \pi[\bar{p}_c, z]$ satisfies the desired properties.

By the definition of $\pi$ it holds that $\pi(e) = z = -\pi(e')$. Let $x \in \{a_1, \dots, a_\lambda\}$. Then we have $x \notin \bigcup_{i=1}^{m-1} \hat{v}_i \cup \hat{w} \cup \hat{w}' \cup \hat{v}$, since none of the vertices $v$, $w$ and $w'$ is marked and since we removed any other marked vertex $y \in V$ from $\mathcal{G}$.

Moreover, for $f \in \{(v,w),(w,v),(v,w'),(w',v)\}$ we have that $x \notin \hat{f}$ by our assumption that $e, e'$ are not marked. Also for $f \in \{(w,v_1),(w',v_{m-1})\}$ we have $x \notin \hat{f}$, since otherwise we had removed the vertices $v_1$ and $v_{m-1}$ from $\mathcal{G}'$. Finally, for $f \in \bigcup_{i=1}^{m-2}\{(v_i,v_{i+1}),(v_{i+1},v_i)\}$ we have $x \notin \hat{f}$, since otherwise we had removed one of the endpoints of each such edge $f$ from $\mathcal{G}'$. Hence $\pi(x) = x$. Finally, since $v \notin V''$ we also have that $\pi(f) = f$ for all $f \notin E(v) \setminus \{e, e'\}$. □

**Lemma 4.33.** *Let $\lambda \leq \ell$ and let $\bar{a}, \bar{b} \in A^\lambda$. Then $(\mathfrak{A}, \bar{a}) \equiv^C_{\ell+2} (\mathfrak{A}, \bar{b})$ if, and only if, there exists $\pi \in Aut(\mathfrak{A})$ such that $\pi(\bar{a}) = \bar{b}$.*

*Proof.* We proceed by induction on the maximal position $1 \leq i \leq \lambda$ up to which the tuples $\bar{a}$ and $\bar{b}$ agree, that is such that for $1 \leq j < i$ we have $a_j = b_j$ and such that $a_i \neq b_i$. Let $a := a_i$ and $b := b_i$. Then we have to show that there exists an automorphism $\pi \in \mathrm{Fix}(a_1 \cdots a_{i-1}) = \mathrm{Aut}(\mathfrak{A}, a_1, \dots, a_{i-1})$ such that $\pi(a) = b$. Since $\bar{a}$ and $\bar{b}$ have the same $\mathrm{C}^{\ell+2}_{\infty\omega}$-type we know that $a$ and $b$ belong to the same $\preceq$-class. We choose $v \in V$ such that $a, b \in \mathrm{gadget}(v)$.

In what follows, whenever we speak of *marked vertices* or *marked edges* then we implicitly refer to a marking with respect to the already fixed part of parameters $\{a_1, \dots, a_{i-1}\}$.

Without loss of generality we may assume that the $v$-gadget is not marked by an element $x \in \{a_1, \dots, a_{i-1}\}$, because otherwise, by Lemma 4.31, every element in $\mathrm{gadget}(v)$ can uniquely be identified in $\mathrm{C}^{\ell+2}_{\infty\omega}$. We distinguish between the two cases where $a$ and $b$ are equation nodes and where $a$ and $b$ are edge nodes.

For the first case let $a, b \in \hat{v}$. There exists a unique $\pi \in \mathbb{F}_q^{E(v)}$ such that $\pi(a) = b$ and such that $\sum_{e \in E(v)} \pi(e) = 0$. Moreover, this vector $\pi$ can easily be defined in $\mathrm{C}^{\ell+2}_{\infty\omega}$ given the elements $a$ and $b$. Now assume that one of the edges $e = (v,w) \in E(v)$ is marked, but that $\pi(e) \neq 0$. Since the edge $e$ is marked, every element in $\hat{e}$ can uniquely be identified in $\mathrm{C}^{\ell+2}_{\infty\omega}$ by Lemma 4.31. However, since $a$ and $b$ are $R$-connected to *different* elements in $\hat{e}$ (as $\pi(e) \neq 0$), this contradicts the fact that $\bar{a}$ and $\bar{b}$ have the same $\mathrm{C}^{\ell+2}_{\infty\omega}$-type. Thus, for every edge $e \in E(v)$ we either have that $\pi(e) = 0$ or that $e$ is not marked. By induction on the number of edges $e \in E(v)$ with $\pi(e) \neq 0$ we show that $\pi$ can be extended to an automorphism in $\mathrm{Fix}(a_1, \dots, a_{i-1})$. Thus let us fix $e \in E(v)$ such that $\pi(e) \neq 0$. Since we have that $\sum_{f \in E(v)} \pi(f) = 0$, there has to be another edge $e' \in E(v)$ with $\pi(e') \neq 0$. We apply Lemma 4.32 to obtain an automorphism $\sigma \in \mathrm{Fix}(a_1, \dots, a_{i-1})$ such that $\sigma(e) = \pi(e)$, $\sigma(e') = -\pi(e)$ and $\sigma(f) = 0$ for all $f \in E(v)$. Now consider $(\pi - \sigma) \in \mathbb{F}_q^{E(v)}$. By the induction hypothesis we can extend this vector to an automorphism $\pi_* \in \mathrm{Fix}(a_1, \dots, a_{i-1})$. But then $(\pi_* + \sigma) \in \mathrm{Fix}(a_1, \dots, a_{i-1})$ is an extension of $\pi$.

For the second case assume that $a, b \in \hat{e}$ for some edge $e \in E(v)$. As above we conclude that the edge $e$ is not marked. Since $\mathrm{con}(\mathcal{G}) > (\ell + 2)$, the minimal

degree of each vertex in $\mathcal{G}$ is at least $(\ell+4)$. Since the vertex $v$ is not marked there has to be another edge $e' \in E(v)$, $e \neq e'$, which is not marked. Thus we can apply Lemma 4.32 to obtain an automorphism $\pi \in \mathrm{Fix}(a_1, \ldots, a_{i-1})$ such that $\pi(a) = b$ and $\pi(f) = 0$ for all $f \in E(v) \setminus \{e, e'\}$. □

It is well-known that the sets of $\mathrm{C}^{\ell+2}_{\infty\omega}$-equivalent tuples can be linearly ordered in FPC, see for example [79]. Hence, it follows from our previous lemma that the class $\mathcal{K}$ satisfies property (II).

**Lemma 4.34.** *The class $\mathcal{K}$ satisfies the properties (I) and (II).*

Let us now turn our attention to property (IV). In the next lemma we are going to show that for each $k \geq 1$ and each sufficiently connected graph $\mathcal{G} \in \mathfrak{G}$, the logic $\mathrm{C}^k_{\infty\omega}$ cannot distinguish between any pair of CFI-structures over $\mathcal{G}$ (although there exist non-isomorphic CFI-structures over $\mathcal{G}$).

**Lemma 4.35.** *Let $k \geq 1$ and let $\mathcal{G} = (V, \leq, E) \in \mathfrak{G}$ be such that $\mathit{con}(\mathcal{G}) > k$. Then for all $\vec{d}, \vec{d}_* \in [q]^V$ it holds that*

$$\mathrm{CFI}_q(\mathcal{G}, \vec{d}) \equiv^C_k \mathrm{CFI}_q(\mathcal{G}, \vec{d}_*).$$

*Thus, the class $\mathcal{K}$ satisfies property (IV).*

*Proof.* Let $\mathfrak{A} = \mathrm{CFI}_q(\mathcal{G}, \vec{d})$ and let $\mathfrak{B} = \mathrm{CFI}_q(\mathcal{G}, \vec{d}_*)$. Without loss of generality we assume that $\mathfrak{A} \not\cong \mathfrak{B}$. We show that Duplicator wins the $k$-pebble bijection game played on $\mathfrak{A}$ and $\mathfrak{B}$. Let $z_a := \sum_{v \in V} \vec{d}(v)$, let $z_b := \sum_{v \in V} \vec{d}_*(v)$ and let $z := z_b - z_a$. As above, for $e = (v, w) \in E$ and $y \in [q]$ we let $\sigma^y[e] \in \Gamma = \Gamma(\mathcal{G})$ denote the isomorphism which shifts the edge class $\hat{e}$ by $y$, the edge class $\hat{f}$ for $f = (w, v)$ by $-y$ and which stabilises all remaining classes, that is

$$\sigma^y[e](f) = \begin{cases} z, & \text{if } f = (v, w), \\ -z, & \text{if } f = (w, v), \\ 0, & \text{else.} \end{cases}$$

Given a position $(\mathfrak{A}, a_1, \ldots, a_\ell, \mathfrak{B}, b_1, \ldots, b_\ell)$ in the $k$-pebble bijection game, we say that a pair $(v, \pi)$ with $v \in V$ and $\pi \in \Gamma(\mathcal{G})$ is *good* if:

- the $v$-gadget is not marked (by the pebbled elements $a_1, \ldots, a_\ell$ in $\mathfrak{A}$ or, equivalently, by the pebbled elements $b_1, \ldots, b_\ell$ in $\mathfrak{B}$),
- $\pi(a_i) = b_i$ for $1 \leq i \leq \ell$,
- $\pi(\mathfrak{A} \setminus \hat{v}) = \mathfrak{B} \setminus \hat{v}$, and
- $(\sigma^z[e] + \pi)(\mathfrak{A} \restriction \mathrm{gadget}(v)) = \mathfrak{B} \restriction \mathrm{gadget}(v)$ for all $e \in E(v)$.

Intuitively this means that $\pi$ is almost an isomorphism between $\mathfrak{A}$ and $\mathfrak{B}$ except for the gadget associated to vertex $v$. Of course $\pi$ itself does not induce a bijection between the universes of the two CFI-structures (as otherwise $\mathfrak{A} \cong \mathfrak{B}$). However, for each $e \in E(v)$ we can associate a bijection $\hat{\pi}_e : A \to B$ to $\pi$ which is defined as

$$\hat{\pi}_e(x) = \begin{cases} \pi(x), & \text{if } x \notin \hat{v}, \\ (\sigma^z[e] + \pi)(x), & \text{if } x \in \hat{v}. \end{cases}$$

In what follows we show that Duplicator can play in such a way that after each round such a good pair $(v, \pi)$ exists. Obviously, if Duplicator can maintain this invariant this suffices for her to win the game.

Indeed we can find such a good pair $(v, \pi)$ by Lemma 4.29 for the initial position $(\mathfrak{A}, \mathfrak{B})$ of the game. Let us now consider one round of the game which starts from a position $(\mathfrak{A}, a_1, \dots, a_\ell, \mathfrak{B}, b_1, \dots, b_\ell)$ for which a good pair $(v, \pi)$ exists. First, Spoiler chooses a pair $i \leq k$ of pebbles which he removes from the game board (if the corresponding pebbles are placed at all). Duplicator then answers Spoiler's challenge by providing a bijection $\hat{\pi}_e$ for some edge $e \in E(v)$ which is not marked. Note that such an edge $e$ exists since $\mathrm{con}(\mathcal{G}) > k$ and thus each vertex has degree at least $k + 2$. Spoiler picks a new pair $(a, \hat{\pi}_e(a)) \in A \times B$ of $\hat{\pi}_e$-related elements on which he places the $i$-th pair of pebbles. By the properties of $\pi$ it immediately follows that the resulting mapping $\bar{a}[i \mapsto a] \mapsto \bar{b}[i \mapsto b]$ is a partial isomorphism. However, it might happen that Spoiler placed the $i$-th pair of pebbles on equation nodes $\hat{v}$ in the gadget associated to vertex $v$. In this case the pair $(v, \pi)$ is not good any longer. So assume that Spoiler pebbled a new pair of elements $(a, \pi_e(a)) \in \hat{v} \times \hat{v}$. Since the edge $e = (v, w)$ was not marked we know that the $w$-gadget is not marked. Thus it is easy to see that the pair $(w, \sigma^z[e] + \pi)$ is good. □

To complete our proof we establish an $\mathrm{FPS}_q$-definable canonisation procedure on the class $\mathcal{K}$. The idea is as follows: given a CFI-structure $\mathfrak{A} = \mathrm{CFI}_q(\mathcal{G}, \vec{d})$ over a graph $\mathcal{G}$ and a value $z \in [q]$ we construct a linear equation system over $\mathbb{F}_q$ which is solvable if, and only if, $\sum \vec{d} = z$. This linear equation system is FO-definable in the structure $\mathfrak{A}$, which shows that $\mathrm{FPS}_q$ can determine the isomorphism class of a CFI-structure over $\mathcal{G}$. Since the graph $\mathcal{G}$ is ordered, it is easy to construct an ordered representative from each of the isomorphism classes of CFI-structures over $\mathcal{G}$, which concludes our argument.

More specifically, let $\mathcal{G} = (V, \leq, E) \in \mathfrak{G}$, let $\mathfrak{A} = \mathrm{CFI}_q(\mathcal{G}, \vec{d}) \in \mathcal{K}$ and let $z \in \mathbb{F}_q$. For our linear equation system we identify each element $e_i \in \hat{E}$ and each vertex $v \in V$ with a variable over $\mathbb{F}_q$, that is we let $\mathcal{V} := \hat{E} \uplus V$ be the set of variables. The equations of the linear equation system are given as follows:

$$e_{i+1} = e_i + 1 \qquad \text{for all } e_i \in \hat{E} \tag{E 4.1}$$
$$e_i = -f_{-i} \qquad \text{for related edges } e, f \in E \tag{E 4.2}$$
$$v = \sum_{e \in E(v)} e_{\rho(e)} \qquad \text{for all } v \in V, \rho \in \hat{v} \tag{E 4.3}$$
$$z = \sum_{v \in V} v. \tag{E 4.4}$$

It is easy to see that this system is FO-definable in $\mathfrak{A}$. First of all, the equation (E 4.4) can be defined as a sum over the ordered set $V$. Moreover, we can express the equations of type (E 4.1) and (E 4.2) by using the cycle and inverse relation, respectively. Finally, the equations of type (E 4.3) can be expressed by using the gadget relation $R$.

**Lemma 4.36.** *The system defined above is solvable if, and only if,* $\sum \vec{d} = z$.

*Proof.* If $\sum \vec{d} = z$, then it is easy to verify that we obtain a solution $\vec{\sigma} \in \mathbb{F}_q^{\mathcal{V}}$ of the linear system by setting $\vec{\sigma}(e_i) = i$ and $\vec{\sigma}(v) = \vec{d}(v)$. For the other direction, we show that a solution $\vec{\sigma} \in \mathbb{F}_q^{\mathcal{V}}$ of this system defines an isomorphism $\pi$ between $\mathfrak{A}$ and $\mathfrak{B} = \mathrm{CFI}_q(\mathcal{G}, \vec{d}_+)$ where $\vec{d}_+(v) \coloneqq \vec{\sigma}(v)$. As a preparation, we let $\delta(e) \coloneqq \vec{\sigma}(e_i) - i$ for $e \in E$ and some $e_i \in \hat{e}$. Since $\vec{\sigma}$ is a solution, $\delta \in \mathbb{F}_q^E$ is well-defined. Now we obtain the isomorphism $\pi$ for $e_i \in \hat{E}$ and $\rho \in \hat{V}$ by setting

$$\pi(e_i) \mapsto e_{\sigma(e_i)}$$
$$\pi(\rho) \mapsto \rho + \delta.$$

Using the equations (E 4.1) and (E 4.2) one easily verifies that $\pi$ respects the cycle relation $C$ and the inverse relation $I$. Moreover, let $(\rho, e_{\rho(e)}) \in R$. Then

$$\pi(e_{\rho(e)}) = e_{\vec{\sigma}(e_{\rho(e)})} \text{ and } \vec{\sigma}(e_{\rho(e)}) = \rho(e) + \delta(e).$$

Thus, $\pi$ also respects $R$. Finally, by the equations of type (E 4.3), for all $v \in V$ and $\rho \in \hat{v}$ we have that

$$\sum \rho + \delta = \sum_{e \in E(v)} \vec{\sigma}(e_{\rho(e)}) = \vec{\sigma}(v).$$

This shows that $\vec{\sigma}(v) = d_+(v)$ and that $\sum \vec{d}_+ = \sum_{v \in V} \vec{\sigma}(v) = z$ because of equation (E 4.4). □

**Lemma 4.37.** *The class $\mathcal{K}$ satisfies property (III).*

This finishes our proof of Theorem 4.19.

## 4.5 Discussion

We have shown that the expressive power of rank operators and solvability quantifiers over different prime fields is incomparable. The important consequence is that the version of rank logic FPR with a distinct rank operator $\mathsf{rk}_p$ for every prime $p \in \mathbb{P}$ fails to capture polynomial time. In particular, we saw that the revised version of rank logic FPR$^*$ with a uniform rank operator is strictly more powerful. We remark that the problem of having non-uniform operators has been noted earlier in [59, 80, 71], but there was no proof showing that uniform operators lead to more expressive power. Moreover, we separated rank operators and solvability quantifiers in the absence of counting.

Of course, the immediate question is whether the extension FPR$^*$ of FPC by the *uniform* rank operator $\mathsf{rk}$ suffices to capture polynomial time. We do not believe that this is the case. A natural candidate to separate FPR$^*$ from PTIME is the solvability problem for linear equation systems over (finite) rings and Abelian groups, see Chapter 3. While the solvability of linear equation systems can efficiently be decided also over (finite) Abelian groups and rings, it is not clear whether rank operators over (finite) *fields* suffice to do so. In particular, can FPR$^*$ define the solvability of linear equation systems over $\mathbb{Z}_4$? In fact, a much simpler instance of this question is open as well: can FPR$^*$ distinguish between different CFI-structures over $\mathbb{Z}_4$? In some sense it seems hard to simulate counting modulo 4 by counting modulo $p$ for $p \in \mathbb{P}$, but on the other hand, it may be possible to reduce the natural linear equation system for CFI-structures over $\mathbb{Z}_4$ to several linear equation systems over $\mathbb{Z}_2$. For instance, it is obvious that FPR$^*$ can distinguish between such pairs of CFI-structures which differ by a unit in $\mathbb{Z}_4$ (just consider the associated linear equation system modulo two).

**Matrix rank over rings** If it turns out that FPR$^*$ cannot define the solvability of linear equation systems over finite rings, then the natural idea would be to generalise rank operators to (finite, commutative) rings. This, however, is a non-trivial task.

First of all, there are different ways to define the matrix rank over (finite) commutative rings, which turn out to be non-equivalent (while all these different notions *are equivalent* over *fields*). In algebra, the most common approach to define the matrix rank over commutative rings is to consider, for a matrix $M$ over a commutative ring $R$, the *$t$-th determinantal ideal* $I_t(M)$ of $M$, which is the ideal generated by all $t \times t$-minors of $M$. Then the rank of $M$ is defined as the maximal $t$ such that $I_t(M)$ is different from $(0)$ (or, for the similar notion of *McCoy rank* one puts the stronger requirement that $I_t(M)$ is not annihilated by any ring element except for 0), see [18, 77]. While many of the natural properties of matrix rank over fields also hold for this notion of matrix rank over commutative rings, there are two problems when we want to apply this notion in our setting. First of all, it is not clear whether this variant of matrix rank over rings can be computed in polynomial time (at least we are

not aware of any algorithm to do so). Moreover, even if it can be computed in polynomial time, then it is not clear whether it helps to decide the solvability of linear equation systems. At least, the simple criterion $\mathsf{rk}(M) = \mathsf{rk}(M \mid \vec{c})$ can not be used to characterise solvable linear equation systems $M \cdot \vec{x} = \vec{c}$.

Let us very briefly comment on two ideas to overcome these problems. First, assume we generalise the notion of matrix rank to commutative rings in a different way, namely by just copying its standard definition over fields. Thus, we define the rank of a matrix $M$ over a commutative ring $R$ as the size of a maximal set of linearly independent columns (we remark that, in general, we then have $\mathsf{rk}(M) \neq \mathsf{rk}(M^T)$). For this variant of matrix rank we can actually prove that it can be computed in polynomial time. The question is as follows: does this notion of matrix rank help to solve linear equation systems over rings? In this case we could consider another revision of rank logic by rank operators which compute this variant of matrix rank over rings. Our preliminary results, though, rather point into the direction of saying that this variant of matrix rank can already be expressed in rank logic $\mathrm{FPR}^*$.

Another way to circumvent the difficulties with the notion of matrix rank over commutative rings is to restrict ourselves to *simple* commutative rings, where simple, for example, means *chain rings* (local rings in which every ideal is principal, see Section 3.2.2). Over chain rings it is indeed possible to define a notion of matrix rank which can be computed in polynomial time and which suffices to solve linear equation systems. However, it is again not clear whether this notion suffices to express the solvability of linear equation system over *all* finite rings (or Abelian groups), see [27] for details.

**Queries from algebra** Rank logic is a very powerful extension of FPC which can express the solvability of linear equations systems over *all* finite fields. This query is not only important as such, but it also occurs as a basic subroutine in many polynomial-time algorithms. Still, in the last two chapters we saw that there are many subtle issues with the precise technical definition of rank logic like the (non-)uniformity of operators or the seemingly strong dependence on the kind of underlying algebraic domain.

More strikingly, there are a lot of similar (though more general) algorithmic problems from the field of algebra for which we do not know whether they can be expressed in rank logic. For instance, can rank logic express the *solvability* of linear equation systems over the *integers* or can rank logic define the *membership problem* for *permutation groups* (see the following paragraph). In some sense, if with rank operators we had really identified an adequate logical mechanisms to express the general algorithmic principles to manipulate succinct representations of algebraically structured objects, then one would expect that rank logic can actually define these problems.

As one of the interesting candidates, let us briefly discuss the *permutation group membership problem*. In its very general form, it asks whether given an (unordered) set of permutations $\pi_1, \ldots, \pi_k \in \mathrm{Sym}(\Omega)$ over an (unordered) set

$\Omega$, and given a test permutation $\sigma \in \mathrm{Sym}(\Omega)$, does it hold that $\sigma$ is contained in the permutation group $\Gamma = \langle \pi_1, \ldots, \pi_k \rangle \leq \mathrm{Sym}(\Omega)$ which is generated by $\pi_1, \ldots, \pi_k$? It is far from being obvious that this problem can be decided in polynomial-time (observe that the generated permutation group can be of exponential size), see [36, 60, 85]. The crucial step for deciding whether "$\sigma \in \Gamma$" holds is to compute a *strong* generating set for $\Gamma$ (which is a generating set with a certain normal form). By using this strong generating set it is then easy to check whether $\sigma$ can be written as a product of (strong) generators. This normal form, and actually the whole approach, resembles the method of Gaussian elimination, and thus, deciding the permutation group membership problem can naturally be seen as a generalisation of the solvability problem for linear equation systems. In fact, one can easily reduce the solvability problem for linear equation systems over *all* finite Abelian groups to the permutation group membership problem. So the obvious question is: can rank logic express membership in permutation groups? As a first step, it makes sense to study this question in the relaxed version for *Abelian* permutation groups, see [78]. In particular, we remark that the algorithmic ideas for solving the permutation group membership problem are central ingredients for the known graph canonisation algorithms for classes of graphs with bounded colour class size and bounded degree. Hence, if we make progress in understanding the logical mechanisms that are required to express the permutation group membership problem, then this might also lead to a *natural* logic for polynomial time on these classes of graphs.

**Solvability quantifiers vs. rank operators** In the absence of counting we proved that rank operators are strictly more powerful than solvability quantifiers. However, the relationship between solvability logic FPS and rank logic FPR (as extensions of fixed-point logic *with* counting) remains unclear. We only know, by our proof of Lemma 4.25, that on *every* class of structures of bounded colour class size the two logics have the same expressive power. However, over general structures our reduction fails. One way to attack this problem might be to combine our algebraic approach with the game-theoretic toolkit proposed by Dawar and Holm in [29]. In particular, we are interested in a variant of their partition games for infinitary logics with solvability quantifiers. Might it be the case that in such games we can determine the winner in polynomial time (this is open for the variant for rank operators)? As a first step, it would also be interesting to see whether a reduction of rank operators to solvability quantifiers can be obtained in a logic which is more powerful than FPC, like for example Choiceless Polynomial Time, see Section 2.4.

Moreover, separating solvability quantifiers from weaker linear-algebraic operators (in the absence of counting) would be very insightful. For instance, it is clear that with matrix rank operators one can easily check whether two matrices have the same rank (that means one can define the *matrix equivalence*

*problem*). Moreover, if we are able to check whether two matrices have the same rank, then we can also decide the solvability of linear equation systems (again recall that $M \cdot \vec{x} = \vec{c}$ is solvable if, and only if, $\mathsf{rk}(M) = \mathsf{rk}(M \,|\, \vec{c})$). So the obvious question is: what happens if we take first-order logic and extend it by quantifiers for the matrix equivalence problem? Is the resulting logic strictly stronger than FOS?

Another question is motivated by our normal form theorem for $\mathrm{FOS}_p$ (Theorem 4.9). One crucial step was to show that nested solvability quantifiers can be reduced to a single solvability quantifier. It would be very interesting to prove, or disprove, that this also holds for solvability quantifiers over general cyclic rings $\mathbb{Z}_d$, $d \geq 2$. In fact, there might be interesting connections with the closure properties of the complexity class $\mathrm{MOD}_k\mathrm{L}$ for integers $k \geq 2$ which are not primes.

**The power of rank operators** There are also more positive directions to explore. For example, it is still open whether rank logic (in its revised version) can express the isomorphism problem on classes of graphs with bounded colour class size, see Chapter 6. More strikingly, until today we do not know whether rank operators can simulate fixed-point inductions, that is we do not know whether FOR is a strict fragment of FPR (although there are strong reasons to believe that this is the case). In this context it is also interesting to study the power of the matrix-equivalence games proposed by Dawar and Holm in [29] which characterise the expressive power of infinitary logic with matrix rank operators (until today, we do not know whether the winner in such games can be determined in polynomial time). Also, we aim to study the definability of more general problems from the field of algebra in rank logic, such as the permutation group membership problem, or certain tractable instances of the constraint satisfaction problem [19]. There are also nice recent results which explore the power of certain algebraic proof systems [13]. Relating such algebraic proof systems to rank logic may provide new insights about the expressive power of rank operators.

# Chapter 5

# Cyclic linear equation systems

A *cyclic linear equation system* (or CES for short) is a linear equation system over a finite ring $\mathbb{Z}_d$, where $d$ is a prime-power, with a strong auxiliary structure: the set of variables $V$ is *almost* totally ordered up to classes $V_i$ in which all pairs of variables linearly depend on each other, and this dependence is explicitly specified by a set $C_i$ of linear equations on $V_i$ (which we call *cyclic constraints*). More precisely, the value of each variable $v \in V_i$ is determined by the value of *any* other variable $w \in V_i$ from the same class, that is for all pairs of variables $v, w \in V_i$ the cyclic constraint $C_i$ contains a linear equation of the form $v - w = z_{vw}$ for a *constant* $z_{vw} \in \mathbb{Z}_d$. Thus, in principle, every class $V_i$ can be resolved by picking an arbitrary variable from the class and by replacing all remaining variables by the equivalent linear terms which are specified through the cyclic constraints. The resulting system would have a *totally ordered* set of variables, and the solvability of such linear systems is definable in fixed-point logic with counting.

However, fixing an arbitrary variable from each class simultaneously is not possible in a logic, since formulas have to respect the symmetries of the input structure. Indeed, in spite of the strong auxiliary structure, a cyclic linear equation system can possess a large automorphism group. In particular, it turns out that the isomorphism problem for Cai, Fürer, Immerman graphs can be rephrased as a cyclic linear equation systems over $\mathbb{Z}_2$ [28] and, as a consequence, the solvability problem for CESs cannot be defined in fixed-point logic with counting. Thus, CESs form a class of structurally quite simple linear equation systems which are powerful enough to separate fixed-point logic with counting from polynomial time. Moreover, they have several natural applications, most importantly, for deciding the isomorphism problem for structures which resemble the CFI-graphs. Incidentally, we used cyclic linear equation systems in our proof of Lemma 4.37 in Chapter 4 to decide the isomorphism problem of generalised CFI-structures.

In this chapter we show that the solvability problem for cyclic linear equation systems can be expressed in Choiceless Polynomial Time (Theorem 5.12). Hence, there is a polynomial-time procedure to decide the consistency of cyclic

linear equation systems which avoids arbitrary choices, but which cannot be formulated in fixed-point logic with counting. This yields an interesting family of polynomial-time queries which separate CPT from FPC.

Besides that, our result is a first step towards the significant open question of whether Choiceless Polynomial Time can express the solvability of general linear equation systems over all finite Abelian groups (recall that this query cannot be expressed in FPC). This question is of particular importance, since an answer would, most probably, clarify the relationship between Choiceless Polynomial Time and rank logic. Maybe, the matrix rank over finite fields can be defined in Choiceless Polynomial Time? Also, in the light of our studies in Chapter 3 and Chapter 4, another scenario seems possible: maybe, rank logic can *not* define the solvability of cyclic linear equation systems over all rings $\mathbb{Z}_d$ where $d$ is a prime power? In this case, our result would provide a CPT-definable query which cannot be defined in FPR (which would show that CPT cannot be embedded into FPR).

Furthermore, the definability result in this chapter is the key to show that Choiceless Polynomial Time can express *all* polynomial-time properties of *structures with Abelian colours* (see Chapter 6). The main observation there is that cyclic linear equation systems can be used to succinctly represent large sets of isomorphisms between structures with Abelian colours.

The whole chapter is strongly based on [1]. In Section 5.1, we start by introducing the notion of cyclic linear equation systems. We then obtain a simple normal form for CESs and we show that it is first-order definable. In Section 5.2, we establish the central notion of *hyperterms* and we uncover strong connections between their syntactic structure (as hereditarily finite sets), their symmetries, and their semantics. We also show that basic operations on the domain of hyperterms, like "addition" and "scalar multiplication", can be expressed in Choiceless Polynomial Time. In Section 5.3, we then obtain a CPT-program which translates cyclic linear equation systems into equivalent and *ordered* systems of *hyperequations*. By adapting the method of Gaussian elimination to finite rings and by applying our knowledge about hyperterms we can finally show that the solvability of such ordered systems of hyperequations can be defined in Choiceless Polynomial Time.

## 5.1 A definable normal form

In this section we introduce the notion of *cyclic linear equation systems* and we establish a simple first-order definable normal form. The crucial feature of cyclic linear equation systems is that the set of variables is almost linearly ordered up to classes in which all pairs of variables directly depend on each other. Formally, cyclic equation systems contain a linear preorder $\preceq$ on their set of variables with the additional requirement that all pairs of $\preceq$-equivalent variables differ by a constant only, and this constant is specified by the equations of the linear system (in a consistent way). Let us give the precise definition.

**Definition 5.1.** A *cyclic linear equation system* (or CES, for short) is a structure $(\mathfrak{A}, \preceq) \in \mathcal{S}(\tau_{\text{les-r}} \uplus \{\preceq\})$ such that $\mathfrak{A} \in \text{Ls}(\tau_{\text{les-r}})$ (cf. Section 3.1) encodes a linear equation system $M \cdot \vec{x} = \vec{c}$ over a ring $\mathbb{Z}_d$, where $M$ is an $I \times J$-coefficient matrix over $\mathbb{Z}_d$, and where $\vec{c}$ is an $I$-vector of constants over $\mathbb{Z}_d$, with the following properties:

- $d$ is a prime power, that is $d = p^\ell$ for some prime $p \in \mathbb{P}$ and $\ell \geq 1$,
- $\preceq$ is a linear preorder on $J$, that is $J = J_0 \preceq J_1 \preceq \cdots \preceq J_{n-1}$, and
- for every pair of $\preceq$-equivalent variables $x_j, x_{j'}$, for $j, j' \in J_i$, the linear equation system contains an equation $x_j - x_{j'} = z$ for some $z \in \mathbb{Z}_d$.

In this chapter we show that the solvability problem for cyclic linear equation systems can be defined in Choiceless Polynomial Time. Since CPT-programs manipulate hereditarily finite *sets*, it is convenient, for the sake of a concise presentation, to adapt our matrix encoding of linear equation systems to a presentation which is based on hereditarily finite sets.

To this end, let us denote by $V = \{x_j : j \in J\}$ the set of variables of a linear equation system $M \cdot \vec{x} = \vec{c}$ over $\mathbb{Z}_d$ as above. Then an *atomic linear term* is either a constant $z \in \mathbb{Z}_d$ or an object $z \cdot v$ for $z \in \mathbb{Z}_d$ and $v \in V$. Moreover, a *linear term* is a set of atomic linear terms, and a *linear equation* is a pair $(t, z)$ where $t$ is a linear term and where $z \in \mathbb{Z}_d$. We usually write a linear equation $e = (t, z)$ in the more convenient way as $e : t = z$. Finally, a *linear equation system* is a set of linear equations.

Let $\alpha : V \to \mathbb{Z}_d$ be an *assignment* of the variables $V$ to values in $\mathbb{Z}_d$. Then the *value* $t[\alpha] \in \mathbb{Z}_d$ of an atomic linear term $t = z \cdot v$ under $\alpha$ is $t[\alpha] = z \cdot \alpha(v)$ (for $t = z$ with $z \in \mathbb{Z}_d$ we set $t[\alpha] = z$). Moreover, the value $t[\alpha] \in \mathbb{Z}_d$ of a non-atomic linear term $t$ with respect to $\alpha$ is $t[\alpha] = \sum_{s \in t} s[\alpha]$. An assignment $\alpha : V \to \mathbb{Z}_d$ *satisfies* a linear equation $e = (t, z)$, denoted as $\alpha \vDash e$, if $t[\alpha] = z$. A linear equation system $S$ is *solvable* (or *consistent*) if there is an assignment $\alpha : V \to \mathbb{Z}_d$ (a *solution*) which satisfies all linear equations in $S$ (for such an assignment $\alpha$ we also write $\alpha \vDash S$).

With this notation, a cyclic linear equation system over $\mathbb{Z}_d$ can be identified with a triple $(V, S, \preceq)$, where $V$ is the set of variables as above, and where

- $\preceq$ is a linear preorder on the variables $V = V_0 \preceq \cdots \preceq V_{n-1}$, and
- the set of linear equations $S$ contains for every block $V_i$ a *cyclic constraint* $C_i \subseteq S$ that is a consistent and maximal set of equations of the form $v - w = z$ for $v, w \in V_i$ and $z \in \mathbb{Z}_d$.

Let $C_i$ be a cyclic constraint for the variable block $V_i$. We write $\mathbb{L}(C_i) = \mathbb{L}_i$ to denote the set of assignments $\alpha : V_i \to \mathbb{Z}_d$ which satisfy $C_i$. Since $C_i \subseteq S$, every solution of the given CES is contained in the space

$$\mathbb{L} := \mathbb{L}_0 \times \cdots \times \mathbb{L}_{n-1} \subseteq \mathbb{Z}_d^{V_0} \times \cdots \times \mathbb{Z}_d^{V_{n-1}}.$$

In other words, if we denote by $\mathbb{L}(S) \leq \mathbb{Z}_d^V$ the solution space of the given CES, then we have $\mathbb{L}(S) \subseteq \mathbb{L}$.

Our next aim is to obtain a normal form for cyclic linear equation systems. Although, a priori, there is no bound on the size of the classes $V_i$, we show in Lemma 5.3 that, by the presence of the cyclic constraints $C_i$, we can assume that $|V_i| = d$ for all $i \in [n]$. Moreover, we want this to be a true equality, that is we do not want that two different variables $v, w \in V_i$ can take the same value in an assignment $\alpha \in \mathbb{L}_i$. In other words, we do not want that $v - w = 0 \in C_i$ if $v \neq w$ (because then we could replace $v, w$ by a single variable).

To this end, we establish a first-order interpretation which transforms a cyclic linear equation system into a system which has this property and, moreover, we achieve this transformation in such a way that a (definable) one-to-one correspondence between the sets of solutions can be maintained. As a first preparation we state a simple observation.

**Lemma 5.2.** *Let $\alpha, \beta \in \mathbb{L}_i$. If $\alpha(v) = \beta(v)$ for some $v \in V_i$, then $\alpha = \beta$. In particular, the solution spaces $\mathbb{L}_i$ are of size $d$, that is $|\mathbb{L}_i| = d$.*

*Proof.* Let $v \in V_i$. Then for every $w \in V_i$ there is a unique $z_w \in \mathbb{Z}_d$ such that $w - v = z_w \in C_i$. Hence, if for $\alpha \in \mathbb{L}_i$ we have $\alpha(v) = z$, then it follows that $\alpha(w) = z + z_w$. Moreover, by the consistency of $C_i$, we have for $w' \in V_i$ that $w' - w = z_{w'} - z_w \in C_i$. Altogether, this shows that for $\alpha : V_i \to \mathbb{Z}_d$ with $\alpha(v) = z$ we have $\alpha \in \mathbb{L}(C_i)$ if, and only if, $\alpha(w) = z + z_w$ for $w \in V_i$. □

**Lemma 5.3.** *There is a first-order interpretation which transforms a cyclic linear equation system $(V, S, \leq)$ over $\mathbb{Z}_d$, where $V = V_0 \leq \cdots \leq V_{n-1}$ (with associated cyclic constraints $C_i$), into an equivalent cyclic linear equation system $(V', S', \leq')$ over $\mathbb{Z}_d$ such that*

- *$V' = V_0' \leq' \cdots \leq' V_{n-1}'$ and $|V_i'| = d$ for all $i \in [n]$ (with associated cyclic constraints $C_i'$), and such that for all $i \in [n]$ and $v, w \in V_i'$, $v \neq w$, we have $v - w = 0 \notin \mathbb{L}(C_i')$, and such that*
- *the interpretation provides bijections $\varphi_i : \mathbb{L}(C_i) \mapsto \mathbb{L}(C_i')$ such that $\varphi = (\varphi_0, \ldots, \varphi_{n-1})$ is a bijection between $\mathbb{L}(S)$ and $\mathbb{L}(S')$.*

*Proof.* We start by defining the new classes of variables $V_i'$. Let $V_i^*$ denote the set of all (pairwise distinct) objects of the form $v + z$ for variables $v \in V_i$ and constants $z \in \mathbb{Z}_d$. Then we have $|V_i^*| = |V_i| \cdot d$. Next, we consider the following equivalence relation $\approx$ on $V_i^*$: we set $v + z \approx w + z'$ if, and only if, $v - w = z' - z \in C_i$. It is straightforward to verify that the consistency and maximality of $C_i$ implies that $\approx$ is an equivalence relation on $V_i^*$. With this preparation we set $V_i' := (V_i^* / \approx)$.

We next define a cyclic constraint $C_i'$ on the set $V_i'$ which contains for every pair $[v + z], [w + z'] \in V_i'$ the constraint $[v + z] - [w + z'] = c + z - z'$ where $c \in \mathbb{Z}_d$ is such that $C_i$ contains the constraint $v - w = c$. Again it is straightforward to show that $C_i'$ is well-defined.

We proceed to define the mappings $\varphi_i : \mathbb{L}(C_i) \to \mathbb{L}(C_i')$. For $\alpha \in \mathbb{L}(C_i)$ we let $\beta = \varphi_i(\alpha) \in \mathbb{L}(C_i')$ be given as $\beta([v+z]) := \alpha(v) + z$. Then $\beta$ is well-defined, $\beta \vDash C_i'$ and $\beta([v+0]) = \alpha(v)$. Moreover, assume that $\varphi_i(\alpha) = \varphi_i(\beta)$. Then we have that $\alpha = \beta$ since for all $v \in V_i$ it holds that

$$\alpha(v) = \varphi_i(\alpha)([v+0]) = \varphi_i(\beta)([v+0]) = \beta(v).$$

Since $|\mathbb{L}(C_i)| = |\mathbb{L}(C_i')| = d$ by Lemma 5.2, we conclude that $\varphi_i$ indeed defines a bijection between $\mathbb{L}(C_i)$ and $\mathbb{L}(C_i')$. Moreover, since $\varphi_i(\alpha)([v+0]) = \alpha(v)$ for all $v \in V_i$ and $\alpha \in \mathbb{L}_i$ we easily obtain an equivalent cyclic linear equation system over $V' := V_0' \trianglelefteq \cdots \trianglelefteq V_{n-1}'$ by substituting each occurrence of a variable $v \in V_i$ in the original linear equation system by the corresponding $\approx$-equivalence class $[v+0] \in V'$, and by replacing each cyclic constraints $C_i$ by $C_i'$ for $i \in [n]$.

It remains to show that $|V_i'| = d$. To this end, we choose an arbitrary variable $v \in V_i$ and claim that $V_i' = \{[v+z] : z \in \mathbb{Z}_d\}$. In fact, if we can show this, then our original claim follows, since for each $v \in V_i$ and $z, z' \in \mathbb{Z}_d$ with $z \neq z'$ we have that $[v+z] \neq [v+z']$. Thus let $w \in V_i$ and $y \in \mathbb{Z}_d$. Then there exists a unique $c \in \mathbb{Z}_d$ such that $w - v = c \in C_i$. But then we have $[w+y] = [v+c+y]$ which finishes our argument.

It is easy to see that the transformations described above can be expressed via a first-order interpretation. □

We remark that the linear equation systems over $\mathbb{Z}_q$ ($q \in \mathbb{P}$) which we defined in the proof of Lemma 4.37 to characterise the isomorphism class of CFI-structures over $\mathbb{Z}_q$ are actually *cyclic* linear equation systems. Since FPC fails to capture PTIME on $\mathcal{K}_q$ (Theorem 4.21) we obtain the following result: in spite of the strong structural properties, the solvability problem for cyclic linear equation systems is not definable in FPC, see also [59, Chapter 7].

**Corollary 5.4.** *The solvability problem for cyclic linear equation systems (even over prime fields) cannot be defined in fixed-point logic with counting.*

## 5.2 Classes of equivalent linear terms

In this section we introduce *hyperterms* which are succinct encodings of large classes of equivalent linear terms. In particular, we discover strong connections between their syntactic structure, their symmetries, and their semantics. Hyperterms will play the central role in our CPT-procedure for solving cyclic linear equation systems in the next section, and thus we show, as a preparation, that certain basic operations for hyperterms, such as addition and scalar multiplication, can be defined in Choiceless Polynomial Time.

For what follows, let us fix a cyclic linear equation system $(V, S, \trianglelefteq)$ over the ring $\mathbb{Z}_d$, where $d = p^k$ for $p \in \mathbb{P}$ and $k \geq 1$, in the representation from the previous section. By Lemma 5.3 we can assume that $V = V_0 \trianglelefteq \cdots \trianglelefteq V_{n-1}$ with associated cyclic constraints $C_i$ and such that $|V_i| = d$ for $i \in [n]$ and for all $v, w \in V_i, v \neq w$ we have that $v - w = z \in C_i$ for certain constants $z \in \mathbb{Z}_d \setminus \{0\}$.

For $z \in \mathbb{Z}_d$ and $v \in V_i$ we denote by $v^{+z} \in V_i$ the (unique) variable such that $C_i$ contains the constraint $v^{+z} - v = z$ (note that by our assumptions such a variable always exists). For convenience, we set $v^+ := v^{+1}$. With this notation we observe that the constraint $C_i$ defines a directed cycle on $V_i$ of length $d$ via the edge relation $E_i = \{(v, v^+) : v \in V_i\}$. This justifies to call $C_i$ a *cyclic* constraint.

Assume that we fix a variable $v \in V_i$. Then we obtain an *ordered* representation of $V_i$ as $V_i = v = v^{+0} \leq v^{+1} \leq v^{+2} \leq \cdots \leq v^{+(d-1)}$, that is, we can order each class $V_i$ by means of a single parameter. As a consequence, a complete order on the set of variables $V$ can be obtained by the parallel choice of a variable $v \in V_i$ for each of the classes $V_i$, $i \in [n]$. Of course, as we explained earlier, fixing such variables $v \in V_i$, for $i \in [n]$, simultaneously is not possible in any reasonable logic, since this would require to take into account all symmetric choices whose number is, in general, exponential in the size of the input structure.

The crucial idea of hyperterms is to avoid this exponential blow up by identifying "equivalent" choices and by succinctly encoding the corresponding equivalence classes as higher-order objects in the universe of hereditarily finite sets $\mathrm{HF}(V)$ over the variables $V$. To illustrate this, let us consider a small example. First of all, let us fix three distinct blocks $V_a, V_b, V_c \in \{V_i : i \in [n]\}$ of variables where $a < b < c$ and let us choose variables $v_a = v_a^{+0} \in V_a$, $v_b = v_b^{+0} \in V_b$ and $v_c = v_c^{+0} \in V_c$. Now consider the linear term $t = v_a + v_b + v_c$. In the presence of the cyclic constraints $C_a, C_b$ and $C_c$ (associated with $V_a, V_b$ and $V_c$, respectively) we conclude that this term is equivalent, for instance, to the linear term $t' = v_a^{+1} + v_b^{+1} + v_c^{+(d-2)}$. Indeed, by using the cyclic constraints $C_a, C_b, C_c$, which imply that $v_a^{+1} - v_a = 1$, $v_b^{+1} - v_b = 1$ and $v_c^{+(d-2)} - v_c = d - 2$, we have that

$$t' - t = (v_a^{+1} - v_a) + (v_b^{+1} - v_b) + (v_c^{+(d-2)} - v_c) = 1 + 1 + (d-2) = 0.$$

For an illustration see Figure 5.1.

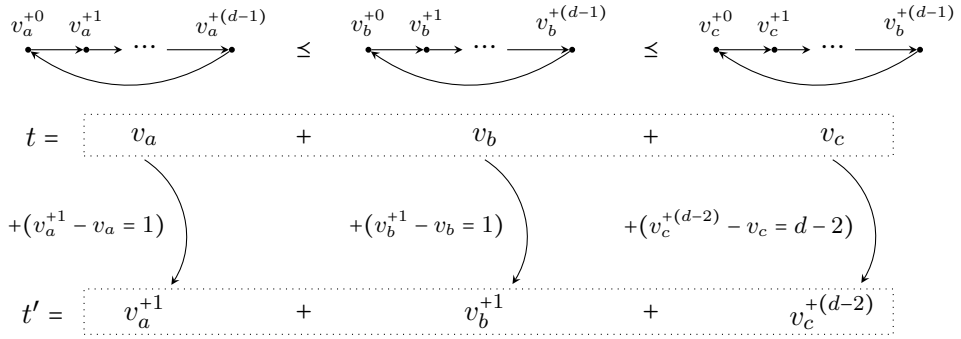


Figure 5.1: Equivalence of linear terms in the presence of cyclic constraints

Of course, besides $t'$, there exist other terms which are equivalent to $t$ as well (again, with respect to the cyclic constraints $C_a, C_b, C_c$). To capture the

set of all such equivalent terms systematically, we let $\pi^{i:+} \in \mathrm{Sym}(V_i)$, for $i \in [n]$, denote the natural $d$-cycle on $V_i$ with respect to the successor relation $(v, v^+)$, that is

$$\pi^{i:+} := (\, v\, v^{+1}\, v^{+2} \cdots v^{+(d-1)}\,) \text{ for some } v \in V_i.$$

Moreover, for $z \in \mathbb{Z}_d$, we denote by $\pi^{i:+z}$ the $z$-th power of $\pi^{i:+}$ and we let $\Gamma_i \leq \mathrm{Sym}(V_i)$ denote the subgroup of *cyclic shifts on* $V_i$, that is $\Gamma_i := \langle \pi^{i:+} \rangle$. We identify $\Gamma_i$ with $\mathbb{Z}_d$ in the obvious way and we set

$$\Gamma := \Gamma_0 \times \Gamma_1 \times \cdots \times \Gamma_{n-1} = \mathbb{Z}_d^n.$$

In other words, the group $\Gamma$ is generated by the set $\{\pi^{i:+} : i \in [n]\}$, and every element $\pi \in \Gamma$ can be written as an $[n]$-vector over $\mathbb{Z}_d$ or, in the notation from above, as

$$\pi = \sum_{i=0}^{n-1} \pi^{i:+z_i} \quad \text{where } \pi(i) = z_i \in \mathbb{Z}_d.$$

By definition, the group $\Gamma$ acts on the set of variables $V$. We consider the natural extension of this action to the class of hereditarily finite sets $\mathrm{HF}(V)$. In particular, $\Gamma$ acts on the set of linear terms over the variables $V$.

We now come back to our example from above. Let $\Delta \leq \Gamma$ denote the subgroup of $\Gamma$ which consists of all vectors $\pi \in \Gamma = \mathbb{Z}_d^n$ such that $\pi(a) + \pi(b) + \pi(c) = 0 \in \mathbb{Z}_d$ (recall that $a, b, c \in [n]$ are the indices of the three variable blocks $V_a, V_b, V_c$ which occur in the linear term $t = v_a + v_b + v_c$). Then the $\Delta$-orbit $\Delta(t)$ of the linear term $t$ contains the term $t'$. Indeed, for every $\pi \in \Delta$ such that $\pi(a) = 1, \pi(b) = 1$ and $\pi(c) = d - 2$ we have $\pi(t) = t'$. Moreover, one can verify, analogously as above, that *every* linear term in $\Delta(t)$ is equivalent to $t$ (of course, again with respect to the cyclic constraints $C_a, C_b, C_c$).

As it turns out, we can establish a more general connection from this observation. Consider for an arbitrary linear term $t$ the linear term $\pi(t)$ for some $\pi \in \Gamma$. Then we claim that the term $s := \pi(t) - t$ can be reduced, using the cyclic constraints $C_i$, to a constant. To verify this, it suffices to observe that whenever $t$ contains an atomic linear subterm $z \cdot v$ for $v \in V_i$ and $z \in \mathbb{Z}_d$, then $\pi(t)$ contains the atomic subterm $z \cdot \pi(v)$. Since $\pi(v) - v = y \in C_i$ for some $y \in \mathbb{Z}_d$, we have that $z \cdot \pi(v) - z \cdot v = z \cdot y \in \mathbb{Z}_d$, which proves our claim.

From this argument we can extract the following general result: the $\Gamma$-orbit of a linear term $t$ can be partitioned into at most $d$ different classes of equivalent linear terms (again, assuming the presence of the cyclic constraints $C_i$, $i \in [n]$). Moreover, since the automorphism group of the cyclic linear equation system is a subgroup of $\Gamma$, these equivalence classes are objects which can, in principle, be manipulated by a CPT-program (in particular, their number is bounded by $d$). The only problem is that the size of such equivalence classes, in an *explicit* representation, is exponential in the number of variables which occur in the linear term $t$. The idea of hyperterms is to succinctly encode these equivalence classes as highly nested objects in the universe of hereditarily finite sets $\mathrm{HF}(V)$ over the variables $V$.

### 5.2.1 The notion of hyperterms

The notion of hyperterms is strongly inspired by the very clever CPT-procedure of Dawar, Richerby, and Rossman for deciding the isomorphism problem of CFI-graphs [32] . Given a cyclic linear equation system $(V, S, \leq)$ as above, the class of associated *hyperterms* is defined via an inductive construction. During this construction we ensure that hyperterms have the following properties.

(S) For every hyperterm $T$ and each $z \in \mathbb{Z}_d$ there is a *$z$-shifted* hyperterm $T^{+z}$. Moreover, for $z_1, z_2 \in \mathbb{Z}_d$ we have $T^{+(z_1+z_2)} = (T^{+z_1})^{+z_2}$.

(V) Given an assignment $\alpha \in \mathbb{L}$, each hyperterm $T$ has a *value* $T[\alpha] \in \mathbb{Z}_d$. Moreover, for $z \in \mathbb{Z}_d$, we have $T^{+z}[\alpha] - T[\alpha] = z$.

(C) In a hyperterm $T$, the blocks $V_i$, $i \in [n]$, appear with *coefficients* $c_i(T) \in \mathbb{Z}_d$. The coefficients are invariant under $z$-shifts, i.e. $c_i(T) = c_i(T^{+z})$ for $z \in \mathbb{Z}_d$.

(L) For each hyperterm $T$ there are variables $v_i \in V_i$, $i \in [n]$, and a constant $z \in \mathbb{Z}_d$ such that $T$ is equivalent to the *linear term* $t := \sum_{i=0}^{n-1} c_i(T) \cdot v_i + z$, that is for all $\alpha \in \mathbb{L}$ we have $t[\alpha] = T[\alpha]$.

We proceed to give the inductive definition of hyperterms.

***Atomic hyperterms*** For $z \in \mathbb{Z}_d$, $T := z$ is a hyperterm. We set $T^{+y} := (z + y)$ for $y \in \mathbb{Z}_d$ and let $c_i(T) = 0$ for $i \in [n]$. For $\alpha \in \mathbb{L}$ we set $T[\alpha] := z$.

For $v \in V_i$, $T := v$ is a hyperterm where $T^{+y} := v^{+y}$ for $y \in \mathbb{Z}_d$. We set $c_j(T) = 1$ if $j = i$ and $c_j(T) = 0$ otherwise. Finally, we let $T[\alpha] := \alpha(v)$.

***Addition of hyperterms*** Let $Q, R$ be hyperterms and let $z \in \mathbb{Z}_d$. Then $T = Q \oplus_z R := \{\langle Q^{+z_1}, R^{+z_2}\rangle : z_1 + z_2 = z\}$ is a hyperterm. The shifted hyperterms $T^{+y}$ are given as $T^{+y} := Q \oplus_{z+y} R$ for $y \in \mathbb{Z}_d$. We set $c_i(T) := c_i(Q) + c_i(R)$ for $i \in [n]$ and $T[\alpha] := Q[\alpha] + R[\alpha] + z$ for $\alpha \in \mathbb{L}$.

For convenience we often write $\oplus$ instead of $\oplus_0$.

***Scalar multiplication*** Let $Q$ be a hyperterm and let $z \in \mathbb{Z}_d$, $z \neq 0$. Then we define the hyperterm $T = z \odot Q := Q \oplus \cdots \oplus Q$ which results by applying the $\oplus$-operation $z$-times to $Q$ (where we implicitly agree on an application from left to right). The definitions of $T^{+y}, c_i(T)$ and $T[\alpha]$ follow from the definition of $\oplus$.

With the inductive definition it is easy to verify that hyperterms satisfy the properties (S), (V), (C) and (L). The only non-trivial part is to check that the properties (V) and (C) hold for hyperterms formed by scalar multiplication. However, from the following Lemma 5.5, in which we summarise some simple observations that we use later on, one can directly infer that this is the case.

We remark that the operation $\oplus$ is not associative. Thus whenever we form a sum $T_0 \oplus T_1 \oplus \cdots \oplus T_{m-1}$ of hyperterms $T_i$, then we implicitly agree that the application of $\oplus$ is from left to right.

**Lemma 5.5.** *Let $Q, R$ be hyperterms and let $z \in \mathbb{Z}_d$ be a constant.*

*(a) For all $y_1, y_2, y \in \mathbb{Z}_d$ such that $y_1 + y_2 + y = z$ we have*

$$(Q \oplus R)^{+z} = Q \oplus_z R = Q^{+y_1} \oplus_y R^{+y_2}.$$

*(b) For hyperterms $T_0, \dots, T_{m-1}$ and values $y_0, \dots, y_{m-1} \in \mathbb{Z}_d$ we have*

$$T_0^{+y_0} \oplus T_1^{+y_1} \oplus \cdots \oplus T_{m-1}^{+y_{m-1}} = (T_0 \oplus T_1 \oplus \cdots \oplus T_{m-1})^{+y_0+y_1+\cdots+y_{m-1}}.$$

*Now let $z \neq 0$ and $T = z \odot Q$.*

*(c) If $z \neq 1$, then for all $y, y_1, y_2 \in \mathbb{Z}_d$ with $(z-1) \cdot y_1 + y_2 = y$ we have*

$$T^{+y} = ((z-1) \odot Q^{+y_1}) \oplus Q^{+y_2}.$$

*(d) Let $y \in \mathbb{Z}_d$. Then $T^{+y \cdot z} = z \odot Q^{+y}$.*

*(e) For $\alpha \in \mathbb{L}$ we have $T[\alpha] = z \cdot Q[\alpha]$.*

*(f) For $i \in [n]$ we have $c_i(T) = z \cdot c_i(Q)$.*

*Proof.* To prove (a) it suffices to recall the definition of the $\oplus_z$-operation:

$$\begin{aligned} Q \oplus_z R &= \{\langle Q^{+x_1}, R^{+x_2}\rangle : x_1 + x_2 = z\} \\ &= \{\langle Q^{+y_1+x_1}, R^{+y_2+x_2}\rangle : x_1 + y_1 + x_2 + y_2 = z\} \\ &= \{\langle (Q^{+y_1})^{+x_1}, (R^{+y_2})^{+x_2}\rangle : x_1 + x_2 = y\} = Q^{+y_1} \oplus_y R^{+y_2}. \end{aligned}$$

To prove (b) we proceed by induction on $m \geq 2$. The case $m = 2$ already follows from (a), so let us assume that $m > 2$. Then we have that

$$\begin{aligned} T_0^{+y_0} \oplus T_1^{+y_1} \oplus \cdots \oplus T_{m-1}^{+y_{m-1}} &= (T_0^{+y_0} \oplus T_1^{+y_1} \oplus \cdots \oplus T_{m-2}^{+y_{m-2}}) \oplus T_{m-1}^{y_{m-1}} \\ \text{(IH)} &= (T_0 \oplus T_1 \oplus \cdots \oplus T_{m-2})^{+y_0+y_1+\cdots+y_{m-2}} \oplus T_{m-1}^{+y_{m-1}} \\ \text{(a)} &= (T_0 \oplus T_1 \oplus \cdots \oplus T_{m-1})^{+y_0+y_1+\cdots+y_{m-1}}. \end{aligned}$$

To prove (c) we proceed by induction on $1 < z \leq d-1$. The claim follows from the above for the case $z = 2$, so assume that $z > 2$. We observe that $T = ((z-1) \odot Q) \oplus Q$ and thus $T^{+y} = ((z-1) \odot Q) \oplus_y Q$. By (a) this means that $T^{+y} = ((z-1) \odot Q)^{+(y-y_2)} \oplus Q^{+y_2}$. Note that $y - y_2 = (z-1) \cdot y_1$. Since $z - 1 \geq 2$, we can use the induction hypothesis to see that

$$((z-1) \odot Q)^{+(z-1)\cdot y_1} = ((z-2) \odot Q^{+y_1}) \oplus Q^{+y_1} = (z-1) \odot Q^{+y_1}.$$

Now (d) directly follows from (c), since $T^{+y \cdot z} = ((z-1) \odot Q^{+y}) \oplus Q^{+y} = z \odot Q^{+y}$.

For the remaining claims we also proceed by induction on $1 \leq z \leq d-1$ where the cases for $z = 1$ are trivial. Thus let us assume that $1 < z < d$. Then $T = ((z-1) \odot Q) \oplus Q$. Hence $T[\alpha] = ((z-1) \odot Q)[\alpha] + Q[\alpha] = z \cdot Q[\alpha]$ by the induction hypothesis. The same reasoning shows the connection for the coefficients $c_i(T)$. □

Our next aim is to uncover a very strong connection between the action of $\Gamma$ on the class of hyperterms, on the class of assignments $\mathbb{L}$, and on the class of linear terms. More specifically, it turns out that the $\Gamma$-orbit $\Gamma(T)$ of a hyperterm $T$ is a subset of the set of shifted hyperterms $\{T^{+z} : z \in \mathbb{Z}_d\}$. Moreover, by property (L), we can associate to every hyperterm $T$ a set of equivalent linear terms $L(T)$, and we show in Lemma 5.8 that the action of $\Gamma$ on $\Gamma(T) \subseteq \{T^{+z} : z \in \mathbb{Z}_d\}$ is in a one-to-one correspondence to the action of $\Gamma$ on $\Gamma(L(T))$. Hence, a hyperterm $T$ indeed is a succinct representation of the class of equivalent linear terms $L(T)$.

**Remark 5.6.** *The group $\Gamma$ acts transitively on $\mathbb{L}$: for $\pi \in \Gamma$ and $\alpha \in \mathbb{L}$ we define $\pi(\alpha) \in \mathbb{L}$ as the assignment given by*

$$\pi(\alpha)(v) := \alpha(\pi(v)).$$

**Definition 5.7.** For $\alpha \in \mathbb{L}$, $i \in [n]$, and $z \in \mathbb{Z}_d$ we let $\alpha^{i:+z} \in \mathbb{L}$ denote the *semantical z-shift of block $V_i$* for the assignment $\alpha$ which is defined as $\alpha^{i:+z}(v) := \alpha(v) + z$ for $v \in V_i$ and $\alpha^{i:+z}(v) = \alpha(v)$ for $v \notin V_i$. In other words, we let $\alpha^{i:+z} = \pi^{i:+z}(\alpha)$.

The following lemma characterises the $\Gamma$-symmetries of hyperterms and makes the strong connection between their syntax and semantics precise.

**Lemma 5.8.** *Let $T$ be a hyperterm and let $c_i = c_i(T) \in \mathbb{Z}_d$ be the coefficient of variable block $V_i$ in $T$ for $i \in [n]$.*

*(a) For $i \in [n]$, $z \in \mathbb{Z}_d$ we have $\pi^{i:+z}(T) = T^{+c_i \cdot z}$. In particular if $c_i = 0$, then $\pi^{i:+z}(T) = T$.*

*(b) For all $\pi \in \Gamma$, for all hyperterms $Q, R$, and for all constants $y, z \in \mathbb{Z}_d$, $z \neq 0$, we have $\pi(Q \oplus_y R) = \pi(Q) \oplus_y \pi(R)$ and $\pi(z \odot Q) = z \odot \pi(Q)$.*

*(c) For $\alpha \in \mathbb{L}$ we have $T[\alpha^{i:+z}] = \pi^{i:+z}(T)[\alpha]$ for $i \in [n]$, $z \in \mathbb{Z}_d$. It follows that $T[\pi(\alpha)] = \pi(T)[\alpha]$ for all $\pi \in \Gamma$.*

*(d) The group $\Gamma$ acts on $\{T^{+y} : y \in \mathbb{Z}_d\}$. Moreover, $\pi \in \Gamma$ pointwise stabilises a hyperterm $T^{+y}$ if, and only if, $\sum_{i=0}^{n-1} \pi(i) \cdot c_i = 0$.*

*Proof.* We first simultaneously prove (a), (b) and (c) by an induction on the structure of hyperterms. Note that in order to prove (c) it suffices to show that $T[\alpha^{i:+z}] = T[\alpha] + c_i \cdot z$ when we assume that (a) holds. The cases for atomic hyperterms are trivial.

- *Addition of hyperterms.* Let $Q, R$ be hyperterms and let $T = Q \oplus_y R$ for $y \in \mathbb{Z}_d$. Then $c_i = q_i + r_i$ where $q_i = c_i(Q)$ and $r_i = c_i(R)$. Moreover,

$$\begin{aligned}
\pi^{i:+z}(T) &= \pi^{i:+z}(\{\langle Q^{+y_1}, R^{+y_2}\rangle : y_1 + y_2 = y\}) \\
&\overset{\text{(IH)}}{=} \{\langle Q^{+y_1 + z\cdot q_i}, R^{+y_2 + z \cdot r_i}\rangle : y_1 + y_2 = y\} \\
&= Q^{+z\cdot q_i} \oplus_y R^{+z\cdot r_i} \\
&\overset{\text{(Lemma 5.5 (a))}}{=} Q \oplus_{y+z\cdot c_i} R = T^{+z\cdot c_i}.
\end{aligned}$$

In particular this shows that $\pi^{i:+z}(Q \oplus_y R) = \pi^{i:+z}(Q) \oplus_y \pi^{i:+z}(R)$. Moreover, we have $T[\alpha^{i:+z}] = Q[\alpha^{i:+z}] + R[\alpha^{i:+z}] + y$. Thus by the induction hypothesis we have $T[\alpha^{i:+z}] = Q[\alpha] + z \cdot q_i + R[\alpha] + z \cdot r_i + y = T[\alpha] + z \cdot c_i$.

- *Scalar multiplication.* Let $T = y \odot Q$ for $y \in \mathbb{Z}_d$ and a hyperterm $Q$. Then, by Lemma 5.5 (f), we have $c_i = y \cdot q_i$ where $q_i = c_i(Q)$. We proceed by induction on $1 \leq y < d$. If $y = 1$, then the claim follows from the induction hypothesis for $Q$. For $y > 1$, let $T = (y-1) \odot Q \oplus Q$. From the above and from the induction hypothesis we know that

$$\begin{aligned}
\pi^{i:+z}(T) &= \pi^{i:+z}((y-1) \odot Q \oplus Q) \\
&\overset{\text{(IH)}}{=} \big((y-1) \odot \pi^{i:+z}(Q)\big) \oplus \pi^{i:+z}(Q) = y \odot \pi^{i:+z}(Q) \\
&\overset{\text{(IH)}}{=} y \odot Q^{+q_i \cdot z} \\
&\overset{\text{(Lemma 5.5 (d))}}{=} T^{+z \cdot c_i}.
\end{aligned}$$

In particular we have $\pi^{i:+z}(y \odot Q) = y \odot \pi^{i:+z}(Q)$ and by Lemma 5.5 (e) we have that $T[\alpha^{i:+z}] = y \cdot Q[\alpha^{i:+z}]$ and $T[\alpha] = y \cdot Q[\alpha]$. Thus by the induction hypothesis we have $T[\alpha^{i:+z}] = y \cdot (Q[\alpha] + q_i \cdot z) = y \cdot Q[\alpha] + c_i \cdot z = T[\alpha] + c_i \cdot z$.

Finally, (d) follows from (a), since $\Gamma$ is generated by the elements $\pi^{i:+z}$. □

### 5.2.2 Hyperterms in Choiceless Polynomial Time

We now turn our attention to the manipulation of hyperterms in Choiceless Polynomial Time. It is obvious that atomic hyperterms can be defined in CPT. In the following lemma we show that also the basic operations of addition and scalar multiplication can be realised by a CPT-program in such a way that the number of newly created hereditarily finite sets is bounded by a polynomial in $d$. This shows that the iterated application of the $\{\oplus, \odot\}$-operations to hyperterms only polynomially increases the size of the resulting hyperterms (seen as objects in $\mathrm{HF}(V)$) with respect to $d$ and the number of such operations. This insight will play an important role in Section 5.3 where we consider a variant of Gaussian elimination applied to hyperterms.

**Lemma 5.9.** *There are* CPT*-programs* $\Pi^{\oplus}$ *and* $\Pi^{\odot}$ *and a polynomial* $p : \mathbb{N} \to \mathbb{N}$ *such that, given a cyclic linear equation* $(V, S, \leq)$ *over* $\mathbb{Z}_d$*, and hyperterms* $Q, R$*,*

- $\Pi^{\oplus}$ *constructs the hyperterms* $T = Q \oplus R$ *and* $T^{+z}$ *for* $z \in \mathbb{Z}_d$*, and*
- $\Pi^{\odot}$ *constructs the hyperterms* $T = y \odot Q$ *and* $T^{+z}$ *for* $y, z \in \mathbb{Z}_d, y \neq 0$*.*

*Moreover, these hyperterms* $T$ *are created by activating at most* $p(d)$ *new objects from* $\mathrm{HF}(V)$*, that is* $\mathrm{TC}(T) \subseteq \bigcup_{z \in \mathbb{Z}_d} \mathrm{TC}(Q^{+z}) \cup \mathrm{TC}(R^{+z}) \cup N$ *where* $N \subseteq \mathrm{TC}(T)$ *and* $|N| \leq p(d)$*.*

*Proof.* We first consider the case $T = Q \oplus R$ (the case $T^{+z} = Q \oplus_z R$ is analogous). By definition we have $T = \{\langle Q^{+z_1}, R^{+z_2}\rangle : z_1 + z_2 = 0\}$. Clearly this object can be created within CPT by using comprehension terms. Moreover, we have

$$\mathrm{TC}(T) \subseteq \bigcup_{z \in \mathbb{Z}_d} \mathrm{TC}(Q^{+z}) \cup \mathrm{TC}(R^{+z}) \cup \{Q^{+z_1}, R^{+z_2}, \langle Q^{+z_1}, R^{+z_2}\rangle : z_1 + z_2 = 0\}.$$

Hence, it suffices to choose the polynomial $p$ such that $p(d) \geq 3 \cdot d$.

We proceed with the case $T = y \odot Q$ for $y \in \mathbb{Z}_d$. Since the $\odot$-operation is defined recursively via the $\oplus$-operation it is clear from the above that there exists a CPT-program which constructs the hyperterm $T$. To show that the number of new objects that have to be created is bounded by a polynomial in $d$, we proceed by induction on $0 < y < d$. More precisely we show that for the hyperterm $S_y := y \odot Q$ we have

$$\bigcup_{z \in \mathbb{Z}_d} \mathrm{TC}(S_y^{+z}) \subseteq \bigcup_{z \in \mathbb{Z}_d} \mathrm{TC}(Q^{+z}) \cup N_y$$

where $N_y \subseteq \bigcup_{z \in \mathbb{Z}_d} \mathrm{TC}(S_y^{+z})$ and $|N_y| \leq y \cdot 3 \cdot d^2$. If we can show this, then it suffices to choose $p \geq 3 \cdot d^3$ to obtain our original claim.

The case $y = 1$ is trivial and the case $y = 2$ follows from the above. Thus let $y > 2$. Then for all $T^{+x} = S_{y-1} \oplus_x Q$ for $x \in \mathbb{Z}_d$ we have

$$\mathrm{TC}(T^{+x}) \subseteq \bigcup_{z \in \mathbb{Z}_d} \mathrm{TC}(S_{y-1}^{+z}) \cup \mathrm{TC}(Q^{+z}) \cup \{S_{y-1}^{+z_1}, Q^{+z_2}, \langle S_{y-1}^{+z_1}, Q^{+z_2}\rangle : z_1 + z_2 = x\}.$$

By the induction hypothesis we can find $N_{y-1} \subseteq \bigcup_{z \in \mathbb{Z}_d} \mathrm{TC}(S_{y-1}^{+z})$ such that $|N_{y-1}| \leq (y-1) \cdot 3 \cdot d^2$ and such that

$$\bigcup_{x \in \mathbb{Z}_d} \mathrm{TC}(T^{+x}) \subseteq \bigcup_{z \in \mathbb{Z}_d} \mathrm{TC}(Q^{+z}) \cup N_{y-1} \cup \{S_{y-1}^{+z_1}, Q^{+z_2}, \langle S^{+z_1}, Q^{+z_2}\rangle : z_1, z_2 \in \mathbb{Z}_d\}.$$

Since for $N_y := N_{y-1} \cup \{S_{y-1}^{+z_1}, Q^{+z_2}, \langle S^{+z_1}, Q^{+z_2}\rangle : z_1, z_2 \in \mathbb{Z}_d\}$ we have $|N_y| \leq (y-1) \cdot 3 \cdot d^2 + 3 \cdot d^2 = y \cdot 3 \cdot d^2$, the claim follows. □

Besides addition and scalar multiplication, another very important basic operation is the evaluation of a hyperterm $T$ with respect to an assignment $\alpha \in \mathbb{L}$. On the other hand, during the run of a CPT-program we will, in general, never be in a situation where we have access to a complete assignment $\alpha \in \mathbb{L}$. The reason is that $\Gamma$ acts transitively on $\mathbb{L}$. Since the automorphism group of the given cyclic linear equation system is a (not necessarily strict) subgroup of $\Gamma$, and since the size of $\mathbb{L}$ is exponential in the number of variables, it is not possible to identify a single assignment $\alpha$ in the run of a CPT-program, since this would require to activate the whole orbit of $\alpha$ (which, as we have just explained, can be the complete solution set $\mathbb{L}$). Still, there is one very important special case in which the evaluation of $T[\alpha]$ makes sense although we cannot access $\alpha \in \mathbb{L}$ explicitly.

**Definition 5.10.** A hyperterm $T$ is *constant* if $c_i(T) = 0$ for all $i \in [n]$.

Let us explain more precisely in which sense a hyperterm $T$ with $c_i(T) = 0$, for all $i \in [n]$, is "constant". To this end, let $\alpha, \beta \in \mathbb{L}$ and let $\pi \in \Gamma$ such that $\pi(\alpha) = \beta$. By Lemma 5.8 (d) we have that $\pi(T) = T$ since $c_i(T) = 0$ for all $i \in [n]$ and, moreover, by Lemma 5.8 (c) we know that

$$T[\beta] = T[\pi(\alpha)] = \pi(T)[\alpha] = T[\alpha].$$

Thus the hyperterm $T$ actually has a *constant value* $c_T = T[\alpha]$ for all assignments $\alpha \in \mathbb{L}$. It turns out that this value can be defined in CPT.

**Lemma 5.11.** *There is a* CPT*-program which defines, given a cyclic linear equation system* $(V, S, \preceq)$ *over* $\mathbb{Z}_d$ *as above, and a constant hyperterm* $T$*, the value* $c_T \in \mathbb{Z}_d$ *of* $T$*.*

*Proof.* We describe a CPT-program which proceeds in two stages: first, it syntactically substitutes every occurrence of a variable $v \in V$ in the hyperterm $T$ by its value $\alpha(v) \in \mathbb{Z}_d$ for some assignment $\alpha \in \mathbb{L}$. Secondly, it takes the resulting hyperterm $T' \in \mathrm{HF}(\mathbb{Z}_d)$ and extracts from it the value $c_T = T[\alpha] \in \mathbb{Z}_d$ of the hyperterm $T$. Since $T[\alpha]$ is independent of the specific assignment $\alpha \in \mathbb{L}$, this approach is sound.

However, we already explained above that we cannot fix an assignment $\alpha \in \mathbb{L}$, and thus the question remains how we can express the first step in Choiceless Polynomial Time. The key idea is to exploit the high degree of symmetry of the hyperterm $T$. In fact, since $c_i = c_i(T) = 0$ for all $i \in [n]$, we know by Lemma 5.8 that $\pi(T) = T$ for all $\pi \in \Gamma$. Hence, the result of syntactically replacing every occurrence of a variable $v \in V$ in $T$ by $\alpha(v)$ gives the same hyperterm as when we substitute every $v \in V$ by $\pi(\alpha)(v)$ for any $\alpha \in \mathbb{L}$. Formally we have

$$T[v \mapsto \alpha(v)] = \pi^{-1}(T)[v \mapsto \pi(\alpha)(v)] = T[v \mapsto \pi(\alpha)(v)].$$

Hence, if we would make the substitution $v \mapsto \alpha(v)$ in parallel for all $\alpha \in \mathbb{L}$, then we would obtain a *single* hyperterm $T' \in \mathrm{HF}(\mathbb{Z}_d)$ whose value coincides with the value $c_T$ of $T$.

Still, the set $\mathbb{L}$ is too large to make this substitution in parallel for all $\alpha \in \mathbb{L}$. Instead we use the decomposition $\mathbb{L} = \mathbb{L}(C_0) \times \cdots \times \mathbb{L}(C_{n-1})$ of $\mathbb{L}$ into the small parts of valid assignments $\mathbb{L}(C_i)$ for block $V_i$ and then proceed inductively: for $i \in [n]$ let $T_i$ denote the hyperterm which results by substituting in $T$ every occurrence of a variable $v \in (V_0 \uplus \cdots \uplus V_i)$ by $\alpha(v)$ for $\alpha \in \mathbb{L}(C_0) \times \cdots \times \mathbb{L}(C_i)$. Then $T_{i+1}$ can easily be constructed from $T_i$ in CPT by substituting all $v \in V_{i+1}$ by $\alpha(v)$ for $\alpha \in \mathbb{L}(C_{i+1})$ in parallel. With the same reasoning as above it follows that the resulting hyperterm $T_{i+1}$ is independent of the specific assignment $\alpha \in \mathbb{L}(C_{i+1})$. We let $T' := T_{n-1} \in \mathrm{HF}(\mathbb{Z}_d)$. While it is clear that the value of $T'$ coincides with the value $c_T$ of $T$, in contrast to $T$, the hyperterm $T'$ does not contain any variables from $V$ as atoms. For such hyperterms we can easily determine their value in CPT.

**Claim:** There is a CPT-definable mapping $\rho : \mathrm{HF}(\mathbb{Z}_d) \to \mathbb{Z}_d$ which maps a hyperterm $R \in \mathrm{HF}(\mathbb{Z}_d)$ to its value $\rho(R) = c_R \in \mathbb{Z}_d$.
*Proof of claim:* If $R$ is a constant, the claim is trivial. Otherwise, $R$ is built from the constants in $\mathbb{Z}_d$ and the $\oplus$-operation (since also the scalar multiplication is a shorthand for an iterated application of $\oplus$). This means that $R$ must contain a subhyperterm of the form $z_1 \oplus z_2 = \{\langle z_1 + y_1, z_2 + y_2\rangle : y_1 + y_2 = 0\}$. By substituting every such subhyperterm in $R$ by its value $(z_1 + z_2) \in \mathbb{Z}_d$ we can iteratively reduce $R$ to a constant $c$ in $\mathbb{Z}_d$ which coincides with its value. Of course the number of such substitutions is bounded by the size of $R$ and introduces at most $|R|$ many new objects for each iteration which shows that $\rho$ can be defined in CPT. ⊣
Finally, by combining the routines to transform $T$ into $T'$ and to evaluate $T'$, we obtain a CPT-program which defines the value of $T$. □

## 5.3 Solving ordered systems of hyperequations

In this section we obtain our main result of this chapter.

**Theorem 5.12.** *The solvability problem for cyclic linear equation systems is definable in Choiceless Polynomial Time.*

Let us briefly sketch our proof plan. First of all, we show that we can transform the equations of a given cyclic linear equation system into equivalent *hyperequations*. Such hyperequations differ from usual linear equations by the simple fact that they are formed from hyperterms instead of usual linear terms. Formally, a *hyperequation* is a pair $T = z$ consisting of a hyperterm $T$ and a constant $z \in \mathbb{Z}_d$. Thus, in view of our results from Section 5.2, a hyperequation is a succinct representation of a set of equivalent linear equations. Since we defined for each $\alpha \in \mathbb{L}$ and each hyperterm $T$ a value $T[\alpha] \in \mathbb{Z}_d$, the notion of solvability (with respect to $\mathbb{L}$-assignments) transfers to hyperequations and systems of hyperequations. Moreover, we will show that we can translate the linear equations of a cyclic linear equation system in such a way that we obtain a *linear order* on the set of resulting hyperequations. Secondly, we use the CPT-definability of basic operations on hyperterms, that is of addition, scalar multiplication, and the evaluation of constant hyperterms, to express in CPT a variant of Gaussian elimination to decide the solvability problem for ordered systems of hyperequations.

We remark that whenever we speak of *equivalent* linear terms or equations, then this equivalence is relativised to assignments in $\mathbb{L}$ as formalised in the following definition.

**Definition 5.13.** Two linear terms $s, t$ are *equivalent* ($s \equiv t$) if $s(\alpha) = t(\alpha)$ for all $\alpha \in \mathbb{L}$. Two linear equations $(s = z_s)$, $(t = z_t)$ are *equivalent* if $s - t \equiv z_s - z_t$.

Our next aim is to establish a CPT-definable preorder $\preceq$ on the set of linear equations $S$ in such a way that classes $S_i$ of $\preceq$-equivalent equations are

equivalent. As explained above we will, in a further step, transform each such class $S_i$ into a single hyperequations $T_i = z_i$. As a matter of fact, the preorder $\preceq$ on $S$ induces a linear order on the set of hyperequations $\{T_i = z_i\}$. In this way we can reduce the solvability problem of a cyclic linear equation system to the solvability problem of an *ordered* system of hyperequations.

The crucial step of the transformation $S_i \mapsto (T_i = z_i)$ is to translate linear terms $t$ into equivalent hyperterms $T_t$ in such a way that equivalent linear terms $t, s$ are mapped to the *same* hyperterm $T_s = T_t$. To guarantee this, we have to establish a certain normal form for linear terms. While, in general, a linear term $t$ may contain more than one variable from each of the blocks $V_i$, it is easy to see that every such linear term $t$ can be rewritten as a linear term $t'$ of the form

$$t' = (\sum_{i=0}^{n-1} z_i \cdot v_i) + y \text{ for } z_i, y \in \mathbb{Z}_d, v_i \in V_i.$$

The reason is that every variable in $V_i$ can be expressed by a linear term in any other variable from $V_i$ due to the cyclic constraint $C_i$. More formally, for every pair $v, w \in V_i$ we can find $z \in \mathbb{Z}_d$ such that $v \equiv w + z$. In particular this shows the following.

**Lemma 5.14.** *Let $t = z_t \cdot v_t$ and $s = z_s \cdot v_s$ be two atomic linear terms with $v_t, v_s \in V_i$ and $z_s, z_t \in \mathbb{Z}_d$. If $t \equiv s$, then $z_t = z_s$.*

*Proof.* Let $v_t \equiv v_s + y$ for $y \in \mathbb{Z}_d$. If $t \equiv s$, then $z_t \cdot (v_s + y) \equiv z_s \cdot v_s$, and hence the linear term $(z_t - z_s) \cdot v_s$ is equivalent to a constant. This, in turn, is only possible if $z_t = z_s$, since otherwise $(z_t - z_s) \cdot 1 \neq (z_t - z_s) \cdot 0$. □

Before we proceed, a small remark is in place. As we explained earlier, we aim to translate linear equations $t = z$ into equivalent hyperequations $T = z$. For this step, we can assume that, without loss of generality, the linear term $t$ does not contain any constant linear terms, because such constants $y \in t$ could clearly be combined with $z$. In particular, for the equivalent representation as $t' = (\sum_{i=0}^{n-1} z_i \cdot v_i) + y$ for $z_i, y \in \mathbb{Z}_d, v_i \in V_i$, from above we can assume that $y = 0$. For what follows, we agree on this implicit assumption.

In general, each linear term $t$ can be decomposed into an ordered sequence of linear subterms $t_{*i} \subseteq t$ over the variable blocks $V_i$. Formally, let $t_{*i}$ denote the linear subterm $t_{*i} := \{z \cdot v \in t : v \in V_i, z \in \mathbb{Z}_d\}$. Then $t = \biguplus_{i \in [n]} t_{*i}$. By the reason explained above, we can assume that each of the subterms $t_{*i}$ only contains a single variable from the block $V_i$ (but this variable is not uniquely determined).

**Lemma 5.15.** *There is a* CPT*-program* $\Pi$ *such that, given a CES as above and a linear term $t$ in which only variables from the block $V_i$ occur, the program* $\Pi$ *outputs a constant $y \in \mathbb{Z}_d$, a coefficient $z \in \mathbb{Z}_d$ and a set of variables $W \subseteq V_i$ such that every linear term $z \cdot v + y$ for $v \in W$ is equivalent to $t$.*

*Proof.* Let $v \in V_i$. For each $w \in V_i$ we find a constant $z_w \in \mathbb{Z}_d$ such that $w \equiv v + z_w$. We replace each variable $w \in V_i$ in the given term $t$ by the equivalent term $v + z_w$ and simplify the resulting expression afterwards. In this way we obtain for every $v \in V_i$ an atomic linear term $z_v \cdot v + y_v$ for $y_v, z_v \in \mathbb{Z}_d$, such that $t \equiv z_v \cdot v + y_v$.

Assume for $v, w \in V_i$ we have that $y_v = y_w$. Then $z_v \cdot v \equiv z_w \cdot w$. By Lemma 5.14 we have $z_v = z_w$. We fix the minimal $y \in \mathbb{Z}_d$ and the corresponding $z \in \mathbb{Z}_d$ such that $(y, z) = (y_v, z_v)$ for some $v \in V_i$ and set $W := \{v \in V_i : (y_v, z_v) = (y, z)\}$. Then $y, z$ and $W$ satisfy the claim. □

We are prepared to specify the CPT-definable preorder $\preceq$ on the set $S$ of linear equations. By Lemma 5.15 we can construct in CPT for a given linear term $t$ and each $i \in [n]$ a unique pair $\sigma_i = (y_i, z_i) \in \mathbb{Z}_d \times \mathbb{Z}_d$ such that $t_{*i} \equiv z_i \cdot v + y_i$ for suitable $v \in V_i$. We define the *signature* $\operatorname{sgn}(t) = (\sigma_i)_{i \in [n]} \in (\mathbb{Z}_d \times \mathbb{Z}_d)^n$ of a linear term $t$ as the sequence consisting of these pairs $\sigma_i = (y_i, z_i)$. In this way we obtain a CPT-definable preorder $\preceq$ on $S$ which is given by

$$(t, z) \preceq (s, z') \text{ if, and only if, } \operatorname{sgn}(t) < \operatorname{sgn}(s) \text{ or } (\operatorname{sgn}(t) = \operatorname{sgn}(s) \text{ and } z \leq z').$$

As usual, we write $S = S_0 \preceq \cdots \preceq S_{m-1}$ and we say that $S_i$ is the $i$-th block of $\preceq$-equivalent equations. Let $(t, z), (s, z) \in S_i$. We claim that either we have $t \equiv s$ or that the given cyclic linear equation system is inconsistent. To see this, first note that the linear equation $t - s = 0$ is a consequence of the two linear equations $t = z$ and $s = z$.

Of course this does not immediately imply that $t \equiv s$. In fact it still might be the case that for some $\alpha \in \mathbb{L}$ we have $t[\alpha] = s[\alpha]$, but that for some $\beta \in \mathbb{L}$ it holds that $t[\beta] \neq s[\beta]$. However, since $\operatorname{sgn}(t) = \operatorname{sgn}(s)$ it easily follows that the linear term $t - s$ is equivalent to a constant in $\mathbb{Z}_d$. Hence, from now on we can assume that for each pair of linear equations $(s, z), (t, z) \in S_i$ the linear terms $s$ and $t$ are equivalent.

### 5.3.1 From linear equations to hyperequations

Our next aim is to establish the CPT-definable mapping $t \mapsto T_t$ which translates linear terms $t$ into equivalent hyperterms $T_t$ with the property that equivalent linear terms $s, t$ with $\operatorname{sgn}(s) = \operatorname{sgn}(t)$ are mapped to the same hyperterm $T_s = T_t$. As a first preparation, let us consider the important special case of atomic linear terms.

**Lemma 5.16.** *Let $s = z \cdot v$ and $t = z \cdot w$ be equivalent linear terms with $z \in \mathbb{Z}_d, z \neq 0$ and $v, w \in V_i$. Then $z \odot w = z \odot v$.*

*Proof.* Let $v - w \equiv y \in \mathbb{Z}_d$. Then $z \cdot y = 0$ and $z \odot w^{+y} = z \odot v$. By Lemma 5.5 (d) we conclude that $z \odot v = z \odot w^{+y} = (z \odot w)^{+z \cdot y} = z \odot w$. □

The preceding lemma allows us to specify the CPT-definable translation $t \mapsto T_t$:

- *Translating subterms $t_{*i} \mapsto T_i$.* Let $i \in [n]$. To translate the subterm $t_{*i}$ into an equivalent hyperterm $T_i$ we first apply Lemma 5.15 to define constants $y_i, z_i \in \mathbb{Z}_d$ and a set $W_i \subseteq V_i$ such that $t_{*i} \equiv z_i \cdot w + y_i$ for all $w \in W_i$. If $z_i = 0$, then we just set $T_i := y_i$.

  Otherwise, we observe that for two different $w, w' \in W_i$ we have $z_i \cdot w \equiv z_i \cdot w'$. By Lemma 5.16 we conclude that $z_i \odot w = z_i \odot w'$. Thus we can set $T_i := (z_i \odot w)^{+y_i}$ for some (any) $w \in V_i$ to obtain a single hyperterm $T_i$ which is equivalent to $t_{*i}$.

- *Combining the partial hyperterms.* We obtain $T_t$ as $T_t := T_1 \oplus \cdots \oplus T_n$.

First of all, by the definition of hyperterms and their semantics it is obvious that $T_t$ is equivalent to $t$ in the sense that for all $\alpha \in \mathbb{L}$ we have $T_t[\alpha] = t[\alpha]$. Moreover, from our previous results it is clear that the above described translation $t \mapsto T_t$ is definable in Choiceless Polynomial Time. In the next lemma we prove that it also has the desired property of merging equivalent linear terms into a single hyperterm.

**Lemma 5.17.** *Let $s, t$ be two linear terms with $sgn(s) = sgn(t)$. Then $t - s \equiv \delta$ for some $\delta \in \mathbb{Z}_d$ and we have $T_t = T_s^{+\delta}$. In particular, if $s \equiv t$, then $T_t = T_s$.*

*Proof.* Let $\mathrm{sgn}(s) = \mathrm{sgn}(t) = ((y_0, z_0), \ldots, (y_{n-1}, z_{n-1}))$. Then for $i \in [n]$ we have that $t_{*i} \equiv z_i \cdot w_t + y_i$ and $s_{*i} \equiv z_i \cdot w_s + y_i$ for appropriate sets $W_i^t, W_i^s \subseteq V_i$ and $w_t \in W_i^t$, $w_s \in W_i^s$. This shows that $t_{*i} - s_{*i} \equiv z_i \cdot c_i =: \delta_i$ for an appropriate $c_i \in \mathbb{Z}_d$ with $w_t - w_s \equiv c_i$. Hence $t - s \equiv \sum_{i \in [n]} \delta_i =: \delta$.

**Claim:** If $t_{*i} \mapsto T_i$ and $s_{*i} \mapsto S_i$, then $T_i = S_i^{+\delta_i}$.
*Proof of claim:* First of all, if $z_i = 0$, then $\delta_i = 0$ and $T_i = S_i = S_i^{+\delta_i}$. Otherwise we have that $T_i = (z_i \odot w_t)^{+y_i}$ and $S_i = (z_i \odot w_s)^{+y_i}$ for all $w_t \in W_i^t$ and $w_s \in W_i^s$. Thus we can conclude by Lemma 5.5 (d) that $T_i = (z_i \odot w_t)^{+y_i} = (z_i \odot w_s^{+c_i})^{+y_i} = (z_i \odot w_s)^{+\delta_i + y_i} = S_i^{+\delta_i}$. ⊣

Finally, by the above claim we can conclude that

$$\begin{aligned} T_t &= T_0 \oplus \cdots \oplus T_{n-1} \\ &= S_0^{+\delta_0} \oplus \cdots \oplus S_{n-1}^{+\delta_{n-1}} \\ \text{Lemma 5.5 (b)} &= (S_0 \oplus \cdots \oplus S_{n-1})^{+\delta_0 + \cdots + \delta_{n-1}} \\ &= T_s^{+\delta}. \end{aligned}$$

□

Lemma 5.17 shows that we can use the CPT-definable translation $t \mapsto T_t$ to map each block $S_i$ of equivalent linear equations into a single hyperequation $T_i = z$ for $i \in [m]$. Moreover, the linear order on the blocks $S_i$ induces a linear order on the set of hyperequations $S^* := \{T_i = z_i : i \in [m]\}$. Altogether, we have shown that there exists a CPT-program which translates each cyclic linear equation system into an equivalent and ordered system of hyperequations. We proceed to prove that we can express the solvability of such ordered systems of hyperequations in Choiceless Polynomial Time.

### 5.3.2 Gaussian elimination for systems of hyperequations

To express the solvability problem for ordered systems of hyperequations in Choiceless Polynomial Time, we use a slightly adapted variant of Gaussian elimination. Besides the fact that we are dealing with hyperequations, the modifications are necessary since we are working with linear equation systems over a finite *ring* $\mathbb{Z}_d$ (instead of a finite *field*).

As a first preparation we note that the elementary transformations which are necessary to implement Gaussian elimination can also be applied to systems of hyperequations by using the operations $\oplus$ and $\odot$ on the class of hyperterms.

**Definition 5.18.** Let $S^*$ be a system of $m$ hyperequations as above. Moreover, let $i, j \in [m]$, $i \neq j$ and let $(T, z), (T', z') \in S^*$ denote the $i$-th and $j$-th hyperequation in $S^*$, respectively, and let $y \in \mathbb{Z}_d \setminus \{0\}$. Then we set

$$S^*[i \mapsto i + y \cdot j] := (S^* \setminus \{(T, z)\}) \cup \{(T \oplus (y \odot T'), z + y \cdot z')\}.$$

To put it in words, the preceding definition formalises what it means to apply the basic operation of "row addition" on the level of hyperequations: we replace in $S^*$ the $i$-th hyperequation by its $\oplus$-sum with a $\odot$-scalar multiple of the $j$-th hyperequation. The following lemma states that this operation preserves the set of solutions.

**Lemma 5.19.** *For every $i, j \in [m]$, $i \neq j$ and every $y \in \mathbb{Z}_d \setminus \{0\}$ the system $S^*[i \mapsto i + y \cdot j]$ is equivalent to $S^*$, that is $\alpha \in \mathbb{L}$ is a solution of $S^*$ if, and only if, it is a solution of $S^*[i \mapsto i + y \cdot j]$.*

*Proof.* This follows by the semantics of hyperterms and Lemma 5.5 (e). □

Of course, we can also change the order of hyperequations in $S^*$ without affecting the set of solutions, and we identify this operation of permuting hyperequations with the elementary transformation of "row permutations". For $\pi \in \mathrm{Sym}([m])$ we let $S^*[i \mapsto \pi(i)]$ denote the result of permuting the hyperequations in $S^*$ according to $\pi$. Essentially, these two basic kinds of transformations, that is "row additions" and "row permutations", suffice to carry out the method of Gaussian elimination.

To make this connection more precise, we associate to the ordered system of hyperequations $S^* = \{T_i = z_i : i \in [m]\}$ the corresponding $m \times n$-coefficient matrix $M[S^*] : [m] \times [n] \to \mathbb{Z}_d$ which is defined as

$$M[S^*](i, j) := c_j(T_i).$$

Then $M[S^*]$ is a matrix over the two *ordered* index sets $[m]$ and $[n]$.

Now, permuting the rows of $M[S^*]$ corresponds to permuting the hyperequations in $S^*$: for $\pi \in \mathrm{Sym}([m])$ we have that $\Pi \cdot M[S^*] = M[S^*[i \mapsto \pi(i)]]$ where $\Pi$ denotes the $[m] \times [m]$-permutation matrix associated to $\pi$. Similarly,

if we denote for $i, j \in [m]$, $i \neq j$ and $y \in \mathbb{Z}_d \setminus \{0\}$ the elementary operation of adding the $y$-multiple of row $j$ to row $i$ by $X[i \mapsto i + y \cdot j] \in \mathbb{Z}_d^{[m]\times[m]}$, then

$$X[i \mapsto i + y \cdot j] \cdot M[S^*] = M[S^*[i \mapsto i + y \cdot j]].$$

We can use this one-to-one correspondence between the application of elementary operations to $M[S^*]$ and $S^*$ to bring the system $S^*$ into a particular kind of normal form. Specifically, we say that the coefficient matrix $M[S^*]$ is in *Hermite normal form* if there exists a permutation matrix $P \in \{0,1\}^{[n]\times[n]}$ such that

$$M[S^*] \cdot P = \begin{pmatrix} a_{0,0} & \cdots & \cdots & \cdots & a_{0,(n-1)} \\ 0 & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & a_{(k-1),(k-1)} & \cdots & a_{(k-1),(n-1)} \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

where $a_{0,0} \mid a_{1,1} \mid \cdots \mid a_{(k-1),(k-1)}$ and such that for all $i \in [k]$ and $j \in [n]$ it holds that $a_{i,i} \mid a_{i,j}$. Accordingly, we say that $S^*$ is in Hermite normal form, if $M[S^*]$ is so.

An important structural property of the ring $\mathbb{Z}_d$ is that divisibility is a total preorder (which is not longer true if $d$ is composed of distinct primes). By using this property one can transform any $[m] \times [n]$-matrix $M$ over $\mathbb{Z}_d$ via a polynomial number of elementary row operations (that is of row additions and row permutations) into an equivalent matrix in Hermite normal form. We remark that this property precisely characterises those commutative rings which are local, cf. Lemma 3.17, and in fact, the following lemma holds for every local ring.

**Lemma 5.20.** *There is a polynomial time algorithm which transforms a given $[m] \times [n]$-matrix $M$ over $\mathbb{Z}_d$ into an equivalent matrix in Hermite normal form.*

*Proof.* We can implement such an algorithm by using the following recursive procedure: in the remaining $[k] \times [\ell]$-matrix $N$, for $k \leq m$ and $\ell \leq n$, we first select an entry $r = M'(i,j) \in \mathbb{Z}_d$ which is minimal with respect to divisibility in $\mathbb{Z}_d$. As a first step, we apply the appropriate row and column transpositions to obtain an equivalent $k \times \ell$-matrix $N'$ which has this entry $r$ in its upper left corner, i.e. $N'(0,0) = r$.

As a second step, we use the first row of $N'$ with the element $r$ in its first position to eliminate all other entries in the first column. After this transformation, the element $r$ still divides every entry in the resulting matrix, since all of its entries are linear combinations of entries of $N'$. We recursively proceed in this way with the $[k-1] \times [\ell-1]$-submatrix that results by deleting the first row and column from $N'$. □

Since the coefficient matrix $M[S^*]$ is a matrix over two ordered index sets, we can express the algorithm from the preceding lemma in CPT. Moreover, since we established a one-to-one correspondence between the application of elementary transformations to $M[S^*]$ and to $S^*$ we can also apply these transformation on the level of hyperequations. In particular, we make use of Lemma 5.9 to see that a polynomial number of elementary transformations of hyperequations can be expressed by a CPT-program since the size of the resulting hyperterms is polynomially bounded. Altogether, this shows that a given system of (ordered) hyperequations $S^*$ can be transformed by a CPT-program into an equivalent system of hyperequations in Hermite normal form.

To complete our proof of Theorem 5.12, it remains to show that Choiceless Polynomial Time can express the solvability of systems of hyperequations in Hermite normal form. For convenience, we say that a hyperequation $T = z$ is *atomic* if the hyperterm $T$ is constant, cf. Definition 5.10. Recall that in this case the hyperterm $T$ has a constant value $c_T = T[\alpha]$ for all $\alpha \in \mathbb{L}$ which means that either every $\alpha \in \mathbb{L}$ satisfies the hyperequation $T = z$ (if $c_T = z$) or the hyperequation has no solution in $\mathbb{L}$. With this notation, we characterise the solvability of systems of hyperequations in Hermite normal form as follows.

**Lemma 5.21.** *Let $S^*$ be a system of hyperequations in Hermite normal form. Then $S^*$ is solvable if, and only if, both of the following conditions are satisfied.*

*(i) Each atomic hyperequation $(T = z) \in S^*$ is consistent.*

*(ii) For each non-atomic hyperequation $(T = z) \in S^*$, the atomic hyperequation $(p^\ell \odot T = p^\ell \cdot z)$ is consistent, where $\ell \geq 1$ is minimal such that $p^\ell \cdot c_i(T) = 0$ for all $i \in [n]$ (in particular, if $p^\ell = 0 \in \mathbb{Z}_d$, then some coefficient $c_i(T)$ is a unit in $\mathbb{Z}_d$).*

*Proof.* Clearly, if $S^*$ is consistent then the conditions (i) and (ii) hold.

For the other direction, we make use of the assumption that $S^*$ is in Hermite normal form. Let $S' \subseteq S^*$ be the subset of non-atomic hyperequations in $S^*$ and let $(T_0 = z_0), \ldots, (T_{k-1} = z_{k-1})$ be an enumeration of $S'$ such that for the $[k] \times [n]$-coefficient matrix $M[S']$ and an appropriate $[n] \times [n]$-permutation matrix $P$ we have

$$M[S'] \cdot P = \begin{pmatrix} a_{0,0} & \cdots & \cdots & \cdots & a_{0,(n-1)} \\ 0 & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & a_{(k-1),(k-1)} & \cdots & a_{(k-1),(n-1)} \end{pmatrix},$$

where $a_{0,0} \mid a_{1,1} \mid \cdots \mid a_{(k-1),(k-1)}$ and where for all $j \in [k]$ and $i \in [n]$ it holds that $a_{j,j} \mid a_{j,i}$. Next, we use property (L) of hyperterms to obtain for every hyperterm $T_j$, $j \in [k]$, an equivalent linear term $t_j = \sum_{i \in [n]} a_{ji} \cdot v_i + y_j$ for appropriate $v_i \in V_i$ and $y_j \in \mathbb{Z}_d$.

Let $j \in [k]$. By condition (ii) we know that $(p^\ell \odot T_j = p^\ell \cdot z_j)$ is consistent where $p^\ell$ is the minimal power of $p$ which annihilates $a_{jj}$. We conclude that

$p^\ell \cdot (z_j - y_j) = 0$ and thus $a_{jj} \mid (z_j - y_j)$ where we use that every element $x \in \mathbb{Z}_d$ can be written as $x = p^e \cdot u$ for an appropriate power $p^e$ of $p$ and a unit $u \in \mathbb{Z}_d^\star$.

Now, the system of hyperequations $S'$ is consistent if, and only if, the system of linear equations $(M[S'] \cdot P) \cdot \vec{v} = \vec{b}$ is consistent where $\vec{v} = (v_0, \ldots, v_{n-1})$ and where

$$\vec{b} = \begin{pmatrix} z_0 - y_0 \\ \vdots \\ z_{k-1} - y_{k-1}. \end{pmatrix}$$

Let $j \leq k$ be minimal such that for some $\alpha \in \mathbb{L}$ all linear equations $t_{j'} = z_{j'}$ with index $j \leq j' \leq k-1$ are satisfied under the assignment $\alpha$. We claim that $j = 0$. Otherwise we assume that $j \geq 1$ and we fix a witnessing $\alpha \in \mathbb{L}$, i.e. $t_{j'}[\alpha] = z_{j'}$ for all $j' \geq j$. The first observation is that we can change $\alpha(v_{j-1})$ without affecting the value $t_{j'}[\alpha]$ for all $j' \geq j$. The second observation is that, since $a_{(j-1),(j-1)} \mid a_{(j-1),i}$ for all $i \in [n]$ and $a_{(j-1),(j-1)} \mid (z_{j-1} - y_{j-1})$, we have that $a_{(j-1),(j-1)} \mid \sum_{i \geq j} a_{(j-1),i} \cdot \alpha(v_i) - (z_j - y_j)$. Hence, $a_{(j-1),(j-1)} \cdot \alpha(v_{j-1}) = \sum_{i \geq j} a_{j,i} \cdot \alpha(v_i) - (z_j - y_j)$ for an appropriate $\alpha(v_{j-1}) \in \mathbb{Z}_d$. This contradicts the minimality of $j$ and finishes the proof of the lemma. □

The preceding lemma shows that the solvability problem for systems of hyperequations in Hermite normal form can be reduced to the consistency check for atomic hyperequations. By Lemma 5.11 this test can be expressed in Choiceless Polynomial Time. This finishes our proof of Theorem 5.12.

## 5.4 Discussion

We introduced cyclic linear equation systems and we discussed their relevance in the quest for a logic for polynomial time. In particular, we saw that the solvability problem for cyclic linear equation system cannot be defined in fixed-point logic with counting although such equation systems are structurally rather simple. Our main result was that Choiceless Polynomial Time can define the solvability of cyclic linear equation systems. This shows that CPT extends FPC by a non-trivial and interesting class of polynomial-time queries.

In these regards, the main open question is whether Choiceless Polynomial Time can express the solvability problem for general linear equation systems over finite (Abelian) groups. We saw that if we put the strong structural assumption of having *cyclic* linear equation systems, then the solvability problem can be defined in CPT. Hence, a natural approach would be to generalise our ideas and to prove that CPT can solve linear equation systems also with more restricted kinds of auxiliary structure. We want to sketch two possible ways of proceeding along such lines.

For both approaches we again consider linear equation systems of the form $(V, S, \preceq)$ which have an auxiliary linear preorder $\preceq$ on their set of variables $V = V_0 \preceq \cdots \preceq V_{n-1}$, but we drop the requirement of having cyclic constraints $C_i$ on the classes $V_i$. Then the first natural idea is to require that all blocks $V_i$

of $\preceq$-equivalent variables are of bounded size, that is $|V_i| \leq b$ for some constant $b \in \mathbb{N}$ and for all $i \in [n]$ (this resembles pretty much the notion of structures with *bounded colours*, see Section 6.1, Definition 6.6). In fact we can prove, and it partly follows from our results in the next chapter, that CPT can define the solvability of such systems for the case where $b \in \{1, 2, 3\}$ (note that the case $b = 1$ corresponds to *ordered* systems, but for $b = 2$ we already capture the linear equation systems for CFI-graphs). For the case $b = 4$ we have some preliminary results which look promising, but for the general case, it is far open whether the solvability problem for such *b-bounded linear equation systems* can be expressed in Choiceless Polynomial Time.

The second approach is based on the insight that for *cyclic* linear equation systems $(V, S, \preceq)$, every block $V_i$ represents a *single* variable. More precisely, if we fix the value of any variable in a block $V_i$, then the value of every other variable in the same block is fixed as well due to the cyclic constraint $C_i \subseteq S$. This suggests to consider *b-determined linear equation systems* for $b \in \mathbb{N}$. For such systems, every block $V_i$ comes with an associated constraint $D_i \subseteq S$ such that whenever we fix the value of at least $b$ different variables from $V_i$, then $D_i$ determines the values of all remaining variables in $V_i$. In particular, for $b = 1$ we obtain the notion of cyclic linear equation systems. We aim to study how far one can get with such an approach in the future.

# Chapter 6

# Canonising structures with Abelian colours

In this chapter we introduce structures with *Abelian colours*. Our main result is that structures with Abelian colours can be canonised in Choiceless Polynomial Time. The immediate consequence is that *every* polynomial-time property of such structures can be expressed in Choiceless Polynomial Time.

Structures with Abelian colours have been considered quite frequently in finite model theory. Most importantly, all structures which are constructed in a way similar to the graphs of Cai, Fürer, and Immerman turn out to be structures with Abelian colours, for example multipedes [55], structures from the construction of Hella [57], and CFI-structures over general Abelian groups [10, 59], see also Chapter 4. Since such families of structures have been introduced to prove limitations of the expressive power of fixed-point logic with counting, our result identifies a general collection of polynomial-time queries which cannot be expressed in fixed-point logic with counting, but in Choiceless Polynomial Time. In particular, our canonisation procedure solves an open question posed by Blass, Gurevich, and Shelah in [16, Question (5.12)]: the isomorphism problem for multipedes can be defined in Choiceless Polynomial Time, see Theorem 6.13.

The notion of structures with *Abelian colours* is motivated by the following reasoning. We know that *ordered* structures are simple in the sense that they are rigid and, more importantly, that they can be identified with a canonical string representation via a first-order transformation. In addition, the Immerman-Vardi Theorem shows that *every* polynomial-time algorithm which operates on strings can be expressed in (least) fixed-point logic. Thus, if we restrict to ordered structures, then the question for a logic for polynomial time is solved. The basic idea of structures with Abelian colours is to relax the assumption of having a *complete* linear order on the universe just as little as necessary to obtain a class of structures on which fixed-point logic (even with counting) fails to capture polynomial time. Strikingly it turns out, that the resulting notion generalises many of the known examples for separating

fixed-point logic with counting from polynomial time.

To be more precise, a *structure with Abelian colours* consists of a usual relational structure $\mathfrak{A}$ extended by a linear preorder $\preceq$ on the universe $A = A_0 \preceq A_1 \preceq \cdots \preceq A_{n-1}$, and for every *colour class* $A_i$, $i \in [n]$, it possesses an *ordered and Abelian* group $\Gamma_i$ which acts *transitively* on $A_i$. In other words, the universe of $\mathfrak{A}$ is *almost* linearly ordered up to the colour classes $A_i$ on which we can access an ordered Abelian group $\Gamma_i$ which relates all pairs of elements in the colour class $A_i$.

The concept of Abelian colours is related to the well studied notion of *bounded colour class size*. Recall that a structure $\mathfrak{A}$ with *bounded colours* also contains a built-in linear preorder $\preceq$ on its universe $A = A_0 \preceq A_1 \preceq \cdots \preceq A_{n-1}$, but instead of having ordered Abelian groups acting transitively on the individual colour classes $A_i$, the requirement is that these colour classes $A_i$ are *small*, that is their size is bounded by some function in $|A|$ (which usually grows very slowly, or which is even constant). We will discuss the precise connections between structures with *Abelian colours* and with *bounded colours* at the end of Section 6.1. However, let us already mention at this point that our initial aim was to develop a CPT-definable canonisation procedure for structures with constantly bounded colours. We discovered the notion of Abelian colours when we studied the following simplification. Assume that for a given structure $\mathfrak{A}$ with bounded colours, all colour classes $A_i$ induce substructures of $\mathfrak{A}$ which have Abelian automorphism groups. As we can show, such structures can easily be transformed into structures with Abelian colours and thus, our canonisation procedure can be applied for such particular structures with bounded colours as well, see Theorem 6.8.

This chapter is strongly based on [1]. In Section 6.1, we introduce structures with Abelian colours. One particular feature of such structures is that they can contain relations of unbounded arity (similar to hypergraphs). Nevertheless, we prove in Theorem 6.5 that every structure with Abelian colours can be transformed, in Choiceless Polynomial Time, into a graph with Abelian colours. Furthermore, we discuss connections between structures with Abelian colours and with bounded colours. In Section 6.2 we establish our main result of this chapter which is a CPT-definable canonisation procedure for structures with Abelian colours. We conclude that Choiceless Polynomial Time captures PTIME on the class of structures with Abelian colours, see Theorem 6.13. For our canonisation procedure we strongly make use of cyclic linear equation systems (see Chapter 5). We close with a discussion in Section 6.3.

## 6.1 Structures with Abelian colours

The aim of this section is to introduce and motivate the notion of structures with *Abelian colours*. In particular, we will see that we already encountered structures with Abelian colours several times in this thesis. For instance, the generalisations of the Cai, Fürer, Immerman graphs which we studied in

Section 4.4 are, up to a CPT-definable preprocessing, structures with Abelian colours. More generally, the cyclic linear equation systems which we defined in Chapter 5 fall into this category as well (and, in some sense, they form the prototype examples of structures with Abelian colours). Secondly, in this section we gather some important properties of structures with Abelian colours. In particular, we show that every structure with Abelian colours can be extended by *any* set of tuples of arbitrary length such that we can maintain the property of having Abelian colours via a CPT-definable transformation. While this result is interesting on its own, it also plays an important role for our CPT-canonisation procedure in the following section. Finally, we discuss connections between structures with *bounded colours* and with Abelian colours.

**Definition 6.1.** A *structure with Abelian colours* is a quadruple $(A, \preceq, X, \Phi)$, where $\preceq$ is a linear preorder on the *universe* $A = A_0 \preceq A_1 \preceq \cdots \preceq A_{n-1}$, where $X \subseteq A^{<\omega}$ is a finite set of tuples over $A$ (of arbitrary length), and where $\Phi = \{(\Gamma_i, \leq) : i \in [n]\}$ is a family of *ordered Abelian* permutation groups $\Gamma_i \leq \mathrm{Sym}(A_i)$ which act *transitively* on the *colour classes* $A_i$ for $i \in [n]$.
Moreover, we denote the class of structures with Abelian colours by $\mathcal{K}_{\mathrm{AC}}$.

At first glance, this definition is not completely satisfactory, since the quadruple $(A, \preceq, X, \Phi)$ is not a relational structure in the usual sense (the elements $X$ and $\Phi$ are higher-order objects in $\mathrm{HF}(A)$). However, at least for $\Phi$ a representation in a standard relational structure can easily be obtained. By definition, $\Phi$ is an *ordered* set which itself consists of *ordered* groups of $A_i$-permutations, that is of bijective functions from $A_i$ to $A_i$. We can represent $\Phi$ in the following way: we extend our universe by an *ordered* index set $A_\Phi$ of new atoms $a_\gamma$ (we add one atom $a_\gamma$ for each permutation $\gamma \in \bigcup \Gamma_i$) and use a ternary relation symbol $R_\Phi$ with the interpretation that for all $a_\gamma \in A_\Phi$, the relation $R_\Phi(a_\gamma, x, y)$ defines the graph of a bijective function from $A_i$ to $A_i$ (the graph of the permutation $\gamma$). The important point to observe is that the set of new indexing elements $A_\Phi$ is totally ordered, which means that the auxiliary atoms $a_\gamma \in A_\Phi$ belong to colour classes of size one (and on colour classes of size one, an action of an ordered Abelian permutation group can trivially be defined). Still, this approach requires some further discussion about the *sizes* of structures with Abelian colours. In fact, if we measure their sizes by $m = |A| + |X|$ (which would be natural since $\preceq$ and $\Phi$ are rather auxiliary objects), then it might seem that the size of $\Phi$ can be exponential in $m$. However, as we require in our definition that the ordered groups $(\Gamma_i, \leq)$ are *Abelian* and *transitive* we know that $|\Gamma_i| = |A_i|$.

The more critical point is the encoding of the *relational part* of the structure, that is of $(A, X)$. For the special case of classes $\mathcal{K}$ of structures with Abelian colours where $X \subseteq A^{\leq k}$ for all $(A, \preceq, X, \Phi) \in \mathcal{K}$ and for a fixed $k \geq 1$, we can easily encode such structures as triples $(\mathfrak{A}, \preceq, \Phi)$ where $\mathfrak{A} \in \mathcal{S}(\tau)$ for a suitable vocabulary $\tau$. In this case we say that $\mathcal{K}$ is a class of *$\tau$-structures* with Abelian colours. Let us denote the class of $\tau$-structures with Abelian colours by $\mathcal{K}_{\mathrm{AC}}(\tau)$.

In general, however, $(A, X)$ cannot be encoded as a $\tau$-structure (for any fixed vocabulary $\tau$), since the tuples in $X$ can be of unbounded length. Hence, we have to use a similar approach as we did for $\Phi$ above, that is we would like to introduce for every tuple $x \in X$ a new atom $a_x$ and to encode the structure of the tuple $x$, that is the entries together with the corresponding components, by auxiliary relations. Moreover, in order to match our definition of structures with Abelian colours, we also have to extend the preorder $\preceq$ to the newly created atoms $a_x$ and, more importantly, also define ordered Abelian groups which act transitively on the resulting colour classes. This, however, turns out to be much more difficult as for the case of $\Phi$. The reason is that, in contrast to the case of $\Phi$, we do *not have* access to a linear order on $X$.

Nevertheless, in Theorem 6.5 we show that this approach is feasible. More precisely, we show that each structure $(A, \preceq, X, \Phi)$ with Abelian colours can be encoded (via an invertible CPT-transformation) as an (undirected) graph with Abelian colours, that is we obtain a CPT-definable encoding of $\mathcal{K}_{\mathrm{AC}}$-structures as $\mathcal{K}_{\mathrm{AC}}(\{E\})$-structures.

In Figure 6.1 we illustrate the notion of structures with Abelian colours.

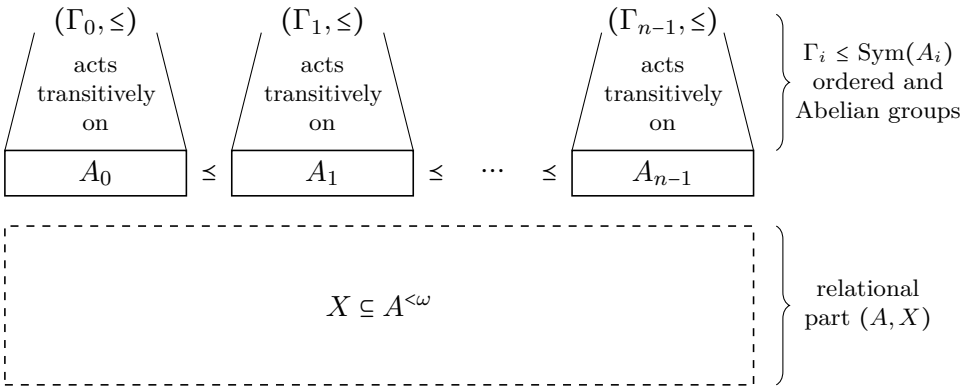


Figure 6.1: Illustration of the notion of structures with Abelian colours

Our main result in this chapter is that there exists a CPT-program $\Pi$ which transform each structure $\mathfrak{A} \in \mathcal{K}_{\mathrm{AC}}$ into an *isomorphic* structure $(\mathfrak{B}, <) := \Pi(\mathfrak{A})$ over an *ordered* universe $B \in \mathrm{HF}(\varnothing)$ such that $\mathfrak{B} \cong \mathfrak{A}$. In other words, there exists a CPT-definable canonisation procedure on the class $\mathcal{K}_{\mathrm{AC}}$ and, as a consequence, Choiceless Polynomial Time captures PTIME on $\mathcal{K}_{\mathrm{AC}}$.

Before we proceed, let us consider a concrete example of a class of structures with Abelian colours. Specifically, we want to show that the generalised CFI-structures over prime fields $\mathbb{F}_q$, which we defined in Section 4.4, are structures with Abelian colours. More precisely, we show that these CFI-structures can be extended by the necessary algebraic components (that is by the ordered, Abelian and transitive groups $\Gamma_i$) in Choiceless Polynomial Time (in fact, the logic DTC suffices). In particular, this holds for the original Cai, Fürer, Immerman graphs [21], since they arise as a special case for $q = 2$.

Let $\mathfrak{A} = \mathrm{CFI}_q(\mathcal{G}, \vec{d})$ be a CFI-structure over $\mathbb{F}_q$ for $q \in \mathbb{P}$, for an (undirected), ordered, and connected graph $\mathcal{G} = (V, \leq, E)$, and for a sequence $\vec{d} \in \mathbb{F}_q^V$ of gadget values as defined in Section 4.4. Recall that $\mathfrak{A}$ possesses a linear preorder $\preceq$ on the set $\hat{E}$ that induces a linear order on the set of edge classes $\{\hat{e} : e \in E\}$ (furthermore, recall that $\hat{E} = \biguplus_{e \in E} \hat{e}$).

In order to represent $\mathfrak{A}$ as a structure $\mathfrak{A}'$ with Abelian colours we can take $\hat{E}$ as our universe and adopt the preorder $\preceq$ on $\hat{E}$ for $\mathfrak{A}'$. Furthermore, it is easy to define ordered Abelian groups on the individual edge classes $\hat{e}$. This is because on every edge class $\hat{e}$ the cycle relation $C$ defines a directed cycle of length $d$. Thus, the automorphism group of this directed cycle is a transitive Abelian group $\Gamma(e) \leq \mathrm{Sym}(\hat{e})$ which can be defined and canonically ordered, for instance, in DTC, see Lemma 3.3. Moreover, the inverse relation $I$ can directly be embedded into the relational part $X$ of $\mathfrak{A}'$, since it consists of undirected edges between edge nodes in $\hat{E}$. Finally, we have to encode the equation nodes $\rho \in \hat{V}$. This we naturally do by using the set $X \subseteq \hat{E}^{<\omega}$ again: for every $\rho \in \hat{v}$ we add the tuple $(e_{\rho(e)})_{e \in E(v)}$ to $X$.

### 6.1.1 From general structures to undirected graphs

In what follows, we solve the question from the beginning: how can we represent the class $\mathcal{K}_{\mathrm{AC}}$ of structures with Abelian colours as a class of $\tau$-structures over a *fixed* vocabulary $\tau$? Specifically, we demonstrate how to transform, in Choiceless Polynomial Time, each structure $\mathcal{A} = (A, \preceq, X, \Phi) \in \mathcal{K}_{\mathrm{AC}}$ into an $\{E\}$-structure $\mathcal{B} = (B, \preceq', E, \Phi')$ with Abelian colours, $E \subseteq B^2$, in such a way that $\mathcal{B}$ encodes $\mathcal{A}$. In particular, we guarantee that from a canonical copy $\mathcal{B}^< \in \mathrm{HF}(\varnothing)$ of $\mathcal{B}$ we can extract in CPT a canonical copy $\mathcal{A}^< \in \mathrm{HF}(\varnothing)$ of the original structure $\mathcal{A}$. This shows that, up to a CPT-definable preprocessing, we can assume that $\mathcal{K}_{\mathrm{AC}} \subseteq \mathcal{S}(\tau)$ where $\tau = \{\preceq, E, R_\Phi\}$ (and where $R_\Phi$ is a ternary relation symbol which encodes the family of ordered Abelian groups $\Phi$, as discussed before). In particular, this result will be very helpful for describing the canonisation procedure in the following section, since it allows us to restrict to *graphs* with Abelian colours (instead of structures which contain relations of unbounded arity).

To establish the described CPT-transformation, we first show that, without loss of generality, we can assume that the tuples in $X$ have a certain kind of normal form. More precisely, we want that each tuple $x \in X$ has at most one entry in each of the classes $A_i$ and, moreover, all tuples in $X$ agree on the specific classes $A_i$ to which their entries belong. Formally this means that there is a set $I \subseteq [n]$ of indices $I = \{i_1, \ldots, i_k\} \subseteq [n]$ of size $|I| = k$ such that $X \subseteq A_{i_1} \times A_{i_2} \times \cdots \times A_{i_k} \subseteq A^{<\omega}$. To put it in different words, all tuples in $X$ belong to the same $\Gamma$-orbit where $\Gamma$ is the following Abelian group $\Gamma \leq \mathrm{Sym}(A)$ which results by composing the Abelian groups $\Gamma_i$ that component-wisely act on individual colour classes $A_i$:

$$\Gamma := \Gamma_0 \times \Gamma_1 \times \cdots \times \Gamma_{n-1} \leq \mathrm{Sym}(A).$$

Note that although, in general, the size of $\Gamma$ is exponential in the size of $A$, we can explicitly access the *ordered* set of generators $\Theta := \biguplus_{i\in[n]} \Gamma_i$. Also, given two group elements $\gamma, \delta \in \Gamma$ we can easily express in CPT that $\gamma \leq \delta$ by using the induced lexicographic ordering on $\Gamma$. We remark that $\Gamma$ also plays an important role in the following section. Now, in order to obtain the described normal form for the set $X \subseteq A^{<\omega}$ we introduce the key notion of *shapes*.

**Definition 6.2.** Let $1 \leq k \leq \ell$. An $(\ell, k)$-*shape* is a sequence $\rho = (\gamma_0, \dots, \gamma_{\ell-1}) \in \Theta^\ell$ for which the set of *relevant colours* $V(\rho) := \{i \in [n] : \rho(j) \in \Gamma_i \text{ for } j \in [\ell]\}$ has size $k = |V(\rho)|$. In this case,

(i) $\mathbb{A}(\rho)$ is the set of assignments $\alpha : V(\rho) \to A$ with $\alpha(i) \in A_i$ for $i \in V(\rho)$,

(ii) and, for $\alpha \in \mathbb{A}(\rho)$, we let $\rho[\alpha] \in A^\ell$ denote the $\ell$-tuple whose entry at position $j \in [\ell]$ results by applying $\gamma = \rho(j) \in \Gamma_i$ to the element $\alpha(i)$.

We aim to use shapes to identify $\ell$-tuples $x \in A^\ell$ by pairs consisting of an ordered object $\hat{x} = \rho \in \mathrm{HF}(\Theta)$ (the *shape* of $x$) and a realising assignment $\alpha \in V(\rho)$. This assignment $\alpha$, in turn, can then be identified with a tuple $\bar{\alpha} \in A^k$ which satisfies the normal form conditions that we described above.

More precisely, since $\Theta$ is an *ordered* set (and since shapes are objects in $\mathrm{HF}(\Theta)$) we can canonically define the *shape of a tuple* $x \in A^\ell$ as the lexicographically minimal $(\ell, k)$-shape $\hat{x} = \rho$ such that $x = \rho[\alpha]$ for a (unique) $\alpha \in \mathbb{A}(\rho)$ where $k = |\{i \in [n] : x(j) \in A_i \text{ for } j \in [\ell]\}|$. More directly, to obtain the shape of $x$ we first collect all indices $i \in [n]$ such that $x$ contains elements from the colour class $A_i$. We then independently consider all subtuples $(x \restriction A_i)$ of $x$ which result by restricting $x$ to those entries which belong to $A_i$. Let $y := (x \restriction A_i) \in A_i^r$ be one of such sequences. To obtain the corresponding subsequence $\hat{y} \in \Gamma_i^r$ of the shape $\hat{x}$ of $x$ we first choose the minimal group element $\gamma \in \Gamma_i$ and put it as the first entry $\hat{y}(0) := \gamma$ of $\hat{y}$. Moreover, we choose $a = \alpha(i) \in A_i$ for our realising assignment such that $\gamma(a) = y_0$. This is possible since $\Gamma_i$ is a transitive group. Then all other entries $\hat{y}(j)$ of $\hat{y}$ are immediately determined as the unique $\delta \in \Gamma_i$ which map $a$ to $y_j$. It is clear that this procedure can be expressed in Choiceless Polynomial Time.

**Lemma 6.3.** *There is a* CPT*-program which, given a structure* $(A, \preceq, X, \Phi)$ *with Abelian colours as above, associates to every tuple* $x \in X$ *a pair* $(\hat{x}, \bar{\alpha}) \in \mathrm{HF}(\Theta) \times A^{<\omega}$*, where* $\Theta := \biguplus_{i\in[n]} \Gamma_i$*, such that*

- $\hat{x} = \rho \in \mathrm{HF}(\Theta)$ *is the shape of* $x$ *where* $V(\rho) = \{i_0 < i_1 < \dots < i_{k-1}\}$*, and*
- *where for* $\alpha : V(\rho) \to A, i_j \mapsto \bar{\alpha}(j)$ *we have* $\alpha \in \mathbb{A}(\rho)$ *and* $\rho[\alpha] = x$*.*

Let $x, y \in A^\ell$ and assume that $x \leftrightarrow (\hat{x}, \bar{\alpha})$ and $y \leftrightarrow (\hat{y}, \bar{\beta})$ (where $\leftrightarrow$ denotes the correspondence from the preceding lemma). If $\hat{x} = \hat{y}$, then $x$ and $y$ belong to the same $\Gamma$-orbit. This is easy to see, since for all $j \in [n]$ there exists at most one entry $i \in [k]$ such that $\bar{\alpha}(i) \in A_j$ or $\bar{\beta}(i) \in A_j$ and, since $\hat{x} = \hat{y}$, in such case we have $\bar{\alpha}(i) \in A_j$ *and* $\bar{\beta}(i) \in A_j$. Moreover, a witnessing group element $\gamma \in \Gamma$

which maps $x$ to $y$ can easily be defined in CPT (it suffices to map $\bar{\alpha}$ to $\bar{\beta}$). In other words, the shape of a tuple $x \in A^\ell$ completely describes its $\Gamma$-orbit.

Before we proceed, we give a small example which illustrates the notion of shapes and how they are applied to encode tuples $x \in A^{<\omega}$. Let $n = 4$, let

$$A = A_0 = \{a_0, a_1\} \preceq A_1 = \{b_0, b_1\} \preceq A_2 = \{c_0, c_1\} \preceq A_3 = \{d_0, d_1\},$$

and let $\Gamma_i = \mathrm{Sym}(A_i) = \{\gamma^i, \delta^i\}$ be the symmetric group acting on the two elements in $A_i$. Here, $\gamma^i = \mathrm{id}_{A_i}$ denotes the identity on $A_i$ and $\delta^i$ denotes the transposition of the two elements in $A_i$. Then $|\Gamma_i| = 2$, $\Gamma_i$ is Abelian and, moreover, we have a canonical order $\gamma^i < \delta^i$ on $\Gamma_i$. Now, consider the tuple $x = (b_0, a_0, a_1, b_0, d_1, a_1, d_0) \in A^7$. Then $x$ is identified, according to Lemma 6.3, with the pair $(\hat{x}, \bar{\alpha}) \in \mathrm{HF}(\Gamma_0 \uplus \cdots \uplus \Gamma_3) \times A^3$ that is given as

$$\hat{x} = (\gamma^1, \gamma^0, \delta^0, \gamma^1, \gamma^3, \delta^0, \delta^3) \text{ and } \bar{\alpha} = (a_0, b_0, d_1) \in A^3.$$

Note that the tuple $y = (b_1, a_1, a_0, b_1, d_1, a_0, d_0) \in A^7$ has the same shape $\hat{y} = \hat{x}$, but the realising assignment $\bar{\beta} = (a_1, b_1, d_1)$ for $y$ would differ from that of $x$ accordingly. Moreover, every group element $\lambda \in \Gamma$ which maps $\bar{\alpha}$ to $\bar{\beta}$ also maps the tuple $x$ to the tuple $y$. For instance, this holds for $\lambda = (\delta^0, \delta^1, \gamma^2, \gamma^3)$.

**Lemma 6.4.** *There is a* CPT*-program which, given a structure $(A, \preceq, X, \Phi)$ with Abelian colours, defines an extension of $\preceq$ to $X = X_0 \preceq \cdots \preceq X_{m-1}$ and a family of ordered, Abelian permutation groups $\Psi = \{(\Delta_i, \preceq) : i \in [m]\}$, $\Delta_i \leq \mathrm{Sym}(X_i)$, which act transitively on the classes $X_i$. In particular, all tuples $x, y \in X_i$ have the same shape $\hat{x} = \hat{y} = \hat{X}_i$ for $i \in [m]$.*

*Proof.* As a first preparation we use Lemma 6.3 to decompose $X$ into an ordered partition of classes $X = Z_0 \preceq Z_1 \preceq \cdots \preceq Z_{r-1}$ such that all tuples $y, z \in Z_i$ have the same shape, i.e. if $y \leftrightarrow (\hat{y}, \bar{\alpha})$ and $z \leftrightarrow (\hat{z}, \bar{\beta})$, then $\hat{y} = \hat{z}$. This refinement process can be expressed in CPT by Lemma 6.3 and the fact that shapes are ordered objects. As explained above, this also implies that all tuples in the refined classes $Z_i$ belong to the same $\Gamma$-orbit. From now on, let us assume that $X = Z_i$ is one of these classes.

What remains is to obtain an Abelian and ordered group $\Delta \leq \mathrm{Sym}(X)$ which acts transitively on $X$. Of course, the natural approach is to consider the induced action of $\Gamma$ on $X$. However, in general $X$ is not a block, that is it might happen that for some $\gamma \in \Gamma$ we have $X \neq \gamma(X), X \cap \gamma(X) \neq \varnothing$. To overcome this problem we make use of the fact that we can access the ordered set $\Theta$ of generators for $\Gamma$. Let $\gamma \in \Theta$ be minimal such that $X \neq \gamma(X), X \cap \gamma(X) \neq \varnothing$. Then we split $X$ into the two parts $X \setminus \gamma(X)$ and $X \cap \gamma(X)$ which we can canonically order, for instance, by setting $(X \setminus \gamma(X)) < (X \cap \gamma(X))$. We continue with this CPT-definable refinement process as long as some of the resulting classes split (i.e. as long as they are not blocks). Summing up, we have refined $X$ into classes $X = X_0 \preceq X_1 \preceq \cdots \preceq X_{m-1}$ such that

- for $y, z \in X_i$ we can fix (in CPT) an element $\gamma \in \Gamma$ such that $\gamma(y) = z$,

- and for all $\gamma \in \Gamma$ we either have that $\gamma(X_i) = X_i$ or $\gamma(X_i) \cap X_i = \varnothing$.

We can now easily define in CPT the ordered Abelian groups $\Delta_i \leq \mathrm{Sym}(X_i)$, for $i \in [m]$, which act transitively on $X_i$. To this end we fix for each pair $y, z \in X_i$ a group element $\gamma \in \Gamma$ such that $\gamma(y) = z$. Then we consider the permutation which is induced by $\gamma$ on $X_i$. The set of these induced permutations yields the desired group $\Delta_i$. □

The preceding lemma brings us quite close to our goal of showing that every structure with Abelian colours can be encoded, via a CPT-definable invertible mapping, as a graph with Abelian colours. Let $\mathcal{A} = (A, \preceq, X, \Phi) \in \mathcal{K}_{\mathrm{AC}}$ and let us assume that we have refined $X = X_0 \preceq X_1 \preceq \cdots \preceq X_{m-1}$ as stated in Lemma 6.4. Then all tuples $x, y \in X_i$ have the same shape $\rho_i := \hat{x} = \hat{y}$ and we can use the position $i \in [m]$ of the block $X_i$ in the order $\preceq$ to store this common information. As a consequence we can represent tuples $x \in X_i$ by their realising assignments $\bar{\alpha} \in A_{i_0} \times \cdots \times A_{i_{k-1}}$ where $V(\rho_i) = \{i_0 < i_1 < \cdots < i_{k-1}\}$. Since the order on the classes $A_i$ is fixed, we can encode these assignments $\bar{\alpha}$ by new atoms $z_{\bar{\alpha}}$ which are connected via undirected edges to the elements $\bar{\alpha}(j) \in A_{i_j}$ for $j \in [k]$. In particular, note that the preorder $\preceq$ on $X$ and the group actions $\Delta_i$ on $X_i$ directly induce corresponding actions on the newly created atoms so that we can maintain the property of having a structure with Abelian colours. In this way we have represented the structure $\mathcal{A}$ as a graph $\mathcal{B} = (B, E \preceq', \Phi')$ with Abelian colours, where $E \subseteq B^2$ is a symmetric edge relation, where $A \subseteq B$ and where $|B| \in \mathcal{O}(|A| + |X|)$.

At this point, we have to discuss another technical issue. While the translation $\mathcal{A} \mapsto \mathcal{B}$ is clearly invertible in CPT, it is not clear whether we can construct, given a canonisation $\mathcal{B}^< \cong \mathcal{B}$ of the graph $\mathcal{B}$, also a canonisation $\mathcal{A}^<$ of the original structure $\mathcal{A}$ in Choiceless Polynomial Time. The difficulty is that this would require to reconstruct tuples from their shapes and their realising assignments. But what happens if the realising assignments take values in another domain (like it would be the case if we were working with assignments over an ordered copy of the universe of $\mathcal{A}$). Then it is unclear how we can evaluate shapes (which contain group elements from the original groups $(\Gamma_i, \leq)$) for such assignments. The solution is to carefully analyse the canonisation procedure in the following section. Then we observe that the canonical copy of $\mathcal{B}^<$ also contains canonical copies $\Gamma_i^<$ of the groups $(\Gamma_i, \leq) \in \Phi'$. These groups $\Gamma_i^<$, in turn, act on the ordered universe of $\mathcal{B}^<$ and we can, moreover, fix a single isomorphism which maps $(\Gamma_i, \leq) \to \Gamma_i^<$ (and this isomorphism is part of an isomorphism between $\mathcal{B}$ and $\mathcal{B}^<$). Hence, we can translate shapes canonically to shapes over the ordered copy $\mathcal{B}^<$ which can then be used to reconstruct the tuples for our canonical copy of the set $X$ from the original structure $\mathcal{A}$.

**Theorem 6.5.** *There is a* CPT*-program* $\Pi$ *which encodes structures* $\mathcal{A} \in \mathcal{K}_{AC}$ *with Abelian colours as graphs* $\mathcal{B} \in \mathcal{K}_{AC}(\{E\})$ *with Abelian colours in such a way that if we apply our canonisation procedure (Theorem 6.13) for* $\mathcal{B}$*, then we can obtain in* CPT *a canonical copy of* $\mathcal{A}$ *from the canonical copy of* $\mathcal{B}$*.*

In other words, the preceding theorem says that the class $\mathcal{K}_{\mathrm{AC}}$ can be identified with a subclass of $\mathcal{S}(\{E, P\})$ via a CPT-definable transformation. This solves our problem of representing structures with Abelian colours as usual relational structures.

### 6.1.2 Structures with bounded colours

At the end of this section we want to discuss connections between structures with Abelian colours and certain classes of structures with *bounded colours*.

**Definition 6.6.** Let $f : \mathbb{N} \to \mathbb{N}$. A *$\tau$-structure with colour class size $f$*, or *a $\tau$-structure with $f$-bounded colours*, is a pair $(\mathfrak{A}, \preceq)$ where $\mathfrak{A} \in \mathcal{S}(\tau)$ and where $\preceq$ is a linear preorder on $A = A_0 \preceq A_1 \preceq \cdots \preceq A_{n-1}$, such that $|A_i| \leq f(|A|)$ for all $i \in [n]$. As before, the classes $A_i$ are called the *colour classes* of $(\mathfrak{A}, \preceq)$.

Again one may think of the linear preorder $\preceq$ as an "approximation" of a linear order on the universe $A$. The difference to the notion of structures with Abelian colours is that we do not have access to an Abelian group action on the colour classes $A_i$. Instead we have the guarantee that the sizes of the colour classes $A_i$ are bounded by $f(|A|)$ (and usually, $f$ will be a function which grows very slowly, or which is even constant).

Let $\mathcal{K}^f_{\mathrm{BC}}(\tau)$ denote the class of all $\tau$-structures with $f$-bounded colours for some function $f : \mathbb{N} \to \mathbb{N}$. We are interested in the (descriptive) complexity of the isomorphism problem on $\mathcal{K}^f_{\mathrm{BC}}(\tau)$ and, more generally, in a (definable) canonisation procedure on this class. Of course, the hardness of these two problems crucially depends on the choice of the function $f : \mathbb{N} \to \mathbb{N}$. For example, while both problems are trivial for $f = 1$, they are as hard as possible if we let $f = \mathrm{id}_{\mathbb{N}}$.

Let us briefly summarise what is known about the algorithmic complexity of these problems on $\mathcal{K}^f_{\mathrm{BC}}(\tau)$. Most importantly, it is known that the structure isomorphism problem is polynomial-time decidable on $\mathcal{K}^f_{\mathrm{BC}}(\tau)$ if $f \in \mathbb{N}$ is a constant [12, 36]. Interestingly, the corresponding isomorphism test is strongly based on algorithmic techniques for handling permutation groups (membership testing, intersection problems, and so on). More precisely, the basic idea is to compute sets of generators for the automorphism group of the given structure by iteratively stabilising small substructures and by successively combining the intermediate results. In some sense, it is fair to regard these permutation group algorithms as generalisations of the method of Gaussian elimination for general, that is non-Abelian groups. It is an easy observation that these isomorphism tests and canonisation procedures on $\mathcal{K}^f_{\mathrm{BC}}(\tau)$ also work for the case where $f$ is not constant, but very slowly growing, for instance if $f(n) = \log(n)/\log(\log(n))$. More precisely, the original algorithms run in time $\mathcal{O}((f(n)! \cdot n)^c)$ for some constant $c$. It is also worth mentioning that, more recently, the notion of bounded colour class size has been studied for hypergraphs and in the context of fixed-parameter tractability, see [76] and

see [6] for an algorithm with an improved running time. However, in this thesis, whenever we speak of structures with bounded colours, then we always assume that we have structures over a *fixed* vocabulary (as opposed to the case of hypergraphs where hyperedges can be of unbounded width).

Despite these nice algorithmic results, fixed-point logic with counting already fails to express the isomorphism problem on $\mathcal{K}^2_{\mathrm{BC}}(\tau)$ if we assume that $\tau$ contains a relation symbol of arity at least three. This is because the CFI-graphs can be represented as relational structures in $\mathcal{K}^2_{\mathrm{BC}}(\tau)$ where $\tau = \{R\}$ consists of a single ternary relation symbol $R$. Also, if we restrict ourselves to graphs, then the bound on the constant does not increase significantly. In fact, the CFI-graphs are already contained in $\mathcal{K}^4_{\mathrm{BC}}(\{E\})$ and this bound is sharp: it follows from [66] that all graphs in $\mathcal{K}^3_{\mathrm{BC}}(\{E\})$ can be canonised in fixed-point logic with counting. Summing up, the tremendous mismatch between the algorithmic tractability of the isomorphism problem on $\mathcal{K}^f_{\mathrm{BC}}(\tau)$ and its definability in fixed-point logic with counting, makes the classes $\mathcal{K}^f_{\mathrm{BC}}(\tau)$ an interesting testing ground for other candidates for logics capturing polynomial time such as rank logic FPR and Choiceless Polynomial Time CPT.

**Theorem 6.7** ([12, 21, 36])**.**

*(a) Let $f : \mathbb{N} \to \mathbb{N}$ be such that $f(n)! \in \mathcal{O}(n^c)$ for $c \geq 1$. Then the isomorphism problem is decidable in polynomial time on $\mathcal{K}^f_{BC}(\tau)$ and, more generally, there exists a polynomial-time canonisation procedure on $\mathcal{K}^f_{BC}$.*

*(b) The isomorphism problem on $\mathcal{K}^4_{BC}(\tau)$ where $\{E\} \subseteq \tau$, and on $\mathcal{K}^2_{BC}(\sigma)$ where $\sigma$ contains a relation symbol of arity at least three, is not definable in fixed-point logic with counting.*

In particular, this leads to the following important question: is the isomorphism problem on $\mathcal{K}^f_{\mathrm{BC}}(\tau)$ definable in Choiceless Polynomial Time (or in rank logic) for all constants $f \in \mathbb{N}$? Indeed, such a result would significantly increase our knowledge about the gain in expressiveness when we pass from fixed-point logic with counting to CPT (or to FPR). Also, it would provide a promising starting point from where one could study the definability of the isomorphism problem on more complicated classes of structures for which efficient isomorphism tests are known, but on which the Weisfeiler-Lehman method, and thus FPC, fails. This includes, for example, classes of graphs with bounded degree [37, 75] or classes with slowly growing treewidth [31, 74].

Interestingly, the CPT-canonisation procedure on $\mathcal{K}_{\mathrm{AC}}$ which we develop in the following section provides a first step towards an answer to the above question. More specifically, if we consider classes $\mathcal{K} \subseteq \mathcal{K}^f_{\mathrm{BC}}(\tau)$ of structures with $f$-bounded colours (for a sufficiently slowly growing function $f$) and if make the the additional assumption that all substructures which are induced on the individual colour classes have *Abelian* automorphism groups, then the structures in $\mathcal{K}$ really are, up to a CPT-definable preprocessing, structures

with Abelian colours. As a consequence, the CPT-definable canonisation procedure on $\mathcal{K}_{\mathrm{AC}}(\tau)$ provides a CPT-definable canonisation procedure on $\mathcal{K}$.

**Theorem 6.8.** *Let $f : \mathbb{N} \to \mathbb{N}$ be such that $f(n)! \in \mathcal{O}(n^c)$ for $c \geq 1$. Moreover, let $\mathcal{K} \subseteq \mathcal{K}^f_{BC}(\tau)$ be a class of $\tau$-structures $(\mathfrak{A}, \preceq)$ with $f$-bounded colours such that for all colour classes $A_i \subseteq A$ the automorphism group $Aut(\mathfrak{A} \restriction A_i)$ of the substructure $(\mathfrak{A} \restriction A_i)$ induced on $A_i$ is Abelian.*

*Then there is a* CPT*-program $\Pi$ which defines, given $(\mathfrak{A}, \preceq) \in \mathcal{K}$, a refinement $\preceq'$ of the preorder $\preceq$ on $A$ and a set $\Phi = \{(\Gamma_i, \leq) : i \in [m]\}$ of ordered and Abelian groups $(\Gamma_i, \leq)$ which act transitively on the colour classes $A'_i$ that are induced by the refined preorder $\preceq'$, that is $(\mathfrak{A}, \preceq', \Phi) \in \mathcal{K}_{AC}(\tau)$.*

*Proof.* To obtain an appropriate CPT-transformation $\Pi$ we use an important algebraic characterisation for the structure of sets of isomorphisms between two relational structures (this characterisation also plays an important role for our canonisation procedure which we establish in the following section).

Let $(\mathfrak{A}, \preceq) \in \mathcal{K}$ be a structure with $f$-bounded colours, let $A_i \subseteq A$ be a colour class of $(\mathfrak{A}, \preceq)$ and let $\mathfrak{B} := (\mathfrak{A} \restriction A_i)$ denote the substructure of $\mathfrak{A}$ induced on $B = A_i$. By our assumption on $\mathcal{K}$ we know that $\Gamma := \mathrm{Aut}(\mathfrak{B}) \leq \mathrm{Sym}(B)$ is an Abelian group which can be constructed in CPT since $|\mathrm{Sym}(B)| = |B|! \in \mathcal{O}(|A|^c)$. However, we neither have a linear order on this group $\Gamma$, nor can we assume that $\Gamma$ acts transitively on $B$ (which are the necessary conditions to obtain a structure with Abelian colours).

To attack these problems, we denote by $B^< := \{0, \dots, |B| - 1\} \subseteq \mathrm{HF}(\varnothing)$ an ordered set of size $|B|$ and we identify the linear orderings on $B$ with the set $\mathcal{O}(B)$ of bijections $\pi : B \to B^<$ in the obvious way. Of course, if we apply any bijection $\pi \in \mathcal{O}(B)$ to the structure $\mathfrak{B}$, then we obtain an *ordered* structure $\pi(\mathfrak{B})$. More importantly, whenever it holds that $\pi(\mathfrak{B}) = \sigma(\mathfrak{B})$ for some $\pi, \sigma \in \mathcal{O}(B)$, then $(\sigma^{-1} \cdot \pi)(\mathfrak{B}) = \mathfrak{B}$, that is $(\sigma^{-1} \cdot \pi) \in \Gamma$. Hence, if we fix an ordered structure $\mathfrak{B}^<$ (for example, the lexicographically minimal one) such that $\pi(\mathfrak{B}) = \mathfrak{B}^<$ for some $\pi \in \mathcal{O}(B)$, then the set of isomorphisms $\pi \in \mathcal{O}(B)$ with $\pi(\mathfrak{B}) = \mathfrak{B}^<$ can be written for every $\sigma \in \mathcal{O}(B)$ with $\sigma(\mathfrak{B}) = \mathfrak{B}^<$ as

$$\{\pi \in \mathcal{O}(B) : \pi(\mathfrak{B}) = \mathfrak{B}^<\} = \sigma\Gamma = {}^{\sigma}\Gamma\sigma, \text{ see also Lemma 6.9.}$$

In particular, this shows that we can define in Choiceless Polynomial Time a set $\sigma\Gamma \subseteq \mathcal{O}(B)$ of bijections between $B$ and $B^<$ with the indicated algebraic structure. We claim that, given this set $\sigma\Gamma$, we can easily obtain a linear order on $\Gamma$. First of all, every bijection $\pi \in \sigma\Gamma$ induces a group isomorphism $\varphi_\pi : \Gamma \to {}^{\pi}\Gamma, \gamma \mapsto {}^{\pi}\gamma$. Since ${}^{\pi}\Gamma$ acts on $B^<$ we can identify ${}^{\pi}\Gamma$ with an ordered permutation group. Finally, the order on ${}^{\pi}\Gamma$ translates via $\varphi_\pi$ into an ordering on $\Gamma$. The problem is that, in general, these group isomorphisms $\varphi_\pi$ may differ for the various possible choices of $\pi \in \sigma\Gamma$ (which also means that we do not obtain a *unique* ordering on $\Gamma$).

However, we claim that this cannot happen in our situation since $\Gamma$ is an *Abelian* group. To see this let $\pi \in \sigma\Gamma$. Then $\pi = \sigma\lambda$ for some $\lambda \in \Gamma$ and we have

$$^{\pi}\gamma = \pi\gamma\pi^{-1} = \sigma\lambda\,\gamma\,\lambda^{-1}\sigma^{-1} \overset{\Gamma \text{ Abelian}}{=} \sigma\gamma\sigma^{-1}.$$

Hence, for all $\pi, \varrho \in \sigma\Gamma$ we have $\varphi_\pi = \varphi_\varrho$. In this way we obtain a definable linear order on $\Gamma$.

Finally, we have to treat the case where $\Gamma$ does not act transitively on $B$. Thus, assume that $C \subset B$ is a $\Gamma$-orbit. Then for all bijections $\pi, \varrho \in \sigma\Gamma$ we have $C^< := \pi(C) = \varrho(C) \subseteq B^<$ since $\Gamma(C) = C$. Moreover, it is easy to see that for every pair of different $\Gamma$-orbits $C_1, C_2 \subset B$ we have $C_1^< \cap C_2^< = \varnothing$. Thus, we can use the natural order on $\mathcal{P}(B^<) \in \mathrm{HF}(\varnothing)$ to obtain an ordering on the $\Gamma$-orbits of $B$ (note that for this last step we have not used that $\Gamma$ is Abelian). Then the induced action of $\Gamma$ on the individual orbits $C \subset B$ yields the desired family of ordered, Abelian and transitive groups. □

In particular, the preconditions of Theorem 6.8 are satisfied for every class $\mathcal{K}^2_{\mathrm{BC}}(\tau)$ of $\tau$-structures with colour class size two. Indeed, if $|A_i| \leq 2$, then $\mathrm{Sym}(A_i)$ is an Abelian group which can be ordered canonically. Thus, given our results from the following section, this observation shows that CPT captures polynomial time on $\tau$-structures with colour class size two (see Corollary 6.14).

## 6.2 Canonising structures with Abelian colours

In this section we establish a CPT-definable canonisation procedure on the class $\mathcal{K}_{\mathrm{AC}}$ of structures with Abelian colours. As an immediate consequence it follows that Choiceless Polynomial Time captures PTIME on $\mathcal{K}_{\mathrm{AC}}$.

We proceed as follows. First, we introduce some further piece of notation and summarise simple facts about the algebraic structure of sets of isomorphisms between two relational structures. We then describe, in a second step, a general canonisation procedure for structures with Abelian colours. Unfortunately, this procedure cannot be expressed in Choiceless Polynomial Time directly, since it requires to manipulate exponential-sized sets of isomorphisms between (parts of) the input structure and a (partially) canonised version. Hence, it is necessary, in a third step, to develop a succinct representation of these sets of isomorphisms in such a way that basic operations (such as testing for emptiness) are definable in Choiceless Polynomial Time. The main idea in this last part will be to use cyclic linear equation systems to obtain a succinct, and CPT-definable, representation of these sets of isomorphisms.

Let $\mathcal{A} \in \mathcal{K}_{\mathrm{AC}}$ be a structure with Abelian colours. By Theorem 6.5 we can assume that $\mathcal{A} = (A, \leq, E, \Phi) \in \mathcal{K}_{\mathrm{AC}}(\{E\})$, that is $\mathcal{A}$ is a *graph* with Abelian colours. As before, let $A = A_0 \leq A_1 \leq \cdots \leq A_{n-1}$ and let $\Phi = \{(\Gamma_i, \leq) : i \in [n]\}$ denote the family of Abelian and ordered permutation groups $\Gamma_i \leq \mathrm{Sym}(A_i)$ which act transitively on the colour classes $A_i$. Recall the definition of the Abelian group $\Gamma := \Gamma_0 \times \Gamma_1 \times \cdots \times \Gamma_{n-1} \leq \mathrm{Sym}(A)$ from the last section. We

already pointed out that, in general, the size of $\Gamma$ is exponential in the size of $A$, but that we can access the *ordered* set of generators $\Theta := \biguplus_{i\in[n]} \Gamma_i$. Also, given two group elements $\gamma, \delta \in \Gamma$ we can express in CPT that $\gamma \leq \delta$ by using the induced lexicographic ordering on $\Gamma$ (which means that we can implicitly define an ordering on $\Gamma$ without having to represent it completely). We also consider the natural extension of the action of $\Gamma$ on $A$ to the class of hereditarily finite sets $\mathrm{HF}(A)$ over $A$. Moreover, we let $q_i := |A_i| = |\Gamma_i|$ denote the size of the $i$-th colour class and we set $A_i^< := \{(i,0), \dots, (i, q_i - 1)\}$ to obtain an *ordered* and distinguished set of size $q_i$ which will serve as an ordered domain for the colour class $A_i$. Accordingly, we let $A^< := A_0^< \uplus \cdots \uplus A_{n-1}^<$ denote an ordered set of size $|A|$ which provides an ordered domain for the complete universe $A$. We also translate the preorder $\preceq$ to the classes $A_i^<$ in the obvious way, that is let $A_i \preceq A_j$ if, and only if, $A_i^< \preceq A_j^<$.

**Lemma 6.9.** *There exists a* CPT*-program which defines, given a graph* $\mathcal{A} = (A, \preceq, E, \Phi)$ *with Abelian colours as above, for every* $i \in [n]$*, a set* $\mathcal{O}(A_i)$ *of bijections between* $A_i$ *and* $A_i^<$ *such that* $\Gamma_i$ *acts transitively on* $\mathcal{O}(A_i)$*. In particular, it holds that* $\mathcal{O}(A_i) = \pi_i \Gamma_i$ *for all* $\pi_i \in \mathcal{O}(A_i)$*.*

*Proof.* Since $\Gamma_i$ is an *Abelian* and *transitive* group, the images $\gamma(a)$ for $a \in A_i$ are different for all $\gamma \in \Gamma_i$. Moreover, we have access to a linear order on $\Gamma_i$, i.e. $\Gamma = \{\gamma_0 < \gamma_1 < \cdots < \gamma_{q_i - 1}\}$. In this way we obtain for every $a \in A_i$ an ordering of $A_i$ as $\gamma_0(a) < \gamma_1(a) < \cdots < \gamma_{q_i-1}(a)$ which corresponds to a bijection $\pi_a : A_i \to A_i^<$. Finally, it is easy to see that $\gamma(\pi_a) = \pi_{\gamma(a)}$ for all $\gamma \in \Gamma_i$. □

In particular, the sets $\mathcal{O}(A_i)$ are of size $q_i = |A_i| = |\Gamma_i|$. In what follows, whenever we refer to the sets $\mathcal{O}(A_i)$, then we mean the canonically constructed set of bijections between $A_i$ and $A_i^<$ (which effectively correspond to linear orderings of $A_i$) of the form $\mathcal{O}(A_i) = \pi_i \Gamma_i$ for all $\pi_i \in \mathcal{O}(A_i)$ as in the previous lemma. We let $\mathcal{O}(A) := \mathcal{O}(A_0) \times \cdots \times \mathcal{O}(A_{n-1})$ denote the set of bijections $\pi : A \to A^<$ which result by composing the individual bijections from $\mathcal{O}(A_i)$. Since $\mathcal{O}(A_i)$ can be written as $\mathcal{O}(A_i) = \pi_i \Gamma_i$ for some $\pi_i \in \mathcal{O}(A_i)$ (although we cannot fix such a $\pi_i$ canonically), we can write $\mathcal{O}(A)$ as $\mathcal{O}(A) = \pi\Gamma$ where $\pi \in \mathcal{O}(A)$. We stress the fact that we can not select such an ordering $\pi \in \mathcal{O}(A)$ during the run of a CPT-program; in fact, if we could, then this would, in particular, render our whole approach useless, since we had access to a *complete linear order* $\pi \in \mathcal{O}(A)$ on $\mathcal{A}$ which makes canonisation very easy.

We make a further crucial observation. Of course, each $\pi \in \mathcal{O}(A)$ maps objects $x \in \mathrm{HF}(A)$ to objects $\pi(x) \in \mathrm{HF}(A^<)$ and, since $A^<$ is an *ordered* set, we can canonically distinguish different objects in $\mathrm{HF}(A^<)$ (in fact, up to a CPT-definable transformation, all objects in $\mathrm{HF}(A^<)$ are strings). In particular, given an object $x \in \mathrm{HF}(A_i)$ we can identify in CPT a (non-empty) subset $M \subseteq \mathcal{O}(A_i)$ of orderings on $A_i$ such that for all $\pi, \sigma \in M$ we have $\pi(x) = \sigma(x)$. In fact, we can just apply all $\pi \in \mathcal{O}(A_i)$ to $x$ and choose $M \subseteq \mathcal{O}(A_i)$ to consist of those $\pi \in \mathcal{O}(A_i)$ which map $x$ to the lexicographically minimal element in

$\mathrm{HF}(A_i^<)$. As we already saw in the previous section, such subsets $M \subseteq \mathcal{O}(A_i)$ have the following algebraic structure:

$$M = \pi\Delta = {}^{\pi}\Delta\pi \text{ for any } \pi \in M \text{ and for } \Delta = \mathrm{Aut}(x) \cap \Gamma_i.$$

Moreover, for each ordering $\pi \in \mathcal{O}(A_i)$ we can consider the induced group isomorphism $\pi : \Gamma_i \to {}^{\pi}\Gamma_i, \gamma \to {}^{\pi}\gamma = \pi\gamma\pi^{-1}$ which maps $\Gamma_i$ to its canonical copy $\Gamma_i^< := {}^{\pi}\Gamma$. Since $\Gamma_i$ is an Abelian group, and since $\Gamma_i$ acts transitively on $\mathcal{O}(A_i)$, these isomorphisms coincide for all $\pi, \sigma \in \mathcal{O}(A_i)$ which means that we obtain a *single* isomorphism $\psi_i : \Gamma_i \to \Gamma_i^<$ which is induced by *any* of the isomorphisms $\pi \in \mathcal{O}(A)$ (or $\pi \in \mathcal{O}(A_i)$).

### 6.2.1 An inductive canonisation scheme

We are ready to give a high level description of our canonisation procedure. The first important step is to split the structure $\mathcal{A}$ into an ordered sequence of small substructures $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{m-1}$, $\mathcal{A}_i \subseteq \mathcal{A}$. The benefit is that these "small" substructures $\mathcal{A}_i$ can be canonised easily. This leads to the idea of constructing the canonisation of $\mathcal{A}$ along the decomposition inductively: first, we determine canonisations $\mathcal{A}_i^<$ of the small substructures $\mathcal{A}_i$ of $\mathcal{A}$ and then, in a second step, we combine the small canonised pieces $\mathcal{A}_i^<$ to obtain a canonisation of the full structure $\mathcal{A}$.

In general, the canonisation $\mathcal{A}_i^<$ of the substructure $\mathcal{A}_i$ is not unique, that is we will obtain *different* ordered structures $\mathcal{A}_i^<$ which are isomorphic to $\mathcal{A}_i$. As a consequence, we have to choose one canonical copy $\mathcal{A}_i^<$ of $\mathcal{A}_i$ among the set of possible candidates (which we can do, since the structures $\mathcal{A}_i^<$ have an *ordered* universe). However, fixing a canonisation for one of the substructures $\mathcal{A}_i$ imposes constraints on the remaining choices of canonisations $\mathcal{A}_j^<$ of the substructures $\mathcal{A}_j$, since different substructures $\mathcal{A}_i$ and $\mathcal{A}_j$ can have common vertices. This means that when we combine the canonised parts $\mathcal{A}_i^<$ for the different substructures $\mathcal{A}_i$, then we have to ensure that the choices we make are compatible.

To guarantee this, we maintain a set of isomorphisms (a subset of $\mathcal{O}(A)$ with a certain algebraic structure) between the processed part of the input structure and the partial canonisation that we constructed so far. As we mentioned before, the main technical step is to succinctly encode this exponential-sized set of witnessing isomorphisms by using the notion of cyclic linear equation systems.

Let us now formulate the approach precisely. First of all, we define the decomposition of $\mathcal{A}$ as $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{m-1}$ where $\mathcal{A}_k$ is the subgraph of $\mathcal{A}$ which is induced on the vertex set $A_{ij} := A_i \cup A_j$ for $0 \leq i \leq j < n$ (and where $k$ is the position of the set $\{i, j\}$ in an enumeration of $\binom{[n]}{2} \cup [n]$). We say that $\mathcal{A}_k$ denotes the *k-th component* of $\mathcal{A}$ and that $A_i$ and $A_j$ are the *relevant* colours. For $s \in [m]$ we denote by $\mathcal{A}[s] \subseteq \mathcal{A}$ the (not necessarily induced) subgraph of $\mathcal{A}$ that consists of the first $s$ components, that is $\mathcal{A}[s] = \mathcal{A}_0 \cup \cdots \cup \mathcal{A}_{s-1}$. For

technical reasons, we sometimes assume that the substructures $\mathcal{A}_i$ are defined over the whole universe $A$ (but that they only contain the part of $E$ which is induced on the relevant colour classes).

**Definition 6.10.** Let $s \leq m$. An *$s$-canonisation of $\mathcal{A}$* is a canonisation of $\mathcal{A}[s]$, that is an ordered structure $\pi(\mathcal{A}[s]) = \pi(\mathcal{A}_0) \cup \cdots \cup \pi(\mathcal{A}_{s-1})$ for some $\pi \in \mathcal{O}(A)$. A non-empty set $C \subseteq \mathcal{O}(A)$ *witnesses* this $s$-canonisation if $\pi(\mathcal{A}_j) = \sigma(\mathcal{A}_j)$ for all $\pi, \sigma \in C$ and $j \in [s]$.

Since $\mathcal{A} = \bigcup_{i \in [m]} \mathcal{A}_i$, an $m$-canonisation of $\mathcal{A}$ is a canonisation of $\mathcal{A}$. Thus we aim to use the decomposition $\mathcal{A} = \mathcal{A}_0 \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_{m-1}$ of $\mathcal{A}$ to design an inductive canonisation procedure. Specifically, we want to iteratively construct $s$-canonisations $\mathcal{A}^<[s]$ of $\mathcal{A}$ for increasing values of $s \leq m$. While doing so, we maintain a set $C_s \subseteq \mathcal{O}(A)$ of isomorphisms that witnesses the partial canonisations $\mathcal{A}^<[s]$ of $\mathcal{A}[s]$. This last step is necessary to guarantee the consistency of our construction.

Before we present the full canonisation procedure in Figure 6.2, we need a further notation. For $C \subseteq \mathcal{O}(A_{ij}) := \mathcal{O}(A_i) \times \mathcal{O}(A_j)$, where $0 \leq i \leq j \leq n$, we define the *extension* of $C$ to $\mathcal{O}(A)$ as the set $\mathrm{ext}(C) = \{(\pi_0, \dots, \pi_{n-1}) \in \mathcal{O}(A) : (\pi_i, \pi_j) \in C\} \subseteq \mathcal{O}(A)$. Similarly, for $0 \leq i \leq j < n$, and for a group $\Delta \leq \Gamma_{ij} := \Gamma_i \times \Gamma_j$, we denote by $\mathrm{ext}(\Delta) \leq \Gamma$ the *extension* of $\Delta$ to $\Gamma$, that is

$$\mathrm{ext}(\Delta) = \{(\gamma_0, \dots, \gamma_{n-1}) \in \Gamma : (\gamma_i, \gamma_j) \in \Delta\}.$$

Let us explain our canonisation procedure (Figure 6.2) more precisely. For the main loop, we assume that $s \geq 1$ and that we have constructed an $(s-1)$-canonisation $\mathcal{A}^<[s-1]$ of $\mathcal{A}$ together with a set $C_{s-1} \subseteq \mathcal{O}(A)$ of witnessing isomorphisms. Our task is to extend this $(s-1)$-canonisation $\mathcal{A}^<[s-1]$ of $\mathcal{A}$ by a canonical copy $\mathcal{A}^<_{s-1} := \pi(\mathcal{A}_{s-1})$ of the $(s-1)$-th component $\mathcal{A}_{s-1}$ in such a way that the choice is consistent with our decisions before, that is we have to ensure that we select $\pi \in C_{s-1}$ to define $\pi(\mathcal{A}_{s-1}) = \mathcal{A}^<_{s-1}$. Moreover, we have to update the set of witnessing isomorphisms as $C_s := \{\sigma \in C_{s-1} : \sigma(\mathcal{A}_{s-1}) = \mathcal{A}^<_{s-1}\}$.

To start, we identify in step (1) the colour classes $A_i, A_j$ which are relevant for the subgraph $\mathcal{A}_{s-1}$. In the next step we would like to choose $\pi \in C_{s-1}$ in order to define $\pi(\mathcal{A}_{s-1}) = \mathcal{A}^<_{s-1}$. However, recall that a CPT-program cannot fix a complete isomorphism $\pi \in C_{s-1} \subseteq \mathcal{O}(A)$. Fortunately, in order to determine $\pi(\mathcal{A}_{s-1})$ it suffices to know how $\pi$ acts on the relevant colour classes $A_i$ and $A_j$. Thus, instead of selecting $\pi \in C_{s-1}$ we can simply choose $\pi$ from the set $\mathcal{O}(A_{ij}) = \mathcal{O}(A_i) \times \mathcal{O}(A_j)$ (which we *can*, in contrast to $\mathcal{O}(A)$, access explicitly), but by doing so we have to guarantee that $\pi$ can be extended to an isomorphism in $C_{s-1}$. This is why we define in step (2) the set $O_{s-1}$ of all possible orderings in $\mathcal{O}(A_{ij})$ which are consistent with $C_{s-1}$ in this sense.

We next argue that the algebraic structure on $C_{s-1}$ and on $\mathcal{O}(A_{ij})$ can be transferred to the set $O_{s-1} \subseteq \mathcal{O}(A_{ij})$. First, we know by our induction assumption that $C_{s-1} = \sigma_{s-1}\Lambda_{s-1}$ for some $\sigma_{s-1} \in \mathcal{O}(A)$ and some group $\Lambda_{s-1} \leq \Gamma$ (again we stress the fact that we do not have access to $\sigma_{s-1} \in \mathcal{O}(A)$).

**Given:** Graph $\mathcal{A} = (A, \preceq, E, \Phi)$ with Abelian colours (notation as above)

(0) Construct sets of $A_i$-orderings $\pi_i\Gamma_i = \mathcal{O}(A_i)$ (according to Lemma 6.9)

(In what follows, construct for $s \in [m]$ an $s$-canonisation $\mathcal{A}^<[s]$ with witnessing set $C_s \subseteq \mathcal{O}(A)$, i.e. for $\pi, \sigma \in C_s$ we have $\pi(\mathcal{A}[s]) = \sigma(\mathcal{A}[s]) = \mathcal{A}^<[s]$. Moreover, $C_s$ has the following algebraic structure $C_s = \sigma_s\Lambda_s$ for $\sigma_s \in \mathcal{O}(A)$ and $\Lambda_s \leq \Gamma$)

$C_0 := \pi\Gamma = \mathcal{O}(A)$ and $\mathcal{A}^<[0] := \varnothing$

**for** $s = 1$ **to** $m$ **do**

(1) Let $\mathcal{A}_{s-1}$ be the subgraph which is induced on the set $A_i \cup A_j$

(2) Define $O_{s-1} := \{\sigma \in \mathcal{O}(A_{ij}) : \mathrm{ext}(\sigma) \cap C_{s-1} \neq \varnothing\}$

(3) Define $\Delta_{s-1} := \{\pi^{-1}\sigma : \pi, \sigma \in O_{s-1}\} \cap \mathrm{Aut}(\mathcal{A}_{s-1}) \leq \Gamma_{ij}$

(4) Partition $O_{s-1}$ as $D_{s-1} := \{\sigma\Delta_{s-1} : \sigma \in O_{s-1}\}$

(5) Fix some set $\sigma\Delta_{s-1} \in D_{s-1}$

(6) Set $C_s := C_{s-1} \cap \mathrm{ext}(\sigma\Delta_{s-1})$

(7) Set $\mathcal{A}^<[s] := \mathcal{A}^<[s-1] \cup \pi(\mathcal{A}_{s-1})$ for some $\pi \in \sigma\Delta_{s-1}$

**end for**

**Return:** The canonisation $\mathcal{A}^< := \mathcal{A}^<[m]$ of $\mathcal{A}$

(and isomorphisms $\psi_i : \Gamma_i \to {}^{\pi_i}\Gamma_i$ which map $\Gamma_i$ to its canonical copy ${}^{\pi_i}\Gamma$)

Figure 6.2: Canonisation procedure for structures with Abelian colours

Let us denote by $\Lambda^{ij}_{s-1} \leq \Gamma_{ij} = \Gamma_i \times \Gamma_j$ the induced action of $\Lambda_{s-1}$ on the pair of colour classes $A_i \times A_j$ and let us similarly denote by $\sigma^{ij}_{s-1} \in \mathcal{O}(A_{ij})$ the restriction of $\sigma_{s-1}$ to the colour classes $A_i$ and $A_j$. Then we have that $O_{s-1} = \sigma^{ij}_{s-1}\Lambda^{ij}_{s-1}$ where $\Lambda^{ij}_{s-1} \leq \Gamma_{ij}$ and $\sigma^{ij}_{s-1} \in \mathcal{O}(A_{ij})$.

Note that $\Lambda^{ij}_{s-1} = \{\pi^{-1}\sigma : \pi, \sigma \in O_{s-1}\}$. In general, two orderings $\pi, \sigma \in O_{s-1}$ define different ordered copies $\pi(\mathcal{A}_{s-1}) \neq \sigma(\mathcal{A}_{s-1})$ of the $(s-1)$-th component $\mathcal{A}_{s-1}$. Thus, in step (3) and step (4), we partition the set $O_{s-1}$ into blocks of isomorphisms which map $\mathcal{A}_{s-1}$ to the same ordered graph. As we elaborated before, the structure of such blocks can be described by the action of the subgroup $\Delta_{s-1} = \mathrm{Aut}(\mathcal{A}_{s-1}) \cap \Lambda^{ij}_{s-1} \leq \Lambda^{ij}_{s-1}$ on $O_{s-1}$, that is the set of orbits $D_{s-1} = \{\sigma\Delta_{s-1} : \sigma \in O_{s-1}\}$ of the action of $\Delta_{s-1}$ on $O_{s-1}$ corresponds to the desired partition. Moreover, since every block $\sigma\Delta_{s-1} \in D_{s-1}$ corresponds to a different ordered graph $\sigma(\mathcal{A}_{s-1})$, we can fix in step (5) such a block canonically.

Next, for some (or equivalently all) $\pi \in \sigma\Delta_{s-1}$ we let $\mathcal{A}^<_{s-1} := \pi(\mathcal{A}_{s-1})$ denote the ordered copy of the component $\mathcal{A}_{s-1}$ which was fixed by our choice of $\sigma\Delta_{s-1} \in D$. Then we can refine in step (6) the set of witnessing isomorphisms as $C_s := C_{s-1} \cap \mathrm{ext}(\sigma\Delta_{s-1})$ (which yields a non-empty set) to guarantee that for all $\pi \in C_s$ we have $\pi(\mathcal{A}_{s-1}) = \mathcal{A}^<_{s-1}$. Finally, we can extend in step (7) the $(s-1)$-canonisation $\mathcal{A}^<[s-1]$ of $\mathcal{A}$ by the canonisation $\mathcal{A}^<_{s-1}$ of the $(s-1)$-th component $\mathcal{A}_{s-1}$ to obtain the desired $s$-canonisation $\mathcal{A}^<[s]$ of $\mathcal{A}$.

Unfortunately, it is not clear how to formulate this canonisation procedure in Choiceless Polynomial Time. The difficulty is to maintain the sets of

witnessing isomorphisms $C_s \subseteq \mathcal{O}(A)$, which might be of exponential size, during our iterative construction of $\mathcal{A}^<$. In other words, we have to find a succinct and CPT-definable representation for these sets of isomorphisms $C_s$ such that certain basic operations (most importantly, testing for emptiness and taking intersections) are CPT-definable as well. This will be the subject of the following subsection where we show how to represent the sets $C_s$ by families of cyclic linear equation systems. Before we proceed, let us summarise the requirements for representations of the sets $C_s$ which are needed to express our canonisation procedure for structures with Abelian colours (Figure 6.2) in Choiceless Polynomial Time.

**Definition 6.11.** A *suitable representation (of witnessing sets of isomorphisms)* is a triple of CPT-programs $(\Pi_\varnothing, \Pi_\cap, \Pi_{\text{ext}})$ such that, given the preconditions as in Figure 6.2 (with the notation from above), the programs manipulate succinct representations $\rho \in \mathrm{HF}(A)$ of sets of isomorphisms $\sigma\Delta \subseteq \mathcal{O}(A)$, where $\Delta \leq \Gamma$, in the following way.

(i) *Consistency.* Given a representation $\rho$ of a set $\sigma\Delta$, the program $\Pi_\varnothing$ defines whether $\sigma\Delta \neq \varnothing$.

(ii) *Intersection.* Given two representations $\rho_1, \rho_2$ of sets $\sigma_1\Delta_1$ and $\sigma_2\Delta_2$, a representation $\rho$ of the set $\sigma_1\Delta_1 \cap \sigma_2\Delta_2$ is defined by $\Pi_\cap$.

(iii) *Representation of basic sets.* Given $\sigma\Delta \subseteq \mathcal{O}(A_{ij})$ for $0 \leq i \leq j < n$ and $\Delta \leq \Gamma_{ij}$, $\Pi_{\text{ext}}$ defines a representation of $\mathrm{ext}(\sigma\Delta) \subseteq \mathcal{O}(A) = \pi\Gamma$.

### 6.2.2 Representing sets of witnessing isomorphisms

What remains is to find suitable representations for the sets of witnessing isomorphisms $C_s = \sigma_s\Lambda_s$, where $\Lambda_s \leq \Gamma$ and where $\sigma_s \in \mathcal{O}(A)$, which satisfy the requirements of Definition 6.11. Basically, we aim to associate single isomorphisms with vectors over finite rings $\mathbb{Z}_d$ in such a way that the resulting sets of vectors inherit the algebraic structure of the sets $C_s$. This will make it possible to describe these sets of vectors as solution spaces of (cyclic) linear equation systems. The crucial step is to show that such a correspondence between isomorphisms and vectors can be defined, in Choiceless Polynomial Time, for the sets $\pi_i\Gamma_i = \mathcal{O}(A_i)$ for each colour class $A_i$.

**Lemma 6.12.** *There exists a* CPT*-program* $\Pi$ *such that, given a set* $B \subseteq \mathrm{HF}(A)$ *together with an ordered and Abelian group* $\Gamma \leq \mathit{Sym}(B)$ *which acts transitively on* $B$*, the program* $\Pi$ *defines the associated set* $\mathcal{O}(B) = \pi\Gamma$ *of* $B$*-orderings (according to Lemma 6.9) and*

- *a decomposition of* $\Gamma$ *into subgroups of prime-power order, that is* $\Gamma = \langle\delta_1\rangle \oplus \cdots \oplus \langle\delta_k\rangle$ *for* $\delta_1, \ldots, \delta_k \in \Gamma$ *where* $|\delta_i| = d_i$ *is a prime-power, and*
- *sets* $W_1, \ldots, W_k \subseteq \mathrm{HF}(B)$ *where* $|W_i| = d_i$ *together with a linear order* $W_1 < W_2 < \cdots < W_k$*, and*

- *if we set* $L_i \coloneqq \mathbb{Z}_{d_i}^{W_i}$ *and let* $e_i \in L_i$ *denote the* $L_i$*-identity vector which is defined as* $e_i(w) = 1$ *for all* $w \in W_i$*, then* $\Pi$ *defines an embedding* $\varphi : \pi\Gamma \to L_1 \times \cdots \times L_k$ *which respects the action of* $\Gamma$ *on* $\pi\Gamma$ *in the following way. For all* $\sigma \in \pi\Gamma$ *and* $\gamma = \ell_1 \cdot \delta_1 \oplus \cdots \oplus \ell_k \cdot \delta_k \in \Gamma$ *we have that*

$$\varphi(\sigma\gamma) = \varphi(\sigma) + (\ell_1 \cdot e_1, \cdots, \ell_k \cdot e_k).$$

*In other words, via the canonical group embedding* $\psi : \Gamma \to L_1 \times \cdots \times L_k, (\ell_1 \cdot \delta_1 \oplus \cdots \oplus \ell_k \cdot \delta_k) \mapsto (\ell_1 \cdot e_1, \cdots, \ell_k \cdot e_k)$*, the action of* $\Gamma$ *on* $\pi\Gamma$ *corresponds to the action of* $\psi(\Gamma)$ *on* $\varphi(\pi\Gamma)$*.*

*Proof.* First of all, we use the linear order on $\Gamma$ to fix in CPT a set of generators $\delta_1, \ldots, \delta_k \in \Gamma$ of $\Gamma$ which yield a decomposition of $\Gamma$ as required (recall from Section 2.5 that this step can be expressed already in fixed-point logic). For the remaining parts we use the following recursive procedure.

If $k = 1$, then $\Gamma = \langle \delta \rangle$ is a cyclic group of prime-power order $d$ which acts transitively on $W \coloneqq B$. Since $\Gamma$ is Abelian, we can define the *unique* group isomorphism $\theta : \Gamma \to {}^{\sigma}\Gamma, \gamma \mapsto {}^{\sigma}\gamma$ for $\sigma \in \pi\Gamma$. We let $B^< \coloneqq \{0, \ldots, |B| - 1\}$ denote an ordered set of size $|B|$ and we define the unique $\rho \in \mathrm{Sym}(B^<)$ such that ${}^{\rho}({}^{\sigma}\delta) = {}^{\rho\sigma}\delta = (0\,1\,2 \cdots |B| - 1) \in \mathrm{Sym}(B^<)$. We let $L = \mathbb{Z}_d^W$ and we denote by $e \in L$ the $L$-identity vector, i.e. $e(w) = 1$ for all $w \in W$. Then the mapping $\varphi : \pi\Gamma \to L, \sigma \mapsto \varphi(\sigma)$ where $\varphi(\sigma)(w) \coloneqq \rho\sigma(w)$ for $w \in W$ is CPT-definable. We claim that $\varphi(\sigma \circ \delta) = \varphi(\sigma) + e$ for all $\sigma = \pi\delta^s \in \pi\Gamma$. To verify this let $w \in W$. Then $(\varphi(\sigma) + e)(w) = \varphi(\sigma)(w) + 1 = (({}^{\rho\sigma}\delta)(\rho\sigma\delta^s))(w) = \rho\sigma\delta^{s+1}(w) = \varphi(\sigma \circ \delta)(w)$.

Now let $k > 1$. Then $\Gamma = \Delta \oplus \Lambda$ where $\Delta = \langle \delta_1 \rangle$ and $\Lambda = \langle \delta_2 \rangle \oplus \cdots \oplus \langle \delta_k \rangle$. We partition $B$ into the set $\bar{X} \coloneqq \{X_0, \ldots, X_{t-1}\}$ of $\Lambda$-orbits and the set $\bar{Y} \coloneqq \{Y_0, \ldots, Y_{s-1}\}$ of $\Delta$-orbits. It holds that the subgroups $\Delta$ and $\Lambda$ act transitively on $\bar{X}$ and on $\bar{Y}$, respectively. Furthermore, note that $t = d_1$ and $s = \Pi_{j=2}^{k} d_j$ and that $t = |\Delta|$ and $s = |\Lambda|$. We claim that for each pair of sets $(X, Y) \in \bar{X} \times \bar{Y}$ it holds that $X \cap Y = \{b\}$ for $b = b_{(X,Y)} \in B$. To see this, assume that $X \cap Y \supseteq \{b, c\}$, $b \neq c$. Then there were a $\delta \in \Delta$ and a $\lambda \in \Lambda$ which both map $b$ to $c$. This, however, is a contradiction to our assumption that the action of $\Gamma$ on $B$ is regular (since $\Gamma$ is an Abelian and transitive group). Moreover, the intersection of $X \cap Y$ cannot be empty because of the pigeonhole principle: all elements of $X$ (note that $|X| = s$) have to appear in some of the orbits in $\bar{Y}$.

In other words, every element $b = b_{(X,Y)} \in B$ can be identified with the unique pair $(X, Y) \in \bar{X} \times \bar{Y}$ such that $X \cap Y = \{b\}$. This is the crucial insight for our following construction. More precisely, we aim to infer from an ordering of the set of $\Lambda$-orbits $\bar{X}$ and from an ordering of the set of $\Delta$-orbits $\bar{Y}$, an ordering of the set $B$.

To describe the formal construction we partition the ordered set $B^< = \{0, \ldots, |B| - 1\}$ into $t$ segments $S_0 = \{0, \ldots, s - 1\}$, $S_1 = \{s, \ldots, 2s - 1\}, \ldots, S_{t-1} = \{(t-1)s, \ldots, ts - 1\}$. Then each of these segments has size $s$ and, modulo $s$, each segment is a disjoint copy of the initial segment $S_0$. Similarly, we partition

$B^<$ into $s$ segments $T_0, T_1, \dots, T_{s-1}$ by setting $T_r := \{0+r, s+r, \dots, (t-1)s+r\}$. Note that each set $T_r$ has size $t$ and, modulo $s$, the set $T_r$ represents $r$ where $0 \le r < s$.

In the next step we determine a permutation $\rho \in \mathrm{Sym}(B^<)$ over the ordered domain $B^<$ such that for every $\sigma \in \mathcal{O}(B) = \pi\Gamma$ the following holds:

(i) for every $\Lambda$-orbit $X \in \bar{X}$ we have $\rho\sigma(X) = S_j$ for a $j = 0, \dots, t-1$, and

(ii) for every $\Delta$-orbit $Y \in \bar{Y}$ we have $\rho\sigma(Y) = T_r$ for some $0 \le r < s$.

Let us first show that such a permutation $\rho \in \mathrm{Sym}(B^<)$ exists and that it can be constructed by a CPT-program. We start by fixing an arbitrary bijection $\sigma \in \mathcal{O}(B)$. Then $\sigma$ induces a linear order on $\bar{X}$. Hence we can easily construct a permutation $\rho_1 \in \mathrm{Sym}(B^<)$ such that $\rho_1\sigma(X) \in \{S_0, \dots, S_{t-1}\}$. Next we want to construct a permutation $\rho_2 \in \mathrm{Sym}(B^<)$ such that

- $\rho_2(S_i) = S_i$ for all $i \in [t]$, and
- $\rho_2\rho_1\sigma(Y) \in \{T_0, \dots, T_{s-1}\}$ for all $Y \in \bar{Y}$.

If we have achieved this, then we can set $\rho := \rho_2\rho_1$ to satisfy our claim from above. We construct $\rho_2$ recursively by using the linear order on $\bar{Y}$. Assume that for some orbit $Y \in \bar{Y}$ (for example, the minimal one with respect to the order induced by $\sigma$) we have $\rho_2\rho_1(Y) \cap T_i \neq \varnothing$ but $\rho_2\rho_1(Y) \neq T_i$ for some minimal $i \in [s]$. Then we can fix $a \in Y$ such that $\rho_2\rho_1(a) \notin T_i$ and $b \in B \setminus Y$ such that $\rho_2\rho_1(b) \in T_i$. Let $\rho_3 = (\,\rho_2\rho_1(a)\,\rho_2\rho_1(b)\,) \in \mathrm{Sym}(B^<)$ denote the transposition of the two elements $\rho_2\rho_1(a)$ and $\rho_2\rho_1(b)$. Then we have $\rho_3\rho_2\rho_1(a) \in T_i$ and thus we made progress in the sense that we increased the number of elements in $Y$ which are mapped to positions in $T_i$. Moreover, since $a, b$ belong to different $\Delta$-orbits we know by our assumption that $\rho_2\rho_1(a), \rho_2\rho_1(b) \in S_j$ for some $j \in [t]$. Thus $\rho_3(S_j) = S_j$ and we can maintain the property that $\rho_3\rho_2\rho_1(S_j) = S_j$ for all $j \in [t]$.

Still, the dependence of our construction on $\sigma \in \pi\Gamma$ seems to be problematic. More strikingly, we have only verified the claimed properties for the single ordering $\sigma \in \mathcal{O}(B) = \pi\Gamma$. However, since $\Gamma$ acts on both sets $\bar{X}$ and $\bar{Y}$, it is easy to see that $\rho$ indeed satisfies the above properties with respect to *every* ordering in $\sigma\Gamma$. Also, if by our construction we really obtain different $\rho$, then we can just canonically choose the (lexicographically) minimal one.

As observed earlier, the action of $\Gamma = \Delta \oplus \Lambda$ on $B$ corresponds to the component-wise action of $\Delta \oplus \Lambda$ on $\bar{X} \times \bar{Y}$. Specifically, we obtain a CPT-definable embedding $\eta : \rho\pi\Gamma \to \mathcal{O}(\bar{X}) \times \mathcal{O}(\bar{Y}), \rho\pi\gamma \mapsto \eta(\rho\pi\gamma)$ if we let $\eta(\rho\pi\gamma) \in \mathcal{O}(\bar{X}) \times \mathcal{O}(\bar{Y})$ be the linear order which assigns to every $\Lambda$-orbit $X \in \bar{X}$ the position $j$ for $0 \le j \le t-1$ such that $\rho\pi\gamma(X) = S_j$ and to each $\Delta$-orbit $Y \in \bar{Y}$ the position $0 \le r \le s-1$ such that $\rho\pi\gamma(Y) = T_r$. For all $\delta \oplus \lambda \in \Delta \oplus \Lambda$ and $\sigma \in \rho\pi\Gamma$ we have $\eta(\sigma \circ (\delta \oplus \lambda)) = (\eta(\sigma) \restriction \bar{X} \circ \delta, \eta(\sigma) \restriction \bar{Y} \circ \lambda)$. In particular we have that $\eta(\rho\pi\Gamma) = \nu_X\Delta \times \nu_Y\Lambda$ where $\nu_X \in \mathcal{O}(\bar{X})$ and $\nu_Y \in \mathcal{O}(\bar{Y})$.

Recursively for the smaller groups $\Delta = \langle \delta_1 \rangle$ and $\Lambda = \langle \delta_2 \rangle \oplus \cdots \oplus \langle \delta_k \rangle$ that act on $\bar{X}$ and $\bar{Y}$, respectively, and for $\nu_X \Delta$ and $\nu_Y \Lambda$ we obtain two ordered sequences of CPT-definable sets $W_1$ and $W_2 < \cdots < W_k$, and for $L_i = \mathbb{Z}_{d_i}^{W_i}$ the CPT-definable embeddings $\varphi^X : \nu_X \Delta \to L_1$ and $\varphi^Y : \nu_Y \Lambda \to L_2 \times \cdots \times L_k$, with the appropriate properties as stated above. Now we put everything together to obtain the desired embedding $\varphi : \pi\Gamma \to L_1 \times \cdots \times L_k$ via

$$\varphi(\pi\gamma) = \varphi^X(\eta(\rho\pi\gamma) \restriction \bar{X}) \times \varphi^Y(\eta(\rho\pi\gamma) \restriction \bar{Y}). \qquad \square$$

Before we proceed, we want to present an alternative proof of Lemma 6.12. The reason is that if we use our (recursive) construction from above, then we obtain index sets $W_i$ whose (set-theoretic) rank depends on the number $k$ of summands of $\Gamma$. For instance, if $k \geq 2$ and if $B \subseteq A$ is a set of atoms, then the index sets $\bar{X}$ and $\bar{Y}$ which are constructed in the first step are *sets of orbit* of elements from $B$, which means that both sets, $\bar{X}$ and $\bar{Y}$, have rank two. More generally, if the set $B$ has rank $\ell$, then the sets $\bar{X}$ and $\bar{Y}$ are sets of rank $\ell + 1$. Since in the following steps we recursively apply our construction to the resulting sets $\bar{Y}$, it follows that the index set $W_k$ has rank $\ell + k$.

Although this is not a problem in the setting of Choiceless Polynomial Time, it can become problematic if we want to express our canonisation procedure in logics which cannot access higher-order objects. Indeed, very recently we studied the question of whether our procedure can be expressed in solvability logic (with solvability operators over all finite rings $\mathbb{Z}_d$). It turns out that this is indeed possible, but we have to adapt our proof from above to avoid index sets of unbounded rank. For details we refer to the Bachelor's thesis of Matthias Voit [88].

*Alternative proof (sketch) of Lemma 6.12, see [88].* Again, we first decompose $\Gamma$ as $\Gamma = \langle \delta_1 \rangle \oplus \cdots \oplus \langle \delta_k \rangle$ for $\delta_1, \ldots, \delta_k \in \Gamma$ where $|\delta_i| = d_i$ is a prime power. The main difference to our proof from above is that we construct the index sets $W_i$ at once rather than using recursive applications of our procedure along the decomposition of $\Gamma$.

Let $\Delta_i = \langle \delta_1 \rangle \oplus \cdots \oplus \langle \delta_{i-1} \rangle \oplus \langle \delta_{i+1} \rangle \oplus \cdots \langle \delta_k \rangle$ be the restriction of $\Gamma$ to all summands which are different from $\langle \delta_i \rangle$. Moreover, let $W_i$ be the set of all $\Delta_i$-orbits on $B$. Then $|W_i| = d_i$ and $\langle \delta_i \rangle$ acts transitively on $W_i$. Hence, we can define a directed cycle on $W_i$ whose automorphism group coincides with the action of $\langle \delta_i \rangle$ on $W_i$. Moreover, for every $\sigma \in \pi\Gamma$ we know that $\sigma(W_i)$ yields a partition of $B^<$. In particular, for each $\sigma \in \pi\Gamma$ we can identify one block $Z_\sigma^i \in W_i$ (say, the unique block such that $0 \in \sigma(Z_\sigma^i)$). With this preparation we can easily define the vector $\varphi(\sigma)$. For $Z \in W_i$ we set

$$(\varphi(\sigma))(Z) = \begin{cases} 0, & \text{if } Z = Z_\sigma^i, \\ \ell, & \text{if } Z = \delta_i^\ell(Z_\sigma^i), 0 < \ell < d_i. \end{cases}$$

It is easy to verify that this embedding satisfies the claimed properties. $\square$

With Lemma 6.12 we are prepared to show that the witnessing sets of isomorphisms $C_s = \sigma_s \Lambda_s \subseteq \mathcal{O}(A) = \pi\Gamma$ can be encoded in Choiceless Polynomial Time by sequences of cyclic linear equation systems. One technical difficulty which remains is that the groups $\Gamma_i$ can contain elements of co-prime order. In contrast, we have defined cyclic linear equation systems only over rings $\mathbb{Z}_d$ where $d$ is a prime power. However, from the previous lemma we already know that we can decompose the groups $\Gamma_i$ into direct sums of subgroups of prime power order. As a consequence we can treat the components for different primes separately. This is why we use a *sequence* of cyclic linear equation systems (instead of a single system) to cover all primes which occur in the factorisation of the order of $\Gamma$.

To proceed, we first apply Lemma 6.12 and decompose $\Gamma_i = \langle \delta_1^i \rangle \oplus \cdots \oplus \langle \delta_{k_i}^i \rangle$ for all $i \in [n]$ into a direct sum of cyclic groups $\langle \delta_j^i \rangle$ where for all $1 \leq j \leq k_i$ the order $d_j^i$ of $\delta_j^i$ is a prime-power and we define (again for all $i \in [n]$) sets $W_1^i < W_2^i < \cdots < W_{k_i}^i$ of size $|W_j^i| = d_j^i$ and for $L_j^i := \mathbb{Z}_{d_j^i}^{W_j^i}$ with corresponding $L_j^i$-identity vectors $e_j^i \in L_j^i$ two embeddings

- $\varphi^i : \pi_i\Gamma_i \to L_1^i \times \cdots \times L_{k_i}^i$,
- $\psi^i : \Gamma_i \mapsto L_1^i \times \cdots \times L_{k_i}^i, (\ell_1 \cdot \delta_1^i \oplus \cdots \oplus \ell_{k_i} \cdot \delta_{k_i}^i) \mapsto (\ell_1 \cdot e_1^i, \ldots, \ell_{k_i} \cdot e_{k_i}^i)$,

such that for all $\sigma \in \pi_i\Gamma_i$ and all $\gamma \in \Gamma_i$ we have

$$\varphi^i(\sigma\gamma) = \varphi^i(\sigma) + \psi^i(\gamma).$$

We set $L = L_1^0 \times \cdots \times L_{k_0}^0 \times \cdots \times L_1^{n-1} \times \cdots \times L_{k_{n-1}}^{n-1}$ and we combine the embeddings $\varphi^i$ of $\pi_i\Gamma_i$ in $L_1^i \times \cdots \times L_{k_i}^i$ to obtain an embedding $\varphi$ of $\pi\Gamma$ in $L$, that is we let $\varphi : \pi\Gamma \to L, (\sigma_0, \ldots, \sigma_{n-1}) \mapsto (\varphi^1(\sigma_0), \ldots, \varphi^{n-1}(\sigma_{n-1}))$. Similarly, we combine the embeddings $\psi^i$ of $\Gamma_i$ in $L$ to obtain an embedding $\psi$ of $\Gamma$ in $L$, that is $\psi : \Gamma \to L$ is defined as $\psi((\gamma_0, \ldots, \gamma_{n-1})) = (\psi^0(\gamma_0), \ldots, \psi^{n-1}(\gamma_{n-1}))$.

Our goal is to represent a subset $\sigma\Delta \subseteq \pi\Gamma$ (a witnessing set of isomorphisms) as the solution space of a sequence of cyclic linear equation systems. Note that via the embedding $\varphi$ we can represent $\sigma\Delta$ as the subset $\varphi(\sigma\Delta)$ of the linear space $L$. What remains is to show that the algebraic structure of $\varphi(\sigma\Delta)$ suffices to encode this set as the solution space of a family of cyclic linear equation systems.

To see this, we first analyse $\varphi(\pi\Gamma)$ restricted to a single component $L_j^i$, that is the set $O_j^i := (\varphi(\pi\Gamma) \restriction L_j^i) = (\varphi(\pi_i\Gamma_i) \restriction L_j^i) \subseteq L_j^i$. If we similarly denote by $E_j^i := (\psi(\Gamma_i) \restriction L_j^i) = \{\ell \cdot e_j^i : 0 \leq \ell \leq d_j^i - 1\} \subseteq L_j^i$ the restriction of the embedding of $\Gamma_i$ into $L$ to the component $L_j^i$, then we get $O_j^i = O_j^i + E_j^i$. This means that for all $W_j^i$-vectors $\vec{x}, \vec{y} \in O_j^i$ over $\mathbb{Z}_{d_j^i}$ it holds that $\vec{x} - \vec{y} \in E_j^i$. This in turn means that for all $\vec{x}, \vec{y} \in O_j^i$ and indices $w, w' \in W_j^i$ we have $\vec{x}(w) - \vec{x}(w') = \vec{y}(w) - \vec{y}(w')$. Hence, we can define a cyclic constraint $C_j^i$

on the set $W_j^i$ such that $O_j^i$ corresponds precisely to the set of assignments $\alpha : W_j^i \to \mathbb{Z}_{d_j^i}$ with $\alpha \vDash C_j^i$ (recall Definition 5.1).

As mentioned before, a technical problem arises from the fact that the linear spaces $L_j^i$ for $i \in [n], 1 \leq j \leq k_i$ are defined over rings $\mathbb{Z}_d$ for co-prime integers $d = d_j^i$. To deal with this issue we let $P \coloneqq \{p_1, \dots, p_s\}$ denote the set of all primes $p_i \in \mathbb{P}$ such that $\Gamma$ contains elements of order $p_i$. For $p \in P$ we further let $\Gamma_i^p \leq \Gamma_i$ denote the subgroup of $\Gamma_i$ which consists of all elements $\gamma \in \Gamma_i$ whose order is a power of $p$. Then it holds that $\Gamma_i = \Gamma_i^{p_1} \oplus \cdots \oplus \Gamma_i^{p_s}$. In particular, every summand $\Gamma_i^p$ for $p \in P$ is the direct sum of all subgroups $\langle \delta_j^i \rangle \leq \Gamma_i$ where $d_j^i$ is a power of $p$. Of course we also have that $\psi(\Gamma_i) = \psi(\Gamma_i^{p_1}) + \cdots + \psi(\Gamma_i^{p_s})$.

Similarly, for any subgroup $\Delta \leq \Gamma$ and prime $p \in P$ we let $\Delta^p \leq \Delta$ denote the subgroup of $\Delta$ which consists of elements $\delta \in \Delta$ whose order is a $p$-power. Then $\Delta = \Delta^{p_1} \oplus \cdots \oplus \Delta^{p_s}$ and $\Delta^p \leq \Gamma_0^p \times \Gamma_1^p \times \cdots \times \Gamma_{n-1}^p =: \Gamma^p$.

From the definition of $\psi$ it further follows that when we embed the subgroup $\Gamma^p$ for $p \in P$ into $L$ via $\psi$, then all components $L_j^i$ where $d_j^i$ is co-prime to $p$ are zero. To capture this observation formally we define for every prime $p \in P$ the following subspace $L[p]$ of $L$,

$$L[p] \coloneqq \{(v_1^0, \dots, v_{k_0}^0, \dots, v_1^{n-1}, \dots, v_{k_{n-1}}^{n-1}) \in L : \text{if } v_j^i \neq 0 \text{ then } d_j^i \text{ is a } p\text{-power}\}.$$

Then our above observation can be phrased as $\psi(\Gamma^p) \leq L[p]$. In particular, we obtain a decomposition $L = L[p_1] \oplus \cdots \oplus L[p_s]$ of $L$ which corresponds to the decomposition of $\Gamma$ as $\Gamma^{p_1} \oplus \cdots \oplus \Gamma^{p_s}$.

For $\sigma \in \mathcal{O}(A)$ let us denote by $\varphi(\sigma)^{L[p]} \coloneqq (\varphi(\sigma) \restriction L[p])$ the projection of the vector $\varphi(\sigma) \in L$ to the subspace $L[p]$. Then for every subgroup $\Delta \leq \Gamma$ we obtain a decomposition of $\varphi(\sigma\Delta)$ as

$$\varphi(\sigma\Delta) = \varphi(\sigma)^{L[p_1]} + \psi(\Delta^{p_1}) \oplus \cdots \oplus \varphi(\sigma)^{L[p_s]} + \psi(\Delta^{p_s}) \subseteq L[p_1] \oplus \cdots \oplus L[p_s].$$

Hence, in order to represent the set $\varphi(\sigma\Delta) \subseteq L$, it suffices to represent each individual component $\varphi(\sigma)^{L[p]} + \psi(\Delta^p) \subseteq L[p]$ as the solution space of a cyclic linear equation system over $\mathbb{Z}_d$ where $d$ is a $p$-power. In what follows we use the cyclic constraints $C_j^i$, which we defined above, to show that this is possible.

Naturally, to define an appropriate equation system we aim to use the (non-trivial) indexing components of vectors from $L[p]$ as variables. Thus, formally, we define the set of variables as $W[p] \coloneqq \biguplus\{W_j^i : d_j^i \text{ is a } p\text{-power}\}$. Moreover, we let $d \coloneqq p^\ell = \max\{d_j^i : d_j^i \text{ is a } p\text{-power}\}$ denote the maximal $p$-power which occurs as the order of a group element $\delta_j^i \in \Gamma$. At this point we have to discuss another small technical difficulty. Of course, in general the vectors in $L[p]$ can have entries in different rings $\mathbb{Z}_{d_1}, \mathbb{Z}_{d_2}$ for $d_1 = p^{\ell_1} \neq p^{\ell_2} = d_2$, while we aim to define a cyclic linear equation system over the *single* ring $\mathbb{Z}_d$. However, by our choice of $d$ we know that for every such $d' = d_j^i = p^k$ we have $d' \mid d$. Hence we can use the embedding $\iota : \mathbb{Z}_{d'} \to \mathbb{Z}_d, z \mapsto (d/d') \cdot z$ to identify vectors in $\mathbb{Z}_{d'}^W$ with vectors in $\mathbb{Z}_d^W$. Of course, this embedding is not surjective and thus not all vectors in $\mathbb{Z}_d^W$ correspond to vectors in $\mathbb{Z}_{d'}^W$. Hence, we somehow have to

ensure that for solutions of our equation system we only allow such vectors from $\mathbb{Z}_d^W$ which are contained in $\mathrm{im}(\iota)(\mathbb{Z}_{d'}^W) \subseteq \mathbb{Z}_d^W$. Fortunately, this is very easy. The only thing we have to do is to add for each set $L_j^i = Z_{d'}^W$, which we lifted via the embedding $\iota : L_j^i \to \mathbb{Z}_d^W$, an auxiliary set of linear constraints $d' \cdot v = 0$ for all $v \in W$. Then precisely the vectors in $\mathrm{im}(\iota) \subseteq \mathbb{Z}_d^W$ satisfy these constraints. Of course, we can similarly lift the cyclic constraints $C_j^i$ on $L_j^i$ to corresponding cyclic constraints on $\mathrm{im}(\iota) \subseteq \mathbb{Z}_d^W$.

By now we have identified $L[p]$, via $\iota$, with a subspace of $\mathbb{Z}_d^{W[p]}$, and it only remains to be shown how we can represent $\iota(\varphi(\sigma)^{L[p]} + \psi(\Delta^p)) \subseteq \mathbb{Z}_d^{W[p]}$ as a cyclic linear equation system with variable set $W[p]$ over $\mathbb{Z}_d$. Recall from above, that we have already defined for every component $L_j^i$ a cyclic constraint $C_j^i$ on the set $W_j^i$. If we let $C[p]$ denote the collection of these constraints for all sets $W_j^i \subseteq W[p]$, then the set of $\mathbb{Z}_d^W$-vectors which satisfies these cyclic constraints (and, of course, the new auxiliary equations which we added above) is precisely $\iota(\varphi(\sigma)^{L[p]} + \psi(\Gamma^p))$. The question remains whether we can add an appropriate set of linear equations to represent $\iota(\varphi(\sigma)^{L[p]} + \psi(\Delta^p)) \subseteq \iota(\varphi(\sigma)^{L[p]} + \psi(\Gamma^p))$.

This question can be answered by taking into account the algebraic structure of the set $\varphi(\sigma)^{L[p]} + \psi(\Delta^p)$. To start, assume that for some set $W$ and some prime power $d = p^\ell$ we have a subgroup $\Delta \leq \mathbb{Z}_d^W$. For an appropriate index set $I$, let us consider a $W \times I$ matrix $A \in \mathbb{Z}_d^{W \times I}$ whose columns generate $\Delta$. Let us write $\langle A \rangle \leq \mathbb{Z}_d^W$ to denote the smallest subgroup of $\mathbb{Z}_d^W$ which contains all columns of $A$. By the choice of $A$ we have $\langle A \rangle = \Delta$. By exploiting the fact that divisibility is a preorder in $\mathbb{Z}_d$, we can find two invertible matrices $Q \in \mathbb{Z}_d^{W \times W}$ and $R \in \mathbb{Z}_d^{I \times I}$ such that $B \coloneqq Q \cdot A \cdot R$ is a diagonal matrix. Now for the diagonal matrix $B$ it is straightforward to find a $J \times W$ matrix $M_B$ such that the linear equation system $M_B \cdot \vec{x} = \vec{0}$ has $\langle B \rangle$ as its solution space. We claim that for $M_A \coloneqq M_B \cdot Q$, the linear equation system $M_A \cdot \vec{x} = \vec{0}$ has $\langle A \rangle = \Delta$ as its solution space. To verify this it suffices to check that $Q \cdot \langle A \rangle = \langle B \rangle$. Then for every $\vec{w} \in \mathbb{Z}_d^W$ we have that $M_A \cdot \vec{w} = \vec{0}$ if, and only if, $Q \cdot \vec{w} \in \langle B \rangle$ if, and only if, $\vec{w} \in \langle A \rangle$. Now, to capture the algebraic structure of the set $\varphi(\sigma)^{L[p]} + \psi(\Delta^p)$ we show how we can represent $\vec{w} + \Delta \subseteq \mathbb{Z}_d^W$ for some vector $\vec{w} \in \mathbb{Z}_d^W$ by a linear equation system with variables in $W$ over $\mathbb{Z}_d$. To this end, we first choose an appropriate $I \times W$-coefficient matrix $M$ such that $M \cdot \vec{x} = \vec{0}$ has the solution space $\Delta$ (this is possible as we saw before). Then we can take the linear equation system $M \cdot \vec{x} = M \cdot \vec{w}$ which has, as one can easily verify, $\vec{w} + \Delta$ as solution space. We conclude that for any prime $p \in P$ the set $\iota(\varphi(\sigma)^{L[p]} + \psi(\Delta^p))$ can be represented as a cyclic linear equation system $S_p$ over $\mathbb{Z}_d$ with variable set $W[p]$.

To sum up, we saw how we can represent any set $\sigma\Delta$ with $\Delta \leq \Gamma$ and $\sigma \in \pi\Gamma$ by a sequence of cyclic linear equation systems $(S_{p_1}, \ldots, S_{p_s})$. However, we have not discussed how a CPT-program can effectively define the appropriate cyclic linear equation systems which is, according to Definition 6.11, necessary at least for the basic sets $\mathrm{ext}(\sigma\Delta)$ where $\sigma\Delta \subseteq \mathcal{O}(A_{ij})$. Hence, let us finally go

through the requirements of Definition 6.11 to verify that our representation is indeed suitable in this sense.

(i) *Consistency.* To express whether $(S_{p_1}, \dots, S_{p_s})$ represents a non-empty set $\sigma\Delta$ we just have to check whether each single cyclic equation systems $S_p$ is consistent. This is possible in CPT by Theorem 5.12.

(ii) *Intersection.* Given two representations of sets $\sigma_1\Delta_1$ and $\sigma_2\Delta_2$ as sequences of CESs $(S_{p_1}, \dots, S_{p_s})$ and $(T_{p_1}, \dots, T_{p_s})$, we can represent $\sigma_1\Delta_1 \cap \sigma_2\Delta_2$ by the sequence $(S_{p_1} \cup T_{p_1}, \dots, S_{p_s} \cup T_{p_s})$ where $S_p \cup T_p$ denotes the CPT-definable cyclic linear equation systems which results by combining the sets of linear equations from $S_p$ and $T_p$.

(iii) *Representation of basic sets.* Given a set $\sigma\Delta \subseteq \mathcal{O}(A_{ij})$ for $0 \leq i \leq j < n$ and $\Delta \leq \Gamma_{ij}$ we want to define in CPT a representation of $\mathrm{ext}(\sigma\Delta) \subseteq \mathcal{O}(A)$. We saw above that, from an algebraic viewpoint, such a representation can be achieved.

To define a representation in Choiceless Polynomial Time, we first fix an auxiliary ordering $\rho \in \mathcal{O}(A_{ij})$ as a parameter. Then $\rho$ induces a linear order on the colour classes $A_i$ and $A_j$ and also (via $\varphi$) a linear order on all sets of variables $W^i_\ell, W^j_\ell$ which are relevant to define the cyclic linear equation systems for representing $\mathrm{ext}(\sigma\Delta)$. Having this, we can now simply follow the steps described above to obtain an appropriate representation as $(S_{p_1}, \dots, S_{p_s})$. Finally, the dependence on $\rho$ does not cause any problems, since for all $\rho$ we obtain linear systems which have the same solution spaces. Hence, we can just combine all systems into a single system without changing the represented set.

**Theorem 6.13.** *Choiceless Polynomial Time captures* PTIME *on the class $\mathcal{K}_{AC}$ of structures with Abelian colours.*

By Theorem 6.8 and the subsequent discussion we further obtain:

**Corollary 6.14.** *Choiceless Polynomial Time captures* PTIME *on every class of structures with* 2*-bounded colours.*

Our canonisation procedure generalises the CPT-definable isomorphism test for CFI-graphs of Dawar, Richerby, and Rossman [32]. Moreover the preceding corollary solves an open question of Blass, Gurevich, and Shelah [16, Question (5.12)]: the isomorphism problem for multipedes can be defined in Choiceless Polynomial Time. This follows from the simple observation that multipedes are structures with 2-bounded colours.

**Corollary 6.15.** *The isomorphism problem for multipedes can be defined in Choiceless Polynomial Time (see [16, 55]).*

## 6.3 Discussion

We introduced structures with Abelian colours and we discussed their importance in the quest for a logic capturing polynomial time. Furthermore, we pointed out connections to the well-studied notion of structures with bounded colour class size. In particular, we saw that many of the known queries which separate fixed-point logic with counting from polynomial time are based on structures with Abelian colours. On the other hand, in our main result of this chapter we proved that Choiceless Polynomial Time captures polynomial time on structures with Abelian colours. Hence, structures with Abelian colours form a very interesting class of structures which separates CPT from FPC.

The notion of Abelian colours is very well-motivated by the fact that it provides interesting insights into the importance of linear-algebraic techniques in descriptive complexity theory. On the other hand, from an algorithmic point of view, this notion does not seem very interesting as it imposes very specific and rather artificial structural properties on the input structures. In fact, there are many more natural classes of structures on which algorithmic techniques from computational (linear) algebra have been applied successfully in order to solve the isomorphism and the canonisation problem. One of the simplest among these classes, which also was the starting point of our studies, are structures of bounded colour class size. Hence, we think that the most interesting open question in this context is: Does CPT capture PTIME on classes with bounded colour class size? In fact, we saw that this holds for classes with 2-bounded colours (see Corollary 6.14), and we can show that this is true for classes with 3-bounded colours as well (this is part of ongoing investigations).

Generalising our techniques from Abelian colours to bounded colours may require to leave the area of *linear* algebra and to consider more general ideas from the field of computational group theory. At least the classical deterministic canonisation algorithm for structures with bounded colour class size is based on techniques to manipulate permutation groups. However, since we do not even know whether linear equation systems (with bounded colours) can be solved in Choiceless Polynomial Time, it seems hard to express such general techniques from computational algebra in CPT. More importantly, in our canonisation procedure we use an *implicit* representation of sets of isomorphisms, that is we represent these sets as the solution spaces of linear equation systems. In contrast, the algorithms for manipulating permutation groups make use of an *explicit* representation of groups by sets of generators. In particular, it is unclear whether our CPT-definable canonisation procedure can be formulated by using (a succinct encoding of) a set of generators to encode the sets of witnessing isomorphisms. Besides this, there is a nice result of Arvind, Kurur, and Vijayaraghavan [7] which puts the isomorphism problem of graphs with bounded colour class size in the #L-hierarchy (see for example [3] for background on this complexity class). We aim to explore how far their ideas can be transferred to Choiceless Polynomial Time.

Another interesting approach is to investigate whether at least all properties of structures with bounded colours which are definable in order-invariant (or successor-invariant) first-order logic can be expressed in Choiceless Polynomial Time. Note that there are properties of structures with bounded colours which can be expressed in order-invariant first-order logic but not in fixed-point logic with counting (the isomorphism problem for multipedes).

Another way to proceed is to generalise our setting from Theorem 6.8. There we saw that if we consider structures with bounded colours such that every colour class induces a substructure with an *Abelian* automorphism group, then we can solve the canonisation problem in Choiceless Polynomial Time (since such structures basically are structures with Abelian colours). This naturally leads to the idea of allowing more general groups which act as automorphism groups on the individual colour classes. For instance, we propose to study structures with bounded colours for which the colour classes induce automorphism groups which are *nilpotent* or, more generally, *solvable* (which are both well-studied concepts from algebra to generalise Abelian groups).

In particular, we studied classes of structures with bounded colours such that every colour class induces a substructure whose automorphism group is the dihedral group $D_k$ acting on $k$ letters (for some constant $k$). For $k = 3$ we obtain, as a special case, precisely the class of structures with 3-bounded colours. In this case we can show that our methods for canonising structures with Abelian colours can be adapted to obtain a canonisation procedure for this class of structures. Moreover, for the case $k = 4$ we have some promising preliminary results. However, this is part of ongoing research.

Finally, we remark that there are several other classes of structures for which efficient canonisation algorithms exist, but for which we don't have a *natural* logic which captures polynomial time. Most importantly, this is the case for graphs with bounded degree. In general, a good benchmark for these classes is the Cai, Fürer, Immerman query. In this thesis we saw that the CFI-query can be defined in Choiceless Polynomial Time. However, we strongly made use of the fact that the underlying graph was *ordered*. Actually it is open whether the CFI-query can be defined in Choiceless Polynomial Time also when we start from *unordered* cubic graphs (and we think it would be a nice result to prove this). Interestingly, for some graph classes, like complete graphs or graphs with bounded colours, one can show that the CFI-problem over such graphs can be defined in Choiceless Polynomial Time.

# Chapter 7

# Conclusion

The current frontier in the search for a logic capturing polynomial time is defined by problems from the field of computational algebra. While fixed-point logic with counting expresses a robust, natural, and rich fragment of polynomial time, it fails to capture important algorithmic techniques to handle large, algebraically-structured objects, which are specified in a succinct way. The main difficulty is to understand how such algorithmic methods, which are often based on computing non-canonical normal forms, can naturally be captured by logical mechanisms. As witnessed by the many new insights that were established during the last years, the solvability problem for linear equation systems over finite algebraic domains is a good starting point for further investigations. In this thesis, we extended our knowledge about this question, and we want to summarise our main contributions together with some important open questions, which can serve as subjects for (ongoing and) future research. For more details, we refer to our "Discussion" sections at the end of the corresponding chapters.

In Chapter 3, we studied the inter-definability of linear equation systems over Abelian groups, rings, modules. We saw that if these algebraic domains have a built-in linear order, then we can reduce, in fixed-point logic, linear equation systems over all domains to equivalent systems over cyclic groups of prime-power order. However, we left open whether a reduction to cyclic groups can also be achieved in the absence of an ordering, which would be nice, since we could then concentrate our studies completely on linear equation systems over cyclic groups (whose algebraic structure is very simple). Moreover, it also remained open whether we can go from cyclic groups of prime-power order further to cyclic groups of *prime* order. This is basically equivalent to the question of whether rank logic can define the solvability problem for linear equation systems over all rings $\mathbb{Z}_m$, where $m \geq 1$ is not necessarily prime. Another interesting aspect, which was raised during our investigations, concerns the definability of simpler problems over Abelian groups. For instance, can fixed-point logic (with counting) define summation over arbitrary sets in Abelian groups? Indeed, while much effort has been invested to understand

the expressive power of logics over various classes of graphs, almost nothing seems to be known for other important classes of structures, such as groups.

In Chapter 4, we studied extensions of fixed-point logic with counting by logical operators, specifically solvability quantifiers and rank operators, which can express the solvability of linear equation systems over finite fields. Our main result solves an open question of Dawar and Holm by showing that these operators over different prime fields cannot simulate each other. This also separated rank logic, in the original definition with a distinct rank operator for every prime, from polynomial time. In particular, a revised version of rank logic $\mathrm{FPR}^*$, with a uniform rank operator $\mathsf{rk}_*$, turns out to be strictly more powerful than the original version FPR with distinct operators $\mathsf{rk}_p$ for every prime $p \in \mathbb{P}$. We further saw that rank operators are strictly more powerful than solvability quantifiers in the absence of counting.

Of course, the main open question is whether the revised version of rank logic $\mathrm{FPR}^*$ suffices to capture polynomial time. There is no reason to believe that this is the case, and, in particular, it remains open whether $\mathrm{FPR}^*$ can express the solvability problem for linear equation systems over *all* Abelian groups. A good starting point would be to answer the following simplified version of this question first: can $\mathrm{FPR}^*$ distinguish between generalised Cai, Fürer, Immerman structures over *all* cyclic groups, for example, over $\mathbb{Z}_4$.

In Chapter 5, we introduced the notion of cyclic linear equation systems. Recall that for such systems, the set of variables is almost completely ordered, up to classes in which all pairs of variables are related via a given set of linear equations. Although such systems are structurally quite simple, they generalise, especially, the isomorphism problem for Cai, Fürer, Immerman graphs. Our main result shows that Choiceless Polynomial Time can express the solvability of cyclic linear equation systems. This yields a new family of queries to separate fixed-point logic with counting from Choiceless Polynomial Time. Of course, the most important open question is, whether Choiceless Polynomial Time can express the solvability of general linear equation systems over finite Abelian groups. Most likely, answering this question would also give new insights about the relation of rank logic and Choiceless Polynomial Time. Our definability result for cyclic linear equation may serve as a starting point to systematically investigate this question by considering linear equation systems with built-in auxiliary structure. For instance, one could study linear equation systems whose variables are linearly ordered up to classes, such that either these classes are of (constantly) bounded size, or such that, fixing the value of (constantly) many variables in each class determines the value of all remaining variables (this is a direct generalisation of cyclic linear systems for which we only need to fix the value of a single variable in each class).

In Chapter 6, we established a CPT-definable canonisation procedure for structures with Abelian colours. One of the central ingredients for this procedure are cyclic linear equation systems which are used to succinctly encode large sets of isomorphisms between the input structure and its (partially)

canonised copy. We saw that structures with Abelian colours appear frequently in finite model theory, most importantly, in connection with constructions which resemble the one of Cai, Fürer, and Immerman. We further found interesting connections to the well-studied notion of structures with bounded colour class size. For example, it follows from our results that Choiceless Polynomial Time captures polynomial time on structures with colour class size two. This also solved an open question of Blass, Gurevich, and Shelah: the isomorphism problem for multipedes is CPT-definable (since multipedes are structures of colour class size two). In fact, our original starting point was the question of whether Choiceless Polynomial Time captures polynomial time on classes of structures with bounded colour class size, and we still believe that it would be very interesting to establish a CPT-definable canonisation procedure on such classes. Again, our notion of Abelian colours may serve as a starting point to guide the search for such a canonisation procedure. Recall that a structure with bounded colours in which every colour class induces a substructure with an Abelian automorphism group has Abelian colours. This gives rise to natural generalisations: what happens if the induced automorphisms groups are nilpotent or solvable?

# List of Figures

# Bibliography


[1] F. Abu Zaid, E. Grädel, M. Grohe, and W. Pakusa. Choiceless Polynomial Time on structures with small Abelian colour classes. In *Mathematical Foundations of Computer Science 2014*, pages 50–62. Springer, 2014.

[2] F. Abu Zaid, E. Grädel, Ł. Kaiser, and W. Pakusa. Model-theoretic properties of omega-automatic structures. *Theory of Computing Systems, Special Issue dedicated to STACS 2012*, 2013.

[3] E. Allender and M. Ogihara. Relationships among PL, #L, and the determinant. *Informatique théorique et Applications*, 30(1):1–21, 1996.

[4] M. Anderson and A. Dawar. On symmetric circuits and fixed-point logics. In *31st International Symposium on Theoretical Aspects of Computer Science*, page 41, 2014.

[5] M. Anderson, A. Dawar, and B. Holm. Maximum matching and linear programming in fixed-point logic with counting. In *LICS 2013*, pages 173–182, 2013.

[6] V. Arvind, B. Das, J. Köbler, and S. Toda. Colored hypergraph isomorphism is fixed parameter tractable. *Algorithmica*, 71(1):120–138, 2009.

[7] V. Arvind, P. Kurur, and T. Vijayaraghavan. Bounded color multiplicity graph isomorphism is in the #L hierarchy. *Electronic Colloquium on Computational Complexity (ECCC)*, 2004.

[8] V. Arvind and T. Vijayaraghavan. The complexity of solving linear equations over a finite ring. In *STACS 2015*, pages 472–484. Springer, 2005.

[9] V. Arvind and T. Vijayaraghavan. Classifying problems on linear congruences and abelian permutation groups using logspace counting classes. *Computational Complexity*, 19(1):57–98, 2010.

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

[11] A. Atserias and E. Maneva. Sherali–adams relaxations and indistinguishability in counting logics. *SIAM Journal on Computing*, 42(1):112–137, 2013.

[12] L. Babai. Monte-Carlo algorithms in graph isomorphism testing. *Université de Montréal Technical Report, DMS*, pages 79–10, 1979.

[13] C. Berkholz and M. Grohe. Limitations of algebraic approaches to graph isomorphism testing. *arXiv preprint arXiv:1502.05912*, 2015.

[14] G. Bini and F. Flamini. *Finite Commutative Rings and Their Applications*. Kluwer Academic Publishers, 2002.

[15] A. Blass, Y. Gurevich, and S. Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100(1):141–187, 1999.

[16] A. Blass, Y. Gurevich, and S. Shelah. On polynomial time computation over unordered structures. *Journal of Symbolic Logic*, 67(3):1093–1125, 2002.

[17] A. Blass, Y. Gurevich, and J. Van den Bussche. Abstract state machines and computationally complete query languages. In *Abstract State Machines-Theory and Applications*, pages 22–33. Springer, 2000.

[18] W. Brown. *Matrices over commutative rings*. M. Dekker, 1993.

[19] A. Bulatov and V. Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM Journal on Computing*, pages 16–27, 2006.

[20] G. Buntrock, U. Hertrampf, C. Damm, and C. Meinel. Structure and importance of logspace-mod-classes. *STACS '91*, pages 360–371, 1991.

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

[22] F. Canavoi, E. Grädel, S. Lessenich, and W. Pakusa. Defining winning strategies in fixed-point logic. In *Proceedings of 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2015.

[23] A. Chandra and D. Harel. Structure and complexity of relational queries. In *Foundations of Computer Science, 1980., 21st Annual Symposium on*, pages 333–347. IEEE, 1980.

[24] A. Dawar. On the descriptive complexity of linear algebra. In *Logic, Language, Information and Computation*, pages 17–25. Springer, 2008.

[25] A. Dawar. The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, pages 8–21, 2015.

[26] A. Dawar and E. Grädel. Properties of almost all graphs and generalized quantifiers. *Fundamenta Informaticae*, 98(4):351–372, 2010.

[27] A. Dawar, E. Grädel, B. Holm, E. Kopczynski, and W. Pakusa. Definability of linear equation systems over groups and rings. *Logical Methods in Computer Science*, 9(4), 2013.

[28] A. Dawar, M. Grohe, B. Holm, and B. Laubner. Logics with Rank Operators. In *LICS '09*, pages 113–122. IEEE Computer Society, 2009.

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

[30] A. Dawar and D. Richerby. A fixed-point logic with symmetric choice. In *Computer Science Logic*, pages 169–182. Springer, 2003.

[31] A. Dawar and D. Richerby. The power of counting logics on restricted classes of finite structures. In *Computer Science Logic 2007, 16th Annual Conference of the EACSL*, pages 84–98, 2007.

[32] 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.

[33] H. Ebbinghaus and J. Flum. *Finite model theory*. Springer Science, 2005.

[34] E. Grädel et. al. *Finite Model Theory and Its Applications*. Springer, 2007.

[35] R. Fagin. Generalised first-order spectra and polynomial time recognizable sets. In R. Karp, editor, *Complexity of Computation. SIAM-AMS Proceedings 7*, pages 43–73, 1974.

[36] M. Furst, J. Hopcroft, and E. Luks. Polynomial-time algorithms for permutation groups. In *Foundations of Computer Science, 1980., 21st Annual Symposium on*, pages 36–41. IEEE, 1980.

[37] M. Furst, J. E. Hopcroft, and E. Luks. A subexponential algorithm for trivalent graph isomorphism. Technical report, Cornell University, 1980.

[38] F. Gire and H. Hoang. An extension of fixpoint logic with a symmetry-based choice construct. *Information and Computation*, 144(1):40–65, 1998.

[39] M. Goldmann and A. Russell. The complexity of solving equations over finite groups. *Information and Computation*, 178:253–262, 2002.

[40] E. Grädel. Finite model theory and descriptive complexity. In *Finite model theory and its applications*, pages 125–230. Springer, 2007.

[41] E. Grädel. Back and forth between logics and games. In *Lectures in Game Theory for Computer Scientists*, pages 99–145. Springer, 2011.

[42] E. Grädel, Ł. Kaiser, W. Pakusa, and S. Schalthöfer. Characterising Choiceless Polynomial Time with First-Order Interpretations. In *Proceedings of 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2015.

[43] E. Grädel and W. Pakusa. Rank logic is dead, long live rank logic! *CoRR*, abs/1503.05423, 2015.

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

[45] M. Grohe. Fixed-point logics on planar graphs. In *Logic in Computer Science, 1998 (LICS'98)*, pages 6–15. IEEE, 1998.

[46] M. Grohe. Definable tree decompositions. In *Logic in Computer Science, 2008, (LICS'08)*, pages 406–417. IEEE, 2008.

[47] M. Grohe. The quest for a logic capturing PTIME. In *Logic in Computer Science, 2008, (LICS'08)*, pages 267–271. IEEE, 2008.

[48] M. Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *Journal of the ACM (JACM)*, 59(5):27, 2012.

[49] M. Grohe. Descriptive complexity, canonisation, and definable graph structure theory. `http://www.automata.rwth-aachen.de/~grohe/cap/all.pdf`, 2015.

[50] M. Grohe, B. Grußien, A. Hernich, and B. Laubner. L-recursion and a new logic for logarithmic space. *Logical Methods in Computer Science*, 2012.

[51] M. Grohe and J. Mariño. Definability and descriptive complexity on databases of bounded tree-width. In *Database Theory—ICDT'99*, pages 70–82. Springer, 1999.

[52] M. Grohe and M. Otto. Pebble games and linear equations. *The Journal of Symbolic Logic*, 80:797–844, 2015.

[53] 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.

[54] Y. Gurevich and S. Shelah. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.

[55] Y. Gurevich and S. Shelah. On finite rigid structures. *The Journal of Symbolic Logic*, 61(02):549–562, 1996.

[56] M. Hall. *The theory of groups*. American Mathematical Soc., 1976.

[57] L. Hella. Logical hierarchies in PTIME. *Information and Computation*, 129(1):1–19, 1996.

[58] U. Hertrampf, S. Reith, and H. Vollmer. A note on closure properties of logspace mod classes. *Information Processing Letters*, 75(3):91–93, 2000.

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

[60] D. Holt, B. Eick, and E. O'Brien. *Handbook of computational group theory*. CRC Press, 2005.

[61] N. Immerman. Relational queries computable in polynomial time. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 147–152. ACM, 1982.

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

[63] N. Immerman. Expressibility as a complexity measure: results and directions. In *Second Structure in Complexity Conference*, pages 194–202. Springer, 1987.

[64] N. Immerman. Languages that capture complexity classes. *SIAM Journal on Computing*, 16:760–778, 1987.

[65] N. Immerman. *Descriptive complexity*. Springer Science, 2012.

[66] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. *Complexity Theory Retrospective*, pages 59 – 81, 1990.

[67] R. Kannan and A. Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979.

[68] J. Köbler. On graph isomorphism for restricted graph classes. In *Logical Approaches to Computational Barriers*, pages 241–256. Springer, 2006.

[69] S. Kreutzer. Expressive equivalence of least and inflationary fixed point logic. In *Proceedings of 17th IEEE Symp. on Logic in Computer Science LICS02*, pages 403–410, 2002.

[70] S. Kreutzer. *Pure and Applied Fixed Point Logic*. PhD thesis, RWTH Aachen University, 2002.

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

[72] L. Libkin. *Elements of finite model theory.* Springer Science, 2013.

[73] P. Lindström. First order predicate logic with generalized quantifiers. *Theoria*, 32(3):186–195, 1966.

[74] D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Fixed-parameter tractable canonization and isomorphism test for graphs of bounded treewidth. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 186–195. IEEE, 2014.

[75] E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982.

[76] E. M. Luks. Hypergraph isomorphism and structural equivalence of boolean functions. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, pages 652–658. ACM, 1999.

[77] B. McDonald. *Linear algebra over commutative rings.* M. Dekker, 1984.

[78] P. McKenzie and S. Cook. The parallel complexity of abelian permutation group problems. *SIAM Journal on Computing*, 16(5):880–909, 1987.

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

[80] W. Pakusa. *Finite Model Theory with Operators from Linear Algebra.* Staatsexamensarbeit, RWTH Aachen University, 2010.

[81] C. Papadimitriou. *Computational complexity.* Addison-Wesley, 1995.

[82] D. Richerby. *Fixed-Point Logics with Choice.* PhD thesis, University of Cambridge, 2003.

[83] B. Rossman. Choiceless computation and symmetry. In *Fields of Logic and Computation*, Lecture Notes in Computer Science, pages 565–580. Springer Berlin Heidelberg, 2010.

[84] S. Schalthöfer. *Computing on Abstract Structures with Logical Interpretations.* Master thesis, RWTH Aachen University, 2013.

[85] Á. Seress. *Permutation group algorithms.* Cambridge University Press, 2003.

[86] J. Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004.

[87] M. Vardi. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on Theory of Computing*, pages 137–146. ACM, 1982.

[88] M. Voit. Logical Canonisation of Structures with Transitive Abelian Colours with Linear-Algebraic Operators. Bachelor thesis, RWTH Aachen University, 2015.