Algorithmic Solutions via Model Theoretic Interpretations

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 Informatiker
Faried Abu Zaid, geb. Schreven
aus Viersen

Berichter: Universitätsprofessor Dr. Erich Grädel
Universitätsprofessor Dr. Dietrich Kuske

Tag der mündlichen Prüfung: 23. September 2016

Abstract

Model theoretic interpretations are an important tool in algorithmic model theory. Their applications range from reductions between logical theories to the construction of algorithms for problems, which are hard in general but efficiently solvable on restricted classes of structures, like 3-Colorability on graphs of bounded treewidth. We investigate this tool in three different areas of Algorithmic Model Theory:

1. automata-based decision procedures for logical theories,
2. algorithmic meta-theorems, and
3. descriptive complexity.

One of the main focus points of this dissertation are automata based presentations of infinite objects, which are closely related to monadic second-order interpretations over set variables. We introduce automatic presentations with advice for several automata models. These are presentations where the automata have access to some fixed auxiliary information. We develop algebraic and combinatorial tools, which enable us to prove that certain structures cannot have an ω-automatic presentation with advice. Our main result is that the field of reals is not ω-automatic with any advice, which has been an open problem since the introduction of ω-automatic presentations. The result can also be understood as an answer to a weakened version of a question posed by Rabin, namely whether the field of reals is interpretable in the infinite binary tree.

Further, we consider uniformly automatic classes of structures, which are classes generated by a fixed presentation and a set of advices. Prototypic examples are the class of all finite graphs of treewidth bounded by some constant, the torsion-free abelian groups of rank 1, and the class of all countable linear orders. Uniformly automatic presentations are also found in the mechanics that build the foundation for several algorithmic meta-theorems. We investigate the efficiency of this approach by analysing the runtime of the generic automata-based model checking algorithm in terms of the complexity of the given presentation. We show that the runtime on a presentation of the direct product closure is only one exponential higher than the runtime on the presentation of the primal class. We apply these findings to show that first-order model checking is fixed parameter tractable on the classes of all finite Boolean algebras and the class of all finite abelian groups. In both cases the parameter dependence of the runtime is elementary. The runtime which we achieve on these classes is either provably optimal or outperforms the previously known approaches. Furthermore, we show that the runtime of the generic automata based algorithm for monadic second-order model checking on graphs of treedepth at most h has a $(h+2)$-fold exponential parameter dependence.

This matches the runtime of the best known algorithms for model checking on these classes.

In the last part of this dissertation we turn our attention to logics with a build-in interpretation mechanism. Polynomial time interpretation logic (PIL) is an alternative characterisation of choiceless polynomial time (CPT). CPT is currently considered the most promising candidate for a logic capturing PTIME. We contribute to the exploration of the expressive power of CPT by showing that there is a CPT-definable canonisation procedure on classes of structures with bounded abelian colours. A structure has bounded abelian colours if it is of bounded colour class size and the automorphism group on every colour class is abelian. Examples emerge from the classical examples that separate fixed point logic with Counting from PTIME. The CFI-construction of Cai, Fürer, and Immerman, as well as the Multipedes of Blass, Gurevich, and Shelah have bounded abelian colours. Consequently, the isomorphism problem on these classes is solvable in CPT. For Multipedes this was an open question. In fact, Blass, Gurevich, and Shelah conjectured that the isomorphism problem for Multipedes might not be solvable by a CPT procedure.

Zusammenfassung

Modelltheoretische Interpretationen gehören zu den unverzichtbaren Werkzeugen der algorithmischen Modelltheorie. Klassische Anwendungen sind beispielsweise Reduktionen zwischen logischen Theorien oder die Verwendung zur Konstruktion von Algorithmen für im allgemeinen schwierige Probleme, die auf eingeschränkten Strukturklassen jedoch effizient lösbar sind, wie etwa 3-Färbbarkeit auf Graphen beschränkter Baumweite. Wir untersuchen dieses Konzept und seine Anwendungen in drei verschiedenen Bereichen der algorithmischen Modelltheorie:

1. Automatenbasierte Entscheidungsverfahren,
2. algorithmische Metatheoreme und
3. deskriptive Komplexität.

Automaten basierte Darstellungen unendlicher Objekte, welche Interpretationen der monadischen Logik zweiter Stufe in Mengenvariablen entsprechen, ist ein besonderes Augenmerk gewidmet.

Wir führen parametrisierte automatische Präsentationen ein, welche gewöhnliche automatische Darstellungen dadurch erweitern, dass die Automaten Zugriff auf eine fixierte zusätzliche Hilfseingabe haben. Wir entwickeln algebraische sowie kombinatorische Methoden zur Analyse automatischer Präsentationen. Wir wenden diese an, um zu beweisen, dass bestimmte Strukturen keine automatische Darstellung besitzen. Das Hauptergebnis in dieser Hinsicht ist, dass der Körper der reellen Zahlen keine solche parametrisierte ω-automatische Darstellung zulässt. Dies war seit der Einführung ω-automatischer Präsentationen durch Blumensath und Grädel ein offenes Problem. Das Ergebnis kann auch als die Antwort auf eine abgeschwächte Version einer Frage von Rabin verstanden werden, nämlich ob der Körper der reellen Zahlen in unendlichen Binärbaum Mengen-interpretierbar ist.

Wir beschäftigen uns zudem mit uniform darstellbaren Klassen. Dies sind Strukturklassen, die durch eine feste parametrisierte automatische Präsentation darstellen lassen, indem man Parameter aus einer festgelegten Menge betrachtet. Solche uniforme automatische Präsentationen bilden implizit den Kern vieler algorithmischer Metatheoreme. Wir interessieren uns für die Effizienz dieses Ansatzes und analysieren die Laufzeit des generischen Algorithmus für das Model Checking Problem von uniform baumautomatischen Strukturen in Abhängigkeit zu der Komplexität der gegebenen Präsentation. Wir wenden unsere Ergebnisse an, um zu zeigen, dass FO Model Checking FPT auf der Klasse der endlichen booleschen Algebren und der endlichen abelschen Gruppen ist. In beiden Fällen ist Laufzeit elementar im Parameter. Die erhaltenen Laufzeiten sind entweder beweisbar optimal oder verbessern die bisher bekannten oberen Schranken. Zusätzlich beweisen wir, dass die Laufzeit für

den generischen FPT Algorithmus für MSO Model Checking auf Graphen mit Baumtiefe höchstens h nur $(h+2)$-fach exponentiell im Parameter ist. Diese Laufzeit entspricht der Laufzeit der zur Zeit besten bekannten Algorithmen für dieses Problem.

Wir betrachten Polynomial Time Interpretation Logic (PIL). Diese wurde als eine alternative Charakterisierung der Logik Choiceless Polynomial Time (CPT) eingeführt, welche zur Zeit der erfolgversprechendste Kandidat für eine Logik ist, die PTIME einfangen könnte. Wir tragen zum Verständnis der Ausdrucksstärke von CPT bei, indem wir eine CPT definierbare Kanonisierungsprozedur für Strukturen mit beschränkt großen abelschen Farbklassen angeben. Eine Struktur hat beschränkt große abelsche Farbklassen, wenn sie nur beschränkt große Farbklassen besitzt und die Automorphismengruppen auf den Farbklassen abelsch sind. Beispiele für solche Strukturklassen erwachsen vor allen Dingen aus den bekannten Beispielen, welche Fixpunktlogik mit Zählen von PTIME trennen. So haben etwa die CFI-Graphen (im wesentlichen) beschränkte abelsche Farbeklassen. Dies gilt auch für die Multipedes von Blass, Gurevich und Shelah. Dementsprechend sind die entsprechenden Isomorphieprobleme im CPT lösbar. Für Multipedes war dies eine offene Frage.

Contents

1 Introduction

Motivation

When I think about what drew me to theoretical computer science in the first place, I remember the awe that I felt as a child when I was first confronted with mathematical thoughts that go beyond simple calculations. I wondered how we could be sure that a statement, say about numbers, is true. After all, the object under consideration is infinite and infinity itself is something that we can only vaguely imagine. It seemed to me that mathematicians must have access to some spheres of infinity to which entrance is denied for a normal mortal being.

As my education progressed, I got acquainted to the axiomatic approach and I learned that the insights that amazed me as a child are actually possible because we condense infinity into a finitely presentable collection of elementary properties which serve as a starting point for the journey on the vehicle of logical deduction.

The fact that we can investigate the process of strict logical deduction itself in mathematical terms is not a natural course of action. It is rather the result of a thousands of years lasting progress. The beginnings can be traced back to ancient philosophers and mathematicians, like Aristotle [67] and Euclid [62], who first studied the laws of thought and applied the axiomatic approach. The torch was passed on to people like Leibniz [52], Boole [17], and De Morgan [31] who developed an algebraic theory of logic. All culminated in the works of Frege [45] and Russel and Whitehead [100] who took the formal approach in its strictness to a new level with the aim to make logic the foundation of every mathematical truth. Finally Gödel [50,51] clarified what can and what cannot be done in these formal systems, which also illuminated the deep interconnections between logic and computability, just to mention a few cornerstones (while unjustly omitting many others).

But there is another approach of deepening the understanding of an object, which is so elementary that every child is familiar with it. Simply put, we can understand an object A by describing it in terms of another object B that we

already understand quite well. However, this process is by no means contrary to the process of gaining knowledge by logical deduction, but rather goes hand in hand with it.

This technique has incarnations in almost every field of science. For instance, in computer science we find this concept in form of reductions, where one transforms an instance of a certain algorithmic problem A to an instance of another problem B. Given that we know how to solve B, we also know how to solve A. But probably the most natural formal realisation is (and in my understanding must be) found in mathematical logic, more precisely in the field of model theory. Here it is known as *model-theoretic interpretation*. Roughly speaking, objects are modelled as structures, and the descriptions are formulated in logic. In its simplest version, an interpretation of a structure $\mathfrak{A} = (A, R_1, \dots, R_n)$ in another structure $\mathfrak{B}$ is a tuple of formulae $(\varphi_A(x), \varphi_{R_1}(\bar{x}_1), \dots, \varphi_{R_n}(\bar{x}_n))$ in the vocabulary of $\mathfrak{B}$, such that the evaluation of these formulae in $\mathfrak{B}$ defines a copy of $\mathfrak{A}$, that is $(\varphi_A^{\mathfrak{B}}, \varphi_{R_1}^{\mathfrak{B}}, \dots, \varphi_{R_n}^{\mathfrak{B}}) \cong \mathfrak{A}$.

While this can be seen as a very abstract formalisation of the general concept, it can also be seen as a special case of the reductions from computer science. Indeed, the existence of an interpretation yields an effective reduction between the theories of $\mathfrak{A}$ and $\mathfrak{B}$. Hence, model-theoretic interpretations an an essential interface between logic and computer science. The most classical application is the transfer of decidability and undecidability results, as described above.

Some of the most influential decidability results, both in computer science and model theory, are the decidability of the monadic second-order (MSO) theory of the orders on the natural numbers $\mathfrak{N} = (\mathbb{N}, \leq)$ and the infinite binary tree $\mathfrak{T}^\omega = (\{0,1\}^*, S_0, S_1)$. Even more influential than the mere statements are their proofs, due to Büchi [10] and Rabin [87], which establish a direct correspondence between MSO formulae and finite state automata. In their seminal papers they already apply these findings to reduce the first-order (FO) theory of structures like $(\mathbb{N}, +)$, $(\mathbb{R}, +)$, and $(\mathbb{N}, \cdot)$ to the MSO theory of these two very simple structures. This is done via a very powerful kind of interpretation, which is today known as *set-interpretation*. In contrast to usual interpretations, set-interpretations use formulae in which the free variables range over sets. This way the elements of the interpreted structure are presented by higher order objects of the original structure. Although implicitly in use for a long time, a systematic study of set-interpretations has not been carried out until a few years ago (see [25]).

In finite model theory, especially in descriptive complexity, interpretations

have been used under the term of logical reducibility to identify complete problems for certain complexity classes and to investigate the expressiveness of logics like fixed point logic (FP). For an overview see [57,66].

In the course of this thesis we examine applications of model-theoretic interpretations (often implicitly through equivalent characterisations) in three different areas of algorithmic model theory:

1. automata-based decision procedures for logical theories,
2. algorithmic meta-theorems, and
3. descriptive complexity.

Outline

In the following we describe the structure and content of this dissertation.

Advice Automatic Structures If one takes a closer look at the standard examples of structures that are set-interpretable in $\mathfrak{N}$ and $\mathfrak{T}^\omega$, it is striking that the list contains already a large part of the prominent examples of structures with decidable first-order theory. This gives way to the obvious question if this is a general phenomenon. In other words, is every structure with decidable FO theory also set-interpretable in a structure with decidable MSO theory? Unfortunately, up to this date we seem to be far from being able to answer these kind of questions. Indeed, in order to prove or disprove such a statement, we probably need to be able to tell whether a given structure $\mathfrak{A}$ is interpretable in a given structure $\mathfrak{B}$. But as a matter of fact, it is already challenging to prove that a structure is not set-interpretable in $\mathfrak{N}$, let alone in $\mathfrak{T}^\omega$.

However, in the case where we consider $\mathfrak{N}$ or $\mathfrak{T}^\omega$ as underlying structures, we have a powerful methodology at hand. Instead of working directly with the formulae, we can go one step further along the line of the standard decision procedure for $\mathfrak{N}$ or $\mathfrak{T}^\omega$ and work with automata that correspond to these formulae. This leads to the notion of *automatic structures*. Roughly speaking, a structure is automatic if it consists of a regular domain, such that all relations of the structure are recognizable by synchronous multi-tape automata. Working with this notion has the advantage that the combinatorics of the problem become more visible. In fact, automatic structures are a very active

field of research on their own. In their present form they were introduced by Khoussainov and Nerode [72] for word-automatic structures and later generalised to infinite words and trees by Blumensath and Grädel [14], leading to ω-automatic, tree-automatic, and ω-tree-automatic structures, respectively. A noteworthy predecessor are automatic groups [37].

Besides $\mathfrak{N}$, there are several other linear orders for which MSO has a corresponding automaton model. In recent years an active line of research was to incorporate these models into the family of automata-based presentations (see for instance [41, 69, 70, 95]) and to compare the classes of structures that are generated by them. One may hope that the investigation of interpretations via an increasing number of combinatorial characterisations will sharpen the view towards the general case.

We also aim at enriching the class of structures on which we apply the set-interpretations while maintaining a useful correspondence to automata-based presentations, although we proceed on a path that is in a certain sense orthogonal to the one described above. More precisely, we consider the expansions of $\mathfrak{N}$ and $\mathfrak{T}^\omega$ by unary predicates, say $P_1, \ldots, P_n$. From the automata-theoretic point of view, the sets $P_1, \ldots, P_n$ can be seen as a fixed advice that the automaton reads while it processes the input. This model of automata with advice is known in the literature as referenced automata [8]. The motivating example is the additive group of rational numbers $\mathfrak{Q}$. For a long time it has been on open problem whether $\mathfrak{Q}$ has a word-automatic presentation until it was finally answered negatively by Tsankov [98]. However, it was noted that $\mathfrak{Q}$ has an automatic presentation where the automata have additional access to a fixed infinite advice string. Moreover, this advice string has a decidable MSO theory. Subsequently, Kruckman, Rubin, Sheridan, and Zax [76] suggested the consideration of automatic presentations with advice.

The goal of Chapter 3 is to enrich our understanding of ω-automatic presentations with advice. While in the classical setting without advice, automatic presentations over finite words and trees are reasonably well understood (see for instance [93] for a survey of the finite string case), automatic presentations over infinite words and infinite trees received a lot less attention. Although there are essays that treat several aspects of ω-automatic presentations, see for instance [9, 63, 79], there have been very few works that focus on the development of techniques to show the non-existence of an ω-automatic presentation for a given structure. One reason for this is certainly that the analysis becomes more complex if automata over infinite objects are considered. Notable excep-

tions are the result by Kuske given in [77] that the ω-automatic ordinals are exactly the ordinals below ω^ω and the result of Finkel and Todorcevic given in [40] which shows that the injectively ω-tree-automatic ordinals are exactly the ordinals below ω^{ω^ω}.

One of the most prominent and important structures with a decidable first-order theory is certainly the field of reals $(\mathbb{R}, +, \cdot)$. The decidability goes back to Tarski [97] and is based on a quantifier elimination argument. Therefore, it is very natural to ask whether the field of reals admits an automatic presentation. Indeed, Rabin asked already in his seminal paper [87] whether we can solve the theory of certain fields by automata-theoretic methods. We show that this is not possible using automata over infinite words with advice, i.e., that the field of reals is not ω-automatic with advice. Additionally we show that no infinite integral domain is ω-automatic. Also the question whether the field of reals is ω-automatic has also been explicitly posed in [63]. Of course our result does not completely settle the question of Rabin because it is still possible that the field of reals has an ω-tree-automatic presentation.

The proof employs algebraic methods via the characterisation of ω-regular languages by ω-semigroups. We show that for every presentation of an uncountable structure $\mathfrak{A}$ there is an uncountable substructure $\mathfrak{B}$ that induces a subpresentation where all ω-semigroups that recognise the relations of $\mathfrak{A}$ have a in some sense nice algebraic structure on $\mathfrak{B}$. This, in turn, allows for an combinatorial analysis which involves an inspection of the behaviour of automatic relations on sets of words that pairwise differ only on finitely many positions.

Furthermore, we use our techniques to lift several known non-automaticity results to the advice setting.

Uniformly Automatic Classes Chapter 4 discusses another interesting twist which comes with the introduction of an advice. Instead of a single structure that is presented using a fixed advice we consider classes which are generated by a fixed automatic presentation and an advice set. We call a class of structures which can be presented in this way *uniformly automatic*. The motivating example here is a generalisation of the advice-automatic presentation of $\mathfrak{Q}$: using the same presentation as for $\mathfrak{Q}$, one can present all torsion-free abelian groups of rank 1 simply by altering the advice. Moreover, the class of all torsion-free abelian groups of rank 1 is generated by an ω-regular advice set.

We observe that this concept forms a natural framework in which many well-

known automata-based approaches to certain algorithmic problems in finite and algorithmic model theory can be expressed uniformly. These applications span from classical results like the decidability of the MSO theory of all countable linear orders [87] to the algorithmic meta-theorems for MSO-definable properties of graphs with bounded treewidth and bounded cliquewidth by Courcelle, Makowsky, and Rotics [26, 27].

We examine basic algorithmic applications with respect to the complexity of the advice set. An interesting special case of uniformly automatic presentations are those for which every isomorphism type of the class is presented via exactly one advice. We call this property the *unique presentation property*. In this case the decidability of FO with modulo and cardinality counting quantifiers for automatic structures can be used to count the models of a formula inside the respective class. We prove, however, that this property is already Π_1^0-complete for classes of finite sets and Σ_1^1-hard over signatures with at least one binary predicate.

Moreover, we investigate which closure operators on classes preserve automaticity. It is known that the class of (ω-)[tree-]automatic structures is closed under the basic composition operators of disjoint union and direct product [14]. For (ω-)tree-automatic structures it can easily be seen that a standard construction is uniform in the sense that for every uniformly (ω-)tree-automatic class the closure under disjoint union and direct product is also uniformly (ω-)tree-automatic. For the string case the picture becomes more diverse. The closure under disjoint unions is uniformly automatic for every uniformly automatic class. For classes of finite structures, uniform automaticity is also preserved when closing the class under direct products. In general this is not the case: we show that the free abelian groups of finite rank and the class of all finite direct products of the interval algebra on ω are not uniformly ω-automatic. Note that both are the closure under direct products of uniformly automatic classes. It remains open whether the closure under disjoint unions preserves automaticity for uniformly ω-automatic classes.

Classes of Finite Structures In Chapter 5 we consider the application of uniform tree-automaticity in finite model theory. More precisely, we investigate the use of automata in algorithmic meta-theorems. Algorithmic meta-theorems are general algorithmic results stating that a class of problems $\mathcal{P}$ can be efficiently solved on a class of instances $\mathcal{C}$. In many cases $\mathcal{P}$ is the class

of problems definable in a certain logic $\mathcal{L}$. *Parameterised complexity theory* provides one of the key notions to establish algorithmic meta-theorems: we say that the model checking problem for a logic $\mathcal{L}$ on a class of structures $\mathcal{C}$ is fixed-parameter tractable (FPT) (in the size of the formula) if the is a computable function f and a constant c such that we can decide for every $\varphi \in \mathcal{L}$ and every $\mathfrak{A} \in \mathcal{C}$ in time $f(|\varphi|) \cdot |\mathfrak{A}|^c$ whether $\mathfrak{A} \models \varphi$.

Prototypical examples of automata-based algorithmic meta-theorems are the theorem of Courcelle [26] for MSO-definable problems on graphs of bounded treewidth and the result of Courcelle, Makowsky, and Rotics [27] for MSO-definable problems on graphs of bounded cliquewidth. The basic idea is in both cases to compute from a graph $\mathfrak{G}$ a tree-like decomposition $t_{\mathfrak{G}}$ and from an MSO-formula φ a tree-automaton $\mathcal{A}_\varphi$ that accepts exactly the tree-like decompositions of graphs that model φ. Since the construction of $t_{\mathfrak{G}}$ from $\mathfrak{G}$ can be performed efficiently, we can efficiently check if $\mathfrak{G} \models \varphi$ by checking if $\mathcal{A}_\varphi$ accepts $t_{\mathfrak{G}}$. Note that many NP-complete problems, such as 3-Colourability, are definable in MSO and hence efficiently solvable on the above mentioned classes.

Interestingly, the presentations which build the core of the FPT algorithms for bounded treewidth and bounded cliquewidth graphs are obtained from MSO-interpretations on trees. Uniformly automatic presentations, however, correspond to set-interpretations, which are strictly more powerful than MSO-interpretations. In fact, it is not hard to construct even uniformly word-automatic classes of graphs which have unbounded tree- and cliquewidth. The power to present more complex classes of structures comes with a trade-off: We cannot hope that MSO model checking is FPT on every uniformly automatic class of structures. Instead we have to restrict our consideration to FO model checking. While every FO-definable problem can obviously be solved in polynomial time, we can still derive some very interesting consequences. For instance, if FO model checking is FPT on a class of graphs $\mathcal{C}$ then Independent Set is FPT on $\mathcal{C}$ in the size of the independent set because we can compute for every $k \in \mathbb{N}$ an FO-formula φ_k such that for every graph $\mathfrak{G}$ it holds that $\mathfrak{G} \models \varphi_k$ if and only if $\mathfrak{G}$ contains an independent set of size k.

Meta-theorems for first-order logic have been studied extensively on classes of sparse graphs. The first result in this direction is due to Seese for graphs of bounded degree [96]. Over the past decades larger and larger classes of sparse graphs have been identified for which FO model checking is FPT. This development has recently found its climax in the result of Grohe, Kreutzer,

and Siebertz for nowhere dense graphs [60]. They proved that under certain complexity theoretic assumptions this is the largest possible subgraph-closed class of graphs where FO model checking is FPT.

We investigate automaticity as a generic notion of simplicity which might bring up new and interesting classes of structures for which FO model checking is FPT. Towards the theory, we are concerned with the efficiency of this approach. Note that in general the non-elementary worst-case runtime of the automaton construction process leads to a non-elementary parameter dependence in the algorithmic meta-theorems. Frick and Grohe [47] showed, unless PTIME = NP, there is no algorithm that solves the model checking problem for MSO on words or trees in time

$$f(|\varphi|) \cdot \mathrm{poly}(|t|)$$

for any elementary function $f\colon \mathbb{N} \to \mathbb{N}$. A similar statement holds for FO on words. As trees have treewidth one, this renders Courcelle's approach to model checking of graphs with bounded treewidth optimal. Moreover, the efficiency of the automata theoretic approach has also been confirmed in practice. For instance, Langer et al. [81] implemented Courcelle's technique and found that their implementation can compete with other approaches for specific problems such as Dominating Set.

Even more interestingly, the automata-based approach also tends to behave tamely when applied to interpretations of structures whose theory is elementary. Eisinger [36] gave a triply-exponential upper bound on the size of the minimal automaton for formulae of integer and mixed-real addition. In [33] Durand-Gasselin and Habermehl showed for word-automatic structures that the runtime of the generic algorithm can be bounded by a function which estimates how well the presentation goes along with the Ehrenfeucht-Fraïssé relations of the structure and gave runtime bounds for integer addition matching Eisinger's bound. Additionally they gave a triply-exponential bound for automatic graphs of bounded degree complementing a result by Kuske and Lohrey who proved, using a specialised algorithm, that model checking for automatic graphs of bounded degree is solvable in doubly-exponential space [80].

We adopt Durand-Gasselin's and Habermehl's technique and generalise their result to uniformly tree-automatic presentations. We apply this technique to the presentations of the direct product closures that we described earlier. We prove that the bound of the runtime of the model checking algorithm is at

most exponential in the bound of the runtime for the primal classes. Further we apply these findings in the context of FPT model checking for first order logic. We demonstrate the efficiency of the automata-theoretic approach by analysing the runtime in terms of the parameter dependence on structurally rather simple classes. Our results are as follows:

- FO model checking is FPT on the class of all finite Boolean algebras that are succinctly encoded by the number of atoms and can be performed in

  $$\exp_2(\text{poly}(|\varphi|)) \cdot \log |\mathfrak{B}|.$$

  Unless NEXP $= \bigcup_{c\in\mathbb{N}} \text{STA}(*, 2^{cn}, n)$, this parameter dependence is optimal.

- FO model checking is FPT on the class of all finite abelian groups that are succinctly encoded by the orders of the direct product factors and can be performed in

  $$\exp_4(\text{poly}(|\varphi|)) \cdot \log |\mathfrak{G}|.$$

  We generalise this result to finite groups of *bounded non-abelian decomposition width*, that is groups whose non-abelian direct product factors are of bounded size. We obtain the same asymptotic runtime on these classes.

  This provides some first results towards Grohe's question on which classes of algebraic structures FO model checking is FPT [58]. The mere FPT result for FO model checking on abelian groups was independently also discovered by Bova and Martin [18]. Their algorithm assumes that the groups are encoded by their multiplication tables and yields a non-elementary parameter dependence. Therefore our approach has the two advantages that it works for succinct encodings and yields an elementary parameter dependence.

- MSO model checking is FPT on every class of graphs with tree-depth at most h and can be performed in

  $$\exp_{h+2}(\text{poly}(|\varphi|)) \cdot \text{poly}(|\mathfrak{G}|).$$

  This matches the runtime of the best known algorithm for these classes, which is due to Gajarsky and Hliněný [49]. Their algorithm uses a kernelisation procedure. Our proof makes use of their analysis.

Extending Fixed-Point Logic By Interpretations In the last part of this dissertation, we take a different view towards interpretations. Namely, we consider interpretations as the description of the behaviour of an abstract machine. Roughly speaking, we think of a structure as a representation of a machine state and the interpretation to describe the transitions from one state to another. On the one hand, this describes a model of computation where the interpretation is repeatedly applied. On the other hand this also describes a logic. Therefore it is natural to investigate this idea in the context of descriptive complexity. The logic that substantiates the above idea is called interpretation logic (IL). It was suggested by Kaiser and formally introduced by Schalthöfer [94]. We investigate the polynomial-time fragment of interpretation logic (PIL) as a candidate logic for capturing PTIME.

Descriptive complexity theory is the branch of finite model theory that tries to characterise complexity classes by the type of logic needed to express exactly the languages inside this complexity class. The first result in this field is Fagins Theorem, which states that the properties of finite structures that are recognisable in non-deterministic polynomial time are exactly the properties expressible in existential second-order logic [38]. Below NP the situation is more delicate. The most interesting question, especially in the light of the PTIME versus NP problem, is whether there is a logic capturing PTIME. If we assume that a linear order is present, then fixed-point logic (FP) captures exactly the PTIME properties of finite structures. This was discovered independently by Immerman [65] and Vardi [99]. Without the presence of a linear order, however, FP is relatively far from capturing PTIME. For instance, FP cannot even express simple properties that involve counting: the class EVEN $= \{(A) \mid |A| \text{ is even}\}$ is not definable in FP. Therefore Immerman suggested to extend FP with a mechanism to count. This leads to fixed-point logic with counting (FPC), which was formally introduced by Grädel and Otto in [55].

Cai, Fürer, and Immerman [22] could prove, however, that FPC still does not capture polynomial time. Their example is basically a tractable instance of the graph isomorphism problem on a special class of graphs, called CFI-graphs. The constructions of the CFI-graphs is tuned so that the resulting graphs have bounded degree and bounded colour class size. This in turn means that FPC fails to capture PTIME on both of these classes. CFI-graphs have been useful way beyond the mere application in [22]. In fact they are now a standard tool in finite model theory to show upper bounds on the expressiveness of logics.

However, the CFI-query is not really a natural problem because the CFI-graphs emerge from a rather artificial construction. Atserias, Bulatov and Dawar noted that there is a very natural problem underlying the CFI-query [6]. They showed that via interpretations the CFI-query can be reduced to the solvability of linear equation systems over a finite field, which implies that FPC cannot express solvability of these systems.

Another quite prominent query known to be not expressible in FPC is the isomorphism problem for the multipedes introduced by Blass, Gurevich, and Shelah [12,61]. In the light of the previous observations it is an interesting fact that the polynomial time algorithm, which decides the isomorphism problem for multipedes, also reduces the problem to the solvability of a linear equation system.

In spite of these shortcomings in expressiveness, FPC still captures a large and natural fragment of PTIME. Therefore, FPC is today an important benchmark for every candidate logic for capturing PTIME. Every such logic should be strictly more expressive than FPC. *Choiceless Polynomial Time* (CPT) is currently the most prominent such candidate. The rough idea about it is that it operates on the hereditary finite expansion of an abstract input structure to allow everything that a normal program could do, for instance to use counting, complex data structures, and parallelism. The only exception is that must not perform any operation that would break symmetries, such as arbitrarily picking an element of the input structure. In other words, the operations that a CPT-program can use are defined such that every automorphism of the input structure extends to an automorphism of every "state of the machine". In order to remain in PTIME one artificially bounds the resources of the machine by a polynomial. Although the formal definition of CPT is quite technical, it is still a natural model of automorphism-invariant computations. Moreover, it turns out that CPT has exactly the same expressiveness as PIL [94], which further underlines that these logics capture a natural fragment of PTIME. Also CPT is indeed more expressive than FPC: while it is easy to see that every FPC-formula can be transformed into an equivalent CPT-program, Dawar, Richerby, and Rossman showed that the CFI-query is definable by a CPT-program [30].

Structures of Bounded Colour Class Size We contribute to the exploration of the expressive power of CPT with respect to several of the shortcomings of FPC. Our main contribution is a CPT-definable canonisation procedure for

structures of bounded colour class size where the automorphism groups of the colour classes are abelian. The strategy is as follows: We iteratively canonise the structure along the linear order on the colour classes. This is done by storing all isomorphisms between the already processed part of the structure and the ordered copy of the processed part. Suppose we have already canonised the substructure that is induced by the first n colour classes. When we add the $(n+1)$-st colour class we search for the lexicographically smallest extension of the ordered copy such that some of the stored isomorphisms extend to an isomorphism between the first $n+1$ colour classes and the newly obtained ordered structure. This is done by iteratively considering the connections to all already processed colours.

One of the main obstacles here is that we cannot store the set of isomorphisms directly without violating the polynomial restrictions. Therefore we show that in the case where the automorphism groups on the colour classes are abelian, we can represent the set of isomorphisms succinctly by a system of linear equations with a total preorder on the variables and cyclic constraints on the bags of the preorder. Checking that the canonisation of a new part of the structure is consistent to the canonisation of the previous parts reduces to a solvability check of a linear equation system over a finite ring.

Our main technical contribution is to show that the linear equation systems that arise in this procedure can be solved in CPT. We implement a version of Gaussian elimination where symmetric choices are succinctly encoded by objects, which we call hyperterms.

An important special case are structures of colour class size two. In this case the automorphism group of every colour class is trivially abelian. Hence, our algorithm yields a CPT-definable canonisation procedure for structures of colour class size two. As multipedes are structures of colour class size two, the multipedes query is definable in CPT, which refutes a conjecture of Blass, Gurevich and Shelah [12].

Acknowledgements I would like to thank my advisor Professor Erich Grädel for the possibility to pursue my research interests under his guiding. I would also like to thank my co-referee and new supervisor Professor Dietrich Kuske. I am grateful for their support and patience.

The results, which form the basis of this thesis would not have been possible without my collaborators. Chapter 3 is based on joint work with Erich Grädel, Lukas Kaiser, and Wied Pakusa [3,4]. Chapter 4 is based on still unpublished work with Erich Grädel and Frederic Reinhardt. Chapters 6 and 7 are based on joint work with Erich Grädel, Martin Grohe, and Wied Pakusa [1].

I also wish to thank all of my colleagues at Mathematische Grundlagen der Informatik and Informatik 7. You all made my time at RWTH Aachen University as enjoyable as one could not dare to wish for his workplace. Special thanks need to go to Simon and Wied. I will not say more than that it was an honour to be part of this noble society. ▲▲

Finally, I would like to thank everyone else who has accompanied me through the ups and downs of this journey.

2 Preliminaries

Throughout this work I will assume that the reader posses a certain background knowledge in logic and theoretical computer science. This chapter is therefore not intended to give a comprehensive overview of all concepts needed to understand this work. Rather, this chapter is used to recall some key ideas and to fix notation. For more information on the subjects the reader is directed to appropriate literature in the respective sections.

2.1 Logic and Structures

2.1.1 Basic Notations

A **signature** τ is a set of relation and function symbols together with a function ar that assigns an arity to every symbol. We usually denote relation symbols with (possibly indexed) capital letters like R_1 and function symbols with lower case letters like f_1. As a convention we denote the arity of a relation symbol R_i by r_i if the context prohibits any danger of confusion with a function symbol. For a relation $R \subseteq A_1 \times \cdots \times A_n$, $k < n$, and $\overline{a} \in A_1 \times \cdots \times A_k$ we write $\overline{a}R$ for the set $\{\overline{b} \in A_{k+1} \times \cdots \times A_n \mid (\overline{a}, \overline{b}) \in R\}$ and if $X \subseteq A_1 \times \cdots \times A_k$ we write XR for $\bigcup_{\overline{a} \in X} \overline{a}R$. Similarly we define for $\overline{b} \in A_{k+1} \times \cdots \times A_n$ the set $R\overline{b} = \{\overline{a} \in A_1 \times \cdots \times A_k \mid (\overline{a}, \overline{b}) \in R\}$ and $RY = \bigcup_{\overline{b} \in Y} R\overline{b}$ for $Y \subseteq A_{k+1} \times \cdots \times A_n$.

A **τ-structure** is a tuple $\mathfrak{A} = (A, (R^{\mathfrak{A}})_{R \in \tau}, (f^{\mathfrak{A}})_{f \in \tau})$ where A is a set, called the **universe** of $\mathfrak{A}$, $R^{\mathfrak{A}} \subseteq A^r$ is a r-ary relation over A, and $f^{\mathfrak{A}} : A^{\mathrm{ar}(f)} \to A$ is an $\mathrm{ar}(f)$-ary function over A. By convention, we denote the universe of a structure $\mathfrak{A}$ by A, the universe of a structure $\mathfrak{B}$ by B, and so on. The class of all τ-structures is denoted by $\mathrm{Str}(\tau)$ and the class of all finite τ-structures is denoted by $\mathrm{finStr}(\tau)$.

For a signature τ, **First-Order Logic** over τ (FO[τ]) is build up from

- a denumerable set of individual variables,
- the symbols of τ,

- the boolean connectives $\wedge, \vee, \neg, \ldots$,
- the quantifiers $\exists$ and $\forall$, and
- the bracket symbols (,).

The atomic formulae are of the form $R(\overline{x})$ or $f(\overline{x}) = y$, where R is a relation symbol, f is a function symbol, y is an individual variable, and $\overline{x}$ is a tuple of individual variables matching the arity of R and f, respectively. Formulae can be connected with the boolean connectives and variables can be bound by the quantifiers $Qx : \varphi$ with $Q \in \{\exists, \forall\}$. We will speak simply of first-order logic (FO) whenever the signature is clear from the context or whenever the specific signature does not matter for the consideration. The **free variables** of a formula are the variables that are not bound by a quantifier. We write $\varphi(x_1, \ldots, x_n)$ to denote that all free variables of φ are among $x_1, \ldots, x_n$. A **sentence** of first-order logic is a formula without free variables. The **quantifier rank** $\mathrm{qr}(\varphi)$ is the maximal nesting depth of quantifiers inside φ. The semantics of first-order logic is defined as usual and we will skip the formal definition here. For more information we refer the reader to [34]. We write $(\mathfrak{A}, a_1, \ldots, a_n) \models \varphi(x_1, \ldots, x_n)$ (or in short form $\mathfrak{A} \models \varphi(a_1, \ldots, a_n)$) to express that the structure $\mathfrak{A}$ satisfies the formula φ if the free variable x_i is interpreted by a_i for all $1 \leq i \leq n$. We write $\varphi(x_1, \ldots, x_n)^{\mathfrak{A}} = \{\bar{a} \in A^n \mid \mathfrak{A} \models \varphi(\bar{a})\}$ for the relation that is defined by φ in $\mathfrak{A}$. We use the shorthand notation

$$
\begin{aligned}
&(\exists x_1 \exists x_2 \ldots \exists x_n).\delta(x_1, \ldots, x_n) : \varphi(x_1, \ldots, x_n) \text{ and} \\
&(\forall x_1 \forall x_2 \ldots \forall x_n).\delta(x_1, \ldots, x_n) : \varphi(x_1, \ldots, x_n) \text{ for} \\
&\exists x_1 \exists x_2 \ldots \exists x_n : (\delta(x_1, \ldots, x_n) \wedge \varphi(x_1, \ldots, x_n)) \text{ and} \\
&\forall x_1 \forall x_2 \ldots \forall x_n : (\delta(x_1, \ldots, x_n) \rightarrow \varphi(x_1, \ldots, x_n)),
\end{aligned}
$$

respectively, to emphasize that the quantification is relativised to tuples satisfying $\delta(x_1, \ldots, x_n)$. We say two τ structures $\mathfrak{A}$ and $\mathfrak{B}$ are **elementary equivalent** ($\mathfrak{A} \equiv \mathfrak{B}$) if no sentence of first-order logic can distinguish between $\mathfrak{A}$ and $\mathfrak{B}$, that is $\mathfrak{A} \models \varphi \Leftrightarrow \mathfrak{B} \models \varphi$ for all sentences $\varphi \in \mathrm{FO}[\tau]$. We write $\mathfrak{A} \equiv_k \mathfrak{B}$ if $\mathfrak{A} \models \varphi \Leftrightarrow \mathfrak{B} \models \varphi$ holds for all sentences $\varphi \in \mathrm{FO}[\tau]$ of quantifier rank at most k.

Monadic second-order logic (MSO) augments first-order logic by the possibility to quantify over monadic relation variables. In order to make a notational distinction we use lower case letters $x, y, z, \ldots$ for individual variables

and capitals $X, Y, Z, \ldots$ for relation variables. The notations introduced in the previous paragraph are adopted for MSO accordingly.

Fixed Point Logic (with Counting) (FP(C)) augments first-order logic by a fixed point operator $[\mathrm{fp}R\overline{x} : \varphi(R, \overline{x}, \overline{y})]$, with free variables $\bar{y}$. For every structure $\mathfrak{A}$ and every interpretation $\bar{a}$ of the free variables $[\mathrm{fp}R\overline{x} : \varphi(R, \overline{x}, \overline{y})]$ defines a relation where the arity r matches the arity of $\bar{x}$ and it is defined as the limit of the sequence $(R_i)_{i \in |A|^+}$ given by

$$\begin{aligned} R_0 &= \emptyset \\ R_{\alpha+1} &= R_\alpha \cup \varphi(R_\alpha, \bar{x}, \bar{a})^{\mathfrak{A}} \\ R_\gamma &= \bigcup_{\alpha<\gamma} R_\alpha \text{ for limit ordinals } \gamma. \end{aligned}$$

Fixed point logic with counting augments FP by operators $\#_x \varphi(x, \bar{y})$, where x is bound by the operator $\#_x$. We define the semantics of FPC only on the class of all finite structures. In order to evaluate the counting terms, a formula of FPC is evaluated in a two sorted expansion of a finite structure $\mathfrak{A}$. More precisely, for a finite structure $\mathfrak{A} = (A, R_1, \ldots, R_n, f, \ldots, f_k)$ one considers the two-sorted expansion $(A, \{0, \ldots, |A|\}, R_1, \ldots, R_k, f_1, \ldots, f_k, <)$, where $<$ is the natural linear order on $\{0, \ldots, |A|\}$. Then $\#_x \varphi(x, \bar{y})$ (where x and $\bar{y}$ might contain variables that range over the first and the second sort) is a term which is evaluated to an element of the second sort by

$$(\#_x \varphi(x, \bar{y}))^{(\mathfrak{A}, \bar{b})} = |\varphi(x, \bar{b})^{\mathfrak{A}}|.$$

The Härtig quantifier H has the syntactic build rule $\mathrm{H}\bar{x}\bar{y}\varphi(\bar{x}, \bar{z})\psi(\bar{y}, \bar{z})$ where $\bar{x}, \bar{y}$, and $\bar{z}$ are disjoint tuples of variables. A structure $\mathfrak{A}$ together with an interpretation $\bar{a}$ of the variables in $\bar{z}$ is a model of $\mathrm{H}\bar{x}\bar{y}\varphi(\bar{x}, \bar{z})\psi(\bar{y}, \bar{z})$ if $|\varphi(\bar{x}, \bar{a})^{\mathfrak{A}}| = |\psi(\bar{y}, \bar{a})^{\mathfrak{A}}|$. The extension of first-order logic by the Härtig quantifier is denoted $\mathrm{FO} + \mathrm{H}$.

2.1.2 Model Theoretic Interpretations

Model theoretic interpretations are a well known and powerful tool of mathematical logic. The idea is to use formulas to define a copy of a structure inside another one. The existence of such an interpretation is particularly useful to

transfer definability, decidability, and complexity results from one structure to another.

While the basic structure of interpretations is rather invariant between the different flavours of interpretations that we consider, the objects that represent the elements of the interpreted structure may vary between different kinds of interpretations. We summarize here the sorts of interpretations that are used in this thesis.

Definition 2.1. Let L be a logic and τ, as well as σ, be signatures, where $\sigma = \{R_1, \dots R_n\}$ is finite and relational. An n-ary $\mathrm{L}[\tau, \sigma]$ **interpretation** (in first order variables) is a tuple $\mathcal{I} = (\delta, \varepsilon, \varphi_{R_1}, \dots, \varphi_{R_n})$, where δ has n free variables, ε has $2n$ free variables, and every φ_{R_i} has $r_i \cdot n$ free variables, for $i \in \{1, \dots, n\}$. If L, τ and σ are clear from the context we will just speak of an interpretation rather then of an $\mathrm{L}[\tau, \sigma]$ interpretation.

The tuple $\mathcal{I}$ interprets $\mathfrak{B}$ in $\mathfrak{A}$ (or $\mathcal{I}(\mathfrak{A}) = \mathfrak{B}$) if there exists a surjective mapping $h : \delta^{\mathfrak{A}} \to B$ such that

- for all $\overline{x}, \overline{y} \in \delta^{\mathfrak{A}}$, $h(\overline{x}) = h(\overline{y})$ if, and only if, $(\overline{x}, \overline{y}) \in \varepsilon^{\mathfrak{A}}$ and
- for all $i \in \{1, \dots, n\}$ and all $\overline{x}_1, \dots, \overline{x}_{r_i} \in \delta^{\mathfrak{A}}$, $(\overline{x}_1, \dots, \overline{x}_{r_i}) \in \varphi^{R_i}$ if, and only if, $(h(\overline{x}_1), \dots, h(\overline{x}_{r_i})) \in R_i^{\mathfrak{B}}$.

Analogously we obtain the notion of an $\mathrm{L}[\tau, \sigma]$ **interpretation in set variables** by replacing the term variable by set-variable in the above definition. In this case we distinguish between full and weak set semantic.

In this dissertation we consider the following instances of interpretations

- $\mathrm{FO}[\tau, \sigma]$ interpretations, where the formulas and free variables are first order,
- $\mathrm{MSO}[\tau, \sigma]$ interpretations in set variables (set-interpretations), where we use MSO formulas and free set variables, and
- $\mathrm{WMSO}[\tau, \sigma]$ interpretations in finite set variables, which are syntactically identical to $\mathrm{MSO}[\sigma, \tau]$ interpretations in set variables, but all occurring set variables range over finite sets.

Remark 2.1. By definition, the target signature of an interpretation is always relational. This is not a severe restriction since we can replace a function

$f : A^n \to A$ by its **graph** $R_f = \{(a_1, \ldots, a_n, f(a_1, \ldots, a_n)) \mid a_1, \ldots, a_n \in A\}$. Whenever we talk about interpreting a structure $\mathfrak{A}$ over a signature σ which contains functions symbols we implicitly mean that we interpret the relational version of $\mathfrak{A}$ where every function is replaced by its graph.

For $L \in \{\mathrm{FO}, \mathrm{MSO}\}$ every $\varphi \in \mathrm{FO}[\sigma]$, an $\mathrm{L}[\tau, \sigma]$ interpretation $\mathcal{I}$ induces a L-formula $\varphi^{\mathcal{I}} \in L[\tau]$ where every quantification Qx is translated to $Qx_1 \ldots Qx_n.\delta(\overline{x})$, equalities $x = y$ are replaced by $\varepsilon(\overline{x}, \overline{y})$, and $R(x, y, z, \ldots)$ is replaced by $\varphi_R(\overline{x}, \overline{y}, \overline{z}, \ldots)$. As a consequence First-order interpretations allow us to transfer the first order theory of structures.

Lemma 2.1 (Interpretation Lemma [54])**.** *For every* $\mathrm{FO}[\tau, \sigma]$ *interpretation* $\mathcal{I}$ *over a structure* $\mathfrak{A}$ *and every* FO*-sentence* $\varphi \in \mathrm{FO}[\sigma]$ *, we have that*

$$\mathfrak{A} \models \varphi^{\mathcal{I}} \Leftrightarrow \mathcal{I}(\mathfrak{A}) \models \varphi.$$

Similarly, For every $\psi \in \mathrm{FO}[\sigma]$, an $\mathrm{MSO}[\tau, \sigma]$ interpretation in set variables $\mathcal{I}$ induces a MSO-formula $\psi^{\mathcal{I}} \in L[\tau]$ where every quantification Qx is translated to $QX_1 \ldots QX_n.\delta(\overline{X})$, equalities $x = y$ are replaced by $\varepsilon(\overline{X}, \overline{Y})$, and $R(x, y, z, \ldots)$ is replaced by $\varphi_R(\overline{X}, \overline{X}, \overline{Z}, \ldots)$. Hence, an MSO interpretation $\mathcal{I}$ in set variables allows to reduce the first order theory of $\mathcal{I}(\mathfrak{A})$ to the MSO theory of $\mathfrak{A}$.

Lemma 2.2. *For every* MSO *(*WMSO*) interpretation in* $\mathcal{I}$ *in (finite) set variables over a structure* $\mathfrak{A}$ *and an* FO*-sentence* φ *it holds that*

$$\mathfrak{A} \models \varphi^{\mathcal{I}} \Leftrightarrow \mathcal{I}(\mathfrak{A}) \models \varphi.$$

Model theoretic interpretations form a nice formalism to describe structures by formulas. From a class of interpretations over a fixed structure $\mathfrak{A}$ one naturally obtains a class of finitely representable structures. Indeed, we can present (the isomorphism type of) a structure $\mathfrak{B}$ by an interpretation $\mathcal{I}$ with $\mathcal{I}(\mathfrak{A}) = \mathfrak{B}$.

Definition 2.2. For a structure $\mathfrak{A}$ the class $\mathrm{SI}(\mathfrak{A})$ ($\mathrm{FSI}(\mathfrak{A})$) is the class of all structures $\mathfrak{B}$ such that there is a MSO-interpretation (WMSO-interpretation) $\mathcal{I}$ over (finite) sets with $\mathcal{I}(\mathfrak{A}) = \mathfrak{B}$.

With Lemma 2.2 we get that the decidability of the first order theory of a structure in $\mathrm{SI}(\mathfrak{A})$ ($\mathrm{FSI}(\mathfrak{A})$) reduces to the decidability of the MSO-theory (WMSO-theory) of $\mathfrak{A}$.

Corollary 2.1. *Let $\mathfrak{A}$ be a structure with decidable* MSO*-theory (*WMSO*-theory). Then every structure in* SI($\mathfrak{A}$) *(*FSI($\mathfrak{A}$)*) has a decidable first order theory.*

So far we have fixed a formalism to describe structures by formulas that yields an effective decision procedure for the first-order model checking problem given that the MSO-theory of the underlying structure is decidable. Now we need to find good candidates for such underlying structure which are rich enough to encode interesting structures by interpretations. Probably the most prominent structures with decidable MSO-theory are the natural numbers with the successor relation $\mathfrak{N} = (\mathbb{N}, S)$ and the infinite binary tree with left and right successor relation $\mathfrak{T}^\omega = (\{0,1\}^*, S_0, S_1)$. The decidability is due to the seminal works of Büchi and Rabin [10,87]. Their results also establish the deep connection between (weak) monadic second-order logic and finite automata, which connects logic and theoretical computer science.

2.2 Automata and Formal Languages

Let Σ be an alphabet. A **word** over Σ is a function $w : \{0, \dots, n-1\} \to \Sigma$ for some $n \in \mathbb{N}$. In this case $|w| := n$ is the **length** of w. We will often use the notation $w = w_0 w_1 w_2 \dots w_{n-1}$, where $w_i = w(i)$ and for $0 \leq i < j \leq |w|$ we write $w[i,j) = w_i w_{i+1} \cdots w_{j-1}$. We denote the set of all finite words over Σ of length n by Σ^n, $\Sigma^{\leq n}$ for $\bigcup_{0 \leq i \leq n} \Sigma^i$, and Σ^* for the set of all finite words over Σ. The class of all words is naturally equipped with the concatenation product, which we write multiplicatively vw. Formally for $v \in \Sigma^n$ and $w \in \Sigma^m$ the word $vw \in \Sigma^{n+m}$ is defined as

$$(vw)(i) = \begin{cases} v(i) & \text{if } i < n \\ w(j) & \text{if } i = n + j. \end{cases}$$

We say a word $u \in \Sigma^*$ is a **prefix** of a word $w \in \Sigma^*$ if $w = uv$ for some $v \in \Sigma^*$. We denote the prefix relation by $u \preceq w$. For $w \in \Sigma^*$ and $a \in \Sigma$ we denote by $|w|_a$ number of occurrences of the letter a in w.

An **infinite word** is a function $\alpha : \mathbb{N} \to \Sigma$. We denote the set of all infinite words over Σ by Σ^ω. As it was the case for finite words, there is a natural concatenation operation between finite and infinite words. For $w \in \Sigma^*$ and

$\alpha \in \Sigma^\omega$ we define

$$(w\alpha)(i) = \begin{cases} w(i) & \text{if } i < |w| \\ \alpha(j) & \text{if } i = |w| + j. \end{cases}$$

2.2.1 Regular Languages

Regular languages play a crucial role in many areas of computer science. One reason for this is certainly that they form a very robust class of languages which is also algorithmically quite well manageable. In this section we will give a short overview over several formalisms that characterize regular languages on finite and infinite words, as well as finite or infinite trees.

Regular Expressions We introduce regular languages by means of regular expressions.

Definition 2.3 (Regular Expression)**.** Let Σ be a (usually) finite alphabet. The regular expressions over Σ are defined inductively by the following rules.

1. ε is a regular expression
2. Every letter $a \in \Sigma$ is a regular expression.
3. If α and β are regular expressions than so are α^*, $(\alpha + \beta)$, and $(\alpha\beta)$.

A regular expression is **ε-free** if it is build up using only the rules (2) and (3). With every regular expression α over Σ we associate a language $L(\alpha) \subseteq \Sigma^*$. the language $L(\alpha)$ is defined by

$$L(\alpha) = \begin{cases} \{a\} & \text{if } \alpha = a \in \Sigma \\ L(\beta) \cup L(\gamma) & \text{if } \alpha = (\beta + \gamma) \\ L(\beta)L(\gamma) & \text{if } \alpha = (\beta\gamma) \\ \{\varepsilon\} \cup \{v_1 v_2 \cdots v_n \mid n \geq 1, v_1, \ldots, v_n \in L(\beta)\} & \text{if } \alpha = \beta^* \end{cases}$$

For the sake of simplicity of notation, we will not distinguish between a regular expression and the language that it represents in the following. We call a language $L \subseteq \Sigma^*$ regular if, and only if it can be represented by a regular expression.

The class of ω-regular languages consists of languages $L \subseteq \Sigma^\omega$. This class of languages can be defined by a similar formalism.

Definition 2.4 (ω-Regular Expression)**.** An ω-regular expression if an expression of the form $\alpha_1(\beta_1)^\omega \cup \ldots \cup \alpha_n(\beta_n)^\omega$ for some $n \in \mathbb{N}$, where α_i is a regular expressions and β_i is an ε-free regular expression. The associated language $L(\alpha_1(\beta_1)^\omega \cup \ldots \cup \alpha_n(\beta_n)^\omega)$ is

$$\{w \in \Sigma^\omega \mid \exists i \in \{1, \ldots, n\}\, \exists (v_j)_{j \in \mathbb{N}} : \\ w = v_0 v_1 v_2 \ldots \wedge v_0 \in L(\alpha_i) \wedge v_j \in L(\beta_i) \text{ for } j > 0\}$$

A language $L \subseteq \Sigma^\omega$ is ω-regular if it can be represented by an ω-regular expression.

Automata Besides regular expressions, automata are probably the most common way to represent regular languages.

Definition 2.5. A finite automaton is a tuple $\mathcal{A} = (Q, q_0, \Sigma, \Delta, F)$ where Q is a finite set of states with initial state $q_0 \in Q$ and accepting states $F \subseteq Q$. The alphabet Σ is a finite set of letters and $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation. A run of $\mathcal{A}$ on a word $w = w_1 w_2 \ldots w_n \in \Sigma^*$ is a word $\rho = \rho_0 \rho_1 \ldots \rho_n \in Q^*$ such that

- the first letter ρ_0 is q_0,
- for all $0 \leq i < n$ we have $(\rho_i, w_{i+1}, \rho_{i+1}) \in \Delta$.

Such a run is called accepting if $\rho_n \in F$. The language accepted by $\mathcal{A}$, denoted $L(\mathcal{A})$, is the set of all $w \in \Sigma^*$ such that there exists an accepting run of $\mathcal{A}$ on w. In the case that $|\Delta \cap \{q\} \times \{a\} \times Q| = 1$ for all $(q, a) \in Q \times \Sigma$ we say that $\mathcal{A}$ is deterministic. In this case we interpret Δ as a function $\delta : Q \times \Sigma \to Q$. We extend δ in a natural way to $Q \times \Sigma^*$ by

$$\delta(q, \varepsilon) = q \text{ and} \\ \delta(q, wa) = \delta(\delta(q, w), a).$$

Thus we can characterise the language accepted by a deterministic automaton via $w \in L(\mathcal{A}) \Leftrightarrow \delta(q_0, w) \in F$.

Fact 2.1. *Let $\mathcal{A} = (Q, q_0, \Sigma, \Delta, F)$ be a finite automaton. Then the is a deterministic finite automaton $\mathcal{B}$ with at most $2^{|Q|}$ states such that $L(\mathcal{A}) = L(\mathcal{B})$*

Fact 2.2. *Let Σ be an alphabet. A language $L \subseteq \Sigma^*$ is regular if and only if $L = L(\mathcal{A})$ for some deterministic finite automaton.*

Definition 2.6. A deterministic Muller automaton (or ω-automaton) is a tuple $\mathcal{A} = (Q, q_0, \Sigma, \delta, \mathcal{F})$. The components Q, q_0, Σ, and δ are defined in the same way as for finite word automata. The acceptance is defined by the set $\mathcal{F} \subseteq \mathcal{P}(Q)$. For $w \in \Sigma^\omega$ and $m \in \mathbb{N}$ we write $\delta_m(q, w) := \delta(q, w[0, m)])$ for the state that is reached by $\mathcal{A}$ from q if the first m symbols of w are read. We further extend to function δ to infinite words by

$$\delta(q, w) = \{q' \mid |\{m \in \mathbb{N} \mid \delta_m(q, w) = q'\}| = \infty\}$$

for all $w \in \Sigma^\omega$. The language accepted by $\mathcal{A}$ is $\{w \in \Sigma^\omega \mid \delta(q_0, w) \in \mathcal{F}\}$.

Sometimes we allow for an even further condensed nation by defining $\delta(w) = \delta(q_0, w)$ for $w \in \Sigma^*$ and also $w \in \Sigma^\omega$.

Fact 2.3. *Let Σ be an alphabet. A language $L \subseteq \Sigma^\omega$ is ω-regular if and only if $L = L(\mathcal{A})$ for some deterministic Muller automaton $\mathcal{A}$.*

Semigroups Besides automata we will also use the characterization of regular languages via homomorphisms to finite semigroups. A **semigroup** is a structure $\mathfrak{S} = (S, \cdot)$, where $\cdot : S \times S \to S$ is associative. A **monoid** is a semigroup that contains a neutral element.

Example 2.1. The structure $(\Sigma^+, \cdot)$, where $\cdot$ is the concatenation product, is the free semigroup over the generators Σ and $(\Sigma^*, \cdot)$ is the free monoid over the generators Σ.

The following theorem gives a characterization of regular languages in terms of semigroup homomorphisms.

Theorem 2.1 ([88])**.** *A language $L \subseteq \Sigma^+$ is regular if, and only if, there exists a finite semigroup $\mathfrak{S} = (S, \cdot)$ and a homomorphism $f : \Sigma^* \to S$ such that $f^{-1}(f(L)) = L$ or equivalently there is a subset X of S with $f^{-1}(X) = L$.*

In the study of finite semigroups idempotence and absorption play an important role. An element e of a semigroup $(S, \cdot)$ is idempotent if $e \cdot e = e$, and e absorbs d from the left if $e \cdot d = e$. For every element s of a *finite* semigroup $(S, \cdot)$ there is a $k \in \mathbb{N}$ such that s^k is idempotent. The smallest number k such that s^k is idempotent for all elements $s \in S$ is called the exponent of the semigroup.

For ω-regular languages, there is an analogous characterisation theorem. In this setting, however, semigroups have to be replaced by ω-semigroups. Since the notion of ω-semigroups is not as widespread as the notion of semigroups, we give a short overview based on Perrin and Pin [86].

An **ω-semigroup** is a two-sorted structure $\mathfrak{S} = (S_f, S_\omega, \cdot, *, \pi)$ with the following properties.

- The structure $(S_f, \cdot)$ is a semigroup.
- The mixed product $* : S_f \times S_\omega \to S_\omega$ satisfies

  $$x * (y * z) = (x \cdot y) * z$$

  for all $x, y \in S_f$, $z \in S_\omega$.
- The infinite product $\pi : (S_f)^\omega \to S_\omega$ satisfies the equation

  $$x_0 * \pi(x_1, x_2, x_3, \ldots) = \pi(x_0, x_1, x_2, \ldots)$$

  for every sequence $(x_i)_{i \in \mathbb{N}}$ of elements from S_f.
- The function π is associative in the sense that for every strictly increasing sequence of positive integers $(k_i)_{i \in \mathbb{N}}$ it holds that

  $$\pi(x_1, x_2, x_3, \ldots) = \pi((x_i)_{[0,k_0)}, (x_i)_{[k_0,k_1)}, (x_i)_{[k_1,k_2)}, \ldots).$$

  Here we use the expression $(x_i)_{[n,m)}$ as abbreviation for $x_n \cdot x_{n+1} \cdot \cdots \cdot x_{m-1}$.

Because of the last two properties we can present a mixed product $x_0 * \pi(x_1, x_2, x_3, \ldots)$ without ambiguity as $x_0 x_1 x_2 \cdots$. For reasons of readability we sometimes denote with $(x_i)_{i \in \mathbb{N}}$ not only the sequence of elements, but also their infinite product $x_0 x_1 x_2 \cdots$.

Example 2.2. For an alphabet Σ, the free ω-semigroup over Σ is the structure $\Sigma^\infty = (\Sigma^+, \Sigma^\omega, \cdot, *, \pi)$, where $\cdot, *$ and π are interpreted as the usual concatenation products.

As mentioned above, we introduced ω-semgigroups to give an algebraic characterisation of ω-regular languages. Similar to the characterisation of regular languages, this will be done by homomorphisms from Σ^∞ to finite ω-semigroups. We state the definition of a homomorphism between ω-semigroups explicitly to avoid confusion because of the two sorted nature of ω-semigroups.

Definition 2.7. Let $\mathfrak{S}$ and $\mathfrak{T}$ be ω-semigroups. A homomorphism $g : \mathfrak{S} \to \mathfrak{T}$ between $\mathfrak{S}$ and $\mathfrak{T}$ is a pair $g = (g_f, g_\omega)$ such that

(1) the function $g_f : S_f \to T_f$ is a homomorphism from $(S_f, \cdot)$ to $(T_f, \cdot)$ and

(2) the function $g_\omega : S_\omega \to T_\omega$ preserves the mixed and the infinite product, i.e.

$$g_\omega(x_1x_2x_3\ldots) = g_f(x_1)g_f(x_2)g_f(x_3)\ldots$$

for every sequence $(x_i)_{i\in\mathbb{N}}$, $x_i \in S_f$ and

$$g_\omega(x * y) = g_f(x) * g_\omega(y)$$

for $x \in S_f$ and $y \in S_\omega$.

If we have given a homomorphism $g : \mathfrak{S} \to \mathfrak{T}$, we usually omit the subscripts of the mappings g_f and g_ω whenever this cannot lead to any confusion. The way we recognise languages by homomorphisms directly transfers from the finite word case.

Definition 2.8. Let $L \subseteq \Sigma^\omega$ be a language and $g : \Sigma^\infty \to S$ a morphism into some finite ω-semigroup S. We say that L is recognised by S via g if, and only if, $g^{-1}(g(L)) = L$ or, in other words, $g^{-1}(X) = L$ for some subset X of S_ω.

Let us take a short look on representation issues for ω-semigroups and homomorphisms before we state the main theorem of this paragraph.

First, because of property (2) of Definition 2.7, every ω-semigroup homomorphism $g = (g_f, g_\omega)$ is completely determined by the semigroup homomorphism g_f. Therefore we can represent every such homomorphism in a finite way. A bit more tricky is the situation for the ω-semigroups. Even in the case where S_f and S_ω are finite sets, the ω-semigroup $(S_f, S_\omega, \cdot, *, \pi)$ is not a finite object, since the domain of π is still uncountable for $|S_f| \geq 2$. But, as Wilke showed in [101], the class of all finite ω-semigroups is in one to one correspondence to a class of finite structures where all functions have finite arity. These structures are called Wilke-algebras.

Definition 2.9. A **Wilke-algebra** is a two sorted structure $(S_f, S_\omega, \cdot, *, {}^\omega)$ with the following properties.

(1) The structure $(S_f, \cdot)$ is a semigroup.

(2) The mixed product $* : S_f \times S_\omega \to S_\omega$ satisfies

$$x * (y * z) = (x \cdot y) * z.$$

for all $x, y \in S_f$, $z \in S_\omega$.

(3) The power operation ${}^\omega : S_f \to S_\omega$ fulfils the equations

$$x(yx)^\omega = (xy)^\omega \text{ and}$$
$$(x^n)^\omega = x^\omega \text{ for all } n \geq 1.$$

for all $x, y \in S_f$.

From a given ω-semigroup one naturally obtains a Wilke-algebra by restricting the infinite product π to the products of the form $\pi(a, a, a, \ldots)$ for $a \in S_f$. But also the converse is true.

Theorem 2.2 (Wilke [101])**.** *Every finite Wilke-algebra can be uniquely extended to a finite ω-semigroup.*

The key to prove this theorem is the theorem of Ramsey. We state it here since we will make use of it in Chapter 3.

Theorem 2.3 (Ramsey's Theorem [89])**.** *Let $G = (\mathbb{N}, \binom{\mathbb{N}}{2})$ be the complete countable undirected graph and $f : \binom{\mathbb{N}}{2} \to C$ a coloring of the edges with some finite set of colors C. Then there is an infinite set $N \subseteq \mathbb{N}$ such that every edge in $\binom{\mathbb{N}}{2} \cap (N \times N)$ has the same color.*

At last, we state the main theorem of this paragraph.

Theorem 2.4 (cf. [86])**.** *A language $L \subseteq \Sigma^\omega$ is ω-regular if, and only if, it is recognisable by a homomorphism into a finite ω-semigroup. Moreover, from a finite ω-semigroup $\mathfrak{S}$, given by its corresponding Wilke-algebra, and a homomorphism $g : \Sigma^\infty \to S$ that recognises the language L one can effectively compute a Muller-automaton that recognises L and vice versa.*

Regular Tree-Languages Besides words we will also consider Σ-labelled trees. A tree is a mapping $t : \mathrm{dom}_t \to \Sigma$ where the domain dom_t is a subset of $\mathbb{N}^*$ with two additional properties:

- for all $w \in \mathrm{dom}_t$, if $v \preceq w$ then also $v \in \mathrm{dom}_t$ and
- for all $w \in \mathbb{N}^*$ and $n, m \in \mathbb{N}$, if $wn \in \mathrm{dom}_t$ and $m < n$ then also $wm \in \mathrm{dom}_t$.

For $w \in \{0, \ldots, m\}^*$ the elements $w0, \ldots, wm$ are called the successors of w.

A tree is finite if its domain is finite. A tree t is finitely branching if for all $w \in \mathrm{dom}_t$ there exists a maximal $n \in \mathbb{N}$ with $wn \in \mathrm{dom}_t$. A tree t has bounded degree if $\mathrm{dom}_t \subseteq \{0, \ldots, d-1\}^*$ for some $d \in \mathbb{N}$ and in this case the **rank** or **maximal degree** of t is the minimal $d \in \mathbb{N}$ such that $\mathrm{dom}_t \subseteq \{0, \ldots, d-1\}^*$. A complete infinite tree of rank d is a mapping $t : \{0, \ldots, d-1\}^* \to \Sigma$. A **ranked alphabet** is an alphabet Σ with an associated rank function rk_Σ that maps every $a \in \Sigma$ to a finite set of natural numbers. We sometimes give rk_Σ implicitly by writing $\Sigma = \bigcup_{0 \le i \le m} \Sigma_i$, where $\Sigma_i = \{a \in \Sigma \mid i \in \mathrm{rk}_\Sigma(a)\}$. With $\mathrm{rk}(\Sigma)$ we denote the maximal rank among the ranks of the letters in Σ. A (finite, Σ-labelled) tree is a mapping $t : \mathrm{dom}_t \to \Sigma$, where dom_t is a tree-domain and if $w \in \mathrm{dom}_t$ has exactly k successors in dom_t then $t(w) \in \Sigma_k$. The set of all trees over Σ is denoted by T_Σ.

Example 2.3. Let Σ be an alphabet. The set Σ^* of all words over Σ can be identified with T_Σ with rank funktion $\mathrm{rk}(a) = \{0, 1\}$ for all $a \in \Sigma$.

For a tree $t : \mathrm{dom}_t \to \Sigma$, $w \in \mathrm{dom}_t$ and $a \in \Sigma$ with $\mathrm{rank}(a) = \mathrm{rank}(t(w))$ we define $t[w \to a]$ to be the tree with $t[w \to a](v) = t(v)$ for $v \in \mathrm{dom}(t) \setminus \{w\}$ and $t[w \to a](w) = a$. If t is a Σ-labelled tree and $v \in \mathrm{dom}_t$ then $(t \restriction v)$ denotes the tree with $\mathrm{dom}_{t \restriction v} = \{w \in \{0, \ldots, m\}^* \mid vw \in \mathrm{dom}(t)\}$ and $(t \restriction v)(w) = t(vw)$.

Definition 2.10. Let $\Sigma = \bigcup_{0 \le i \le m} \Sigma_i$ be a ranked alphabet. A (deterministic bottom up) tree-automaton is a tuple $\mathcal{A} = (Q, \Sigma, \delta, F)$, where Q is a finite set of states, $\delta : \left(\bigcup_{0 \le i \le m} Q^i \times \Sigma_i\right) \to Q$ is the transition function, and $F \subseteq Q$ is the set of accepting states.

For a tree $t \in T_\Sigma$ the unique run of $\mathcal{A}$ on t is the Q labelled tree $\rho : \mathrm{dom}_t \to Q$ with $\rho(v) = \delta(t(v))$ if v is a leaf of t and if v has as successors $v_0, \ldots, v_{r-1}$ in t then $\rho(v) = \delta(\rho(v_0), \ldots, \rho(v_{r-1}), t(v))$. The run ρ is accepting if $\rho(\varepsilon) \in F$. The language $L(\mathcal{A})$ is the set of all trees t such that the run of $\mathcal{A}$ on t is accepting.

The extended transition function $\delta^* : T_\Sigma \to Q$ is defined inductively by

$$\delta^*(t) = \begin{cases} \delta(t(\varepsilon)) & \text{if } \mathrm{dom}_t = \{\varepsilon\} \\ \delta(\delta^*(t \restriction 0), \ldots, \delta^*(t \restriction (k-1), t(\varepsilon)) & \text{the root of } t \text{ has } k \text{ successors.} \end{cases}$$

That is $\delta^*(t) = \rho(\varepsilon)$ for the unique run ρ of $\mathcal{A}$ on t.

Definition 2.11. A **Σ-context** (or just context if Σ is clear) is a $(\Sigma \uplus \{x\})$-labelled tree c such that all inner nodes are labelled by symbols from Σ and there is exactly one leaf that is labelled with x. We denote this leaf by $x(c)$. The class of all Σ-contexts is denoted by C_Σ. For a Σ-context c and a tree $t \in T_\Sigma$ the **composition** $c \circ t$ is a Σ-labelled tree with $\mathrm{dom}_{c \circ t} = \mathrm{dom}_c \cup \{x(c)\}\mathrm{dom}_t$ and labelling

$$(c \circ t)(w) = \begin{cases} c(w) & ; w \in \mathrm{dom}_c - \{x(c)\} \\ t(v) & ; w = x(c)v, v \in \mathrm{dom}_t \end{cases}$$

The Myhill-Nerode Theorem for tree-languages is build upon the following equivalence.

Definition 2.12. For a tree-language $T \subseteq T_\Sigma$ the relation $\equiv_T \subseteq T_\Sigma \times T_\Sigma$ is defined by

$$t \equiv_T t' \Leftrightarrow c \circ t \in T \text{ iff } c \circ t' \in T, \text{ for all } c \in C_\Sigma.$$

The relation $\equiv_T$ allows us to formulate an analogue to the Myhill-Nerode Theorem for regular tree-languages.

Theorem 2.5 (Brainerd, [19])**.** *For $T \subseteq T_\Sigma$ the following are equivalent:*

1. *the language T is regular,*
2. *there exists a finitely generated left congruence of finite index with respect to composition with contexts, which respects T, and*
3. *the relation $\equiv_T$ has finite index.*

In some situations, especially in Chapter 4, we will need a more general notion of context. More precisely we want to be able to plug several trees simultaneously into one schema.

Definition 2.13. Let Σ be a ranked alphabet. A n-Σ-context is a $\Sigma \uplus \{c_1, \dots, c_n\}$ labelled tree x such that all inner nodes are labelled by symbols from Σ and for every $i \in \{1, \dots, n\}$ there is exactly one c_i labelled leaf. We denote the c_i labelled leaf of x with $c_i(x)$.

For $t_1, \dots, t_n \in T_\Sigma$ we denote with $x[c_1/t_1, \dots, c_n/t_n] \in T_\Sigma$ the tree with domain $\mathrm{dom}_x \cup \bigcup_{1 \leq i \leq n} c_i(x)\mathrm{dom}_{t_i}$ and labelling

$$x[c_1/t_1, \dots, c_n/t_n](w) = \begin{cases} x(w) & w \in \mathrm{dom}_x \setminus \{c_1(x), \dots, c_n(x)\} \\ t_i(v) & w = c_i(x)v, v \in \mathrm{dom}_{t_i} \end{cases}$$

Convention. If we do not state explicitly otherwise we will always assume that for a ranked alphabet $\Sigma = \bigcup_{0 \leq i \leq m} \Sigma_i$ we have $\Sigma_0 = \Sigma_1 = \cdots = \Sigma_m$.

Automatic Relations Automata can also be used to recognise relations of higher arity. Actually the automata do not recognise the relation itself but the language of **convolutions** of the tuples in the relation. The convolution maps two words to a single word by putting the words "on top of each other". For the case that the two words have a different length a new padding symbol $\square$ is introduced and the shorter word is padded to the length of the other word. More precisely, for two words $v, w \in \Sigma^*$ the convolution $v \otimes w$ is a word over the alphabet $(\Sigma \uplus \{\square\})^2$ of length $\max(|v|, |w|)$ with

$$(v \otimes w)(i) = \begin{cases} (v(i), w(i) & \text{if } i < \min(|v|, |w|) \\ (\square, w(i)) & \text{if } |v| < i < |w| \\ (v(i), \square) & \text{if } |w| < i < |v|. \end{cases}$$

The convolution of two ω-words is defined analogously with the difference that we don't need to introduce a padding symbol.

The convolution of trees follows the same idea. Let Σ be a ranked alphabet. For two trees $s, t \in T_\Sigma$ the convolution $s \otimes t$ is the $(\Sigma \uplus \{\square\})^2$-labelled tree with $\mathrm{dom}_{s \otimes t} = \mathrm{dom}_s \cup \mathrm{dom}_t$ and the labelling

$$t(w) = \begin{cases} (s(w), t(w)) & \text{if } w \in \mathrm{dom}_s \cap \mathrm{dom}_t \\ (\square, t(w)) & \text{if } w \in \mathrm{dom}_t \setminus \mathrm{dom}_s \\ (s(w), \square) & \text{if } w \in \mathrm{dom}_s \setminus \mathrm{dom}_t. \end{cases}$$

Instead of $w_1 \otimes w_2 \otimes \cdots \otimes w_n$ we will often write $\langle w_1, w_2, \dots, w_n \rangle$ and for a language L we write $L^{\otimes n}$ for $\underbrace{L \otimes \cdots \otimes L}_{n \text{ times}} = \{\langle w_1, \dots, w_n \rangle \mid w_1, \dots, w_n \in L\}$.

Definition 2.14. Let $R \subseteq (\Sigma^*)^n$ ($(\Sigma^\omega)^n$, $(T_\Sigma)^n$, or $(T_\Sigma^\omega)^n$, respectively). The relation R is (ω-)[tree-]regular, if the language $\{x_1 \otimes x_2 \otimes \cdots \otimes x_n \mid (x_1, x_2, \ldots, x_n) \in R\}$ is (ω-)[tree-]regular.

2.2.2 Automatic Structures

One of the most important notions throughout this thesis are automatic presentations. The rough idea is to present a possibly infinite object by automata in such a way that the elementary properties of the object under consideration are effectively retrievable from the presentation. This idea was already present in the early works of Büchi and Rabin, where it was used to show the decidability of certain first-order theories like the theories of $(\mathbb{N}, +)$,$(\mathbb{R}, +)$, and $(\mathbb{N}, \cdot)$. A systematic study of automatic presentations was, however, initiated much later by Khoussainov and Nerode [72] for word automatic structures and later generalised to (infinite) words and trees by Blumensath and Grädel [14], leading to ω-automatic, tree-automatic, and ω-tree-automatic structures, respectively. For an overview we refer the reader to [21, 93].

Definition 2.15. Let $\tau = \{R_1, \ldots, R_n\}$ be a finite relational signature with $\approx \notin \tau$. A τ-structure $\mathfrak{A}$ is (ω-)[tree-]automatic if there is a $\tau \cup \{\approx\}$-structure $\mathfrak{B} = (L, \approx^{\mathfrak{B}}, R_1^{\mathfrak{B}}, \ldots, R_n^{\mathfrak{B}})$ and a surjective mapping $\pi : L \to A$ such that

1. L is a regular (ω-)[tree-]language,
2. the relations $\approx^{\mathfrak{B}}, R_1^{\mathfrak{B}}, \ldots, R_n^{\mathfrak{B}}$ are (ω-)[tree-]regular,
3. for all $v, w \in L$ we have $v \approx^{\mathfrak{B}} w \Leftrightarrow \pi(v) = \pi(w)$, and
4. for all $\overline{v} \in L^k$ it holds that $\overline{v} \in R^{\mathfrak{B}} \Leftrightarrow \pi(\overline{v}) \in R^{\mathfrak{A}}$.

An (ω-)[tree-]automatic presentation of $\mathfrak{A}$ is given by a tuple of (ω-)[tree-]automata $\mathfrak{d} = (\mathcal{A}_L, \mathcal{A}_\approx, \mathcal{A}_{R_1}, \ldots, \mathcal{A}_{R_n})$ that recognize the universe and the relations of $\mathfrak{B}$.

The class of all (ω-)[tree-]automatic is denoted by (ω-)[tree-]AutStr.

All these classes of automatic structures have equivalent descriptions as (finite) set interpretable structures

Fact 2.4. *[13]*

- $\mathrm{FSI}(\mathfrak{N}_<) = \mathrm{AutStr}$,

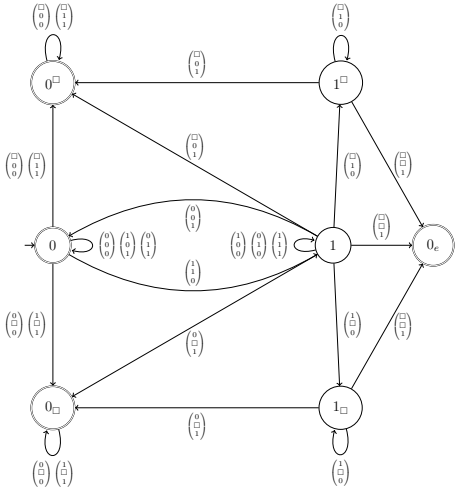


Figure 2.1: The automaton $\mathcal{A}'_+$

- $\mathrm{SI}(\mathfrak{N}_<) = \omega$-AutStr,
- $\mathrm{FSI}(\mathfrak{T}^\omega) = \mathit{tree}$-AutStr, *and*
- $\mathrm{SI}(\mathfrak{T}^\omega) = \omega$-*tree*-AutStr.

Examples We review a few classical examples of various types of automatic presentations in order to get acquainted to the concept.

Example 2.4. The natural numbers with addition $\mathfrak{N}_+ = (\mathbb{N}, +)$ are word automatic. We represent a number $n \in \mathbb{N}$ in reverse binary representation, that is least significant bit first, and use the school algorithm that adds numbers bit wise while storing the necessary carry to perform the operation. From this idea we get a presentation with $L = \{0,1\}^*\{1\} \cup \{\varepsilon\}$ and addition is performed by an automaton that recognises $L(\mathcal{A}'_+) \cap (L \otimes L \otimes L)$, where $\mathcal{A}'_+$ given in Figure 2.4.

In contrast, it is well known that the natural numbers with multiplication $\mathbb{N}_\times$ are not word-automatic [14]. However, $\mathbb{N}_\times$ is tree automatic. This is shown in the following example, which also nicely demonstrates the possibility the use parallelism in tree-automatic presentations.

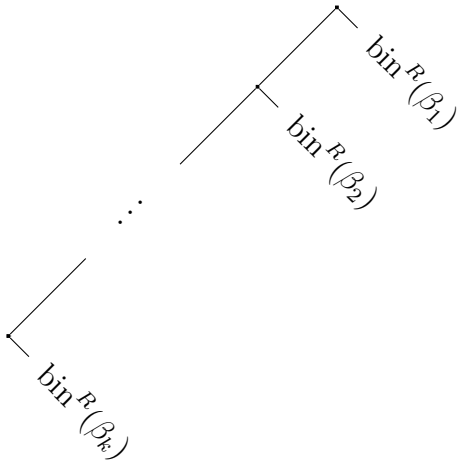


Figure 2.2: Encoding of $n = p_1^{\beta_1} p_2^{\beta_2} \cdots p_k^{\beta_k}$

We quickly recall the definition of a direct sum of monoids. Let I be some index set and $(\mathfrak{M}_i)_{i \in I}$ with $\mathfrak{M}_i = (M_i, \circ^{\mathfrak{M}_i}, 1^{\mathfrak{M}_i})$ be a family of monoids. The direct sum $\oplus_{i \in I} \mathfrak{M}_i$ is the monoid

$$\left(\left\{ (a_i)_{i \in I} \in \prod_{i \in I} M_i \mid a_i = 1^{\mathfrak{M}_i} \text{ for all but finitely many } i \in I \right\}, \circ, \mathbf{1} \right)$$

with $(a_i)_{i \in I} \circ (b_i)_{i \in I} = (a_i \circ^{\mathfrak{M}_i} b_i)_{i \in I}$ and $\mathbf{1} = (1^{\mathfrak{M}_i})_{i \in I}$.

Example 2.5. The natural numbers with multiplication $\mathfrak{N}_\times = (\mathbb{N}, \times)$ are tree-automatic. We make use of the fact that $\mathfrak{N}_\times$ is isomorphic to the infinite direct sum $\oplus_{n \in \mathbb{N}} \mathfrak{N}_+$ for example via the isomorphism $n = p_1^{\beta_1} p_2^{\beta_2} \cdots p_k^{\beta_k} \mapsto (\beta_1, \beta_2, \ldots, \beta_k, 0, \ldots)$, where $\{p_1, p_2, \ldots\} = \mathbb{P}$ are the prime numbers in their natural enumeration. This is reflected in the presentation of a number $n = p_1^{\beta_1} p_2^{\beta_2} \cdots p_k^{\beta_k}$ as depicted in Figure 2.5. The exponents $\beta_1, \beta_2, \ldots$ are encoded in reverse binary on the right branches of a left-growing comb. It is a simple task to check that the trees of this form are recognisable by a tree-automaton. For the multiplication we can now use the idea of Example 2.4 for every right branch separately. Note that $(p_1^{\beta_1} p_2^{\beta_2} \ldots)(p_1^{\gamma_1} p_2^{\gamma_2} \ldots) = p_1^{\beta_1 + \gamma_1} p_2^{\beta_2 + \gamma_2} \ldots$, which establishes the correctness of our construction.

Because there are only countably many finite words or trees, every word- or tree-automatic structure must also be countable. If we want to represent an

uncountable structure we need to switch to automata that read infinite words or trees.

So far all structures in the examples had injective presentations. The following example gives a structure with a non-injective ω-automatic presentation.

Example 2.6. First observe that the Boolean Algebra over the power set of the natural numbers $(\mathcal{P}(\mathbb{N}), \cup, \cap, \bar{\ })$ is ω-automatic. Every word $w \in \{0,1\}^\omega$ represents a set of numbers in probably the most natural way, i.e. $\pi(w) = \{n \in \mathbb{N} \mid w(n) = 1\}$. The operations $\cup, \cap$, and $\bar{\ }$ then correspond to taking bitwise the maximum, minimum, and complement, respectively. Define an equivalence relation on $\mathcal{P}(\mathbb{N})$ by $X \sim Y \Leftrightarrow |X \Delta Y| < \infty$, where Δ is the symmetric difference. One can verify that $\sim$ is a congruence on $(\mathcal{P}(\mathbb{N}), \cup, \cap, \bar{\ })$. The Boolean Algebra $(\mathcal{P}(\mathbb{N}), \cup, \cap, \bar{\ })/\sim$ is atomless. Moreover the congruence $\sim$ is automatic in our presentation because it corresponds to the relation $\sim_e = \{(v, w) \mid |\{i \in \mathbb{N} \mid v[i] \neq w[i]\}| < \infty\}$.

The relation $\sim_e$ is known as the equal-ends relation which will also play an important role later on.

2.3 Logic and Complexity

2.3.1 Parametrised Complexity

Parametrised complexity theory aims to measure the complexity of a problem in terms of functions that may depend on several parameters of the input. The idea is to capture the phenomenon that many presumably hard problems are computable in time that grows only polynomially in the size of the input, but exponential or worse in another parameter of the input. The framework was introduced by Downey and Fellows [32].

Definition 2.16. A **parametrised problem** is a set $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is some finite alphabet.

Definition 2.17. A parametrised problem L is **fixed parameter tractable** (FPT) if $(w, k) \in L$ can be decided in time $f(k)|w|^c$ for some computable function $f : \mathbb{N} \to \mathbb{N}$ and some constant $c \in \mathbb{N}$. We also denote the class of all fixed parameter tractable problems with FPT.

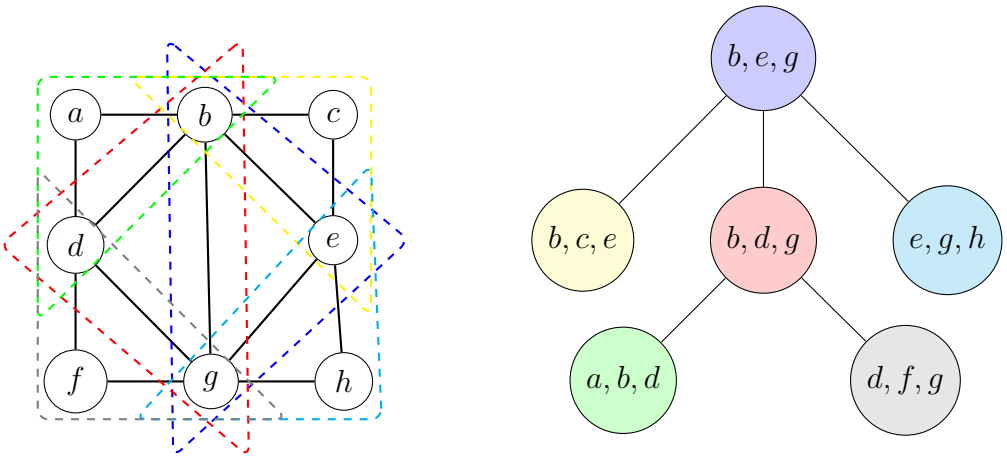


Figure 2.3: A graph of treewidth 2 together with a proper tree decomposition

Often we consider problems in a parameter that is only given implicitly. A good example for this are structural properties of graphs. An important structural parameter of a graphs is its treewidth

Definition 2.18. A **tree decomposition** of a finite graph $G = (V, E^G)$ is a tree $\mathcal{T} = (T, E^{\mathcal{T}})$ with $T \subseteq \mathcal{P}(V)$ such that the following conditions hold:

- $\bigcup T = V$, i.e. every $v \in V$ is contained in some $X \in T$,
- the set $T \restriction v := \{X \in T \mid v \in X\}$ forms a connected subtree in $\mathcal{T}$ for all $v \in V$, and
- if $(v, w) \in E^G$ then $T \restriction v \cap T \restriction w \neq \emptyset$.

The width of a tree decomposition is $\text{width}(\mathcal{T}) = \max\{|X| \mid X \in T\} - 1$. The **treewidth** of a graph G is the smallest width of a tree decomposition of G, that is

$$\text{tw}(G) = \min\{\text{width}(\mathcal{T}) \mid \mathcal{T} \text{ is a tree decomposition of } G\}.$$

Example 2.7. Figure 2.3 shows a graph G with a tree decomposition of width 3. It can be shown that this is optimal, hence $\text{tw}(G) = 2$.

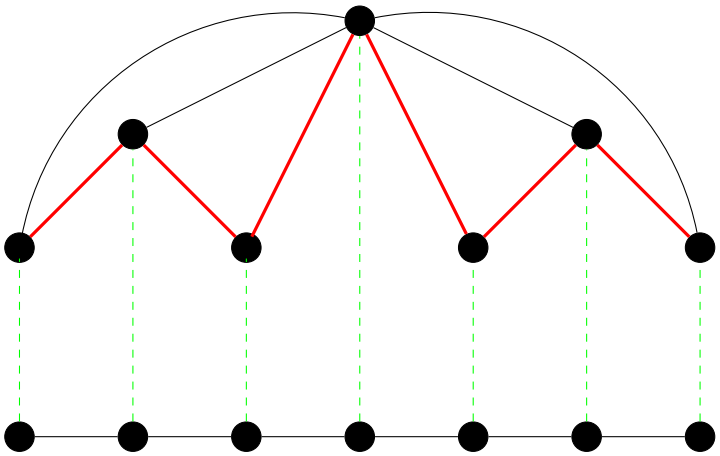


Figure 2.4: The tree-depth of a path with seven nodes

A related, but more restrictive parameter of graphs is the tree-depth

Definition 2.19. The **tree-depth** of a graph $G = (V, E)$ is recursively defined as

$$\mathrm{td}(G) := \begin{cases} 1, & \text{if } |V| = 1 \\ \min\{\mathrm{td}(G \restriction V \setminus \{v\}) \mid v \in V\} + 1 & \text{if } G \text{ is connected and } |V| > 1 \\ \max_{1 \leq i \leq n} \mathrm{td}(G_i) & G \text{ has components } G_1, \ldots, G_n \end{cases}$$

An equivalent characterisation is the minimal height of a rooted forest such that G is isomorphic to a subgraph of the symmetric closure of the ancestor-descendant graph of that forest.

Example 2.8. The tree-depth of a single path with n vertices is $\lfloor \log n \rfloor + 1$. Figure 2.4 shows an optimal embedding of a path with seven vertices.

Logic plays an important role in parametrised complexity theory. An important area of application are algorithmic meta-theorems. Intuitively, an algorithmic meta-theorem states that a whole class of problems is efficiently solvable on a certain class of instances. This is exactly where logic enters the game. If the model checking problem for a logic $\mathcal{L}$ is FPT in the size of the formula on a class $\mathcal{C}$, then this means that all $\mathcal{L}$ definable problems can be

efficiently solved on $\mathcal{C}$. The prototype of such an algorithmic meta-theorem is the theorem of Courcelle for MSO and graphs of bounded treewidth.

Theorem 2.6 (Courcelle [26])**.** MSO *model checking is* FPT *on every class of graphs of bounded treewidth. There is a function f such that for a graph $\mathfrak{G}$ of treewidth at most d and an* MSO *sentence φ one can decide in time*

$$f(d, |\varphi|) \cdot |\mathfrak{G}|$$

whether $\mathfrak{G} \models \varphi$.

For a comprehensive overview with a special emphasis on the connections to logic we refer to [43].

2.3.2 Descriptive Complexity

In descriptive complexity theory one tries to characterise the complexity of a problem in terms of the expressiveness of the language that one needs to describe the problem. A major goal is to identify logics that exactly capture classical complexity classes such as NP or PTIME. In order to make sense, we need to clarify what we mean by a logic capturing a complexity class. Indeed, complexity classes are usually defined in terms of resource bounds to a Turing machine. But a Turing machine operates on strings, while logics describe properties of abstract structures. However, the input string is often just a presentation of a mathematical structure, say for instance a graph, and the Truing machine rather checks a property of the graph than a property of the presentation. In this situation we can say very well that a formula and a Turing machine describe the same property. For instance, a graph may be given by an adjacency matrix, which is written line-wise on the tape of the Turing machine. Then the actual string that we obtain naturally depends on the order that we choose on the vertices, but the property that the graph is Hamiltonian will be invariant among all possible presentations. In this section we want to make this notion precise. For a short historical overview to the field we refer to Chapter 6 and for a broader introduction to the topic we refer to [66] and [53].

Definition 2.20. For a class of Turing machines (possibly nonderterministic or alternating) T, the **complexity class** $\mathrm{C}(T)$ is the class of languages $\{L \mid \exists M \in T : M \text{ decides } L\}$.

Note that most of the usual complexity classes such as PTIME, PSPACE, etc. can be seen as $\mathrm{C}(T)$ for suitable classes of Turing machines T. In all these cases the class of Turing machines is defined via bounds on the resources (computation steps, tape cells, number of alternations, etc.) that the machine may use on an input of size n. Usually one demands that the consumed resources are bounded by some function from a certain class $\mathcal{F} \subseteq \mathbb{N}^{\mathbb{N}}$.

Definition 2.21. The **Berman complexity class** $\bigcup_{c\in\mathbb{N}} \mathrm{STA}(*, 2^{cn}, n)$, is the class of languages accepted by alternating Turing machines running in time 2^{cn} for some constant c and making at most n alternations on inputs of length n.

The algorithmic complexity of a property might vary under different ways to encode structures. Hence, we need to fix a generic way to present arbitrary structures.

Definition 2.22. Let $\tau = \{R_1, \ldots, R_n\}$ be a finite relational signature and $\mathfrak{A}$ be a τ-structure. For every linear order $\leq$ on A we define the function $\pi_< : \{0, \ldots, |A| - 1\} \to A$, which maps $i \in \{0, \ldots, |A| - 1\}$ to the i-th element of A with respect to the order $<$ (starting to count from 0). $\pi_<$ naturally extends to a function $\pi^k_< : \{0, \ldots, |A|^k - 1\} \to A^k$, where the elements of A^k are ordered lexicographically with respect to $<$.

For $1 \leq i \leq n$ we define $\mathrm{code}(R_i^{\mathfrak{A}}, \leq)$ to be the string $w \in \{0,1\}^{|A|^{r_i}}$ with $w[j] = 1$ if, and only if, $\pi^{r_i}_<(j) \in R_i^{\mathfrak{A}}$ for all $0 \leq j < |A|^{r_i}$. Finally we define

$$\mathrm{code}(\mathfrak{A}, \leq) = 1^{|A|} \# \,\mathrm{code}(R_1^{\mathfrak{A}}, \leq) \# \cdots \# \,\mathrm{code}(R_n^{\mathfrak{A}}, \leq),$$

$\mathrm{code}(\mathfrak{A}) = \{\mathrm{code}(\mathfrak{A}, \leq) \mid \leq \text{ linear order on } A\}$, and $\mathrm{code}(\mathcal{C}) = \bigcup_{\mathfrak{A}\in\mathcal{C}} \mathrm{code}(\mathfrak{A})$ for every class of *finite* τ-structures $\mathcal{C}$.

Definition 2.23. A Turing machine M decides a property of τ-structures if there is a class of finite τ-structures $\mathcal{C}$ such that M decides the language $\mathrm{code}(\mathcal{C})$.

We can now define what it means that a logic $\mathcal{L}$ captures a complexity class C.

Definition 2.24. A logic $\mathcal{L}$ captures the complexity class $\mathrm{C}(T)$ (where T is a class of Turing machines) if

1. $\mathcal{L}$ has a decidable syntax,

2. there is an algorithm, which computes for every $\varphi \in \mathcal{L}$ a Turing machine $M \in T$ that decides the language code($\{\mathfrak{A} \mid \mathfrak{A}$ finite, $\mathfrak{A} \models \varphi\}$) and

3. for every $M \in T$ that decides a property of structures over some fixed vocabulary τ there is a sentence $\varphi \in \mathcal{L}$ such that for all finite τ-structures $\mathfrak{A}$ the Turing machine M accepts all codes of $\mathfrak{A}$ if, and only if, $\mathfrak{A} \models \varphi$.

3 Advice Automatic Structures

One line of research in the field of automatic structures is the incorporation of increasingly powerful automata models into the family of automatic presentations. Especially automata that operate on linear orders were extensively investigated, see for instance [41,69,70,95]. In all these cases automatic presentations correspond to set-interpretations in the respective linear order. One can hope that through the consideration of a richer spectrum of structures with a decidable MSO-theory we obtain a better understanding of what can actually be achieved by set-interpretations in general.

We also aim to enrich the class of structures where the set-interpretations are applied to while maintaining useful correspondences to automata based presentations. However, we walk on a path that is somewhat orthogonal to the one described above. More precisely, we consider the expansions of $\mathfrak{N}$ and $\mathfrak{T}^\omega$ by unary predicates, say $P_1, \ldots, P_n$. From the automata theoretic point of view, the sets $P_1, \ldots, P_n$ can be seen as a fixed advice that the automaton reads while it processes the input. This model of automata with advice is known in the literature as referenced automata [8].

The goal of this chapter is to sharpen our techniques that allow us to determine when a structure cannot have an automatic presentation of some sort. Our main results in this direction address a question of Rabin. In his classical paper [87] he asked whether the decidability of the first order theory of certain fields, like the field of reals, can be reduced via interpretations to the decidability of the MSO-theory of $\mathfrak{T}^\omega$. In other words, he asked whether, for instance, the field of reals is ω-tree-automatic. For the restricted case of ω-automatic presentations we can answer this question negatively.

Theorem 3.1. *The field of reals is not ω-automatic with advice.*

Theorem 3.2. *An integral domain is ω-automatic if, and only if, it is finite.*

3.1 Automata, Advices and Interpretations

We start by giving a formal definition of advice automatic presentations and state some basic properties about them. We need to clarify first what a parameterised automaton is. Formally, parameterised automata are just the usual automata over a composed alphabet. The difference is only in the semantics that we impose on them.

Definition 3.1. A **parameterised automaton** is a finite state automaton $\mathcal{A}$, such that $L(\mathcal{A}) \subseteq \Gamma^* \otimes \Sigma^*$. The language that $\mathcal{A}$ recognizes with advice $\alpha \in \Gamma^*$ is $L(\mathcal{A}[\alpha]) = \{w \in \Sigma^* \mid \alpha \otimes w \in L(\mathcal{A})\}$. A language L is called regular with advice α if there is a parameterised automaton $\mathcal{A}$ with $L = L(\mathcal{A}[\alpha])$.

Definition 3.2. A **parameterised Muller automaton** is a Muller automaton $\mathcal{A}$, such that $L(\mathcal{A}) \subseteq \Gamma^\omega \otimes \Sigma^\omega$. the language that $\mathcal{A}$ recognizes with advice $\alpha \in \Gamma^\omega$ is $L(\mathcal{A}[\alpha]) = \{w \in \Sigma^\omega \mid \alpha \otimes w \in L(\mathcal{A})\}$. A language L is called ω-regular with advice α if there is a Muller automaton $\mathcal{A}$ with $L = L(\mathcal{A}[\alpha])$.

The definitions of parameterised tree-automata und ω-tree-automata are analogous and we omit them here.

Definition 3.3. Let τ be a finite relational signature. A **paramerterised (ω-)[tree-]automatic presentation** is a tuple $\mathfrak{d} = (\mathcal{A}, \mathcal{A}_\approx, (\mathcal{A}_R)_{R\in\tau})$ of parameterised (Muller) [tree-]automata. If for a given parameter α the recognised relations are compatible in the sense that if

- $A_\alpha := L(\mathcal{A}[\alpha])$,
- $\approx_\alpha := \mathcal{R}(L(\mathcal{A}_\approx[\alpha])) \subseteq A_\alpha \times A_\alpha$, and
- $R_\alpha := \mathcal{R}(L(\mathcal{A}_R[\alpha])) \subseteq A_\alpha^r$ for $R \in \tau$ with arity r,

then $\mathfrak{d}$ induces the structure $\mathcal{S}_\approx(\mathfrak{d}[\alpha]) = (A_\alpha, \approx_\alpha, (R_\alpha)_{R\in\tau})$. Moreover, if $\approx_\alpha$ is a congruence on $\mathcal{S}_\approx(\mathfrak{d}[\alpha])$, we say that $\mathfrak{d}[\alpha]$ **presents** the structure $\mathcal{S}(\mathfrak{d}[\alpha]) = (A_\alpha, (R_\alpha)_{R\in\tau})/\approx_\alpha$. Being a bit more permissive, we say that $\mathfrak{d}[\alpha]$ presents a structure $\mathfrak{A} \in \mathrm{Str}[\tau]$ if there is a surjective function $\pi : A_\alpha \to A^{\mathfrak{A}}$ such that

1. $v \approx_\alpha w$ implies $\pi(v) = \pi(w)$ for all $v, w \in A_\alpha$, and
2. $(v_1, \ldots, v_r) \in R_\alpha \iff (\pi(v_1), \ldots, \pi(v_r)) \in R^{\mathfrak{A}}$ for all $v_1, \ldots, v_r \in A_\alpha$.

Hence the function π fixes an encoding of the elements of $\mathfrak{A}$ by mapping every $x \in L(\mathcal{A}[\alpha])$ to the element $a \in A^{\mathfrak{A}}$ that is encoded by x. If we want to fix such an encoding, we say that $\mathfrak{A}$ is presented by $(\mathfrak{d}[\alpha], \pi)$. Note that π is in general not uniquely dertermined by $\mathfrak{d}[\alpha]$ and $\mathfrak{A}$. Indeed, $\mathfrak{A}$ might have non-trivial automorphisms. In the case that $\approx_\alpha$ is just the identity we say that the presentation is **injective** and omit $\mathcal{A}_\approx$ in our notation.

A structure $\mathfrak{A}$ is (ω-)[tree-]automatic with advice α, or simply α-automatic, if there is a parameterised (ω-)[tree-]automatic presentation $\mathfrak{d}$ with $\mathfrak{A} \cong \mathcal{S}(\mathfrak{d}[\alpha])$. If we do not want to specify the advice we say that a structure is (ω-)[tree-]automatic with advice or advice (ω-)[tree-]automatic.

The class of (ω-)[tree-]automatic structures over an advice from some advice set P is denoted (ω-)[tree]AutStr[P] or (ω-)[tree]AutStr[α], if $P = \{\alpha\}$ is a singleton set.

We adopt the convention from Remark 2.1 and say that a structure $\mathfrak{A}$ which might contain functions is automatic with advice if this is true for the structure that is obtained by replacing every function f of $\mathfrak{A}$ by its graph R_f.

We also want to point out that the classes AutStr[Γ^*] and tree-AutStr[T_Γ] simply coincide with the classes AutStr and tree-AutStr, respectively. Indeed, if the advice is finite, it can be incorporated into the states of the automata and therefore effectively be dropped. However, the corresponding presentations will become relevant in Chapter 4 & 5 when we consider classes of structures with a uniform presentation.

Fact 2.4 has an obvious analogue in the advice setting. For $\alpha \in \Gamma^\omega$ we define the corresponding **word structure** $\mathfrak{N}_\alpha = (\mathfrak{N}_<, (P^\alpha_\gamma)_{\gamma\in\Gamma})$, where $P^\alpha_\gamma = \{n \in \mathbb{N} \mid \alpha[n] = \gamma\}$. Similarly, for $\alpha \in T^\omega_\Gamma$ over a ranked alphabet Γ we define the tree structure $\mathfrak{T}_\alpha = (\mathrm{dom}_\alpha, S_0, \ldots, S_{\mathrm{rk}(\Gamma)-1}, (P^\alpha_\gamma)_{\gamma\in\Gamma})$ with $S_i = \{(w, wi) \mid w \in \{0, \ldots, \mathrm{rk}(\Gamma)-1\}^*, wi \in \mathrm{dom}_\alpha\}$ for all $i \in \{1, \ldots, \mathrm{rk}(\Gamma)-1\}$ and $P^\alpha_\gamma = \{w \in \mathrm{dom}_\alpha \mid \alpha(w) = \gamma\}$. With these definitions in mind we will sometimes speak about the MSO-theory of an advice. In such cases we always mean the MSO-theory of the corresponding structure.

Fact 3.1. *Let $\mathfrak{A}$ be a structure over a finite relational signature. Then*

- $\mathfrak{A} \in \mathrm{SI}(\mathfrak{N}_\alpha)$ *if, and only if, $\mathfrak{A}$ is ω-automatic with advice α,*
- $\mathfrak{A} \in \mathrm{SI}(\mathfrak{T}_\alpha)$ *if, and only if, $\mathfrak{A}$ is ω-tree-automatic with advice α.*

The classification of the automatic models inside restricted classes of structures is one of the main directions of research in the field of automatic structures. This is especially interesting for classes of structures where we already know that the subclass of automatic models is relatively rich. One such class is certainly the class of abelian groups. Therefore there was a great interest in whether addition on the rationals is automatic. This question was finally answered negatively by Tsankov [98]. However, it was noted before that the additive group of rationals is almost automatic in the sense that there is a presentation, based on the factorial expansion of rational numbers, in which addition is automatic. The only flaw of this presentation is that the domain is not a regular set. It was later noted by Kruckman et al. [76] that the domain becomes regular if the automaton has access to a fixed advice string. The next example explains the presentation in a bit more detail. The idea goes back to Stephan and independently Miller.

Example 3.1 (See also [82])**.** We use the fact that every rational $r \in \mathbb{Q}$ has a unique factorial expansion $r = z + \sum_{i=2}^{\infty} a_i/(i!)$, where $z \in \mathbb{Z}, 0 \leq a_i \leq i$, and only finitely many a_i are different from 0. Every rational number q can thus be represented by an integer z and a sequence $(a_i)_{i\geq 2}$ that satisfies the above conditions. The addition of two rationals r and q, represented by $(z, \overline{a})$ and $(s, \overline{b})$ can be performed by a carry procedure. The sum

$$(r + q) = (z + s) + \sum_{i\geq 2} \frac{a_i + b_i}{i!}$$

is equal to $t + \sum_i \frac{k_i}{i!}$ (where $\overline{k}$ satisfies the above conditions) if, and only if, there exists a sequence of carry bits $(c_i)_{i\geq 1}$, such that

$$c_i = \begin{cases} 0 & \text{if } \forall j > i(a_j = b_j = 0) \\ 0 & \text{if } a_{i+1} + b_{i+1} + c_{i+1} < i + 1 \\ 1 & \text{otherwise.} \end{cases}$$

$$k_i = a_i + b_i + c_i \mod i, \text{ and}$$

$$t = z + s + c_1.$$

The correctness is obtained by the equality

$$\frac{a + b + c}{n!} = \frac{1}{(n-1)!} + \frac{a + b + c - n}{n!}.$$

We convert the factorial expansions of a rational $r = z + \sum_{i=2}^{\infty} a_i/(i!)$ to an infinite string (written on two tapes). On the first tape we write $\mathrm{bin}^{\mathrm{R}}(z)$ and write

$$f(a_2, 2)\#f(a_2, 3)\#f(a_4, 4)\#\dots$$

on the second tape, where $f(a, i)$ is the string of the form $0^n \mathrm{bin}(a)$ with $|0^n \mathrm{bin}(a)| = \lceil log_2(i) \rceil + 1$ (remember we assume $a_i < i$ and hence $|\mathrm{bin}(a_i)| \leq \lceil log_2(i) \rceil + 1$). Consider

$$\alpha = \mathrm{bin}(2)\#\,\mathrm{bin}(3)\#\,\mathrm{bin}(4)\#\dots$$

as a parameter. We can construct an automaton $\mathcal{A}$ that verifies this encoding when α is given as advice. Indeed, $\mathcal{A}$ can read $\mathrm{bin}(i)$ as reference in the ith cell and therefore check whether the given string is a valid encoding of a rational number. For the same reason, the carry procedure described above can be implemented by an automaton with advice α.

Note that the advice α is a veriation of the Champernowne-Smarandache string. Strings of this kind belong to the class of k-lexicographic strings, a class of infinite words that was introduced in [20] where it was also shown that these words have a decidable MSO-theory.

The fact that the advice $\alpha = \mathrm{bin}(2)\#\,\mathrm{bin}(3)\#\cdots$ in Example 3.1 has a decidable MSO-theory is particularly interesting. Indeed, Example 3.1 together with Fact 2.4 allows us to reduce the FO-theory of $(\mathbb{Q}, +)$ to the MSO-theory of $\mathfrak{N}_\alpha$ and therefore yields an effective decision procedure.

One of the main reasons why the various flavours of automatic presentations have drawn so much attention is that the existence of an automatic presentation directly yields a decision procedure for the first-order theory of the structure. Analogously the existence of an advice automatic presentation directly yields a reduction of the first-order theory of the structure to the MSO-theory of the advice. We state the fundamental theorem that first-order formulae can be translated into corresponding automata.

Theorem 3.1. *Given a parameterised (ω-)[tree-]automatic presentation* $\mathfrak{d}$ *and an* FO*-formula* $\varphi(\overline{x})$ *with* m *free variables over the signature of* $\mathfrak{d}$ *one can effectively construct an (ω-)[tree-]automaton* $\mathcal{A}_\varphi$ *with*

$$L(\mathcal{A}_\varphi) = \{\langle \alpha, \overline{a} \rangle \mid \mathcal{S}(\mathfrak{d}[\alpha]) \models \varphi(\overline{a})\}.$$

Since there is no technical difference to the proof of the corresponding theorem for ordinary automatic structures, we omit it here. A detailed description of the algorithm is given in Chapter 5. For more information on the classical case without advice the reader is referred to [14].

Corollary 3.1.

1. *The class of (ω-)[tree-]automatic structures with advice α is effectively closed under* FO*-interpretations.*

2. *If a structure $\mathfrak{A}$ is (ω-)[tree-]automatic with advice α and the* MSO*-theory of α is decidable then $\mathfrak{A}$ has a decidable* FO*-theory.*

Theorem 3.1 shows that we can extend every advice automatic presentation of a structure to a presentation of the expansion by a first-order definable relation. This motivates to notion of inherent regularity. Let $\mathfrak{A}$ be (ω-)[tree-]automatic with advice. We say that a relation R on A is **inherently (ω-)[tree-]regular** if every (ω-)[tree-]automatic presentation $\mathfrak{d}$ with some advice α can be extended to a presentation of $(\mathfrak{A}, R)$. If the automaton model under consideration is clear from the context then we will just say that R is inherently regular.

MSO Model Checking We want to remark here that under certain conditions automatic presentations can be used beyond first-order model checking. Obviously, automatic structures do in general not have a decidable MSO-theory. A simple example is the $\mathbb{N} \times \mathbb{N}$ grid, which is word automatic, even without advice, but has an undecidable MSO-theory. However, depending on the presentation, we can sometimes encode also sets of elements by words or trees and employ them to obtain an automata based decision procedure for the MSO-theory. Simple examples are presentations where the elements are encoded in unary.

Definition 3.4. Let $\mathfrak{d} = (\mathcal{A}, \mathcal{A}_{R_1}, \ldots, \mathcal{A}_{R_n})$ be a parameterised (ω-)automatic presentation and α be some advice. We say that $\mathfrak{d}[\alpha]$ uses a **unary encoding** if $L(\mathcal{A}[\alpha]) \subseteq \{0\}^*\{1\}$ ($\{0\}^*\{1\}\{0\}^\omega$, respectively).

Definition 3.5. Let $\mathfrak{d} = (\mathcal{A}, \mathcal{A}_{R_1}, \ldots, \mathcal{A}_{R_n})$ be a parameterised (ω-)tree-automatic presentation and α be some advice. We say that $\mathfrak{d}[\alpha]$ uses a **unary**

encoding if for all $t \in L(\mathcal{A}[\alpha])$ it holds that $\mathrm{dom}_t = \mathrm{dom}_\alpha$ and there is a $v \in \mathrm{dom}_t$ such that

$$t(w) = \begin{cases} 0 & w \neq v \\ 1 & w = v \end{cases}.$$

That is all $t \in L(\mathcal{A}[\alpha])$ are completely labelled with 0s except for exactly one position.

It is an easy exercise to show that the ω-(tree-)automatic structures with a unary presentation over an advice α are exactly the structures that are interpretable by a unary MSO-interpretation in $\mathfrak{N}_\alpha$ ($\mathfrak{T}_\alpha$).

If an (ω-)[tree-]automatic presentation $\mathfrak{d}$ uses a unary encoding with advice α, then we can enrich the presentation to present also all sets.

Definition 3.6. Let $\mathfrak{A} = (A, R_1, \ldots, R_n)$ be a τ-structure. The **power set structure** $\mathcal{P}(\mathfrak{A})$ is the $(\tau \uplus \{\subseteq\})$-structure $(\mathcal{P}(A), R_1^{\mathcal{P}(\mathfrak{A})}, \ldots, R_n^{\mathcal{P}(\mathfrak{A})}, \subseteq)$, where $(\mathcal{P}(A), \subseteq)$ is the powerset lattice on A and

$$R_i^{\mathcal{P}(\mathfrak{A})} = \{(\{a_1\}, \ldots, \{a_{r_i}\}) \in \mathcal{P}(A)^{r_i} \mid (a_i, \ldots, a_{r_i}) \in R_i\}$$

for all $i \in \{1, \ldots, n\}$.

Corollary 3.2. *If a structure $\mathfrak{A}$ has an (ω-)[tree-]automatic presentation $\mathfrak{d}$ with advice α and $\mathfrak{d}[\alpha]$ uses a unary encoding then the structure $\mathcal{P}(\mathfrak{A})$ is also (ω-)[tree-]automatic with advice α.*

Clearly the MSO-theory of $\mathfrak{A}$ is reducible to the FO-theory of $\mathcal{P}(\mathfrak{A})$ and vice versa.

Corollary 3.3. *Let $\mathfrak{A}$ be a structure. If $\mathcal{P}(\mathfrak{A})$ is (ω-)[tree-]automatic with advice α, then the* MSO*-theory of $\mathfrak{A}$ is reducible to the* MSO*-theory of α.*

3.2 Model Theoretic Properties of Advice Automatic Structures

As we have argued in the previous chapter, adding an advice to the presentation is only interesting for automata over infinite words or trees (as long as we are only interested in presenting a single structure). While presentations over finite

words have been quite intensively studied in the literature, there is much less known about their counterparts over infinite words. One major obstacle is that not every ω-automatic structure has an injective presentation. But even in the case of ω-automatic presentations with advice of countable structures, where (as we will see later in this chapter) injective presentations exist, we still cannot employ the techniques that have been developed for automatic presentations. Indeed, most of these techniques rely on the pumping lemma, which we can no longer apply once an advice is present. In this section we aim therefore to develop new techniques that overcome both of these hurdles. In the second part of this chapter we apply these techniques to prove that certain structures cannot have an ω-automatic presentation (with advice).

On the technical side we follow two paths. First, we perform a combinatorial analysis of equal-end-classes of regular languages. Secondly, we investigate transitive relations by means of ω-semigroups which recognise their presentation.

3.2.1 Bounding the End-Index

In this section we want to develop some technical tools to show that certain structures are not advice automatic. For our investigations we will make heavy use of the equal-ends relation $\sim_e$ for infinite words.

Definition 3.7. For $m \in \mathbb{N}$ two words $v, w \in \Sigma^\omega$ are *m-end-equivalent* ($v \sim_e^m w$) if, and only if, $v[m, \infty) = w[m, \infty)$. That is v and w are equal except for possibly the first m symbols. We say v and w are *end-equivalent* if, and only if, $v \sim_e^m w$ for some $m \in \mathbb{N}$.

Clearly the equivalence relation $\sim_e^m$ partitions any language into finite classes, each of size at most $|\Sigma|^m$.

End-equivalence plays a crucial role in the study of ω-regular languages. We first observe that every infinite ω-regular language has an infinite $\sim_e$-class.

In the following we examine which elements of a structure can be encoded by words from the same $\sim_e$-class. To this end it is convenient to lift the notion of end-equivalence from words in a given presentation to elements of the encoded structure.

Definition 3.8. Let $\sim$ be an equivalence relation on Σ^ω and $\mathfrak{A}$ be a structure with advice automatic presentation $\mathfrak{d}[\alpha]$. Fix a witness $\pi : L(\mathcal{A}[\alpha]) \to A^{\mathfrak{A}}$ for

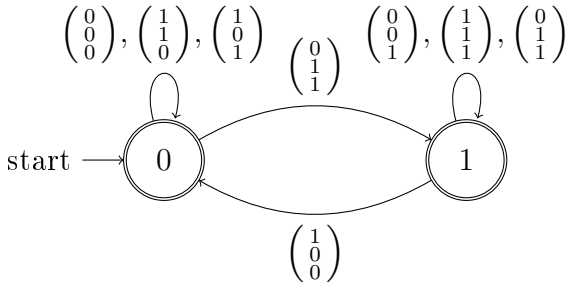
$\mathfrak{d}[\alpha]$ presenting $\mathfrak{A}$ as in Definition 3.3. The $\sim$-*index* of a set $B \subseteq A$ *in* $(\mathfrak{d}[\alpha], \pi)$, $I^{\sim}_{(\mathfrak{d}[\alpha],\pi)}(B)$, is the least $\sim$-index of a set $X \subseteq L(\mathcal{A}[\alpha])$ with $\pi(X) = B$. That is

$$I^{\sim}_{(\mathfrak{d}[\alpha],\pi)}(B) = \min_{\{X \mid \pi(X)=B\}} |\{[x]_{\sim} \mid x \in X\}|.$$

We are mostly interested in the equivalence relations $\sim_e$ and $\sim_e^m$ for $m \in \mathbb{N}$, hence we introduce the shorthand notations $I_{(\mathfrak{d},\pi)}(B)$ and $I^m_{(\mathfrak{d},\pi)}(B)$ for $I^{\sim_e}_{(\mathfrak{d},\pi)}(B)$ and $I^{\sim_e^m}_{(\mathfrak{d},\pi)}(B)$, respectively. We will also omit $(\mathfrak{d}, \pi)$ in the subscript whenever the presentation under consideration is clear from the context.

We turn our attention to the refined relations $\sim_e^m$. These relations are especially interesting as they relate, in some sense, to the concept of same length for finite words. Indeed for finite words one can consider the mapping $w \mapsto w\square^\omega$ (where $\square$ is a fresh symbol), then all words of length at most m form a $\sim_e^m$-class. Our aim is to provide a property of ω-automatic functions that is in the spirit of the Growth-Rate Lemma for automatic functions. Remember that for an automatic function f there is always a constant c such that $|f(\bar{x})| \leq \max_i |x_i| + c$. Or in other words: for all sets $X \subseteq \Sigma^{\leq m}$, the image $f(X^k)$ is a subset of Σ^{m+c}. However, a direct translation of this result fails already without advice. In general even if f is an ω-automatic function and X a $\sim_e^m$-equivalent set, the set $f(X^k)$ is not necessarily $\sim_e$-equivalent.

Example 3.2. Consider the function $f(x, y)$ realised by the following Büchi-automaton:



Then $0^\omega \sim_e^1 10^\omega$, but $f(0^\omega, 10^\omega) = 0^\omega \not\sim_e 1^\omega = f(10^\omega, 0^\omega)$.

Although the function f, given in Example 3.2, produces outputs which are not $\sim_e$-equivalent, one might notice that the set $f(X^2)$ is distributed only over

a constant number of $\sim_e^m$-classes (in this case two) for all $\sim_e^m$-equivalent sets X. This is indeed a general property of automatic presentations with advice, which is formulated in the following lemma.

Lemma 3.1. *Let $\mathcal{A} = (Q, q_0, \Gamma \times (\Sigma)^{k+\ell+1}, \delta, F)$ be a parametrised Muller automaton. Then there is a constant c such that for every $\alpha \in \Gamma^\omega, \bar{a} \in (\Sigma^\omega)^k$, and every $m \in \mathbb{N}$ the following holds:*

Let $R \subseteq (\Sigma^\omega)^{k+\ell+1}$ be recognised by $\mathcal{A}$ with advice α. Then for every set $X \subseteq \Sigma^\omega$ with $I^m(X) = n$ there is a partition of X^ℓ into cn^ℓ many sets $Y_1, \ldots, Y_{cn^\ell}$ with the property:

$$\bar{x}, \bar{x}' \in Y_i \Rightarrow \left\{[y]_{\sim_e^m} \mid y \in \bar{a}\bar{x}R\right\} = \left\{[y]_{\sim_e^m} \mid y \in \bar{a}\bar{x}'R\right\}.$$

Proof. We carry out the proof for the case $k = 0$. The general case is completely analogous. Let q be the number of states of $\mathcal{A}$. For every $\bar{x} \in X^\ell$ we consider the set

$$\delta_m(\bar{x}, \cdot) := \{\delta_m(\alpha \otimes \bar{x} \otimes y) \mid (\bar{x}, y) \in R\} \subseteq \mathcal{P}(Q).$$

The number c of possible sets $\delta_m(x, \cdot)$ is bounded by 2^q. We claim that the partition of X^ℓ with respect to the equivalence relation

$$\bar{x} \sim \bar{x}' :\Leftrightarrow \bigwedge_{1 \leq i \leq \ell} x_i \sim_e^m x_i' \wedge \delta_m(\bar{x}, \cdot) = \delta_m(\bar{x}', \cdot)$$

is suitable for our purpose.

First observe that $\sim$ partitions X into at most cn^ℓ many equivalence classes. Now suppose that $\delta_m(\bar{x}, \cdot) = \delta_m(\bar{x}', \cdot)$ for some $\bar{x}, \bar{x}' \in X^\ell$. We need to show that if $\bar{x} \sim \bar{x}'$ then

$$\left\{[y]_{\sim_e^m} \mid y \in \bar{x}R\right\} \subseteq \left\{[y]_{\sim_e^m} \mid y \in \bar{x}'R\right\}$$

or, in other words, that for every $y \in \bar{x}R$ the set $[y]_{\sim_e^m} \cap \bar{x}'R$ is not empty.

Because of $\delta(\bar{x}, \cdot) = \delta(\bar{x}', \cdot)$ there is for every $y \in \bar{x}R$ a $y' \in \bar{x}'R$ with

$$\delta_m(\alpha \otimes \bar{x} \otimes y) = \delta_m(\alpha \otimes \bar{x}' \otimes y').$$

Consider $y'' := y'[0,m)y[m,\infty)$. The element y'' is in $x'R$ because

$$\begin{aligned}\delta(\alpha \otimes \bar{x}' \otimes y'') &\overset{\bar{x} \sim_e^m \bar{x}'}{=} \delta((\alpha \otimes \bar{x}' \otimes y')[0,m)(\alpha \otimes \bar{x} \otimes y)[m,\infty) \\ &= \delta(\delta_m(\alpha \otimes \bar{x}' \otimes y'), (\alpha \otimes \bar{x}' \otimes y)[m,\infty)) \\ &= \delta(\delta_m(\alpha \otimes \bar{x} \otimes y), (\alpha \otimes \bar{x} \otimes y)[m,\infty)) \\ &= \delta(\alpha \otimes \bar{x} \otimes y) \in F.\end{aligned}$$

This establishes the claim since $y \sim_e^m y''$. □

Lemma 3.2. *Let τ be a signature with a $(k+\ell)$-ary function symbol f and $\mathfrak{d}$ be a parametrised ω-automatic presentation. Then there is a constant c such that for every τ-structure $\mathfrak{A}$ with presentation $(\mathfrak{d}[\alpha], \pi)$ for some $\alpha \in \Gamma^\omega$ we have*

$$I^m(f(\bar{a}, X^\ell)) \leq c \cdot (I^m(X))^\ell$$

for all $m \in \mathbb{N}$, $\bar{a} \in A^k$, and all finite $X \subseteq A$.

Proof. The graph of f is regular with advice α in the presentation $(\mathfrak{d}[\alpha], \pi)$. We apply Lemma 3.1 and obtain $c \in \mathbb{N}$. For every finite $X \subseteq A$ we can fix a witness $Y \subseteq \Sigma^\omega$ for the m-index of X, i.e. $\pi(Y) = X$ and $|Y/\sim_e^m| = I^m(X) =: n$. We partition $Y^\ell = T_1 \uplus \cdots \uplus T_{cn^\ell}$ according to Lemma 3.1, which ensures that for $1 \leq i \leq cn^k$ there exists a class $[y_i]_{\sim_e^m}$ such that $f(\bar{a}, \pi(T_i)) \subseteq \pi([y_i]_{\sim_e^m} \cap L(\mathcal{A}[\alpha]))$. Consequently $f(\bar{a}, X^\ell) \subseteq \pi\left(\bigcup_{1 \leq i \leq cn^\ell} [y_i]_{\sim_e^m} \cap L(\mathcal{A}[\alpha])\right)$, thus $I^m(f(\bar{a}, X^\ell)) \leq cn^\ell = cI^m(X)^\ell$. □

Intuitively one could say that if in some α-automatic presentation $\mathfrak{d}$ of a structure $\mathfrak{A}$ the elements of some finite $X \subseteq A$ are not distributed too widely in $L(\mathcal{A}[\alpha])$, then no function f of $\mathfrak{A}$ will distribute the image $f(X^k)$ to widely in $L(\mathcal{A}[\alpha])$. This property is especially useful to bound the size of the image of certain sets under f.

Lemma 3.3. *Let τ be a signature with a $(k+\ell)$-ary function symbol f and $\mathfrak{d}$ be a parametrised ω-automatic presentation. Then there is a constant c such that for every τ-structure $\mathfrak{A}$ with presentation $(\mathfrak{d}[\alpha], \pi)$ for some $\alpha \in \Gamma^\omega$, every substructure $\mathfrak{B} \subseteq \mathfrak{A}$, every $m \in \mathbb{N}$, and every finite set $X \subseteq B$ there is a finite set Y with $X \subseteq Y \subseteq B$ and*

$$|f(\bar{a}, Y^\ell)| \leq c(I^m(X) + 1)^\ell \cdot |Y|$$

for all $\bar{a} \in B^k$.

Proof. By Lemma 3.2, there is a constant c such that $I^m(f(\bar{a}, S^\ell)) \leq c(I^m(S))^\ell$ for all for all $m \in \mathbb{N}$, $\bar{a} \in A^k$, and all finite $S \subseteq A$. Let $\mathfrak{B} \subseteq \mathfrak{A}$ be a substructure of $\mathfrak{A}$ and $X \subseteq B$ some finite subset. Choose $y \in L(\mathcal{A}[\alpha])$ such that $Y' := \pi([y]_{\sim_e^m} \cap L(\mathcal{A}[\alpha])) \cap B$ is of maximal cardinality among all possible choices for y and set $Y := X \cup Y'$. Then $I^m(Y) \leq I^m(X) + 1$.

Now, consider $\bar{a} \in B^\ell$. Let $W \subseteq L(\mathcal{A}[\alpha])$ be a witness for the m-index of $f(\bar{a}, Y^\ell)$. Then $|\{[w]_{\sim_e^m} \mid w \in W\}| \leq c(I^m(X) + 1)^\ell$. By the maximality of Y' it holds that

$$|\pi([w]_{\sim_e^m} \cap L(\mathcal{A}[\alpha])) \cap B| \leq |Y'| \leq |Y|$$

for all $w \in W$ and because f is closed on B we can bound the size of $f(\bar{a}, Y^\ell)$ by

$$|f(\bar{a}, Y^\ell)| \leq |\bigcup_{w \in W} (\pi([w]_{\sim_e^m} \cap L(\mathcal{A}[\alpha])) \cap B)| \leq c(I^m(X) + 1)^\ell |Y|.$$

□

An important special case is when the end-index of a set is one in an advice-automatic presentation.

Corollary 3.4. *Let τ be a signature with a $(k+\ell)$-ary function symbol f and $\mathfrak{d}$ be a parametrised ω-automatic presentation. Then there is a constant c such that for every τ-structure $\mathfrak{A}$ with presentation $(\mathfrak{d}[\alpha], \pi)$ for some $\alpha \in \Gamma^\omega$, every substructure $\mathfrak{B} \subseteq \mathfrak{A}$ and every finite set $X \subseteq B$ with $I(X) = 1$ there is a finite set Y with $X \subseteq Y \subseteq B$ and*

$$|f(\bar{a}, Y^\ell)| \leq c \cdot |Y|$$

for all $\bar{a} \in B^k$.

Proof. If $I(X) = 1$ then there is also an $m \in \mathbb{N}$ with $I^m(X) = 1$. The claim follows from Lemma 3.3 with $c = c'2^k$, where c' is the constant from Lemma 3.3. □

3.2.2 Injective Presentations

Our investigation of the end-index in the previous section turns out to be especially useful if the underlying presentation is injective. Indeed, injective advice automatic presentations enforce large sets with small end-index.

Lemma 3.4. *Let $L \subseteq \Sigma^\omega$ be infinite and ω-regular with advice α. Then L has an infinite $\sim_e$-equivalence class.*

Proof. We consider two cases. If $X := L/\sim_e$ is finite then obviously there must be an infinite $x \in X$. In the other case X is infinite. Fix a parametrised Muller automaton $\mathcal{A} = (Q, \Gamma \times \Sigma, \Delta, \mathcal{F})$ that recognises L with advice α and let $k := |Q|^2 + 1$. There are $v_1, \ldots, v_k \in L$ such that $v_i \not\sim_e v_j$ for $1 \leq i < j \leq k$. Let ρ_i be the accepting run of $\mathcal{A}$ on $\alpha \otimes v_i$. We define $c_\ell : \binom{\mathbb{N}}{2} \to Q \times Q$ by

$$c_\ell(\{i, j\}) := (\rho_\ell[i], \rho_\ell[j])$$

for $1 \leq \ell \leq k$ and $i < j \in \mathbb{N}$. Combine all c_ℓ to a colouring c of all two element subsets $X \in \binom{\mathbb{N}}{2}$ by $c(X) = (c_1(X), \ldots, c_k(X))$.

Observe that the range of c is finite and therefore we can apply Ramsey's Theorem to c. There exists an infinite set $N = \{n_1 < n_2 < \cdots\} \subseteq \mathbb{N}$ such that c is monochromatic on $\binom{N}{2}$. By definition of c this means $c_\ell(X) = c_\ell(X')$ for every $X, X' \in \binom{N}{2}$ and every $\ell \in \{1, \ldots, k\}$. Further $k > |Q \times Q|$ and therefore there must be $1 \leq i < j \leq k$ such that $c_i \restriction \binom{N}{2} = c_j \restriction \binom{N}{2}$. Assume w.l.o.g. this holds for $i = 0$ and $j = 1$. Additionally, since $v_0 \not\sim_e v_1$, we can assume that N is coarse enough to ensure $v_0[n_i, n_{i+1}) \neq v_1[n_i, n_{i+1})$ for all $i \in \mathbb{N}$.

To complete the proof define $v_i^j := v_i[n_j, n_{j+1})$ and consider the language

$$L' = \{v(w) = v_0[0, n_1)v_{w_1}^1 \ldots v_{w_m}^m v_0[n_{m+1}, \infty) \mid w = w_1 \ldots w_m \in \{0, 1\}^*\}.$$

Obviously L' is an infinite end-equivalent set. We show that the language L' is contained in L. By a simple induction one shows that

$$\delta(\alpha[0, n_{m+1}) \otimes v_0[0, n_1)v_{w_1}^1 \ldots v_{w_m}^m) = \delta((\alpha \otimes v_0)[0, n_{m+1}))$$

for all $w_1 \ldots w_m \in \{0, 1\}^*$. Hence

$$\begin{aligned} \delta(\alpha \otimes v(w)) &= \delta(\delta_{n_{m+1}}(\alpha \otimes v(w)), (\alpha \otimes v_0)[n_{m+1}, \infty)) \\ &= \delta(\delta((\alpha \otimes v_0)[0, n_{m+1}), (\alpha \otimes v_0)[n_{m+1}, \infty)) \\ &= \delta(\alpha \otimes v_0) \in \mathcal{F} \end{aligned}$$

for all $w \in \{0, 1\}^*$. □

Lemma 3.5. *For $\alpha \in \Gamma^\omega$ let $\mathfrak{A}$ be an infinite structure with injective advice automatic presentation $(\mathfrak{d}[\alpha], \pi)$. Then there is an infinite set $B \subseteq A$ with $I(B) = 1$.*

Proof. Since A is infinite, $L = L(\mathcal{A}[\alpha])$ must also be infinite and therefore by Lemma 3.4 there must be an infinite class $X \in L/\sim_e$. By assumption, the mapping π is injective whence $\pi(X)$ is infinite and $I(\pi(X)) = 1$. □

However, this is not true for non-injective ω-automatic presentations with advice. Indeed, Example 2.6 provides an ω-automatic presentation where $I(X) = |X|$ for every subset X.

The above observations enforce that the images of finite sets under a injectively presentable function cannot always be large. This is captured via the notion of the *minimal image size*.

Definition 3.9. For every function $f : A^k \to A$ over an infinite set A we define the *minimal image size* $\mathrm{MIS}_f : \mathbb{N} \to \mathbb{N}$ by

$$\mathrm{MIS}_f(n) = \min\{|f(X^k)| : X \subseteq A, |X| = n\}.$$

We now show that for injectively presentable structures the minimal image size of every function grows at most linearly with n.

Lemma 3.6. *Let $\mathfrak{A}$ be an infinite structure with injective automatic presentation with advice. Then for every inherently regular function f of $\mathfrak{A}$ it holds that* $\mathrm{MIS}_f(n) = \mathcal{O}(n)$.

Proof. We proceed similar to the proof Lemma 3.3 but this time we have to ensure that the set which we choose has the correct size. Fix an injective automatic presentation $(\mathfrak{d}[\alpha], \pi)$ (over some alphabet Σ) of $\mathfrak{A}$ and suppose $f : A^k \to A$ is an inherently regular function on $\mathfrak{A}$ where MIS_f grows super-linearly. We can extend $\mathfrak{d}$ by a Muller automaton $\mathcal{A}_f$ that recognises $\pi^{-1}(R_f)$. Let q be the constant from Lemma 3.2 with respect to f and $((\mathfrak{d}, \mathcal{A}_f), \pi)$. Now choose n such that $\mathrm{MIS}_f(n) > |\Sigma| \cdot q \cdot n$. This is possible since MIS_f grows super-linearly. By Lemma 3.5 there is an infinite set $M \subseteq A$ with $I(M) = 1$. Therefore there is also a smallest m such that there is a set $N \subseteq A$ with $I^m(N) = 1$ of size at least n. Let $N \subseteq A$ be a set with $I^m(N) = 1$ of maximal cardinality.

The size of N is bounded from above by $|\Sigma| \cdot n$. Otherwise we could partition N into the $|\Sigma|$ many sets $(N_a)_{a \in \Sigma}$ with $I^{m-1}(N_a) = 1$ for all $a \in \Sigma$. But then, because $|N| > |\Sigma| \cdot n$, one set N_a must contain more than n elements, which contradicts the choice of m.

Now let N_0 be some subset of N of size exactly n. By Lemma 3.2, the set $f(N_0^k)$ can be partitioned into q many $\sim_e^m$-equivalent sets. One of these sets has size at least

$$\frac{|f(N_0^k)|}{q} > \frac{|\Sigma| \cdot q \cdot n}{q} = |\Sigma| \cdot n \geq |N|.$$

But this contradicts the maximality of N among all $\sim_e^m$-equivalent sets. □

3.2.3 Transitive Automatic Relations

It is well-known that from every word or tree-automatic presentation one can effectively construct an injective presentation. Indeed, every (tree-)automatic equivalence relation has a (tree-)automatic system of representatives. In contrast, the class of injectively presentable ω-automatic structures is strictly included in the class of all ω-automatic structures [63]. This remains true in the presence of an advice string [91]. The problem that we face when we want to apply the results of the previous section to non-injective presentations is that, in general, large sets with small $\sim_e$-index do not need to exist. As seen in Example 2.6, the relation $\sim_e$ is an ω-automatic equivalence relation and thus an automatic presentation might indeed identify all end-equivalent words.

In this section we continue our investigation of the interplay between advice automatic relations and the $\sim_e$ relation. The goal is to obtain a better understanding of how the regularity of the presentation restricts the way in which the elements are encoded. In particular we will examine transitive relations, especially the equality relation $\approx$ and linear orders.

The main technical result is that every advice ω-automatic presentation of an uncountable linear order contains an injective ω-automatic sub-presentation of the lexicographic order on all infinite binary strings $(\{0,1\}^\omega, <_{\text{lex}})$. Note that for ω-automatic structures without advice Kuske has already shown that $(\{0,1\}^\omega, <_{lex})$ is embeddable into any ω-automatic uncountable linear order [77]. More specifically, he constructs from a given ω-automatic presentation of such an order a sub-presentation that is a presentation of $(\{0,1\}^\omega, <_{lex})$. This sub-presentation is not ω-automatic but its domain is the complement of a language $\bigcup_{i \leq n} V_i U_i^\omega$ where the V_i are context free and the U_i are regular. Our result strengthens Kuske's result in two ways. First we extend the result to presentations with advice and second the sub-presentation is much simpler. This will be an essential property for the application in the following sections.

Our construction is to a large part inspired by [68]. The proof makes heavy use of the algebraic characterisation of ω-regular languages by finite ω-semigroups. The key is to show that every ω-semigroup that recognises an uncountable transitive relation contains a certain sub-semigroup with nice algebraic properties. These properties can, in turn, be translated into meaningful properties of the recognised relation. Moreover the presentation maps a regular sub-language into all of these semigroups in a "synchronised" way.

Theorem 3.2. *For every $\alpha \in \Gamma^\omega$ and every ω-automatic presentation with advice α, $\mathfrak{d}[\alpha] = ((\alpha, \mathcal{A}_L, \mathcal{A}_\approx, \mathcal{A}_<), \pi)$, of an uncountable linear order there is an α-automatic subset L' of $L(\mathcal{A}_L[\alpha])$ such that the restriction of $\mathfrak{d}[\alpha]$ to L', is an injective α-automatic presentation of $(\{0,1\}^\omega, <_{lex})$.*

Proof. We fix ω-semigroups $S_\delta = (S_f^\delta, S_\omega^\delta)$, for $\delta \in \{L, \approx, <\}$ such that there are ω-semigroup homomorphisms

$$h_L : (\Gamma \times \Sigma)^\infty \to S_L,$$
$$h_\approx : (\Gamma \times \Sigma \times \Sigma)^\infty \to S_\approx, \text{ and}$$
$$h_< : (\Gamma \times \Sigma \times \Sigma)^\infty \to S_<,$$

which recognise the languages $L(\mathcal{A}_L), L(\mathcal{A}_\approx)$, and $L(\mathcal{A}_<)$, respectively. Note that we treat the automata of $\mathfrak{d}$ here as ordinary Muller-automata, that means h_L recognises the language $\{\alpha \otimes w \in (\Gamma \times \Sigma)^\omega \mid w \in L(\mathcal{A}[\alpha])\}$ and so on. We will use this convention whenever we speak about an ω-semigroup homomorphism recognising an α-automatic relation. For $\delta \in \{L, \approx, <\}$ we set $F_\delta := h_\delta(\{\alpha\} \otimes L(\mathcal{A}_\delta[\alpha])) \subseteq S_\omega^\delta$. Accordingly, an ω-word $v \in \Sigma^\omega$ ($v \in \Sigma^\omega \times \Sigma^\omega$, respectively) is in $L(\mathcal{A}_\delta[\alpha])$ if, and only if, $h_\delta(\alpha \otimes v) \in F_\delta$. We define $C := |S_L| \cdot |S_\approx| \cdot |S_<|$ and k as the least common multiple of the exponents of the semigroups $S_L, S_\approx, S_<$.

We break down the remaining proof into several parts. Our intermediate goal is to find words and an α-automatic factorisation such that the morphisms behave in a manageable fashion with respect to this factorisation. First we give an automatic version of Ramsey's Theorem for α-automatic languages.

Definition 3.10. Let $h : (\Sigma)^\infty \to S = (S_f, S_\omega)$ be an ω-semigroup homomorphism, $v \in \Sigma^\omega$, and $G = \{g_1 < g_2 < g_3 < \ldots\} \subseteq \mathbb{N}$. We say that G is an h-homogeneous factorization of v if, and only if, for some $e \in S_f$ the homomorphism h maps $v[g_i, g_j)$ to e for all $1 \leq i < j$. Making the element e explicit we also say that G is an h, e-homogeneous factorisation of v.

Note that if G is an h, e-homogeneous factorisation then e is necessarily idempotent. Hence we could equivalently demand that e is idempotent and $h(v[g_i, g_{i+1})) = e$ for all $i \geq 1$.

In the following we will use the notion of being α-automatic in a slightly more permissive way than just for infinite words. In all the cases we mean that a suitable encoding of the object under consideration as an infinite word is α-automatic. In particular, a set $G \subseteq \mathbb{N}$ is α-automatic if the characteristic string w_G, that is the infinite $\{0,1\}$-string with $w_G[i] = 1$ if, and only if, $i \in G$, is α-automatic. Further a sequence $(w_i)_{i\in\mathbb{N}}$ of (non-empty) finite words is α-automatic if the word $w_0 w_1 w_2 \ldots$ and the set $\left\{\sum_{i<m} |w_i| \mid m \in \mathbb{N}\right\}$, which is the set of positions where some w_i starts in $w_0 w_1 w_2 \ldots$, is α-automatic .

Lemma 3.7. *For $\alpha \in \Gamma^\omega$ let $\Phi = \{R_1, \ldots, R_n\}$ be a finite set of α-automatic relations where R_i is recognised by a morphism $h_i : (\Gamma \times \Sigma^{r_i})^\infty \to S_i$ for some finite ω-semigroup S_i. Then for every finite set $V = \{v_1, \ldots, v_k\} \subseteq \Sigma^\omega$ there is an automaton $\mathcal{A}_{(\Phi,V)}$ that recognises*

$$\mathrm{hfac}(\Phi, V) = \{w_G \in \{0,1\}^\omega \mid \forall 1 \leq i \leq n \, \forall \overline{v} \in V^{r_i} : \\ G \text{ is an } h_i\text{-homogeneous factorisation of } \alpha \otimes \overline{v}\}$$

with advice $(\alpha, v_1, \ldots, v_k)$.

Proof. Fix the sets $\Phi = \{R_1, \ldots, R_n\}$ and $V = \{v_1, \ldots, v_k\}$. We describe the behaviour of a non-deterministic automaton $\mathcal{A}_{(\Phi,V)}$ that recognises $\mathrm{hfac}(\Phi, V)$. In the beginning $\mathcal{A}_{(\Phi,V)}$ guesses for every $R_i \in \Phi$ and every $\overline{v} \in V^{r_i}$ an idempotent element $e(i, \overline{v}) \in S_i^f$ and then checks for all possible pairs $(i, \overline{v})$ simultaneously that $h_i((\alpha \otimes \overline{v})[g_j, g_{j+1})) = e(i, \overline{v})$ for each $j \in \mathbb{N}$. This can be implemented by an automaton with advice $(\alpha, v_1, \ldots, v_k)$ because the images of $(\alpha \otimes \overline{v})[g_j, g_{j+1})$ under h_i can clearly be computed while reading the input on the intervals $[g_j, g_{j+1})$ which are explicitly distinguished by w_G. □

Ramsey's Theorem ensures that $\mathrm{hfac}(\Phi, V)$ is never empty.

Lemma 3.8. *Let $\Phi = \{R_1, \ldots, R_n\}$ be a finite set of α-automatic relations with associated morphisms $h_1, \ldots, h_n$ into finite ω-semigroups $S_1, \ldots, S_n$. Further let $V \subset \Sigma^\omega$ be a finite set of words. Then* $\mathrm{hfac}(\Phi, V) \neq \emptyset$.

Proof. We need to show that there is a factorisation G that is h_k-homogeneous for $\overline{v}$ for all $1 \leq k \leq n$ and all $\overline{v} \in V^{r_k}$. In order to apply Ramsey's Theorem we colour every $\{i,j\} \in \binom{\mathbb{N}}{2}$, where $i < j$, with the set

$$\{(k, \overline{v}, h_k((\alpha \otimes \overline{v})[i,j])) \mid 1 \leq k \leq n, \overline{v} \in V^{r_k}\} \subseteq \bigcup_{1 \leq k \leq n} \{k\} \times V^{r_k} \times S_k^f.$$

There are only finitely many colours and hence Ramsey's Theorem implies that there is an infinite set $G = \{n_1 < n_2 < \cdots\} \subseteq \mathbb{N}$ such that every $\{i,j\} \in \binom{N}{2}$ have the same colour, say $\{(k, \overline{v}, e_k) \mid 1 \leq k \leq n, \overline{v} \in V^{r_k}\}$. That is $h_k((\alpha \otimes \overline{v})[i,j)) = e_k$ for all $1 \leq k \leq n$, $\overline{v} \in V^{r_k}$, and all $i < j \in N$. Hence G has the intended property. □

We use the Uniformisation Theorem to obtain two α-automatic words v_0 and v_1 and an α-automatic factorisation H that is homogeneous for all possible combinations of v_0 and v_1 in the relations under consideration.

Lemma 3.9. *There are α-automatic $v_0, v_1 \in L(\mathcal{A}[\alpha])$ with $[v_0]_{\sim_e} \cap [v_1]_{\approx} = \emptyset$ and an α-automatic factorisation $G = \{g_1 < g_2 < g_3 < \ldots\} \subseteq \mathbb{N}$ such that for $\delta \in \{<, \approx\}$, the set G is*

- *an h_L, e_L-homogeneous factorisation of v_0 and v_1,*
- *an h_δ, e_δ-homogeneous factorisation of (v_0, v_0) and (v_1, v_1),*
- *an h_δ, e_δ^{01}-homogeneous factorisation of (v_0, v_1), and*
- *an h_δ, e_δ^{10}-homogeneous factorisation of (v_1, v_0)*

for idempotent elements $e_L, e_\delta, e_\delta^{01}$, and e_δ^{10} in the respective semigroups.

Proof. Since $\mathfrak{d}$ is a presentation of an uncountable structure, there is an infinite set $\{w_0, w_1, w_2, \ldots\} \subseteq L(\mathcal{A}[\alpha])$ such that for all $i < j$: $[w_i]_{\sim_e} \cap [w_j]_{\approx} = \emptyset$. To see this, note that every $\sim_e$-class contains only countably many elements.

The relations $\sim_e$ and $\approx$ are α-automatic and hence the $(C+1)$-ary relation defined by $\psi(x_0, \ldots, x_C) = \bigwedge_{0 \leq i < j \leq C} [x_i]_{\sim_e} \cap [x_j]_{\approx} = \emptyset$ is also α-automatic. With the previous argument we also see that $\psi^{\mathcal{S}(\mathfrak{d})}$ is non-empty. We apply the Uniformisation Theorem to the relation defined by ψ and obtain an α-automatic tuple $(v_0, \ldots, v_C)$ such that for $0 \leq i < j \leq C$ the end-class of v_i does not intersect the $\approx$-class of v_j.

Let Φ consist of the relations defined by $\mathcal{A}[\alpha], \mathcal{A}_{\approx}[\alpha]$, and $\mathcal{A}_{<}[\alpha]$ and $V = \{v_0, \ldots, v_C\}$. Then the relation $\mathrm{hfac}(\Phi, V)$ is $(\alpha, v_0, \ldots, v_C)$-automatic and hence α-automatic. As $\mathrm{hfac}(\Phi, V)$ is not empty, we can again by applying the Uniformisation Theorem conclude that there is an α-automatic factorisation G that is homogeneous for all possible combinations of elements from V under the morphisms $h_L, h_{\approx}$, and $h_{<}$.

For each $v \in V$ let the profile of v be the unique tuple $(e_L, e_{\approx}, e_{<}) \in S_L \times S_{\approx} \times S_{<}$ such that G is a h_L, e_L-homogeneous factorisation of v and for $\delta \in \{\approx, <\}$ a h_δ, e_δ-homogeneous factorisation of v. Because $|V| = C+1$ is larger than the number of possible profiles, there must be two distinct elements with the same profile. Without loss of generality assume this is true for v_0 and v_1. Then it is easily verified that v_0,v_1, and G have the postulated properties. □

Since $v_0 \not\sim_e v_1$, we may also assume, without losing the property of G being α-automatic, that G is coarse enough such that $\alpha_0[g_\ell, g_{\ell+1}) \neq \alpha_1[g_\ell, g_{\ell+1})$ for all $\ell \in \mathbb{N}$. From v_0, v_1 we define two α-automatic sequences $(w_{0i})_{i\in\mathbb{N}}$ and $(w_{1i})_{i\in\mathbb{N}}$ of *finite* words, which will be the "templates" for the language that is constructed hereinafter.

Lemma 3.10. *There are α-automatic sequences $(w_{0i})_{i\in\mathbb{N}}$ and $(w_{1i})_{i\in\mathbb{N}}$ and an α-automatic factorisation $H = \{h_0 < h_1 < \cdots\} \subseteq \mathbb{N}$ such that:*

- *$w_{00} = w_{10}$, $w_{0i} \neq w_{1i}$, and $|w_{0i}| = |w_{1i}|$ for all $i > 0$,*
- *$h_i = \sum_{j<i} |w_{0j}| = \sum_{j<i} |w_{1j}|$ for all $i \in \mathbb{N}$, and*
- *for $\delta \in \{\approx, <\}$ there is an element $\to_\delta \in S_f^\delta$ and idempotent elements $\square_\delta, \uparrow_\delta, \downarrow_\delta \in S_f^\delta$ such that*

$$\begin{aligned} h_\delta(\alpha[0,h_1) \otimes w_{00} \otimes w_{00}) &= h_\delta(\alpha[0,h_1) \otimes w_{10} \otimes w_{10}) \\ &= h_\delta(\alpha[0,h_1) \otimes w_{00} \otimes w_{10}) \\ &= h_\delta(\alpha[0,h_1) \otimes w_{10} \otimes w_{00}) \\ &= \to_\delta \end{aligned}$$

and for every $i > 0$

- $h_\delta(\alpha[h_i, h_{i+1}) \otimes w_{0i} \otimes w_{0i}) = h_\delta(\alpha[h_i, h_{i+1}) \otimes w_{1i} \otimes w_{1i}) = \square_\delta$,
- $h_\delta(\alpha[h_i, h_{i+1}) \otimes w_{0i} \otimes w_{1i}) = \uparrow_\delta$,

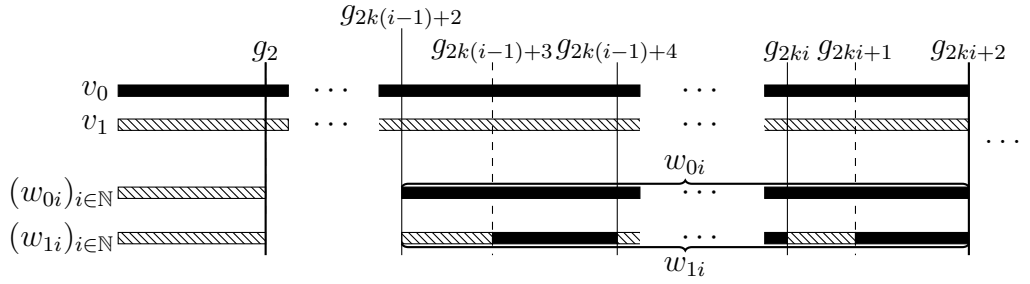


Figure 3.1: Definition of $(w_{0i})_{i\in\mathbb{N}}$ and $(w_{1i})_{i\in\mathbb{N}}$

- $h_\delta(\alpha[h_i, h_{i+1}) \otimes w_{1i} \otimes w_{0i}) = \downarrow_\delta$, *and*
- $\rightarrow_\delta, \uparrow_\delta$ *and* $\downarrow_\delta$ *absorb* $\square_\delta$ *from the right.*

Proof. Let us first define the sequences $(w_{0i})_{i\in\mathbb{N}}$ and $(w_{1i})_{i\in\mathbb{N}}$. We define $w_{00} = w_{10} = v_1[0, g_2)$ and for $i > 0$ we define

$$w_{0i} = v_0[g_{2k(i-1)+2}, g_{2ki+2}) \text{ and}$$
$$w_{1i} = v_1[g_{2k(i-1)+2}, g_{2k(i-1)+3})v_0[g_{2k(i-1)+3}, g_{2k(i-1)+4}) \cdots v_1[g_{2ki}, g_{2ki+1})v_0[g_{2ki+1}, g_{2ki+2}).$$

Using that v_0 and v_1 are α-automatic, we see that both sequences are α-automatic. Indeed, the word $w_{00}w_{01}\ldots$ differs only in finitely many positions from v_0 and the word $w_{10}w_{11}\ldots$ coincides, from position g_2 onward, with v_0 on intervals of the form $[g_{2\ell}, g_{2\ell+1})$ and with v_1 on intervals of the form $[g_{2\ell+1}, g_{2(\ell+1)})$. Therefore from automata that recognise v_0, v_1, and G with advice α we are able to define automata that recognise $w_{00}w_{01}w_{02}\ldots$ and $w_{10}w_{11}w_{12}\ldots$ with advice α, respectively. Similarly, we can recognise the set $H = \{\sum_{i\leq n} |w_{0i}| \mid n \in \mathbb{N}\} = \{g_{2kn+2} \mid n \in \mathbb{N}\} = \{\sum_{i\leq n} |w_{1i}| \mid n \in \mathbb{N}\}$ with advice α because G is α-automatic and k is constant. The assertion that $w_{0i} \neq w_{1i}$ for all $i > 0$ follows from the fact that $v_0[g_\ell, g_{\ell+1}) \neq v_1[g_\ell, g_{\ell+1})$ for all $\ell \in \mathbb{N}$.

Let us now compute the values $\rightarrow_\delta, \uparrow_\delta, \downarrow_\delta$ and $\square_\delta$ for $\delta \in \{\approx, <\}$ and verify

their properties:

$$\begin{aligned}\rightarrow_\delta &= h_\delta(\alpha[0,h_1)\otimes w_{00}\otimes w_{00})\\ &= h_\delta(\alpha[0,g_2)\otimes v_1[0,g_2)\otimes v_1[0,g_2))\\ &= h_\delta(\alpha[0,g_1)\otimes v_1[0,g_1)\otimes v_1[0,g_1))\cdot h_\delta(\alpha[g_1,g_2)\otimes v_1[g_1,g_2)\otimes v_1[g_1,g_2))\\ &= h_\delta(\alpha[0,g_1)\otimes v_1[0,g_1)\otimes v_1[0,g_1))e_\delta\end{aligned}$$

We use the fact that $h_\delta(\alpha[g_j,g_{j+1})\otimes v_i[g_j,g_{j+1})\otimes v_i[g_j,g_{j+1}))=e_\delta$ for $i\in\{0,1\}$ and all $j>0$ and compute

$$h_\delta(\alpha[h_j,h_{j+1})\otimes w_{ij}\otimes w_{ij})=(e_\delta)^{2k}=e_\delta=:\square_\delta$$

for $i\in\{0,1\}, j>0$. Similar computations reveal $\uparrow_\delta=(e_\delta^{01}e_\delta)^k$ and $\downarrow_\delta=(e_\delta^{10}e_\delta)^k$.

Finally the elements $\square_\delta,\uparrow_\delta$ and $\downarrow_\delta$ are idempotent and $\rightarrow_\delta,\uparrow_\delta$ and $\downarrow_\delta$ absorb $\square_\delta$ because e_δ is idempotent and k is a multiple of the exponent of S_f^δ. □

With Lemma 3.10 we are prepared to set up our sub-presentation. Let $L':=\{w_{j_0 0}w_{j_1 1}w_{j_2 2}\ldots \mid j_k\in\{0,1\}\text{ for all }k\in\mathbb{N}\}$. Having established that the sequences $(w_{0i})_{i\in\mathbb{N}}$ and $(w_{1i})_{i\in\mathbb{N}}$ are α-automatic, we can conclude that L' is also α-automatic. We use w_i as a shorthand for $w_{i0}w_{i1}w_{i2}\ldots\in L'$ where i ranges over $\{0,1\}$.

Our ultimate goal is to show that the restriction of $\mathcal{S}(\mathfrak{d}[\alpha])$ to L' is isomorphic to $(\{0,1\}^\omega,<_{\text{lex}})$ and hence we need to show that $<$ behaves accordingly on L'. In fact we claim that there is a bijection function $\rho:\{0,1\}\to\{0,1\}$ such that the function $\pi_\rho: L'\to\{0,1\}^\omega; w_{j_0 0}w_{j_1 1}w_{j_2 2}\ldots\mapsto\rho(j_0)\rho(j_1)\rho(j_2)\ldots$ is an isomorphism between the two structures. The key to establish this is two exploit the transitivity of $<$ and $\approx$ in combination with the properties stated in Lemma 3.10. To ease the notation in this process we will omit the subscript for the semigroup elements mentioned in Lemma 3.10. The context in which they are used should prevent any danger of confusion.

Lemma 3.11. *The language L' is contained in L.*

Proof. We compute $h_L(\alpha\otimes w)$ for every $w\in L'$:

$$\begin{aligned}h_L(\alpha\otimes(w_{j_i i})_{i\in\mathbb{N}}) &= h_L(\langle\alpha[0,h_1),w_{j_0 0}\rangle)\overbrace{h_L(\langle\alpha[h_1,h_2),w_{j_1 1}\rangle)}^{=e_L}\overbrace{h_L(\langle\alpha[h_2,h_3),w_{j_2 2}\rangle)}^{=e_L}\ldots\\ &= h_L(\alpha[0,h_1)\otimes v_1[0,h_1))(e_L)^\omega\\ &= h_L(\alpha\otimes v_1)\in F_L.\end{aligned}$$

□

We define a *convolution* for the semigroup elements $\uparrow, \downarrow$ and $\square$ (with respect to $<$ and $\approx$). We define $\odot : \{\uparrow, \downarrow, \square\} \times \{\uparrow, \downarrow, \square\} \to \{\uparrow, \downarrow, \square, \bot\}$ by

$$x \odot y = \begin{cases} \uparrow & \text{if } (x = \square \text{ and } y = \uparrow) \text{ or } (x = \uparrow \text{ and } y = \square) \\ \downarrow & \text{if } (x = \square \text{ and } y = \downarrow) \text{ or } (x = \downarrow \text{ and } y = \square) \\ \square & \text{if } x = y = \square \text{ or } (x = \uparrow \text{ and } y = \downarrow) \text{ or } (x = \downarrow \text{ and } y = \uparrow) \\ \bot & \text{else.} \end{cases}$$

For two sequences $(x_i)_{i\geq 1}, (y_i)_{i\geq 1} \in \{\uparrow, \downarrow \square\}^\omega$ we apply the convolution elementwise, i.e. $(x_i)_{i\geq 1} \odot (y_i)_{i\geq 1} = (x_i \odot y_i)_{i\geq 1}$. The reason for this definition is that we want to argue about the transitivity of the relations on the level of the corresponding semigroup-morphisms. This is stated more precisely in the following lemma.

Lemma 3.12. *For $\delta \in \{<, \approx\}$ let $(x_i)_{i\geq 1}, (y_i)_{i\geq 1} \in \{\uparrow, \downarrow \square\}^\omega \subseteq (S_f^\delta)^\omega$ be two sequences of ω-semigroup elements such that $x_i \odot y_i \in \{\uparrow, \downarrow, \square\}$ for all $i \geq 1$. If $\to (x_i)_{i\geq 1} \in F_\delta$ and $\to (y_i)_{i\geq 1} \in F_\delta$ then $\to (x_i \odot y_i)_{i\geq 1} \in F_\delta$.*

Proof. Define words

$$\begin{aligned} u_0 &= w_{00} w_{j_{01}1} w_{j_{02}2} \dots, \\ u_1 &= w_{00} w_{j_{11}1} w_{j_{12}2} \dots, \text{ and} \\ u_2 &= w_{00} w_{j_{21}1} w_{j_{22}2} \dots, \end{aligned}$$

where

$$j_{0i} = \begin{cases} 0 & x_i = \uparrow \text{ or } (x_i = \square \text{ and } y_i = \uparrow) \\ 1 & \text{else} \end{cases}$$

$$j_{1i} = \begin{cases} 0 & (x_i = \downarrow \text{ or } y_i = \uparrow) \\ 1 & \text{else} \end{cases}$$

$$j_{2i} = \begin{cases} 0 & y_i = \downarrow \text{ or } (y_i = \square \text{ and } x_i = \downarrow) \\ 1 & \text{else} \end{cases}$$

Note that $j_{0i}, j_{1i},$ and j_{2i} are defined so that

$$h_\delta(\alpha[h_i, h_{i+1}) \otimes w_{j_{0i}i} \otimes w_{j_{1i}i}) = x_i, h_\delta(\alpha[h_i, h_{i+1}) \otimes w_{j_{1i}i} \otimes w_{j_{2i}i}) = y_i,$$

and $h_\delta(\alpha[h_i, h_{i+1}) \otimes w_{j_{0i}i} \otimes w_{j_{2i}i}) = x_i \odot y_i$. Hence, if $h_\delta(\alpha \otimes u_0 \otimes u_1) = \rightarrow (x_i)_{i\geq 1} \in F_\delta$ and $h_\delta(\alpha \otimes u_1 \otimes u_2) = \rightarrow (y_i)_{i\geq 1} \in F_\delta$ then, by the transitivity of δ, it must also be true that $h_\delta(\alpha \otimes u_0 \otimes u_2) = \rightarrow (x_i \odot y_i)_{i\geq 1} \in F_\delta$.

□

We claimed that L' induces an injective presentation and hence we want to show that all words in L' represent distinct elements. We do so by first showing that at least some words cannot encode the same element. For this step it is crucial that we have chosen v_0 and v_1 such that $[v_0]_{\sim_e} \cap [v_1]_\approx = \emptyset$.

Lemma 3.13. *The ω-semigroup elements $\rightarrow\uparrow^\omega$ and $\rightarrow (\uparrow\downarrow)^\omega$ are not in $F_\approx$.*

Proof. Observe that $h_\delta(\alpha, w_0, w_1) = \rightarrow\uparrow^\omega$ and also

$$\begin{aligned}
\rightarrow\uparrow^\omega &= h_\approx(\langle\alpha, v_1, v_1\rangle[0, g_2)(h_\approx(h_\approx(\langle\alpha, v_0, v_1\rangle[g_{2i}, g_{2i+1})\langle\alpha, v_1, v_1\rangle[g_{2i+1}, g_{2i+2}))_{i>1} \\
&= h_\approx(\langle\alpha, v_1, v_1\rangle[0, g_1)e^{11}(e^{01}e^{11})^\omega \\
&= h_\approx(\langle\alpha, v_1, v_1\rangle[0, g_1)e^{11}e^{11}(e^{01}e^{11})^\omega \\
&= h_\approx(\langle\alpha, v_1, v_1\rangle[0, g_1)e^{11}(e^{11}e^{01})^\omega \\
&= h_\approx(\langle\alpha, v_1, v_1\rangle[0, g_2)(h_\approx(\langle\alpha, v_1, v_1\rangle[g_{2i}, g_{2i+1})\langle\alpha, v_0, v_1\rangle[g_{2i+1}, g_{2i+2}))_{i>1} \\
&= h_\approx(w_1, v_1).
\end{aligned}$$

So, if $\rightarrow\uparrow^\omega \in F_\approx$ then $w_0 \approx w_1$ and also $w_1 \approx v_1$ and therefore by transitivity $w_0 \approx v_1$. But $w_0 \sim_e v_0$ which means $\rightarrow\uparrow^\omega \in F_\approx$ implies $[v_0]_{\sim_e} \cap [v_1]_\approx \neq \emptyset$. Because we have chosen v_0 and v_1 such that $[v_0]_{\sim_e} \cap [v_1]_\approx = \emptyset$ we may conclude that $\rightarrow\uparrow^\omega \notin F_\approx$.

Now suppose $\rightarrow (\uparrow\downarrow)^\omega \in F_\approx$. Observe that

$$\rightarrow (\uparrow\downarrow)^\omega = \rightarrow (\uparrow\uparrow\downarrow)^\omega = \rightarrow\uparrow (\uparrow\downarrow\uparrow)^\omega = \rightarrow (\uparrow\downarrow\uparrow)^\omega$$

and also

$$\rightarrow (\uparrow\downarrow)^\omega = \rightarrow \square(\uparrow\downarrow \square)^\omega = \rightarrow (\square \uparrow\downarrow)^\omega.$$

But $\rightarrow (\uparrow\downarrow\uparrow)^\omega \odot \rightarrow (\square \uparrow\downarrow)^\omega = \rightarrow (\uparrow \square\square)^\omega = \rightarrow\uparrow^\omega$ whence with Lemma 3.12 we conclude that $\rightarrow\uparrow^\omega \in F_\approx$. Contradiction! □

We will use similar arguments as in the proof of Lemma 3.13 to characterise the elements in $\to\{\uparrow,\downarrow,\square\}^\omega \cap F_<$ to complete the proof of the main theorem. In the next lemma we establish a normal form for the elements of $\to\{\uparrow,\downarrow,\square\}^\omega$ in order to ease this task.

Lemma 3.14. *The set* $\to\{\uparrow,\downarrow,\square\}^\omega$ *is equal to the union of the sets*

1. $\to\{(\uparrow\downarrow)^\omega,(\downarrow\uparrow)^\omega\}$,
2. $\to\{(\uparrow\downarrow)^{<2k},(\uparrow\downarrow)^{<2k}\uparrow,(\downarrow\uparrow)^{<2k},(\downarrow\uparrow)^{<2k}\downarrow\}\{\uparrow^\omega,\downarrow^\omega,\square^\omega\}$

Proof. The containment from right to left is trivial and hence we only need to show the containment in the other direction. Let us first remark that every finite product $\to x_1x_2\cdots x_n$, where $x_i\in\{\uparrow,\downarrow,\square\}$ for $1\leq n$, is equal to an element in the set $\to\{(\uparrow\downarrow)^{<2k},(\downarrow\uparrow)^{<2k},(\uparrow\downarrow)^{<2k}\uparrow,(\downarrow\uparrow)^{<2k}\downarrow\}$. Indeed, the element $\square$ can be eliminated because it is absorbed by $\uparrow,\downarrow$, and $\to$. The elements $\uparrow,\downarrow$ are idempotent whence consecutive occurrences of the same element can be eliminated. Finally, k is a multiple of the exponent of the semigroup (i.e. $(\uparrow\downarrow)^{2k}=(\uparrow\downarrow)^k$ and $(\downarrow\uparrow)^{2k}=(\downarrow\uparrow)^k$).

Let $(x_i)_{i\geq1}$ be any sequence from $\{\uparrow,\downarrow,\square\}^\omega$. Consider the set $P=\{i\geq1\mid x_i\neq\square\}$. We distinguish two cases.

Case P is finite. In this case $(x_i)_{i\geq1}=x_1x_2\cdots x_n\square^\omega$ for some n and with our initial remark we conclude that $\to x_1x_2\cdots x_n\square^\omega$ is equal to some element in the set $\to\{(\uparrow\downarrow)^{<2k},(\downarrow\uparrow)^{<2k},(\uparrow\downarrow)^{<2k}\uparrow,(\downarrow\uparrow)^{<2k}\downarrow\}\square^\omega$.

Case P is infinite. Let $p_1,p_2,\ldots$ be the elements of P in ascending order. Define $y_i:=(x_{p_i}x_{p_i+1}\cdots x_{p_{i+1}-1})=x_{p_i}\square^{p_{i+1}-p_i-1}=x_{p_i}$. By the associativity of the infinite product $\to(x_i)_{i\geq1}=\to(y_i)_{i\geq1}\in\to\{\uparrow,\downarrow\}^\omega$. If both sets $\{i\mid y_i=\uparrow\}$ and $\{i\mid y_i=\downarrow\}$ are infinite then we can again utilise the associativity of the infinite product and the idempotence of $\uparrow$ and $\downarrow$ to see that $\to(y_i)_{i\geq1}\in\to\{(\uparrow\downarrow)^\omega,(\downarrow\uparrow)^\omega\}$. Otherwise exactly one of the sets is finite and we can argue analogously to the first case that

$$\to(y_i)_{i\geq1}\in\to(\{\uparrow\downarrow\}^{<2k}\cup\{\downarrow\uparrow\}^{<2k}\cup\{\uparrow\downarrow\}^{<2k}\{\uparrow\}\cup\{\downarrow\uparrow\}^{<2k}\{\downarrow\})\uparrow^\omega \text{ or}$$
$$\to(y_i)_{i\geq1}\in\to(\{\uparrow\downarrow\}^{<2k}\cup\{\downarrow\uparrow\}^{<2k}\cup\{\uparrow\downarrow\}^{<2k}\{\uparrow\}\cup\{\downarrow\uparrow\}^{<2k}\{\downarrow\})\downarrow^\omega,$$

respectively. □

Lemma 3.15. *The structure* $\mathcal{S}(\mathfrak{d}[\alpha])\restriction L'$ *is isomorphic to* $(\{0,1\}^\omega,<_{lex})$.

Proof. Because $\rightarrow(\uparrow\downarrow)^\omega \notin F_\approx$ we know that the words $x_0 := w_{10}w_{01}w_{12}\ldots$ and $x_1 := w_{00}w_{11}w_{02}\ldots$ encode distinct elements. Therefore either

$$h_<(\langle\alpha, x_0, x_1\rangle) = \rightarrow(\uparrow\downarrow)^\omega \in F_< \text{ or } h_<(\langle\alpha, x_1, x_0\rangle) = \rightarrow(\downarrow\uparrow)^\omega \in F_<.$$

We carry out the proof for the case $\rightarrow(\uparrow\downarrow)^\omega \in F_<$. For the other case one just has to interchange the roles of $\uparrow$ and $\downarrow$ in the following.

Recall that we claimed $w_{00}w_{j_11}w_{j_22}\ldots < w_{00}w_{j'_11}w_{j'_22}\ldots$ if, and only if, $j_1j_2\ldots <_{\text{lex}} j'_1j'_2\ldots$, which is with our previous observations equivalent to the claim that $\rightarrow\uparrow\{\uparrow,\downarrow,\square\}^\omega \subseteq F_<$. With Lemma 3.14 it is sufficient to consider elements of the form $\rightarrow x\gamma$, where $x \in \{(\uparrow\downarrow)^{<2k}, (\uparrow\downarrow)^{<2k}\uparrow\}$ and $\gamma \in \{\uparrow^\omega, \downarrow^\omega, \square^\omega\}$. We make a case distinction with respect to the value of γ.

Case $\gamma = \uparrow^\omega$. In this case we can assume that $x = (\uparrow\downarrow)^c$, for some $0 \leq c < 2k$. We observe that

$$\begin{aligned}\rightarrow(\uparrow\downarrow)^\omega &= \rightarrow(\uparrow\downarrow)^c\square(\uparrow\downarrow\square)^\omega\\ &= \rightarrow(\uparrow\downarrow)^c(\square\uparrow\downarrow)^\omega\end{aligned}$$

but also

$$\begin{aligned}\rightarrow(\uparrow\downarrow)^\omega &= \rightarrow\square^{2c}(\uparrow\uparrow\downarrow)^\omega\\ &= \rightarrow\square^{2c}\uparrow(\uparrow\downarrow\uparrow)^\omega\\ &= \rightarrow\square^{2c}(\uparrow\downarrow\uparrow)^\omega.\end{aligned}$$

Further

$$\begin{aligned}\rightarrow((\uparrow\downarrow)^c(\square\uparrow\downarrow)^\omega \odot \square^{2c}(\uparrow\downarrow\uparrow)^\omega) &= \rightarrow(\uparrow\downarrow)^c(\uparrow\square\square)^\omega\\ &= \rightarrow(\uparrow\downarrow)^c\uparrow^\omega\\ &= \rightarrow x\gamma.\end{aligned}$$

Lemma 3.12 allows us to conclude $\rightarrow x\gamma \in F_<$.

Case $\gamma = \downarrow^\omega$. Here we can assume that $x = (\uparrow\downarrow)^c\uparrow$, for some $0 \leq c < 2k$. Again we compute suitable identities for $\rightarrow(\uparrow\downarrow)^\omega$.

$$\begin{aligned}\rightarrow(\uparrow\downarrow)^\omega &= \rightarrow\square^{2c+2}(\uparrow\downarrow\square)^\omega\\ &= \rightarrow\square^{2c+1}(\square\uparrow\downarrow)^\omega \text{ and}\\ \rightarrow(\uparrow\downarrow)^\omega &= \rightarrow(\uparrow\downarrow)^c\uparrow(\downarrow\uparrow)^\omega\\ &= \rightarrow(\uparrow\downarrow)^c\uparrow(\downarrow\downarrow\uparrow)^\omega.\end{aligned}$$

We compute the convolution

$$\begin{aligned}\rightarrow(\square^{2c+1}(\square\uparrow\downarrow)^\omega \odot (\uparrow\downarrow)^c\uparrow(\downarrow\downarrow\uparrow)^\omega) &= \rightarrow(\uparrow\downarrow)^c\uparrow(\downarrow\square\square)^\omega\\ &= \rightarrow(\uparrow\downarrow)^c\uparrow(\downarrow)^\omega\\ &= \rightarrow x\gamma\end{aligned}$$

and get that $\rightarrow x\gamma \in F_<$.

Case $\gamma = \square^\omega$. Here either $x = (\uparrow\downarrow)^c$, for some $1 \leq c < 2k$, or $x = (\uparrow\downarrow)^c\uparrow$, for some $0 \leq c < 2k$. We split this case into sub-cases with respect to the value of x.

Subcase $x = (\uparrow\downarrow)^c$.

$$\begin{aligned}\rightarrow(\uparrow\downarrow)^\omega &= \rightarrow\square^{2c}(\uparrow\downarrow)^\omega \text{ and}\\ \rightarrow(\uparrow\downarrow)^\omega &= \rightarrow(\uparrow\downarrow)^{c-1}\uparrow\downarrow(\uparrow\downarrow)^\omega\\ &= \rightarrow(\uparrow\downarrow)^{c-1}\uparrow\downarrow\downarrow(\uparrow\downarrow)^\omega\\ &= \rightarrow(\uparrow\downarrow)^c(\downarrow\uparrow)^\omega\end{aligned}$$

$$\begin{aligned}\rightarrow(\square^{2c}(\uparrow\downarrow)^\omega \odot (\uparrow\downarrow)^c(\downarrow\uparrow)^\omega) &= \rightarrow(\uparrow\downarrow)^c(\square\square)^\omega\\ &= \rightarrow(\uparrow\downarrow)^c(\square)^\omega\end{aligned}$$

Subcase $x = (\uparrow\downarrow)^c\uparrow$.

$$\begin{aligned}\rightarrow(\uparrow\downarrow)^\omega &= \rightarrow\square^{2c+1}(\uparrow\downarrow)^\omega \text{ and}\\ \rightarrow(\uparrow\downarrow)^\omega &= \rightarrow(\uparrow\downarrow)^c\uparrow\downarrow(\uparrow\downarrow)^\omega\\ &= \rightarrow(\uparrow\downarrow)^c\uparrow(\downarrow\uparrow)^\omega.\end{aligned}$$

For the last time we compute the convolution

$$\begin{aligned}\rightarrow(\square^{2c+1}(\uparrow\downarrow)^\omega \odot (\uparrow\downarrow)^c\uparrow(\downarrow\uparrow)^\omega) &= \rightarrow(\uparrow\downarrow)^c\uparrow(\square\square)^\omega\\ &= \rightarrow(\uparrow\downarrow)^c\uparrow(\square)^\omega\end{aligned}$$

and conclude that also in this case $\rightarrow x\gamma \in F_<$.

□

Taking all together we get that $\mathfrak{d}$ restricted to L' is an injective ω-automatic-presentation of $(\{0,1\}^\omega, <_{lex})$. □

Corollary 3.5 (Kuske [77]). *$(\{0,1\}^\omega, <_{lex})$ is embeddable into every uncountable ω-automatic linear order.*

Another important consequence is that our end-index analysis from Section 3.2.1 can be usefully applied to uncountable structures with a linear order.

Corollary 3.6. *Let $\mathfrak{d}$ be an α-automatic presentation of an uncountable linearly ordered structure $\mathfrak{A}$. Then there is an infinite set $B \subseteq A$ with $I_e(B) = 1$.*

Corollary 3.7. *Let f be an inherently regular function on a linearly ordered advice ω-automatic structure. Then $\mathrm{MIS}_f(n) \in \mathcal{O}(n)$.*

But even without the linear order, we learn something about the encoding of elements from our construction. Note that for defining u, w_0, w_1 we did not make use of the linear order, but only of the equivalence relation $\approx$. Indeed, the semigroup elements found in the previous lemmas for $\approx$ still have all the stated properties independent of the presence of a linear order, which allows us to re-prove the following theorem, already mentioned in [68] in a slightly different context.

Theorem 3.3. *Let $\mathfrak{d} = (\mathcal{A}, \mathcal{A}_\approx, \ldots)$ be a parametrised ω-automatic presentation, which presents an uncountable structure with advice $\alpha \in \Gamma^\omega$. Then there are α-automatic sequences $(w_{0i})_{i\in\mathbb{N}}$, $(w_{1i})_{i\in\mathbb{N}}$ of finite words with*

- $w_{00} = w_{10}$,
- $|w_{0i}| = |w_{1i}|$ *and* $w_{0i} \neq w_{1i}$ *for* $i > 0$,
- $L' := \{w_{j_0 0} w_{j_1 1} w_{j_2 2} \ldots \mid j_0 j_1 j_2 \ldots \in \{0,1\}^\omega\} \subseteq L(\mathcal{A}[\alpha])$, *and*
- $v_0 \not\sim_e v_1 \Rightarrow v_0 \not\approx v_1$ *for all* $v_0, v_1 \in L'$.

Proof. Let $(w_{0i})_{i\in\mathbb{N}}$ and $(w_{1i})_{i\in\mathbb{N}}$ be constructed as before. By Lemma 3.13 $\to (\uparrow)^\omega \notin F_\approx$ and $\to (\uparrow\downarrow)^\omega \notin F_\approx$.

It is easy to calculate that for all $v_0, v_1 \in L'$ with $v_0 \not\sim_e v_1$

$$h_\approx(\alpha \otimes v_0 \otimes v_1) \in \to\{(\uparrow\downarrow)^\omega, (\downarrow\uparrow)^\omega\} \cup \to \{(\uparrow\downarrow)^{<2k}, (\uparrow\downarrow)^{<2k}\uparrow, (\downarrow\uparrow)^{<2k}, (\downarrow\uparrow)^{<2k}\downarrow\}\{\uparrow^\omega, \downarrow^\omega\}.$$

Because of the symmetry of $\approx$, it is sufficient to show that

$$(\{\rightarrow(\uparrow\downarrow)^\omega\}\cup\rightarrow(\uparrow\downarrow)^{<2k}\uparrow^\omega \cup\rightarrow\{\uparrow\downarrow(\uparrow\downarrow)^{<2k-1},(\uparrow\downarrow)^{<2k}\uparrow\}\downarrow^\omega)\cap F_\approx=\emptyset.$$

We use the same strategy as in Lemma 3.15. We already know that $\rightarrow(\uparrow\downarrow\ )^\omega\notin F_\approx$. The other elements are handeled by case distinction.

Case $\rightarrow(\uparrow\downarrow)^n\uparrow^\omega$, $0\leq n<2k$:

$$\rightarrow(\uparrow\downarrow)^n\uparrow^\omega=\!\!\rightarrow(\uparrow\downarrow)^n(\uparrow\Box)^\omega$$
$$\rightarrow(\uparrow\downarrow)^n\uparrow^\omega=\!\!\rightarrow\Box(\uparrow\downarrow)^n(\uparrow\Box)^\omega.$$

But also

$$\rightarrow((\uparrow\downarrow)^n(\uparrow\Box)^\omega)\odot(\Box(\uparrow\downarrow)^n(\uparrow\Box)^\omega)=\!\!\rightarrow\uparrow\Box^{2n}\uparrow^\omega=\!\!\rightarrow\uparrow^\omega\ .$$

Hence, $\rightarrow\uparrow^\omega\notin F_\approx$ implies $\rightarrow(\uparrow\downarrow)^n(\uparrow)^\omega\notin F_\approx$.

Case $\rightarrow(\uparrow\downarrow)^n(\downarrow)^\omega$, $1\leq n<2k$:

Remember, $\rightarrow(\uparrow\downarrow)^n(\downarrow)^\omega\in F_\approx$ if, and only if, $\rightarrow(\downarrow\uparrow)^n(\uparrow)^\omega\in F_\approx$. We compute

$$\rightarrow(\uparrow\downarrow)^n(\downarrow)^\omega=\!\!\rightarrow(\uparrow\downarrow)^n(\downarrow\Box)^\omega,$$
$$\rightarrow((\downarrow\uparrow)^n(\uparrow)^\omega\odot(\uparrow\downarrow)^n(\downarrow\Box)^\omega)=\!\!\rightarrow\Box^{2n}(\Box\uparrow)^\omega=\!\!\rightarrow\uparrow^\omega\ .$$

Again, we conclude from $\rightarrow\uparrow^\omega\notin F_\approx$ that also $\rightarrow(\uparrow\downarrow)^n(\downarrow)^\omega\notin F_\approx$.

The case $\rightarrow(\uparrow\downarrow)^n\uparrow(\downarrow)^\omega$ is completely analogous to the previous one. □

3.2.4 ω-Automatic Functions

We present a consequence of Theorem 3.3 for ω-automatic structures without advice. In this case the sub-presentation, described in the theorem, has an especially simple form because ω-regular relations always contain ultimately periodic words.

Lemma 3.16 (See for instance [63, Lemma 3.10])**.** *Let $R\subseteq\Sigma^\omega\times\Gamma^\omega$ be an ω-regular relation recognised by a Büchi-automaton $\mathcal{A}=(Q,\Sigma\times\Gamma,q_0,\Delta,F)$. Further let $\alpha\in\Sigma^\omega$ be some ultimately periodic word with period length $p\in\mathbb{N}$. Then if $\alpha R\neq\emptyset$ then there is a word $\beta\in\alpha R$ with period length at most $|Q|\cdot p$.*

Corollary 3.8. *Let $\mathfrak{d} = (\mathcal{A}, \mathcal{A}_\approx, \ldots)$ be an ω-automatic presentation of an uncountable structure. Then there are words $u, v_0, v_1 \in \Sigma^+$ with*

- *$|v_0| = |v_1|$ and $v_0 \neq v_1$,*
- *$L' := \{uv_{j_0}v_{j_1}v_{j_2}\ldots \mid j_0j_1j_2\ldots \in \{0,1\}^\omega\} \subseteq L(\mathcal{A})$, and*
- *$w_0 \not\sim_e w_1 \Rightarrow w_0 \not\approx w_1$ for all $w_0, w_1 \in L'$.*

Proof. Let $(w_{0i})_{i\in\mathbb{N}}$ and $(w_{0i})_{i\in\mathbb{N}}$ be the ω-automatic sequences, which are described by Theorem 3.3. Then these two sequences must be ultimately periodic. Let n and k be such that

$$w_{i(n+\ell k)}w_{i(n+\ell k+1)}\cdots w_{i(n+(\ell+1)k-1)} = w_{i(n+(\ell+1)k)}w_{i(n+(\ell+1)k+1)}\cdots w_{i(n+(\ell+2)k-1)}$$

for all $i \in \{0,1\}$ and $\ell \in \mathbb{N}$. Define $u := w_{00}$ and $v_i := w_{in}\cdots w_{i(n+k-1)}$ for $i \in \{0,1\}$. Then u, v_0, and v_1 have the properties as stated. □

We shall now extend our techniques by considering intrinsically ω-regular functions on uncountable ω-automatic structures. Our main technical result shows that there is no ω-automatic structure with a family of intrinsically ω-regular functions of unbounded arity such that the preimage of every element is at most countably infinite for all of these functions. Recall that a function is intrinsically regular, if it is ω-regular in any ω-automatic presentation of $\mathfrak{A}$.

Theorem 3.4. *Let $\mathfrak{A}$ be an uncountable ω-automatic structure. Then there is a $k \in \mathbb{N}$ such that for every intrinsically ω-regular extension of $\mathfrak{A}$ by a $(k+1)$-ary function $f(\overline{x}, y)$ there exist uncountable sets $M \subseteq A^k$ and $N \subseteq A$ with $f(\overline{a}, b) = f(\overline{a}', b)$ for all $\overline{a}, \overline{a}' \in M, b \in N$.*

Proof. Fix an ω-automatic presentation $(\mathfrak{d}, \pi)$ over the alphabet $\{0,1\}$. Since $\mathfrak{A}$ is uncountable, we can apply Corollary 3.8 and obtain a language $L' = w\{v_0, v_1\}^\omega \subseteq L$ such that $|v_0| = |v_1|$ and for all $\alpha, \beta \in L'$ it holds that $\alpha \not\sim_e \beta$ implies $\alpha \not\approx \beta$. Set $k = |v_0| + 1$. Let $\mathcal{A}_f = (Q, \{0,1\}^{k+1}, q_0, \Delta, F)$ be the automaton that recognises f in $(\mathfrak{d}, \pi)$ and let p denote the number of possible transition profiles of $\mathcal{A}_f$ i.e. $p = |\{\Delta_w : w \in (\{0,1\}^{k+2})^*\}|$, where

$$\Delta_w = \{(p, q, P) \in Q\times Q\times\mathcal{P}(Q) \mid \Delta(p, w) = q, P = \{q \mid \exists v \preceq w : \Delta(p, v) = w\}\}.$$

We proceed as follows: first we define two languages L_p and L_{id} of ultimately periodic words. In L_p and L_{id} respectively, we will find suitable encodings of

pairs of distinct tuples which can be combined to encodings of uncountably many pairwise distinct tuples. These will then be the encodings for our sets M and N. To ensure that the combinations of tuples from L_p and the combinations of tuples from L_{id} do not interfere with each other, we define these languages in such a way that the set of positions where the words of L_p "encode their information" is disjoint from the set of positions where this is the case for the words in L_{id}. More precisely we are going to guarantee that whenever two words of L_p differ at a given position $i \in \mathbb{N}$, then all pairs of words in L_{id} have the same "dummy letter" at position i, and vice versa. We set

$$L_p := \{wv^\omega : v \in \{v_0, v_1\}^p\overline{\square}\} \text{ and } L_{id} := \{wv^\omega : v \in \underline{\square}\{v_0, v_1\}^{2kp+p}\}$$

where $\overline{\square} := (v_0)^{2kp+p}$ and $\underline{\square} := (v_0)^p$. Then $L_p, L_{id} \subseteq L'$ and every word in $L_p \cup L_{id}$ represents a distinct element from the domain A, i.e. for every pair of words $\alpha, \beta \in L_p \cup L_{id}$ we have $\alpha \not\approx \beta$ if $\alpha \neq \beta$. Furthermore, every word $\alpha \in L_p \cup L_{id}$ is completely determined by the unique finite word $v[\alpha] \in \{v_0, v_1\}^p \cup \{v_0, v_1\}^{2kp+p}$ with $\alpha = w(v[\alpha]\overline{\square})^\omega$ or $\alpha = w(\underline{\square}v[\alpha])^\omega$, respectively. We extend this notation to tuples of words $\overline{\alpha} \in (L_p)^k$ as words $\overline{v}[\overline{\alpha}] \in (\{v_0, v_1\}^{2kp+p})^k$ in the obvious way. In particular we have $|L_p| = 2^p$ and $|L_{id}| = 2^{2kp+p}$.

Let us now consider $(k+1)$-tuples of the form $(\overline{\alpha}, \beta)$ with $\overline{\alpha} \in (L_p)^k$ and $\beta \in L_{id}$ as inputs of the automaton $\mathcal{A}_f$. We note that all words in $(L_p)^k \times L_{id}$ are periodic from position $|w|$ onwards and that the length of their periods divides $r := (2k+1)p|v_0|$. Both values $|w|$ and r are independent of the particular word in $(L_p)^k \times L_{id}$.

By Lemma 3.16 we can infer that for every $\overline{\alpha} \in (L_p)^k, \beta \in L_{id}$ the element $f(\pi(\overline{\alpha}), \pi(\beta))$ has an ultimately periodic encoding $\gamma[\overline{\alpha}, \beta]$ with the following properties:

- the length of the non-periodic prefix of $\gamma[\overline{\alpha}, \beta]$ is $s = |w| + c \cdot r$ for $c \in \mathbb{N}$, and
- the length of the period of $\gamma[\overline{\alpha}, \beta]$ is $\ell = d \cdot r$ for some $d \in \mathbb{N}$, and
- both constants c and d are independent of the particular choice of $\overline{\alpha}, \beta$.

We illustrate the situation in Figure 3.2.

By the choice of k, the number of tuples in $(L_p)^k$ is $2^{pk} \geq 2^{p|v_0|} \cdot 2^p > 2^{p|v_0|} \cdot p$, hence the number of tuples in $(L_p)^k$ exceeds the number of tuples of words in $\Sigma^{p|v_0|}$ and transition profiles of the automaton $\mathcal{A}_f$. Therefore, for every $\beta \in L_{id}$ there exist two distinct words $\overline{\alpha}, \overline{\alpha}' \in (L_p)^k$ such that for some $\delta \in \Sigma^{p|v_0|}$

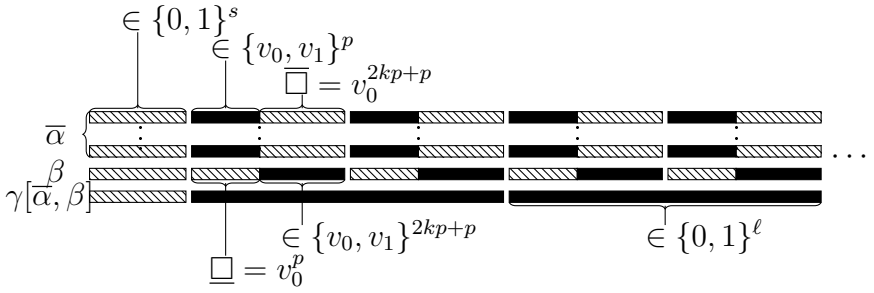


Figure 3.2: Construction of the Inputs for $\mathcal{A}_f$

- the *finite* words $(\overline{v}[\overline{\alpha}], \underline{\square}, \delta)$ and $(\overline{v}[\overline{\alpha}'], \underline{\square}, \delta)$ occur at the positions $s + i\ell$ for all $i \in \mathbb{N}$ in $(\overline{\alpha}, \beta, \gamma[\overline{\alpha}, \beta])$ and $(\overline{\alpha}', \beta, \gamma[\overline{\alpha}', \beta])$, respectively, and
- these infixes $(\overline{v}[\overline{\alpha}], \underline{\square}, \delta)$ and $(\overline{v}[\overline{\alpha}'], \underline{\square}, \delta)$ have the same $\mathcal{A}_f$-transition profiles.

For every $\beta \in L_{id}$ we fix such a pair $(\overline{\alpha}, \overline{\alpha}')$. Furthermore, we recall that the number of words in L_{id} is 2^{2kp+p}. Since the number of pairs of $(L_p)^k$-tuples is 2^{2pk} there are at least $2^{2kp+p}/2^{2pk} = 2^p > p$ words $\beta \in L_{id}$ to which the same pair $(\overline{\alpha}, \overline{\alpha}')$ is assigned. Hence, we can also find two distinct $\beta, \beta' \in L_{id}$ with this property such that for some $\lambda, \lambda' \in \Sigma^{(2kp+p)|v_0|}$ the *finite* words $\left(\overline{\square}^k, v[\beta], \lambda\right)$ and $\left(\overline{\square}^k, v[\beta'], \lambda'\right)$ have the same $\mathcal{A}_f$-transition profile, and occur at positions $h_i = (s + |v_0|p) + i\ell, i \in \mathbb{N}$ in $(\overline{\alpha}, \beta, \gamma[\overline{\alpha}, \beta])$ and $(\overline{\alpha}, \beta', \gamma[\overline{\alpha}, \beta'])$, respectively.

The claim follows by examining the properties of $\overline{\alpha}, \overline{\alpha}' \in (L_p)^k$ and $\beta, \beta' \in L_{id}$. Let us consider the input $(\overline{\alpha}, \beta, \gamma[\overline{\alpha}, \beta])$ which is accepted by $\mathcal{A}_f$. By the properties stated above we can replace the infix $(\overline{v}[\overline{\alpha}], \underline{\square}, \delta)$ in $(\overline{\alpha}, \beta, \gamma[\overline{\alpha}, \beta])$ by $(\overline{v}[\overline{\alpha}'], \underline{\square}, \delta)$ at infinitely many positions. In this way we can obtain an uncountably infinite set $L_M \subseteq (L')^k$ with

- $\overline{\alpha}_0 \not\approx \overline{\alpha}_1$ if $\overline{\alpha}_0 \neq \overline{\alpha}_1$ (i.e. $\overline{\alpha}_0$ and $\overline{\alpha}_1$ encode different elements from A^k) for all $\overline{\alpha}_0, \overline{\alpha}_1 \in L_M$, and
- $(\overline{\alpha}_0, \beta, \gamma[\overline{\alpha}, \beta])$ is accepted by the automaton $\mathcal{A}_f$ for all $\overline{\alpha}_0 \in L_M$.

This is done in the following way: choose a set $X \subseteq \{0,1\}^\omega$ of the size 2^ω such that $x \not\sim_e y$ for all $x, y \in X$. This is possible since $\sim_e$ partitions $\{0,1\}^\omega$ into

countable equivalence classes and $\{0,1\}^\omega$ has continuum many elements. For every $x \in X$ we define $\overline{\alpha}_x$ by

$$\overline{\alpha}_x[0,s) := \overline{\alpha}[0,s)$$
$$\overline{\alpha}_x[h_i,h_{i+1}) := \begin{cases} \overline{\alpha}[h_i,h_{i+1}) & \text{if } x[i]=0 \\ \overline{\alpha}'[h_i,h_{i+1}) & \text{if } x[i]=1. \end{cases}$$

We then define $L_M := \{\overline{\alpha}_x : x \in X\}$. It is easy to check that L_M has the claimed properties: $\overline{\alpha}_x, \overline{\alpha}_{x'} \subseteq L'$ and $\overline{\alpha}_x \not\sim_e \overline{\alpha}_{x'}$ for all $\overline{\alpha}_x \neq \overline{\alpha}_{x'} \in L_M$ and therefore $\overline{\alpha}_x \not\approx \overline{\alpha}_{x'}$. Since $\Delta_{((\overline{v}[\overline{\alpha}],\square,\delta))} = \Delta_{((\overline{v}[\overline{\alpha}'],\square,\delta))}$, $(\overline{\alpha}_x, \beta, \gamma[\overline{\alpha},\beta])$ is accepted by $\mathcal{A}_f$. In particular, we can still interchange the infixes $(\overline{\square}^k, v[\beta], \lambda)$ and $(\overline{\square}^k, v[\beta'], \lambda')$ in every input $(\overline{\alpha}_x, \beta, \gamma[\overline{\alpha},\beta])$ in any way without affecting the acceptance behaviour of $\mathcal{A}_f$. We obtain a set $L_N \subseteq L'$ of uncountably many different tuples with the following properties

- $\beta \not\approx \beta'$ if $\beta \neq \beta'$ (i.e. β and β' encode different elements from A), and
- for every $\beta \in L_N$ there exists $\gamma \in \Sigma^\omega$ such that $(\overline{\alpha}, \beta, \gamma)$ is accepted by the automaton $\mathcal{A}_f$ for all for all $\overline{\alpha} \in L_M$.

Altogether for $M = \pi(L_M) \subseteq A^k$ and $N = \pi(L_N) \subseteq A$, we have $f(m,n) = f(m',n)$ for all $m, m' \in M$ and $n \in N$. Since both sets M and N are uncountable, the claim follows.

□

Sometimes it is convenient to apply Theorem 3.4 in the following simplified version.

Corollary 3.9. *Let $\mathfrak{A} = (A, R_1, \ldots, R_n)$ be an uncountable ω-automatic structure. Then there is an $\ell \in \mathbb{N}$ such that for every definable ℓ-ary function $f(\overline{x})$ there is an uncountable set $M \subseteq A^\ell$ with $f(\overline{a}) = f(\overline{a}')$ for all $\overline{a}, \overline{a}' \in M$.*

Proof. Let k be the constant from Theorem 3.4. We set $\ell = k+1$. Let f be an ℓ-ary function that is definable in $\mathfrak{A}$. Then by Theorem 3.4 there exist uncountably infinite sets $M' \subseteq A^k, N' \subseteq A$ with $f(\overline{a}, b) = f(\overline{a}', b)$ for all $\overline{a}, \overline{a}' \in M', b \in N'$. Hence, we can simply choose any of the uncountably infinite sets $M_b = \{(\overline{a}, b) : \overline{a} \in M'\}$ for some $b \in N'$ to satisfy the claim. □

3.3 Structures Without Automatic Presentation

In this last section we apply our results to identify structures, which cannot have an ω-automatic presentation (with advice). We start with the main theorems of this chapter.

3.3.1 Integral Domains and the Field of Reals

To illustrate the applications of our results, consider the case of an integral domain. Recall that an integral domain is a commutative ring that has no zero divisors.

Lemma 3.17. *Let $\mathfrak{d}$ be an α-automatic presentation of an infinite integral domain $\mathfrak{A}$ for some $\alpha \in \Gamma^\omega$. Then there is no infinite set $B \subseteq A$ with finite index $I_e(B)$ in $\mathfrak{d}$.*

Proof. Suppose otherwise. Then we could extend $\mathfrak{d}$ to a presentation $\mathfrak{d}'$ of the extension of $\mathfrak{A}$ by the first-order definable function $f(x, a, b) = ax + b$, which defines a linear polynomial for every fixed $a, b \in A$. The structure $\mathfrak{A}$ is an integral domain and hence for two distinct pairs $(a, b), (a', b')$ the equation $ax + b = a'x + b'$ has as most one solution. This implies that for every finite set $B \subseteq A$ and every set X with $|X| > |B|^4$ there must be an $x \in X$ such that $|f(x, B, B)| = |B|^2$. Otherwise we could fix for every $x \in X$ a tuple $(a_x, b_x, a'_x, b'_x) \in B^4$ with $(a_x, b_x) \neq (a'_x, b'_x)$ and x is a solution for the equation $a_x x + b_x = a'_x x + b'_x$. But $|X| > |B^4|$ and consequently there would be $x \neq y \in X$ with $(a_x, b_x, a'_x, b'_x) = (a_y, b_y, a'_y, b'_y)$, that is x and y would be two distinct solutions of the same equation, which is impossible.

However, $(\mathfrak{A}, f)$ contains an infinite set with end-index one in $\mathfrak{d}'$ and, consequently, there are arbitrarily large finite sets $B \subseteq A$ with $I(B) = 1$. According to Corollary 3.4 there are arbitrarily large finite sets $B \subseteq A$ with $|f(x, B, B)| \leq c \cdot |B|$ for some constant c and all $x \in A$, which contradicts the above calculations. □

Corollary 3.10. *No infinite integral domain has an injective ω-automatic presentation with advice.*

Proof. If the presentation $\mathfrak{d}$ is injective then Lemma 3.4 gives us directly an infinite set with end-index 1 in $\mathfrak{d}$. □

With all that we have learned so far, Theorem 3.1 also becomes an easy corollary.

Theorem 3.1. *The field of reals is not ω-automatic with advice.*

Proof. The linear order on the reals is first order definable in $(\mathbb{R}, +, \cdot)$ and therefore every presentation $\mathfrak{d}$ of $(\mathbb{R}, +, \cdot)$ would induce an infinite set with end-index 1 in contradiction to Lemma 3.17. □

Of course we would like to generalise this result to all infinite integral domains. But for now we are only able to generalise the theorem for ω-automatic presentations without advice.

Theorem 3.2. *An integral domain is ω-automatic if, and only if, it is finite.*

Proof. One direction is trivial since all finite structures are ω-automatic. For the other direction, we recall from [73] that the (finite word) automatic integral domains are exactly the finite ones. Hence, by [68], there are no countably infinite ω-automatic integral domains.

Suppose now that $\mathfrak{A} = (A, +, \cdot)$ is an uncountable ω-automatic integral domain. Fix a presentation of $\mathfrak{A}$ and let k be the constant from Theorem 3.4 with respect to this presentation. Consider the family of polynomials of degree k. This family can be presented by the function $(a_0, \ldots, a_k, x) \mapsto \sum_{i=0}^{k} a_i x^i$ with k parameters $a_0, \ldots, a_{k-1} \in A$ and input x. It is obvious that this function is first-order definable.

On one hand, it is a well-known fact from algebra that, on an integral domain, two different polynomials of degree at most $k-1$ agree on at most $k-1$ inputs. On the other hand, $\mathfrak{A}$ is uncountable and therefore Theorem 3.4 implies that there are $\overline{a} \neq \overline{b} \in A^k$ such that $\sum_{i=0}^{k-1} a_i x^i = \sum_{i=0}^{k-1} b_i x^i$ for even uncountably many $x \in A$. □

We close this section with a few applications concerning other types of algebraic structures.

Theorem 3.5. *There is no infinite ω-automatic structure with pairing function.*

Proof. Towards a contradiction, suppose there is an ω-automatic structure $\mathfrak{A}$ in which a pairing function f is definable. First we note that $\mathfrak{A}$ cannot be countable. Otherwise, by [68], $\mathfrak{A}$ would have an injective presentation. But

$\mathrm{MIS}_f(n) = n^2$ in contradiction to Lemma 3.9. Therefore $\mathfrak{A}$ must be uncountable. In this case we obtain a contradiction to Corollary 3.9 by constructing a family of definable injective functions of unbounded arity. We let

$$f_1(x,y) := f(x,y) \text{ and}$$
$$f_{n+1}(x_1,\ldots,x_{2^n},y_1,\ldots,y_{2^n}) := f(f_n(x_1,\ldots,x_{2^n}),f_n(y_1,\ldots,y_{2^n})).$$

Then f_n is injective and FO-definable in $\mathfrak{A}$ for every $n \geq 1$. □

Another example to which we can apply our techniques are lattices.

Lemma 3.18. *Every uncountable ω-automatic lattice contains an element such that uncountably many elements are smaller than this element, and contains an element such that uncountably many elements are greater than this element.*

Proof. Let $\mathfrak{A} = (A, <, \wedge, \vee)$ be an uncountable ω-automatic lattice. Fix an ω-automatic presentation and let c be the constant from Lemma 3.9. We consider the definable functions $f(x_1,\ldots,x_c) := \bigwedge_{1\leq i\leq c} x_i$ and $g(x_1,\ldots,x_c) := \bigvee_{1\leq i\leq c} x_i$. By Corollary 3.9 there must be elements $a, b \in A$ such that $f^{-1}(a)$ and $g^{-1}(b)$ are uncountable. But this is only possible if the sets

$$\{x : x \text{ appears in some } (x_1,\ldots,x_c) \in f^{-1}(a)\} \subseteq \{x : x \leq a\}$$

and

$$\{x : x \text{ appears in some } (x_1,\ldots,x_c) \in g^{-1}(b)\} \subseteq \{x : x \geq b\}$$

are uncountable. □

As a consequence we can reprove Kuske's theorem that no uncountable ordinal is ω-automatic.

Theorem 3.6 (Kuske [77])**.** *There is no uncountable ω-automatic ordinal.*

Proof. First note that Lemma 3.18 directly implies that ω_1, the first uncountable ordinal, is not ω-automatic. Every ordinal is a lattice ($\wedge$ and $\vee$ can be defined) and for every element of ω_1 the number of elements below it is countable. But this implies that no larger ordinal α can be automatic either, since ω_1 is definable in all of these ordinals by the formula $\varphi(x) := \exists^{\leq\aleph_0} y(y < x)$. □

3.3.2 Countable Structures

This section is devoted to advice automatic presentations of countable structures. It turns out that our findings of the Sections 3.2.1 and 3.2.2 are especially well suited for the analysis of countable structures. The reason for this is that countable advice automatic structures have presentations where the end-index of the whole domain is one. We say that a parametrised ω-automatic presentation has a **finite word encoding** over the advice α, if $\Sigma = \Sigma' \uplus \{\Box\}$ and $L(\mathcal{A}[\alpha]) \subseteq \Sigma'^*\{\Box^\omega\}$.

Theorem 3.7. *A countable structure $\mathfrak{A}$ has an ω-automatic presentation $\mathfrak{d} = (\mathcal{A}, \mathcal{A}_\approx, (\mathcal{A}_R)_{R\in\tau})$ with advice α, if, and only if, it has an injective ω-automatic presentation $\mathfrak{d}' = (\mathcal{A}', (\mathcal{A}'_R)_{R\in\tau})$ with advice α over finite words. Moreover, $\mathfrak{d}'$ can be effectively constructed from $\mathfrak{d}$.*

Proof. Since $\mathfrak{A}$ is countable, it follows from [68, Proposition 3.1] that there is a constant c computable from $\mathfrak{d}$ such that $I_e(A) \leq c$. Hence, the formula

$$\varphi(x_1, \ldots, x_c) := \forall y(\bigvee_{1\leq i\leq c} \exists z(z \approx y \wedge z \sim_e x_i))$$

is satisfiable in the structure $(\mathcal{S}_\approx(\mathfrak{d}[\alpha]), \sim_e)$. This in turn means that φ defines an ω-automatic relation $R_\varphi \subseteq \Gamma^\omega \times (\Sigma^\omega)^c$ with $\alpha R \neq \emptyset$. By the Uniformisation Theorem for ω-automatic relations [24], there is an ω-automatic function $f_R : R(\Sigma^\omega)^c \to (\Sigma^\omega)^c$ with $f_R(\alpha) \in \alpha R$ for all $\alpha \in R(\Sigma^\omega)^c$.

We are now prepared to construct $\mathfrak{d}'$. Intuitively we are going to use f_R to pick $x_1, \ldots, x_c$ from the original representation such that all elements are encoded in $L := L(\mathcal{A}[\alpha]) \cap \bigcup_{1\leq i\leq c}[x_i]_{\sim_e}$. Then for every $y \in L$ we just cut y from the point where it coincides with some x_i and annotate the resulting string with the respective end-class. More formally we first expand the alphabet Σ by new symbols $\{1, \ldots, c\}$. The domain automaton is constructed from the formula:

$$\varphi_A(\alpha, x) := \exists y \in \Sigma^*, i \in \{1, \ldots, c\}(x = yi\Box^\omega \wedge (y(f_R(\alpha)_i[|y|, \infty)) \in L(\mathcal{A}[\alpha])).$$

Similarly for $S \in \{\approx\} \cup \tau$ we construct an automaton by the formula

$$\varphi_S(\alpha, y_1 i_1 \Box^\omega, \ldots, y_k i_k \Box^\omega) := S(y_1(f_R(\alpha)_{i_1}[|y_1|, \infty)), \ldots, y_k(f_R(\alpha)_{i_k}[|y_k|, \infty))).$$

The corresponding relations can easily be implemented by Muller automata. As always, the resulting presentation can be made injective by taking the length-lexicographic smallest representative of any $\approx$-class. □

For ω-automatic structures, it has been shown that Theorem 3.1 remains true for the extension FOC of first-order logic by the quantifiers $\exists^{\infty}/\exists^{>\aleph_0}/\exists^{(k,m)}$, meaning "there exists infinitely / uncountably / k mod m many". Theorem 3.1 thus also remains valid for FOC over automatic structures with advice.

The *k-Ramsey quantifier* $\exists^{\text{k-ram}}$ is defined by $\mathfrak{A} \models \exists^{\text{k-ram}} \overline{x} \varphi(\overline{x}, \overline{c})$, iff there is an infinite $X \subseteq A$ so that $\mathfrak{A} \models \varphi(a_1, \ldots, a_k, \overline{c})$ for all pairwise different $a_1, \ldots, a_k \in X$, where $\varphi(\overline{x}, \overline{p})$ is a τ-formula with $|\overline{x}| = k$.

Lemma 3.19. *Let $S \subseteq (\Sigma^*)^{k+l}$ be a regular relation with advice α and $P = \{\overline{p} \in (\Sigma^*)^l \mid \exists^{\text{k-ram}} \overline{x} S \overline{x} \overline{p}\}$. There are infinite sets $(A_{\overline{p}})_{\overline{p} \in P}$ with $(a_1, \ldots, a_k, \overline{p}) \in S$ for all pairwise different $a_1, \ldots, a_k \in A_{\overline{p}}$ so that $A = \{(a, \overline{p}) \mid a \in A_{\overline{p}}, \overline{p} \in P\}$ is regular with advice α.*

Proof. Consider ω-words of the form $s \otimes t$ with $s = s_0 s_1 \ldots$ and $t = t_0 t_1 \ldots$ such that $|s_i| = |t_i|$ and $s_i \in \Sigma^* \{\dot{a} \mid a \in \Sigma\}, t_i \in \Sigma^*$. Say that a word $x \in \Sigma^*$ is *on* $s \otimes t$, if there is an i so that $x = \pi(s_0 \ldots s_i t_{i+1})$, where π is the projection of the alphabet, which maps $\dot{a}$ to a for all $a \in \Sigma$ and leaves all other symbols unchanged. It is not hard to construct a Muller automaton $\mathcal{B}$, so that $L(\mathcal{B})[\alpha]$ consists of those $s \otimes t \otimes \overline{p}$, so that $(x_1, \ldots, x_k, \overline{p}) \in S$ for all $(x_1, \ldots, x_k)$ such that $x_1, \ldots, x_k$ are pairwise different words on $s \otimes t$. Applying the Uniformisation Theorem to $\mathcal{B}$, we get a Muller automaton $\mathcal{B}'$ so that for every $\overline{p} \in P$ there is at most one $s \otimes t$ with $s \otimes t \otimes p \otimes \alpha \in L(\mathcal{B}')$. Using $\mathcal{B}'$ we can easily construct another Muller automaton that recognizes with advice α the set of all words that are on this unique $s \otimes t$.

It remains to show that for each $\overline{p} \in P$ there is at least one $s \otimes t$ with $s \otimes t \otimes p \in L(\mathcal{B})[\alpha]$. Let $\overline{p} \in P$ and $X \subseteq \Sigma^*$ be an infinite set with $(x_1, \ldots, x_k, \overline{p}) \in S$ for all pairwise distinct $x_1, \ldots, x_k \in X$. Consider the subtree of $\mathfrak{T}_{|\Sigma|}$ that is generated by the prefix-closure of X. According to König's Lemma there is an infinite path γ in this tree so that from every node on the path a node in X is reachable. We define words $s_i, t_i \in \Sigma^*$ with the following properties:

1. $|s_i| = |t_i|$ for all $i \in \mathbb{N}$
2. $s_0 \ldots s_i$ is a prefix of γ for all $i \in \mathbb{N}$.
3. $s_0 \ldots s_i t_{i+1} \in X$ for all $i \in \mathbb{N}$.

For this define recursively $s_0, t_0 := \varepsilon$, t_{i+1} as a shortest path from $s_0 \ldots s_i$ to a node in X and s_{i+1} as the path of length $|t_{i+1}|$, so that $s_0 \ldots s_{i+1}$ remains a prefix of γ for all $i \in \mathbb{N}$.

□

Corollary 3.11. *If $\alpha \in \Gamma^\omega$ has a decidable* MSO*-theory then the* FOC$+\exists^{\text{k-ram}}$*-theory of any countable α-automatic structure with advice α is decidable.*

Richness and Limitation Revisited While it is known that the countable ω-automatic structures are exactly the automatic structures [68], Example 3.1 describes an important and natural example of a structure that is not automatic but α-automatic for some suitable advice $\alpha \in \Gamma^\omega$. In the light of this fact, there is a natural need to reexamine the classical examples of non-automatic structures with a decidable theory. Indeed, an inspection of the corresponding proofs reveals that most of them rely on applications of the pumping lemma for finite automata. This is, however, no longer an option in the presence of an advice string because such an application of the pumping lemma would also alter the advice string. Note that the only interesting parameters α are the non ultimately periodic ones. But for such α there is no factorisation $\alpha = uw\beta$ such that $\alpha = uw^n\beta$ with $w \neq \varepsilon$ and $n \neq 1$. As mentioned before, the methods developed in this chapter provide a suitable replacement for the pumping lemma, which leads to alternative non-automaticity proofs for many of the known examples of non-automatic structures, which also apply in the presence of an advice. In this sence all of these structures are far from being automatic.

Corollary 3.12. *No countably infinite integral domain $\mathfrak{A}$ is ω-automatic with advice.*

Proof. Countable advice automatic structures have presentations with finite word encodings, as stated by Theorem 3.7. For such presentations $I_e(A) = 1$. □

In the following, we want to give a collection of structures which cannot be substructure of any countable structure advice ω-automatic presentation.

Corollary 3.13. *Let $\mathfrak{A}$ be a countable advice ω-automatic structure with binary function f. Then there is a constant c such that for every substructure $\mathfrak{B} \subseteq \mathfrak{A}$ and every finite set $C \subseteq B$ there is a finite set $D \subseteq B$ such that $|f(C, D)| \leq c \cdot |D|$.*

Proof. This is a direct consequence of Lemma 3.4. First, because $\mathfrak{A}$ is countable there is an advice $\alpha \in \Gamma^\omega$ and an α-automatic presentation $\mathfrak{d}$ such that $I(A) = 1$ with respect to $\mathfrak{d}$. By Corollary 3.4 there is a constant c such that for every finite set $C \subseteq B$ there is a finite set $D \supseteq C$ with $c \cdot |D| \geq |f(D, D)| \geq |f(C, D)|$. □

Lemma 3.20. *No countable advice automatic structure contains the free semigroup with two generators as substructure.*

Proof. Suppose $\mathfrak{A} = (A, \cdot)$ contains the free semigroup $(\{a, b\}^+, \cdot)$ over the generators a, b. Consider the sets $B_n = \{a, b\}^n$ for $n \in \mathbb{N}$. For every finite set $C \subseteq \{a, b\}^+$ the product $B_n \cdot C$ has size $|B_n| \cdot |C| = 2^n \cdot |C|$ because for every $v \in \{a, b\}^n$ the mapping $f_v : c \mapsto vc$ is obviously injective and also for $v \neq v'$ it is clear that $f_v(c) \neq f_{v'}(c')$ for all $c, c' \in C$. Therefore $\mathfrak{A}$ can not be advice automatic since this clearly violates the property given in Corollary 3.13. □

Corollary 3.14. *The free group with at least two generators is not advice automatic.*

Next we will show that the same holds for $(\mathbb{N}, \cdot)$. For this last application recall Freiman's Theorem. Let $(G, +)$ be an abelian group. A **generalised arithmetic progression of rank** d is a set $P = \{a_0 + \sum_{i=1}^{d} z_i a_i \mid 0 \leq z_i \leq k_i\}$ for some $a_0, \ldots, a_d \in G$, $k_1, \ldots, k_d \in \mathbb{N}$.

Theorem 3.8 (Freiman [46])**.** *Let G be a torsion-free abelian group. For every constant $c > 0$ there are $k, d \in \mathbb{N}$ such that for all subsets $A \subseteq G$ with $|A + A| \leq c|A|$ there is a generalized arithmetic progression P of rank d that contains A with $|P| \leq k \cdot |A|$.*

Theorem 3.9. *The structure $(\mathbb{N}, \cdot)$ is not a substructure of a countable advice automatic structure.*

Proof. Suppose $(\mathbb{N}, \cdot) \subseteq \mathfrak{A}$. We want to apply Freiman's Theorem to subsets of $(\mathbb{N}, \cdot)$. For this purpose we consider the embedding ι from $(\mathbb{N}, \cdot)$ to $\mathbb{Z}^{<\omega} := \oplus_{n \in \mathbb{N}}(\mathbb{Z}, +)$ given by $\iota(p_1^{n_1} p_2^{n_2} \cdots p_m^{n_m}) = (n_1, n_2, \ldots, n_m, 0, 0, \ldots)$, where $\{p_1, p_2, p_3, \ldots\} = \mathbb{P}$ is the canonical enumeration of all primes. The application of Freiman's Theorem to $\mathbb{Z}^{<\omega}$ yields a function $k(r)$ such that every finite set $X \subseteq \mathbb{Z}^{<\omega}$ with $|X + X| \leq r \cdot |X|$ is is contained in a subgroup with rank at most $k(r)$. Consider the sets $\mathbb{P}_n := \{p_1, \ldots, p_n\}$. The

group generated by $\iota(\mathbb{P})$ is isomorphic to $\mathbb{Z}^n$. Therefore we can conclude that if $n > k(r)$ then $\text{rank}(\langle \iota(X) \rangle) > k(r)$ for every finite set $X \subseteq \mathbb{N}$ with $\mathbb{P}_n \subseteq X$, because $\mathbb{Z}^n \cong \langle \iota(\mathbb{P}_n) \rangle \leq \langle \iota(X) \rangle$ and $\text{rank}(\mathbb{Z}^n) = n > k(r)$. Here we use that $\text{rank}(G') \leq \text{rank}(G)$ for every subgroup G' of a torsion free abelian group G. Using the contraposition of Freiman's theorem we get that $|\iota(X) + \iota(X)| > r \cdot |\iota(X)|$ and hence $|X \cdot X| > r \cdot |X|$. It follows that $\mathfrak{A}$ can not be ω-automatic with any advice because Corollary 3.4 is not fulfilled for the sets $\mathbb{P}_n$. □

Corollary 3.15. *The following structures have no ω-automatic presentation with advice:*

- $(\mathbb{N}, \cdot)$,
- $(\mathbb{Q}, \cdot)$, *and*
- *any countable torsion-free abelian group of infinite rank.*

Note that Theorem 3.9 does no longer hold if one considers uncountable structures. For instance, the multiplication over the reals $(\mathbb{R}, \cdot)$ is ω-automatic and clearly contains $(\mathbb{N}, \cdot)$ as substructure. This is possible because an ω-automatic presentation of $(\mathbb{R}, \cdot)$ can and also must scatter the encodings of the natural numbers across infinitely many end-classes.

3.4 Discussion

In this chapter we developed an algebraical and combinatorial toolbox, which allows us to show that certain structures do not have an ω-automatic presentation with advice. As stated before, one of our higher-order motivations is to become a broader understanding of set-interpretations by successively enriching the structures to which the interpretation is applied. A good benchmark for our progress is that we could give a negative answer to a weakened version of a question by Rabin. The field of reals is not set-interpretable in $\mathfrak{N}$. We showed that this is not the case, even if we consider expansions with arbitrary unary predicates. However the original question remains unsolved.

Open Problem 3.1. *Is the field of reals set-interpretable in $\mathfrak{T}^\omega$ or, equivalently, is the field of reals ω-tree-automatic?*

While we do not expect that it is, our methods still seem to be insufficient to settle this problem. The fact that this question was first raised in 1968 [87], demonstrates how challenging these kind of questions tend to be.

Working with automatic presentations has proven to be a powerful methodological approach. However, in order to make the theory more independent of specific structures, it would be desirable to have a framework in which the characterisation of MSO is even more uniform. An interesting approach towards such a framework was recently proposed by Bojańczyk [16]. He describes the concept of regularity in the language of category theory. It turns out that many theorems which have their respective variants for almost all kinds of automata models can be proven uniformly in this setting. Most interestingly MSO appears in this context as some kind of algebra, similar to the ω-semigoups, which we have already worked with. It would therefore be very interesting to see to which extend concepts like set-interpretations can be studied in this abstract framework.

Another possibility is to consider hierarchies of structures, which are generated by operations that go along well with MSO. Probably the most prominent such hierarchy is the Caucal Hierarchy [23]. It is constructed by iterating the application of unfolding and reverse rational mappings, starting from the class of all finite graphs. Most notable, all structures in the Caucal Hierarchy have a decidable MSO-theory. Further, one could hope that the inductive definition also allows some kind of uniform handling of all structures in this hierarchy. One should, however, observe that the structures $\mathfrak{N}$ and $\mathfrak{T}^\omega$ are already in the first level of the Caucal Hierarchy. This means that in order to make progress along this route we probably still have to be able to analyse set-interpretation in $\mathfrak{T}^\omega$. Nevertheless, Colcombet and Löding considered finite set-interpretation on structures of the Caucal hierarchy and coined the term higher order tree-automatic structures. They were able to show that the corresponding hierarchy of interpretable structures is strict [25].

In summary, although our understanding has significantly increased over the last decades, there is still an infinite path (or even infinite tree of possibilities) ahead. Hence, this branch of algorithmic model theory offers exciting possibilities for future research.

4 Uniformly Automatic Classes

In the previous chapters we investigated which structures can (or cannot) be presented by automata that have access to a fixed advice. As the reader might have already noticed, the advice is not part of the finite presentation itself, although the underlying advice is essential for the mechanics of the presentation (nevertheless, in order to obtain an effective decision procedure, one needs an algorithm that decides the MSO-theory of the advice string). Henceforth, the same presentation is capable of presenting several structures by changing the underlying advice. If one considers a whole set of possible advices, we naturally obtain a presentation of a class of structures. Let us formulate this idea more precisely.

Definition 4.1. Let $\mathfrak{c}$ be a parametrised (ω-)[tree-]automatic presentation and let $P \subseteq \Gamma^*$ ($P \subseteq \Gamma^\omega$, $P \subseteq T_\Gamma$, or $P \subseteq T_\Gamma^\omega$, respectively) be a set of advices. The tuple $(P, \mathfrak{c})$ is a *uniform (ω-)[tree-]automatic* presentation of a class $\mathcal{C}$ if for all $\mathfrak{A} \in \mathcal{C}$ there is an $\alpha \in P$ with $\mathcal{S}(\mathfrak{c}[\alpha]) \cong \mathfrak{A}$ and vice versa.

A class of structures $\mathcal{C}$ is *uniformly (ω-)[tree-]automatic*, if there exists an uniform (ω-)[tree-]automatic presentation of $\mathcal{C}$. We denote by (ω-)[tree-]AutCl the class of all uniformly (ω-)[tree-]automatic classes.

In order to obtain effective decision procedures for the first-order or monadic second-order theory of presentable classes, we need to refine the notion of uniformly automatic with respect to the complexity of the advice set P.

Definition 4.2. Let $\mathcal{C}$ be a uniformly (ω-)[tree-]automatic class. We say that

- $\mathcal{C}$ is strongly (ω-)[tree-]automatic ($\mathcal{C} \in$ (ω-)[tree-]SAutCl), if it has an (ω-)[tree-]automatic presentation over an advice set P with decidable MSO-theory.
- $\mathcal{C}$ is regularly (ω-)[tree-]automatic ($\mathcal{C} \in$ (ω-)[tree-]RAutCl), if it has an (ω-)[tree-]automatic presentation over a regular advice set P. If the advice set P of a given presentation is indeed regular, we will usually assume that P is given by an automaton $\mathcal{A}_P$.

This definition generates 12 classes of presentations that are all interesting in their own right. Because of this vast combinatoric explosion and to keep the language concise, we will adopt the convention to speak only of uniformly automatic classes or presentations whenever we want to express a thought that might apply to several kinds of presentations and only make the exact distinction clear in the formal statements.

Examples We give a few motivating examples, which also demonstrate the variety of possible applications. The first example is due to Reinhardt [91]. Here we apply the idea formulated in the beginning of the chapter to the advice automatic presentation of $(\mathbb{Q}, +)$ from Example 3.1.

Example 4.1. The class of all torsion-free abelian groups of rank 1 is regularly ω-automatic. By the famous characterisation of Baer [7], the torsion-free abelian groups of rank 1 are, up to isomorphism, exactly the subgroups of $(\mathbb{Q}, +)$. Moreover, every subgroup of $(\mathbb{Q}, +)$ is isomorphic to a subgroup of $(\mathbb{Q}, +)$ that contains the element 1. Every such subgroup $G \leq (\mathbb{Q}, +)$ is characterised by the number of times 1 can be divided by each prime. More precisely, the isomorphism type of G is completely determined by the characteristic sequence $c = (c_p)_{p \in \mathbb{P}}$ with

$$c_p = \begin{cases} n & \text{if } n = \max\{n \in \mathbb{N} \mid p^n \big| 1 \text{ in } G\} \text{ exists} \\ \infty & \text{otherwise.} \end{cases}$$

For any characteristic sequence c the canonical subgroup G_c with characteristic sequence c is the subgroup generated by the set

$$\bigcup_{p \in \mathbb{P}} \left\{ \frac{1}{p^n} \mid n \in \mathbb{N}, n \leq c_p \right\}.$$

To any sequence $k = (k_i)_{i \in \mathbb{N}}$ we assign the characteristic sequence $c(k) = (c(k)_p)_{p \in \mathbb{P}}$ with

$$c(k)_p = \begin{cases} \sum_{i=0}^{m} \max\{n \in \mathbb{N} \mid p^n \big| k_i \text{ in } \mathbb{Z}\} & \text{if } m = \max\{i \in \mathbb{N} \mid p \big| k_i\} \text{ exists} \\ \infty & \text{otherwise.} \end{cases}$$

It can be shown that every element x in $G_{c(k)}$ has a presentation

$$x = e \cdot \left(m + \sum_{i=0}^{n} \frac{a_i}{\prod_{j=0}^{i} k_i} \right)$$

with $e \in \{1, -1\}$, $m, n \in \mathbb{N}$, and $0 \leq a_i < k_i$ for all $0 \leq i \leq n$.

Now let $\mathfrak{d}$ be the presentation given in Example 3.1 and let α_k be the ω-word that consists of the binary expansions of the sequence k, that means the advice is $\alpha_k = \mathrm{bin}(k_0)\#\,\mathrm{bin}(k_1)\#\,\mathrm{bin}(k_2)\#\cdots$. With the above facts, one can verify that $\mathcal{S}(\mathfrak{d}[\alpha_k]) \cong G_{c(k)}$. Hence together with the ω-regular set

$$P = \{\mathrm{bin}(k_0)\#\,\mathrm{bin}(k_1)\#\,\mathrm{bin}(k_2)\#\cdots \mid (k_i)_{i\in\mathbb{N}} \in \mathbb{N}^\omega\}$$

we obtain the regularly ω-automatic presentation $(P, \mathfrak{d})$ of the torsion-free abelian groups of rank 1.

From Rabin's proof of the decidability of the MSO-theory of the class of all linear orders we extract the following example.

Example 4.2. The class $\mathcal{C} = \{\mathcal{P}((A, <)) \mid (A, <) \text{ is a countable linear order}\}$ is regularly ω-tree-automatic. We use the fact that every countable linear order is embeddable into $(\mathbb{Q}, <)$ and that there is an automatic order on the nodes of the infinite binary tree that is isomorphic to the order on $\mathbb{Q}$.

Let $<_{\text{in}}$ be the in-order on $\{0,1\}^*$, that is

$$\begin{aligned} v <_{\text{in}} w :\Leftrightarrow & (v \not\prec w \wedge w \not\prec v \wedge v <_{\text{lex}} w) \\ & \vee (v \prec w \wedge \exists x : w = v1x) \\ & \vee (w \prec v \wedge \exists x : v = w0x,) \end{aligned}$$

where $\prec$ is the prefix relation and $<_{\text{lex}}$ is the lexicographic order. A back and forth argument shows $(\{0,1\}^*, <_{\text{in}}) \cong (\mathbb{Q}, <)$ (Cantor). Note that $<_{\text{in}}$ is an ω-tree-automatic relation on the nodes of the infinite binary tree (i.e. on the $\{0,1\}$-labelled trees that have the label 1 on exactly one position). Hence there is an injective set-interpretation $\mathcal{I} = (\delta(X), \varphi_{\text{in}}(X,Y), \varphi_{\subseteq}(X,Y))$ of $\mathcal{P}(\mathbb{Q}, <)$ in the infinite binary tree $\mathfrak{T}_2^\omega$ where the singleton sets over $\mathbb{Q}$ are exactly the singleton subsets of $\{0,1\}^*$. We modify $\mathcal{I}$ to obtain a uniform set-interpretation $\mathcal{I}'$ of the class of all powerset structures of countable linear orders in $\{(\mathfrak{T}_2^\omega, P) \mid P \subseteq \{0,1\}^*\}$. The interpretation $\mathcal{I}'$ is defined as follows:

$$\begin{aligned} \delta'(X) &= \forall y : y \in X \to y \in P, \\ \varphi'_{<_{\text{in}}}(X,Y) &= \delta'(X) \wedge \delta'(Y) \wedge \varphi_{<_{\text{in}}}(X,Y), \text{ and} \\ \varphi'_{\subseteq}(X,Y) &= \delta'(X) \wedge \delta'(Y) \wedge \varphi_{\subseteq}(X,Y). \end{aligned}$$

In other words $\mathcal{I}'$ interprets in $(\mathfrak{T}_2^\omega, P)$ the powerset structure of the substructure of $(\{0,1\}^*, <_{\text{in}})$ that is induced by P. Then $\mathcal{I}'$ is a uniform set-interpretation of $\mathcal{C}$ in the class of all labelled infinite binary trees and therefore the class $\mathcal{C}$ is regularly ω-tree-automatic.

The last example connects our concept with the celebrated theorem of Courcelle.

Example 4.3. Fix some $d \in \mathbb{N}$ and let $\mathcal{C}$ be the class of all finite graphs with treewidth at most d. Then the class $\{\mathcal{P}(G) \mid G \in \mathcal{C}\}$ is regularly automatic.

A bit of preparation is necessary to define a suitable set of advice trees. Let G be a graph of treewidth at most d and $\mathcal{T}$ a tree decomposition of G with width$(\mathcal{T}) \leq d$. We fix a root for $\mathcal{T}$ and a colouring $f : V \to \{0, \ldots, d\}$ such that all distinct $u, v \in V$ that appear together in some bag of $\mathcal{T}$ get distinct colours assigned. Such a colouring exists and can easily be computed by traversing $\mathcal{T}$ from top to bottom and assign a colour to a node v when it first appears in some bag X. The colour can be chosen as the first colour that is not already occupied by another node from X.

To every node X of $\mathcal{T}$ we associate the structure $\mathfrak{A}_X = (f(X), P^{\mathfrak{A}_X}, E^{\mathfrak{A}_X})$ where $P^{\mathfrak{A}_X} = \{f(v) \mid \text{ no ancestor of } X \text{ in } \mathcal{T} \text{ contains } v\}$ is the set of all colours $f(v)$ such that v appears in X for the first time in the tree and $E^{\mathfrak{A}_X}$ is such that f restricted to X is an isomorphism between the subgraph of G induced by X and $(f(X), E^{\mathfrak{A}_X})$, that is $E^{\mathfrak{A}_X} = \{(f(u), f(v)) \mid u, v \in X, (u, v) \in E^G\}$.

We construct an advice tree for G from $\mathcal{T}$ and f. Our advice trees will have unbounded degree in general, but, of course, we can encode these trees with standard techniques by trees of bounded degree. The alphabet consists of all $\{P, E\}$-structures whose universe is a subset of $\{0, \ldots, d\}$: $\Gamma = \{\mathfrak{A} \in \text{Str}[\{P, E\}] \mid A^{\mathfrak{A}} \in \mathcal{P}(\{1, \ldots, d\})\}$. The shape of the advice tree α is isomorphic to $\mathcal{T}$ and the labelling is defined by $\alpha(w) := \mathfrak{A}_X$, where X is the node of $\mathcal{T}$ that corresponds to w. To illustrate this construction recall the graph G from Example 2.7. Figure 4.1 shows an advice for the tree decomposition given in Figure 2.3.

The set of trees that originate from a proper tree decomposition is regular because one only has to check that the labelling is consistent with the edge

relation of a graph. This is expressed by the formula

$$\forall x \preceq y \in \mathrm{dom}(\alpha) \forall i, j \in \{1, \ldots, d\} : \\ \mathit{NoChange}(i, j, x, y) \rightarrow ((i, j) \in E^{\alpha(x)} \leftrightarrow (i, j) \in E^{\alpha(y)}),$$

where $\mathit{NoChange}(i, j, x, y)$ states that both i and j represent the same vertices of G in x and y:

$$\mathit{NoChange}(i, j, x, y) = \exists z_i, z_j \preceq x : i \in P^{\alpha(z_i)} \wedge j \in P^{\alpha(z_j)} \\ \wedge \forall x \prec w \preceq y : i \notin P^{\alpha(w)} \wedge j \notin P^{\alpha(w)}.$$

Having fixed the advice α, the elements of the structure are encoded over the alphabet $\Sigma = \mathcal{P}(\{1, \ldots, d\})$ by trees over the same domain as α. The singleton sets are encoded by trees $t : \mathrm{dom}(\alpha) \rightarrow \Sigma$ such that

- all nodes $w \in \mathrm{dom}(\alpha)$ with $t(w) \neq \emptyset$ have the same label $\{i\}$ for some $i \in \{1, \ldots, d\}$,
- there is a node v with $t(v) = \{i\}$ and $i \in P^{\alpha(v)}$, and
- the other nodes w with $t(w) = \{i\}$ are exactly the nodes with $v \prec w$ and $i \notin P^{\alpha(u)}$ for all $v \prec u \preceq w$.

Intuitively, the singleton set $\{v\} \subseteq V$ is encoded by the colour $f(v)$ and the subtree of bags of $\mathcal{T}$ that contain v.

Let t_U and t_H be trees that encode sets U and H, respectively. Then the set $U \cup H$ is represented by the tree t with $t(w) = t_U(w) \cup t_H(w)$ for all $w \in \mathrm{dom}(\alpha)$. Note that, by the properties of the colouring function f, every tree that can be derived in this way from the representations of the singleton sets encodes exactly one set U without any ambiguity. The construction of the automata for the domain and the relations of the power set structure is left as an exercise for the reader.

Remark 4.1. In our examples we used presentations of power set structures $\mathcal{P}(\mathfrak{A})$ to translate MSO-sentences over $\mathfrak{A}$ first to FO-sentences over $\mathcal{P}(\mathfrak{A})$ and then, via the interpretation, back to MSO-sentences. It would also be possible to go the direct way of using MSO interpretations instead of set-interpretations. As we already mentioned in Chapter 3, MSO-interpretations correspond to

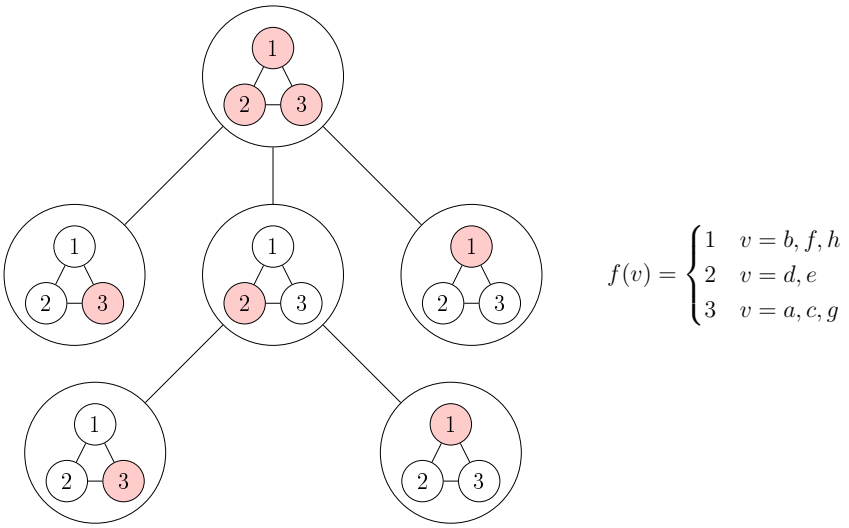


Figure 4.1: An advice tree derived from Figure 2.3

unary automatic presentations. This means elements are encoded by $\{0,1\}$-labelled trees, which are labelled with 1 on exactly one position. We remark here that it does not matter which way we choose in terms of which classes can be presented. In fact, Colcombet and Löding have proved that $\mathcal{P}(\mathfrak{A})$ is set-interpretable in a tree $\mathfrak{T}$ if, and only if, $\mathfrak{A}$ is MSO-interpretable in $\mathfrak{T}$ [25]. Also the translation from one interpretation to the other is uniform. While the direction from right to left is trivial, the other direction is a deep and technical result.

4.1 Decision Problems

Let us first state the fundamental decidability theorem for the theory of strongly automatic classes.

Corollary 4.1. *Every class $\mathcal{C} \in$ (ω-)[tree-]*SAutCl *has a decidable first-order theory.*

Proof. Let $\mathcal{C} \in$ (ω-)[tree-]SAutCl and φ be a first-order sentence in the language of $\mathcal{C}$. Then there is an (ω-)[tree-]automatic presentation $(P, \mathfrak{c})$ of $\mathcal{C}$ such that P has a decidable MSO-theory. Using the algorithm in Theorem 3.1 we

construct an (ω-)[tree-]automaton $\mathcal{A}_\varphi$ that recognises the set $\{\alpha \mid \mathcal{S}(\mathfrak{c}[\alpha]) \models \varphi\}$. By Büchi's [Rabin's] Theorem we can construct an MSO-formula ψ that is true in the advice structure if and only if the advice is accepted by $\mathcal{A}_\varphi$. At last we use the decision procedure for the MSO-theory of P to decide whether $\psi \in \mathrm{Th}_{\mathrm{MSO}}(P)$. □

A classical problem in model theory is to determine the number of non-isomorphic models of a formula within a certain class of structures. Due to the fact that we can decide many kinds of cardinality queries for regular languages this can easily be handled if the presentation of the class fulfills the following uniqueness condition: A uniform (ω-)[tree-]automatic presentation $(P, \mathfrak{c})$ of a class $\mathcal{C}$ has the **unique presentation property**, if for any $\mathfrak{A} \in \mathcal{C}$ there is exactly one advice $\alpha \in P$ such that $\mathcal{S}(\mathfrak{c}[\alpha]) \cong \mathfrak{A}$.

Lemma 4.1. *If a class $\mathcal{C}$ has a regularly (ω-)automatic presentation $(P, \mathfrak{c})$ with the unique presentation property then the following questions are uniformly decidable given $\varphi \in \mathrm{FO}, k, m \in \mathbb{N}$:*

1. *Is the number of pairwise non-isomorphic models of φ in $\mathcal{C}$ a finite number n with $n \equiv k \mod m$?*

2. *Is the number of pairwise non-isomorphic models of φ in $\mathcal{C}$ at most countably infinite?*

3. *Is the number of pairwise non-isomorphic models of φ in $\mathcal{C}$ uncountable?*

Proof. We can construct from $(P, \mathfrak{c})$ an automatic presentation of

$$\mathfrak{A}_\mathcal{C} = ((\uplus_{\alpha \in P} \mathcal{S}(\mathfrak{c}[\alpha])) \cup \{\mathcal{S}(\mathfrak{c}[\alpha]) \mid \alpha \in P\}, \sim),$$

where $\mathcal{S}(\mathfrak{c}[\alpha]) \sim a$ holds if $a \in A^{\mathcal{S}(\mathfrak{c}[\alpha])}$. Let ψ be the formula obtained from φ by relativising all quantifiers Qx by $a \sim x$. Because of the unique representation property the Questions 1 - 3 reduce to checking whether $\mathfrak{A}_\mathcal{C}$ fulfils $\exists^{(k \mod m)} a(\psi)$, $\exists^{\leq \aleph_0} a(\psi)$, or $\exists^{> \aleph_0} a(\psi)$, respectively. The extension of first-order logic by the quantifiers $\exists^{(k \mod m)}$,$\exists^{\leq \aleph_0}$ (, and $\exists^{> \aleph_0} a(\psi)$) is decidable for ($\omega$-)automatic structures [68, 74, 79]. □

Unfortunately even in the simplest possible case, i.e. uniformly automatic classes of finite sets, the unique presentation property is undecidable.

Theorem 4.1. *Let σ be a finite relational signature. The problem to decide whether a regularly automatic presentation of a class of σ-structures has the unique presentation property is*

1. *complete for the class Π^0_1 if σ contains only monadic predicates (even for classes of finite structures) and*

2. *hard for the class Π^1_1 if σ contains a predicate of arity at least two.*

Proof. Proposition 2 follows directly from the fact that the isomorphism problem for automatic structures is Σ^1_1-complete [73]. Obviously, given automatic presentations $\mathfrak{d}_0, \mathfrak{d}_1$, one can construct a parametrised presentation $\mathfrak{c}$ over the advice set $\{0,1\}$ such that $\mathcal{S}(\mathfrak{c}[0]) = \mathcal{S}(\mathfrak{d}_0)$ and $\mathcal{S}(\mathfrak{c}[1]) = \mathcal{S}(\mathfrak{d}_1)$. Then $\mathfrak{c}$ has the unique presentation property if, and only if, $\mathcal{S}(\mathfrak{d}_0) \not\cong \mathcal{S}(\mathfrak{d}_1)$.

In order to establish proposition 1 we adopt a technique used by Kuske et al. to show that the isomorphism problem for automatic equivalence relations is Π^0_1-complete. More precisely we use encodings of polynomials by automata to reduce Hilbert's 10th problem to the uniqueness problem. The problem can be formulated as follows: given polynomials $p, q \in \mathbb{N}[x_1, \ldots, x_k]$ decide whether $p(\overline{a}) = q(\overline{a})$ for some $\overline{a} \in \mathbb{N}^k$. In [78, Lemma 2] it is shown that for every polynomial p with non-negative coefficients one can construct an automaton $\mathcal{A}_p$ such that on input $1^{n_1} \otimes \ldots \otimes 1^{n_k}$ the automaton $\mathcal{A}_p$ has exactly $p(n_1, \ldots, n_k)$ accepting runs. Given such an automaton $\mathcal{A}_p$ we can construct the following automatic presentation $\mathfrak{c}_p$ of the class $\{(\{0, \ldots, m-1\}) \mid \exists \overline{n}(p(\overline{n}) = m)\}$. The advice language of $\mathfrak{c}$ is $\{1^{n_1} \otimes \ldots \otimes 1^{n_k} \mid \overline{n} \in \mathbb{N}^k\}$. For a advice α the domain language is $\{w \in Q_p^* \mid w \text{ is an accepting run of } \mathcal{A}_p \text{ on } \alpha\}$, which is uniformly automatic since an automaton can check while reading $\alpha \otimes w$ if w is an accepting run of $\mathcal{A}_p$ on α. To complete the proof we consider the injective polynomial $C(x,y) = 2y + (x+y)(x+y+1)$. By nesting C with different inputs, we obtain injective polynomials C_k for any arity k. For $p, q \in \mathbb{N}[x_1, \ldots, x_n]$ define

$$p' := C_{k+1}(x_1, \ldots, x_k, p(x_1, \ldots, x_k))$$
$$q' := C_{k+1}(x_1, \ldots, x_k, q(x_1, \ldots, x_k)).$$

Observe that p' and q' are both injective and $p'(\overline{a}) = q'(\overline{b})$ holds if, and only if, $\overline{a} = \overline{b}$ and $p(\overline{a}) = q(\overline{b})$. Now let $\mathfrak{c}$ be the advice disjoint union of $\mathfrak{c}_{p'}$ and $\mathfrak{c}_{q'}$. By the aforementioned properties of p', q', $\mathfrak{c}$ has the unique representation property if, and only if, $p(\overline{a}) \neq q(\overline{a})$ for all $\overline{a} \in \mathbb{N}^k$. This establishes the

hardness for Π_1^0. Further, the isomorphism problem is decidable for automatic structures over purely monadic signatures. This holds because two such structures are isomorphic if and only if all atomic 1-types have the same number of realisations in both structures. The elements of a structure which realise a specific atomic 1-type are definable by a first-order formula and hence form a regular set in the presentation. As the number of atomic 1-types depends only on the signature σ we can compare the number of realisations for each type individually. Hence the uniqueness problem is in Π_1^0 since we can just enumerate all pairs of distinct advices (α, β) from the regular advice set and check if $\mathcal{S}(\mathfrak{c}[\alpha]) \cong \mathcal{S}(\mathfrak{c}[\beta])$. □

4.2 Closure Operators

Automatic structures and their variants are effectively closed under disjoint unions and direct products. In this section we ask whether these constructions can be made uniform. In other words, if a class $\mathcal{C}$ is uniformly automatic is also the closure of $\mathcal{C}$ under disjoint unions and the closure under direct products uniformly automatic? We will see that the picture becomes quite diverse.

Definition 4.3. Let $\mathcal{C}$ be a class of τ-structures. Then $\mathcal{C}^\times$ denotes the closure of $\mathcal{C}$ under direct products and, in case that τ is relational, $\mathcal{C}^\uplus$ denotes the closure of $\mathcal{C}$ under disjoint unions. That is

$$\mathcal{C}^\times = \{\mathfrak{A}_1 \times \cdots \times \mathfrak{A}_n \mid n \geq 1, \mathfrak{A}_1, \ldots, \mathfrak{A}_n \in \mathcal{C}\} \text{ and}$$
$$\mathcal{C}^\uplus = \{\mathfrak{A}_1 \uplus \cdots \uplus \mathfrak{A}_n \mid n \geq 1, \mathfrak{A}_1, \ldots, \mathfrak{A}_n \in \mathcal{C}\}.$$

It is not hard to see that uniformly (ω-)tree-automatic classes behave very well under the two closure operators that we defined above.

Lemma 4.2.

1. *Let $\mathcal{C}$ be a uniformly (ω-)tree-automatic class of structures. From a given (ω-)tree-automatic presentation $(P, \mathfrak{c})$ of $\mathcal{C}$ one can effectively construct an (ω-)tree-automatic presentation $(P^\times, \mathfrak{c}^\times)$ of $\mathcal{C}^\times$. Moreover, regularity of the advice set is preserved.*

2. *Let $\mathcal{C}$ be a uniformly (ω-)tree-automatic class of structures. From a given (ω-)tree-automatic presentation $(P, \mathfrak{c})$ of $\mathcal{C}$ one can effectively construct an (ω-)tree-automatic presentation $(P^\uplus, \mathfrak{c}^\uplus)$ of $\mathcal{C}^\uplus$. Moreover, regularity of the advice set is preserved.*

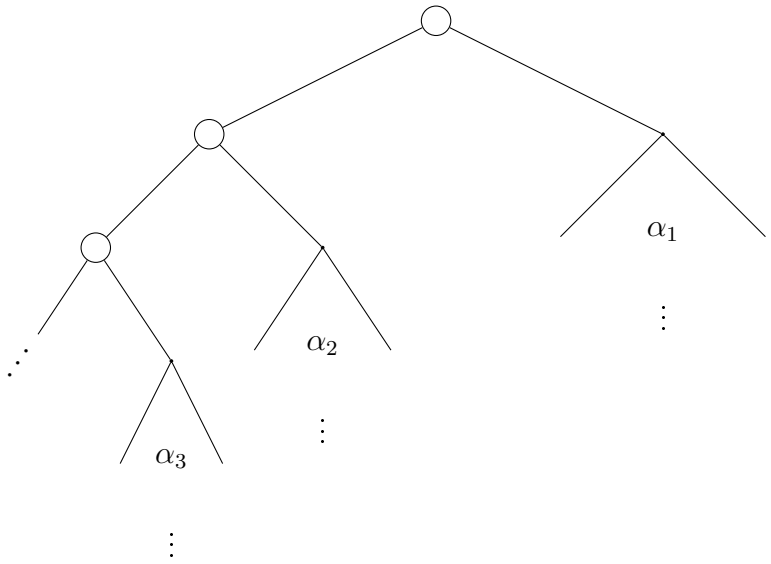


Figure 4.2: The Parameters for the class $\mathcal{C}^\times$

Proof. Proof of (1): Suppose $\mathcal{C}$ is presented by the uniform (ω-)tree-automatic presentation $\mathfrak{c}$ over the advice set P. As the construction is rather straightforward we only give the advice set for the presentation and the idea for the encoding. The advice set consists of all trees where the right child of every node in the left-most branch induces a subtree which is in P. This is depicted in Figure 4.2 . Such an advice presents the structure $\mathcal{S}(\mathfrak{c}[\alpha_1]) \times \mathcal{S}(\mathfrak{c}[\alpha_2]) \times \cdots \times \mathcal{S}(\mathfrak{c}[\alpha_n])$. Let $t_1, \ldots, t_n$ be elements of $\mathcal{S}(\mathfrak{c}[\alpha_1]), \ldots, \mathcal{S}(\mathfrak{c}[\alpha_n])$, respectively. Then the element $(t_1, \ldots, t_n)$ is put together in the same way as the advices.

Proof of (2): We use the same advice set as in (1). The elements are encoded by trees where all except one node 0^n1 of the form $\{0\}^*1$ are leafs labelled with a new dummy symbol and 0^n1 induces a subtree t such that $t \in L(\mathcal{A}[\alpha_n])$. The construction of the automata that recognise the relations in the class is now straight forward. □

Next we consider uniformly automatic classes. While the closure under disjoint unions can be treated by a rather simple construction, we have to restrict our attention to classes of finite structures in order to handle the closure under direct products. For the case of uniformly automatic classes of

finite structures we can ensure that there is a presentation where for every structure of the class the length of the encodings of the elements match the length of the respective advice. This is formulated in the following lemma.

Lemma 4.3. *Let $\mathcal{C}$ be a uniformly automatic class of finite structures. Then there is a uniformly automatic presentation $\mathfrak{c} = (\mathcal{A}, (\mathcal{A}_R)_{R\in\tau})$ of $\mathcal{C}$ with advice set P such that $|w| = |\alpha|$ for all $\alpha \in P$ and all $w \in L(\mathcal{A}[\alpha])$. Further if $\mathcal{C}$ is regularly automatic then P is a regular set.*

Proof. Let $\mathfrak{c}' = (\mathcal{A}', (\mathcal{A}'_R)_{R\in\tau})$ be an injective uniform automatic presentation of $\mathcal{C}$ with advice set P'. Let $Q' \supseteq P$ be the set of all advices α such that $\mathcal{S}(\mathfrak{c}[\alpha])$ is a finite structure. Note that presenting a finite structure under a fixed presentation is a regular property and hence Q' is a regular set. Then $L(\mathcal{A}'[\alpha])$ is finite for all $\alpha \in Q'$. Consider the injective mapping $\pi : Q' \to Q'\#^*$ given by $\alpha \mapsto \alpha\#^k$ with $k = \max\{0, \max\{|w| - |\alpha| \mid \alpha \otimes w \in L(\mathcal{A}'_p)\}\}$. The language $Q := \pi(Q')$ is also regular. Similarly we can construct automata $\mathcal{A}, (\mathcal{A}_R)_{R\in\tau}$ with

$$L(\mathcal{A}) = \{\alpha\#^n \otimes w\#^k \mid \alpha \otimes w \in L(\mathcal{A}') \wedge \alpha\#^n \in Q \wedge |\alpha\#^n| = |w\#^k|\} \text{ and}$$
$$L(\mathcal{A}_R) = \{\alpha\#^n \otimes w_1\#^{k_1} \otimes \ldots \otimes w_r\#^{k_r} \mid \alpha \otimes w_1 \otimes \ldots \otimes w_r \in L(\mathcal{A}'_R) \wedge \alpha\#^n \in Q \wedge |\alpha\#^n| = |w_i\#^{k_i}|\}$$

for all $R \in \tau$. These automata together with the advice set $P = \{\pi(\alpha) \mid \alpha \in P'\}$ form the presentation we are looking for. At last observe that if P' is a regular set then so is P. □

Lemma 4.4.

1. *Let $\mathcal{C}$ be a uniformly automatic class of finite structures. From a given automatic presentation $(P, \mathfrak{c})$ of $\mathcal{C}$ one can effectively construct an automatic presentation $(P^\times, \mathfrak{c}^\times)$ of $\mathcal{C}^\times$. Moreover, regularity of the advice set is preserved.*

2. *Let $\mathcal{C}$ be a uniformly automatic class of structures. From a given automatic presentation $(P, \mathfrak{c})$ of $\mathcal{C}$ one can effectively construct an automatic presentation $(P^\uplus, \mathfrak{c}^\uplus)$ of $\mathcal{C}^\uplus$. Moreover, regularity of the advice set is preserved.*

Proof. Let $\mathfrak{c} = (\mathcal{A}, (\mathcal{A}_R)_{R\in\tau}, \pi)$ be an automatic presentation of $\mathcal{C}$ over the advice set P.

Proof of (1): By Lemma 4.3, we might assume that for all $\alpha \in P$ and all $w \in L(\mathcal{A}[\alpha])$ we have $|\alpha| = |w|$. As advice set for $\mathcal{C}^\times$ we take $(P\#)^*P$, where $\alpha_1\# \dots \#\alpha_n$ is an advice for $\mathcal{S}(\mathfrak{c}[\alpha_1]) \times \cdots \times \mathcal{S}(\mathfrak{c}[\alpha_n])$. The construction of a uniform presentation $\mathfrak{c}^\times$ of $\mathcal{C}^\times$ from $\mathfrak{c}$ is straight forward. On reading $\alpha_1\# \dots \#\alpha_n$ as advice, the automaton $\mathcal{A}^\times$ should accept exactly the words $w_1\# \dots \#w_n$ with $w_i \in L(\mathcal{A}[\alpha_i])$ for all $i \in \{1, \dots, n\}$. This can obviously be done by an automaton since all words in $L(\mathcal{A}[\alpha_i])$ have the same length as α_i. The same idea allows us to construct automata that recognise the relations $R \in \tau$.

Proof of (2): We construct a presentation over the advice set $(P\#)^*P$, where $\alpha_1\# \cdots \#\alpha_n$ should be an advice for $\biguplus_{1\leq i\leq n} \mathcal{S}(\mathfrak{c}[\alpha_i])$. For an advice $\alpha_1\# \cdots \#\alpha_n$ we encode the elements of $\mathcal{S}(\alpha_i)$, $1 \leq i \leq n$, by the language $L_i = \#^{|\alpha_1\#\cdots\#\alpha_{i-1}\#|} L(\mathcal{A}[\alpha_i])$. Intuitively we shift the encodings of the elements in the copy of the i-th summand so that they match with the beginning of the i-th advice. Obviously one can construct a parametrised automaton with $L(\mathcal{A}[\alpha]) = \bigcup_{1\leq i\leq n} L_i$ for all $\alpha = \alpha_1\# \cdots \#\alpha_n \in (P\#)^*P$. It is an easy exercise to construct the rest of the presentation. □

At last we show that Proposition (1) of Lemma 4.4 cannot be extended to arbitrary uniformly automatic classes.

Theorem 4.2. *The class of all free abelian groups of finite rank is not uniformly ω-automatic.*

Proof. The statement is a consequence of the following two propositions.

Lemma 4.5. *There exists an increasing function $f : \mathbb{Q} \to \mathbb{N}$ such that for every free abelian group of finite rank $(G, +)$ the following is true: For every finite set $X \subseteq G$ the rank of the subgroup generated by X is bounded by*

$$\operatorname{rank}(\langle X\rangle) \leq f\left(\frac{|X + X|}{|X|}\right).$$

Proof. Consider the torsion-free abelian group $(\mathbb{Z}^\omega, +)$. By Freiman's Theorem there is a increasing function $g : \mathbb{Q} \to \mathbb{N}$ such that every finite subset $X \subseteq \mathbb{Z}^\omega$ the following holds: Let $n := g\left(\frac{|X+X|}{|X|}\right)$. Then $\langle X\rangle$ is contained in an n-dimensional generalised arithmetical progression $P = \{a_0 + k_1a_1 + \cdots + k_na_n \mid$

$k_1, \ldots, k_n \in \mathbb{Z}\}$ for some $a_0, a_1, \ldots, a_n \in \mathbb{Z}^\omega$. Hence, $\langle X \rangle \subseteq \langle \{a_0, a_1, \ldots, a_n\} \rangle$ and therefore $\mathrm{rank}(\langle X \rangle) \leq \mathrm{rank}(\langle \{a_0, a_1, \ldots, a_n\} \rangle) \leq n + 1$.

Now every free abelian group G of finite rank is isomorphic to $(\mathbb{Z}^n, +)$ for some $n \in \mathbb{N}$. Fix some isomorphism $\iota : G \to (\mathbb{Z}^n, +)$ and let κ be the natural embedding of $(\mathbb{Z}^n, +)$ into $(\mathbb{Z}^\omega, +)$ given by $\kappa : (z_1, \ldots, z_n) \mapsto (z_1, \ldots, z_n, 0, 0, \ldots)$. Then $\lambda := \kappa \circ \iota$ is an embedding of G into $(\mathbb{Z}^\omega, +)$ and for all finite subsets X of G we can bound the rank of $\langle X \rangle_G$ by $\mathrm{rank}(\langle X \rangle_G) = \mathrm{rank}(\langle \lambda(X) \rangle_{(\mathbb{Z}^\omega, +)}) \leq g\left(\frac{|\lambda(X)+\lambda(X)|}{|\lambda(X)|}\right) + 1 = g\left(\frac{|X+X|}{|X|}\right) + 1 =: f\left(\frac{|X+X|}{|X|}\right)$. □

Next we observe that Corollary 3.13 is uniform in the following sense.

Corollary 4.2. *For every uniformly ω-automatic class of countable τ-structures $\mathcal{C}$ and every $(k+\ell)$-ary function $f \in \tau$ there exists a constant c such that for every $\mathfrak{A} \in \mathcal{C}$ and every finite set $X \subseteq A^{\mathfrak{A}}$ there exists a finite set $Y \supseteq X$ such that for every $\overline{a} \in (A^{\mathfrak{A}})^k$ it holds that $|f(\overline{a}, Y^\ell)| \leq c|Y|$.*

Proof. It suffices to observe that the constant c in Corollary 3.13 does not depend on the advice α □

To complete the proof of Theorem 4.2 we show that every uniformly ω-automatic class $\mathcal{C}$ of free abelian groups of finite rank has in fact bounded rank. As there are free abelian groups of arbitrary finite rank, we conclude that the class of all free abelian groups of finite rank is not uniformly ω-automatic.

Now let $\mathcal{C}$ be such a class a free abelian groups and let c be the constant from Lemma 4.2 with respect to $\mathcal{C}$. We claim that the rank of every group $G \in \mathcal{C}$ is bounded by $f(c)$ where f is the function from Lemma 4.5. For every group $G \in \mathcal{C}$ there is a finite set X with $\langle X \rangle = G$ and hence $\mathrm{rank}(\langle Y \rangle) = \mathrm{rank}(G)$ for every $Y \supseteq X$. But Lemma 4.2 implies that there is a finite set $Y \supseteq X$ with $|Y + Y| > c|Y|$. From Lemma 4.5 we get $\mathrm{rank}(G) = \mathrm{rank}(\langle Y \rangle) \leq f(\frac{|Y+Y|}{|Y|}) \leq f(c)$. □

Corollary 4.3. *There is a regularly automatic class $\mathcal{C}$ such that the closure under direct products $\mathcal{C}^\times$ is not uniformly ω-automatic.*

Proof. The free abelian groups of finite rank are up to isomorphism the finite direct products of the automatic structure $(\mathbb{Z}, +)$. □

Corollary 4.4. *The class $\{(\mathbb{N}^n, +) \mid n \geq 1\}$ is not uniformly ω-automatic.*

Proof. The free abelian groups of finite rank are first-order interpretable in $\mathcal{C}$. Since ω-AutCl is closed under first-order interpretations, $\{(\mathbb{N}^n, +) \mid n \geq 1\}$ cannot be uniformly ω-automatic. □

Another natural witness for Corollary 4.3 is the class of all finite direct products of the interval algebra on ω, $\mathfrak{B}_\omega$. Note that this is also the class of all automatic infinite boolean algebras: in [73] it was shown that an infinite boolean algebra is automatic if and only if it is isomorphic to $\mathfrak{B}_\omega^n$ for some $n \geq 1$. In the following we will identify $\mathfrak{B}_\omega$ with the set algebra over all finite and co-finite sets of natural numbers, $\mathcal{P}_{\mathrm{fc}}(\mathbb{N}) = \{X \subseteq \mathbb{N} \mid |X| < \infty \vee |\mathbb{N} \setminus X| < \infty\}$. Indeed, these two boolean algebras are isomorphic.

Lemma 4.6. *There exists an* FOC*-interpretation $\mathcal{I}$ such that the following holds: If $\mathfrak{A} = (\mathfrak{B}_\omega^n, P, \preceq)$ is (isomorphic to) the n-fold cartesian product of the interval algebra $\mathfrak{B}_\omega$ expanded by the unary relation*

$$P = \{(\mathbb{N}, \emptyset, \ldots, \emptyset), (\emptyset, \mathbb{N}, \emptyset, \ldots, \emptyset), \ldots, (\emptyset, \ldots, \emptyset, \mathbb{N})\}$$

and a linear order $\preceq$ of order-type ω then $\mathcal{I}(\mathfrak{A}) \cong (\mathbb{N}^n, +)$.

Proof. It is not too hard to see that $(\mathbb{N}, +)$ is FOC-interpretable in $(\mathcal{B}_\omega, P, \preceq)$. Without loss of generality we can assume that $\preceq$ orders the atoms in the natural way, that is $\{0\} \prec \{1\} \prec \cdots$. The idea is to identify the finite sets in $\mathcal{B}_\omega$ with the binary expansions of natural numbers. Accordingly, a finite set X presents the number $n(X) = \sum_{i \in X} 2^i$. The domain formula has to express that x is a finite set. This can be done by the formula $\varphi_{\mathbb{N}}(x) := \neg\exists^\infty y(y \subseteq x)$. Addition can be performed by quantifying carry-bits and checking the correctness using the linear order on the atoms. This is done by the following formula

$$\mathsf{add}(x, y, z) \equiv \exists c(0 \notin c \wedge \varphi_{\mathrm{carry}}(c) \wedge \forall i \in \mathbb{N}(i \in z \leftrightarrow (i \in x \oplus i \in y \oplus i \in c))),$$

where $\varphi_{\mathrm{carry}}(x, y, c)$ expresses that the carry bit are set correctly:

$$\varphi_{\mathrm{carry}}(x, y, c) \equiv \forall i \in \mathbb{N} \setminus \{0\} : i \in c \leftrightarrow \bigvee_{a \neq b \in \{x,y,c\}} (i-1) \in a \wedge (i-1) \in b.$$

Obviously add can be formulated in the language of Boolean algebras expanded by the linear order.

We describe how to transform the interpretation into a uniform interpretation of $\{(\mathbb{N}^n, +) \mid n \geq 1\}$ in $(\mathfrak{B}_\omega^n, P, \preceq)$. The idea for the general case is to use the predicate P to perform addition in every component separately. An element of $(m_1, \ldots, m_n)$ is encoded by the tuple $(X_1, \ldots, X_n) \in \mathcal{P}_{\mathrm{fc}}(\mathbb{N})^n$ of finite sets with $n(X_i) = m_i$ for all $1 \leq i \leq n$. The correctness of the addition has to be checked for every component separately. We obtain the interpretation

$$\begin{aligned} \delta(x) &\equiv \forall p \in P : \varphi'_{\mathbb{N}}(x \cap p, p) \\ \varphi_+(x, y, z) &\equiv \forall p \in P : \mathsf{add}'(x \cap p, y \cap p, z \cap p, p) \end{aligned}$$

where $\varphi'_{\mathbb{N}}$ and add' are obtained from $\varphi_{\mathbb{N}}$ and add by restricting all quantifications to elements below p. □

Lemma 4.7. *Let $(X_{ij})_{1 \leq i,j \leq n}$ be a collection of finite subsets of $\mathbb{N}$ such that for all $i \in \{1, \ldots, n\}$:*

- *$X_{ki} \cap X_{\ell i} = \emptyset$ and all $1 \leq k < \ell \leq n$ with $k, \ell \neq i$ and*
- *$X_{ii} = \bigcup_{j \neq i} X_{ji}$.*

Then there is an automorphism of $\mathfrak{B}_\omega^n$ which maps the tuple

$$((X_{i1}, \ldots, X_{i(i-1)}, \mathbb{N} \setminus X_{ii}, X_{i(i+1)}, \ldots, X_{in}))_{1 \leq i \leq n} \in (\mathcal{P}_{fc}(\mathbb{N})^n)^n$$

to the tuple

$$((\mathbb{N}, \emptyset, \ldots, \emptyset), (\emptyset, \mathbb{N}, \emptyset, \ldots, \emptyset), \ldots, (\emptyset, \ldots, \emptyset, \mathbb{N})).$$

Proof. The automorphism is constructed as follows: For all $1 \leq i \leq n$ fix a bijection π_i between the atoms below $(X_{i1}, \ldots, X_{i(i-1)}, \mathbb{N} \setminus X_{ii}, X_{i(i+1)}, \ldots, X_{in})$ and the atoms below $(\emptyset, \ldots, \emptyset, \underbrace{\mathbb{N}}_{\text{position } i}, \emptyset, \ldots, \emptyset)$. Because of the properties of $(X_{ij})_{1 \leq i,j \leq n}$, every atom appears in the domain and the range of exactly one π_i. Thus, we can combine $\pi_1, \ldots, \pi_n$ to a permutation $\pi = \bigcup_{1 \leq i \leq n} \pi_i$ on the atoms of $\mathfrak{B}_\omega^n$. We lift π to a permutation ρ on $\mathcal{P}(\mathbb{N})^n$ by

$$\rho((X_1, \ldots, X_n)) = \bigcup_{a \text{ atom below } (X_1, \ldots, X_n)} \pi(a).$$

Then ρ is an automorphism on $(\mathcal{P}(\mathbb{N}), \cup, \cap, \mathbb{N}, \emptyset)^n$ because ρ is derived from a permutation of the atoms. To see that this is true, note that $(\mathcal{P}(\mathbb{N}), \cup, \cap, \mathbb{N}, \emptyset)^n$ is isomorphic to $(\mathcal{P}(\mathbb{N}), \cup, \cap, \mathbb{N}, \emptyset)$ via the automorphism

$$(X_1, \ldots, X_n) \mapsto \bigcup_{1 \leq i \leq n} \{na + (i-1) \mid a \in X_i\}.$$

Hence every permutation of atoms extends to an automorphism of the boolean algebra $(\mathcal{P}(\mathbb{N}), \cup, \cap, \mathbb{N}, \emptyset)^n$. Further, the restriction of ρ to $\mathcal{P}_{\mathrm{fc}}(\mathbb{N})^n$ is a permutation on $\mathcal{P}_{\mathrm{fc}}(\mathbb{N})^n$, hence ρ is an automorphism on $\mathfrak{B}_\omega^n \subseteq (\mathcal{P}(\mathbb{N}), \cup, \cap, \mathbb{N}, \emptyset)^n$. □

Lemma 4.8. *There is an* FOC*-interpretation $\mathcal{J}$ such that the following is true: If a structure $\mathfrak{A}$ is isomorphic to $(\mathfrak{B}_\omega^n, \leq)$, where $\leq$ is a linear order on $\mathcal{P}_{fc}(\mathbb{N})^n$ of order type ω then the structure $\mathcal{J}(\mathfrak{A})$ is isomorphic to a structure $(\mathfrak{B}_\omega^n, P, \preceq)$, where P and $\preceq$ are as in Lemma 4.6.*

Proof. By Lemma 4.7 it suffices to show that a set of the form

$$\{(X_{i1}, \ldots, X_{i(i-1)}, \mathbb{N} \setminus X_{ii}, X_{i(i+1)}, \ldots, X_{in}) \mid 1 \leq i \leq n\},$$

where $(X_{ij})_{1 \leq i,j \leq n}$ are as described in Lemma 4.7, is definable by an FOC-formula.

First, we define the elements of $\mathfrak{B}_\omega^n$ that are finite in all but exactly one component. This is done by the following formula

$$\mathrm{Comp}(x) := \exists^\infty z(z \subseteq x) \wedge \neg \exists y(y \subseteq x \wedge \exists^\infty z(z \subseteq y) \wedge \exists^\infty z(z \subseteq x \setminus y)).$$

The formula Comp states that x is infinite and there is no infinite subset y of x such that $x \setminus y$ is also infinite. This ensures that $x = (X_1, \ldots, X_{i-1}, \mathbb{N} \setminus X_i, X_{i+1}, \ldots, X_n)$ for some finite sets $X_1, \ldots, X_n \subseteq \mathbb{N}$ and some $i \leq n$.

We employ the linear order to preselect n such elements, which are infinite in pairwise different components.

$$\mathrm{Sel}(x) := \mathrm{Comp}(x) \wedge \forall y((\mathrm{Comp}(y) \wedge y < x) \rightarrow \neg \exists^\infty z(z \subseteq x \cap y)).$$

The elements that are defined by Sel might still be not quite of the from that we want. First the elements might have finite non-empty intersections and

second there might be finitely many atoms that are not below any element of $\mathrm{Sel}^{(\mathfrak{B}_\omega^n,\leq)}$. Therefore we need a last modification step. We define

$$\begin{aligned}\mathrm{Disjoin}(x) \equiv \exists y.\mathrm{Sel}(y)\forall a.\mathrm{Atom}(a) :\\ a \in x \leftrightarrow (a \in y \wedge \forall z.(\mathrm{Sel}(z) \wedge z < y) \rightarrow a \notin z)\\ \mathrm{Uncov}(a) \equiv \mathrm{Atom}(a) \wedge \neg\exists x : \mathrm{Disjoin}(x) \wedge a \in x\end{aligned}$$

Finally, we define the formula φ_P:

$$\begin{aligned}\varphi_P(x) \equiv (\mathrm{Disjoin}(x) \wedge \exists y.\mathrm{Disjoin}(y) : y < x) \vee\\ (\exists x'.(\mathrm{Disjoin}(x') \wedge \neg\exists y.\mathrm{Disjoin}(y) : y < x')\forall a.\mathrm{Atom}(a) :\\ a \in x \leftrightarrow a \in x' \vee \mathrm{Uncovered}(a))\end{aligned}$$

□

Corollary 4.5. *The class $\{\mathfrak{B}_\omega\}^\times$ is not uniformly ω-automatic.*

Proof. If $\{\mathfrak{B}_\omega\}^\times$ would be uniformly ω-automatic then there would be a presentation $\mathfrak{c}$ with finite word encoding. But then we could extend this presentation by the length-lexicographic order and use the Interpretations $\mathcal{I}$ and $\mathcal{J}$ from Lemma 4.6 and Lemma 4.8 to obtain a uniformly ω-automatic presentation of the class $\{(\mathbb{N}^n, +) \mid n \geq 1\}$, contradicting Lemma 4.8. □

The reader has probably noticed that we have left one case open: the closure of uniformly ω-automatic classes under disjoint unions. As a matter of fact, this case is still open. On the one hand there seems to be no simple construction to obtain a presentation of the disjoint union closure from a given presentation. But on the other hand we are also not aware of a possible counter example.

Open Problem 4.1. *Is there a uniformly ω-automatic class $\mathcal{C}$ such that $\mathcal{C}^{\uplus}$ is not uniformly ω-automatic?*

Table 4.1 summarises the results of this section.

4.3 Discussion

In this chapter we introduced uniformly automatic presentations. The purpose of this chapter was mainly to establish basic facts and to get the reader

Table 4.1: Closure Operators on Uniformly Automatic Presentations

Presentation	$\times$	$\uplus$
automatic	✗	✓
ω-automatic	✗	?
tree-automatic	✓	✓
ω-tree-automatic	✓	✓

acquainted with the concept. We have seen that, at least implicitly, these presentations have been very successfully applied in various areas of theoretical computer science. The applications range from automata based decision procedures, like for the theory of linear orders, to algorithmic meta-theorems on classes of bounded treewidth or cliquewidth. In the upcoming chapter we will have a deeper look into the applications in finite model theory. We shortly comment on a few things that we have seen in the previous sections.

An interesting fact about having automata, which present a whole class of structures, is that we are able to relate the different models in this class. In Section 4.1 we just briefly hinted at one such application. One could ask whether there are interesting examples of uniformly automatic classes where we could also express certain relations between two models of the class, for instance that one model is embeddable into another. Of course, such a presentation might as well be understood as one large automatic structure. However, the shift in the semantics might give rise to applications which have not been considered so far.

With regard to Section 4.2, we want to remark on the fact that we had to leave the case of closure under disjoint unions open for uniformly ω-automatic classes. This might actually hint at a complexity restriction inherent to these kinds of presentations, which is not yet fully understood. We do not want to rule out that there is always a way to present the closure under disjoint unions with a uniformly ω-automatic presentation or that there might be a nice and simple argument why this is not the case. But at least our consideration did not give a strong indication in either direction.

5 Uniformly Automatic Classes of Finite Structures

While it is not very interesting to ask whether a single finite structure is automatic (because every finite structure is automatic), the situation changes when we ask which classes of finite structures admit a uniform automatic presentation. If we restrict ourselves to finite structures we operate in the field of finite model theory. Here the question is no longer if one can decide first-order or monadic second-order properties of a specific structure but rather how efficient these properties can be decided. Finite model theory has therefore strong connections to complexity theory. Especially in the field of parametrised complexity automata based approaches to model checking play an important role.

We have already seen that uniform tree-automaticity of a class of graphs is a property that goes beyond bounded treewidth and bounded cliquewidth with the obvious drawback that we can handle only first-order logic instead of monadic second-order logic. Still, if computing an advice from a given structure can be performed efficiently, model checking for first-order logic is fixed parameter tractable in the size of the formula on the class. Meta-theorems for first-order logic have a long history on classes of sparse graphs. Starting with Seese and graphs of bounded degree [96] to the resent result of Grohe, Kreutzer, and Siebertz for nowhere dense graphs [60]. Automatic presentations might bring up new and interesting classes of structures with a fixed parameter tractable model checking problem.

In this chapter, we are concerned with the efficiency of this approach. Note that the non-elementary worst-case runtime of the automaton construction leads in general to a non-elementary parameter dependence in the algorithmic meta-theorems. For the direct application of MSO model-checking on words or trees this is probably optimal. Frick and Grohe [47] showed, unless PTIME = NP, there is no algorithm that solves the model checking problem for MSO on finite words or trees in time

$$f(|\varphi|) \cdot \mathrm{poly}(|t|)$$

for any elementary function $f : \mathbb{N} \to \mathbb{N}$. The same holds for first-order logic on words (under the modified assumption FPT $\neq$ AW*). But what about our setting? It might be that the class which is given by the automatic presentation is actually very simple and there are other approaches which lead to FPT-algorithms with elementary parameter dependence. In this chapter we will show that runtime of the automata based model checking algorithm scales in many situations very nicely the the complexity of the class under consideration.

5.1 Model Checking Revisited

It is well known that there are (word-)automatic structures with non-elementary first-order theory [13]. A common example is the infinite binary tree with two successor relations and the descendant relation $(\{0,1\}^*, S_0, S_1, \preceq)$. Hence every algorithm that solves the model checking problem for structures given by a (tree-)automatic presentation has an unavoidable non-elementary worst-case runtime behaviour. On the other hand, for many important examples of automatic structures the situation is much better. For instance it is known that the first-order theory of Presburger Arithmetic can be decided in three-fold exponential time [83]. It is therefore very natural to analyse the runtime of a given model checking algorithm for automatic structures with respect to some fixed presentation.

In [33] Durand-Gasselin and Habermehl proposed a method to estimate the time that the generic automata based model checking algorithm for structures given by a *word-automatic* presentation needs when it is used to solve the first order theory of a single structure. They showed that for certain presentations of $(\mathbb{Z}, +)$ the running time of the algorithm is only triply exponential in the formula. Similar bounds where established for arbitrary word-automatic presentations of structures of bounded degree.

In the following we want to extend their method to *uniformly tree-automatic* presentations of *classes of structures*. Fortunately this generalisation goes through very well because of the nice analogue of the Myhill-Nerode congruence for regular tree-languages (see Theorem 2.5). Nevertheless, we also make some assumptions about the presentations in order to reduce the number of necessary case distinctions.

Definition 5.1. Let t be the convolution of trees $t_1, \ldots, t_n$. a **padded convolution** of $t_1, \ldots, t_n$ is a tree t' such that $\mathrm{dom}(t) \subseteq \mathrm{dom}(t')$, $t(w) = t'(w)$ for

all $w \in \mathrm{dom}(t)$, and $t'(w) = (\Box, \dots, \Box)$ for all $w \in \mathrm{dom}(t') \setminus \mathrm{dom}(t)$. In this case we also say that t' is a padding of t. Note that t is a padded convolution of some trees $t_1, \dots, t_n$ if and only if t is the convolution of some paddings of $t_1, \dots, t_n$. For a tree $t \in T_{\Sigma_1 \Box \times \cdots \times \Sigma_n \Box} \setminus T_{\{\Box^n\}}$ let $\mathrm{cut}(t)$ denote the unique tree with minimal domain (with respect to set inclusion) such that

- $\mathrm{dom}_{\mathrm{cut}(t)} \supseteq \{w \in \mathrm{dom}_t \mid t(w) \neq \Box^n\}$,
- $\mathrm{cut}(t)(w) = t(w)$ for all $w \in \mathrm{dom}_{\mathrm{cut}(t)}$, and

A padding of a uniformly tree-automatic presentation $\mathfrak{c} = (P, \mathcal{A}, (\mathcal{A}_R)_{R \in \tau})$ is a tuple $\mathfrak{c}' = (P, \mathcal{A}', (\mathcal{A}'_R)_{R \in \tau})$ such that $L(\mathcal{B}')$ is the closure under paddings of $L(\mathcal{B})$ for all $\mathcal{B} \in \{\mathcal{A}\} \cup \{\mathcal{A}_R \mid R \in \tau\}$. In this case we also say that $\mathfrak{c}'$ is a padded uniformly tree-automatic presentation.

A padding of an injective presentation is no longer injective in the sense that every padding of a tree t will encode the same element as t. However, it is still the case that equality is simply the identity relation.

Definition 5.2. An injective parametrised tree-automatic presentation $\mathfrak{c} = (\mathcal{A}, \mathcal{A}_{R_1}, \dots, \mathcal{A}_{R_n})$ is **good** if for all $\alpha \in T_\Gamma$ either $L(\mathcal{A}[\alpha]) = L(\mathcal{A}_{R_1}) = \cdots = L(\mathcal{A}_{R_n}) = \emptyset$ or $\mathfrak{c}[\alpha]$ presents a structure. We say that a good parametrised tree-automatic presentation presents the class $\{\mathcal{S}(\mathfrak{c}[\alpha]) \mid L(\mathcal{A}[\alpha]) \neq \emptyset\}$ and denote the corresponding advice set with $P^{\mathfrak{c}} = \{\alpha \in T_\Gamma \mid L(\mathcal{A}[\alpha]) \neq \emptyset\}$.

Since presenting a structure under a given tree-automatic presentation is a regular property of advice-trees, a class $\mathcal{C}$ is presented by a good presentation if and only if $\mathcal{C}$ is regularly tree-automatic. In the remainder of this chapter we will always work with padded good uniformly tree-automatic presentations.

We start with a detailed description of the model checking algorithm on structures given by an advice α from a uniform tree-automatic presentation $\mathfrak{c}$. Up to small optimisations it resembles the standard algorithm that constructs from $\mathfrak{c}$ and an FO-formula φ an automaton $\mathcal{A}_\varphi$ such that $\alpha \in L(\mathcal{A}_\varphi) \Leftrightarrow \mathcal{S}(\mathfrak{c}[\alpha]) \models \varphi$ and then checks whether α is accepted by $\mathcal{A}_\varphi$. The automaton $\mathcal{A}_\varphi$ is constructed recursively using standard constructions to compute automata for the intersection, complement, and projection of tree-automatic languages. The exact procedure is given by Algorithm 1. Analogous to [33], the only crucial optimisation is to construct only the reachable part of every automaton that appears in the process (see procudures Intersect and Determinise).

Algorithm 1 Modelchchecking on Uniformly Tree-Automatic Classes

Input: Padded good tree-automatic presentation $\mathfrak{c} = (\mathcal{A}, (\mathcal{A}_R)_{R\in\tau})$, FO-formula φ
Output: Tree-automaton $\mathcal{A}_\varphi$

procedure COMPOSE($\mathfrak{c}$, φ)
  if $\varphi(x_1, \ldots, x_m) = R(x_{i_1}, \ldots, x_{i_k}), R \in \tau \cup \{=\}$ **then**
    ▷ Construct the minimal automaton that accepts a tree t iff $t = \langle(\alpha, t_1, \ldots, t_r)\rangle$ with $(t_{i_1}, \ldots, t_{i_k}) \in \mathcal{S}(\mathfrak{d}[\alpha])^m$
    $\mathcal{A}'_R \leftarrow$ EXTEND($\mathcal{A}_R, r, i_1, \ldots, i_k$)
    $\mathcal{A}_D \leftarrow$ DOMAIN($\mathcal{A}$, m)
    $\mathcal{A}_\varphi \leftarrow$ INTERSECT($\mathcal{A}'_R$, $\mathcal{A}_D$)
    minimise $\mathcal{A}_\varphi$
    return $\mathcal{A}_\varphi$
  else if $\varphi(x_1, \ldots, x_m) = \psi(x_1, \ldots, x_m) \wedge \theta(x_1, \ldots, x_m)$ **then**
    ▷ Recursively construct $\mathcal{A}_\psi$,$\mathcal{A}_\theta$ and build the intersection automaton
    $\mathcal{A}_\psi \leftarrow$ COMPOSE($(\mathcal{A}, (\mathcal{A}_R)_{R\in\tau})$, ψ)
    $\mathcal{A}_\theta \leftarrow$ COMPOSE($(\mathcal{A}, (\mathcal{A}_R)_{R\in\tau})$, θ)
    return INTERSECT($\mathcal{A}_\psi$, $\mathcal{A}_\theta$)
  else if $\varphi(x_1, \ldots, x_m) = \neg\psi(x_1, \ldots, x_m)$ **then**
    ▷ Construct the complement automaton of $\mathcal{A}_\psi$ and build the intersection automaton with the minimal domain automaton
    $\mathcal{A}_\psi \leftarrow$ COMPOSE($(\mathcal{A}, (\mathcal{A}_R)_{R\in\tau})$, $\psi(x_1, \ldots, x_m)$)
    $\mathcal{A}_D \leftarrow$ DOMAIN($\mathcal{A}, r$)
    return INTERSECT($\overline{\mathcal{A}_\psi}$, $\mathcal{A}_D$)
  else if $\varphi(x_1, \ldots, x_m) = \exists x_{m+1} : \psi(x_1, \ldots, x_{m+1})$ **then**
    ▷ Construct $\mathcal{A}_\psi$, apply the projection $(\sigma_1, \ldots, \sigma_{m+1}) \mapsto (\sigma_1, \ldots, \sigma_m)$ to the alphabet, and determinise the result
    $\mathcal{A}_\psi \leftarrow$ COMPOSE($(\mathcal{A}, (\mathcal{A}_R)_{R\in\tau})$, $\psi(x_1, \ldots, x_m)$)
    $\mathcal{A}'_\varphi \leftarrow$ PROJECT($\mathcal{A}_\psi$)
    $\mathcal{A}_\varphi \leftarrow$ DETERMINISE($\mathcal{A}'_\varphi$)
    return $\mathcal{A}_\varphi$
  end if
end procedure

The subroutine DOMAIN constructs the minimal tree-automaton that recognises exactly those trees in $T_{\Gamma_\Box \times (\Sigma_\Box)^m}$ that are padded convolutions of trees $t_0 \in T_\Gamma$ and $t_1, \ldots, t_m \in T_\Sigma$ such that $t_1, \ldots, t_m \in \mathcal{S}(\mathfrak{d}[t_0])$.

Algorithm 2 Adjust the arity of a relation

Input: Tree-automaton $\mathcal{A} = (Q, \Gamma_\Box \times \Sigma_\Box^k, \delta, F)$, $m \in \mathbb{N}$, $1 \leq i_1, \ldots, i_k \leq m$
Output: $\mathcal{A}'$ s.t. $L(\mathcal{A}') = \{\langle t_p, t_1, \ldots, t_m \rangle | \langle t_p, t_{i_1}, \ldots, t_{i_k} \rangle \in L(\mathcal{A})\}$

procedure EXTEND($\mathcal{A}, r, i_1, \ldots, i_k$)
  for $\overline{a} = (\gamma, \sigma_1, \ldots, \sigma_m) \in \Gamma_\Box \times (\Sigma_\Box)^m, \ell \in \mathrm{rk}(\overline{a}), q_1 \ldots, q_\ell \in Q^\mathcal{A}$ **do**
    $\delta'(q_1, \ldots, q_\ell, \overline{a}) := \delta^\mathcal{A}(q_1, \ldots, q_\ell, (\gamma, \sigma_{i_1}, \ldots, \sigma_{i_k}))$
  end for
  $\mathcal{A}' \leftarrow (Q, \Gamma_\Box \times \Sigma_\Box^m, \delta', F)$
  return $\mathcal{A}'$
end procedure

Algorithm 3 Constructing a Domain Automaton

Input: Parametrised tree-automaton $\mathcal{A}$, $m \in \mathbb{N}$
Output: Tree-automaton $\mathcal{A}_D$ s.t. $L(\mathcal{A}_D) = \{\langle t_p, t_1, \ldots, t_m \rangle \mid t_1, \ldots, t_m \in \mathcal{S}(\mathfrak{d}[t_p])\}$

procedure DOMAIN($\mathcal{A}, m$)
  for $(\gamma, \sigma_1, \ldots, \sigma_m) \in \Gamma \times \Sigma^m, k \in \mathrm{rk}((\gamma, \sigma_1, \ldots, \sigma_m))$ **do**
    for $((q_{11}, \ldots, q_{1m}), \ldots, (q_{k1}, \ldots, q_{km})) \in (Q^m)^k$ **do**
      $\delta_D((q_{11}, \ldots, q_{1m}), \ldots, (q_{k1}, \ldots, q_{km}), (\gamma, \sigma_1, \ldots, \sigma_m)) :=$
        $(\delta(q_{11}, \ldots, q_{1k}, (\gamma, \sigma_1)), \ldots, \delta(q_{m1}, \ldots, q_{mk}, (\gamma, \sigma_m)))$
    end for
  end for
  $\mathcal{A}_D \leftarrow (Q^m, \Gamma \times \Sigma^m, \delta_D, F^m)$
  return $\mathcal{A}_D$
end procedure

In order to analyse Algorithm 1 with respect to the given presentation, we observe the following runtime bounds for the subroutines. We omit the proofs here as they are easily obtainable by a straight forward analysis of the respective routines.

Lemma 5.1. *The procedure* INTERSECT$(\mathcal{A}_1, \mathcal{A}_2)$ *computes a tree-automaton* $\mathcal{A}$ *with* s *states and* $L(\mathcal{A}) = L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ *in time* $\mathcal{O}(|\Sigma| \cdot \mathrm{rk}(\Sigma) \cdot s^{\mathrm{rk}(\Sigma)})$, *where* s *is the number of states reachable from the initial state in the product automaton* $\mathcal{A}_1 \times \mathcal{A}_2$.

Lemma 5.2. *The procedure* DETERMINISE$(\mathcal{A})$ *computes a deterministic tree-automaton* $\mathcal{A}'$ *with* s *states and* $L(\mathcal{A}') = L(\mathcal{A})$ *in time* $\mathcal{O}(|\Sigma| \cdot \mathrm{rk}(\Sigma) \cdot s^{\mathrm{rk}(\Sigma)})$, *where* s *is the number of states reachable from the initial state in the power set automaton of* $\mathcal{A}$.

Algorithm 4 Intersection

Input: Tree-automata $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, F_1)$, $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, F_2)$
Output: Tree-automaton (Q, Σ, δ, F) that recognises $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$

procedure INTERSECT($\mathcal{A}_1$, $\mathcal{A}_2$)
  $Q \leftarrow \emptyset$
  $Q' \leftarrow \{(\delta_1(a), \delta_2(a)) \in Q_1 \times Q_2 \mid a \in \Sigma :\}$
  while $Q \neq Q'$ **do**
    $Q_{\text{last}} \leftarrow Q$
    $Q \leftarrow Q'$
    for $a \in \Sigma, k \in \text{rk}(a)$ **do**
      for $((q_1^1, q_1^2), \ldots, (q_k^1, q_k^2)) \in Q^k \setminus Q_{\text{last}}^k$ **do**
        $Q' \leftarrow Q' \cup \{(\delta_1(q_1^1, \ldots, q_k^1, a), \delta_2(q_1^2, \ldots, q_k^2, a))\}$
      end for
    end for
  end while
  for $a \in \Sigma, k \in \text{rk}(a)$ **do**
    for $(q_1^1, q_1^2), \ldots, (q_k^1, q_k^2) \in Q$ **do**
      $\delta((q_1^1, q_1^2), \ldots, (q_k^1, q_k^2), a) := (\delta_1(q_1^1, \ldots, q_k^1, a), \delta_2(q_1^2, \ldots, q_k^2, a))$
    end for
  end for
  $F \leftarrow Q \cap (F_1 \times F_2)$
  return (Q, Σ, δ, F)
end procedure

Algorithm 5 Projection

Input: Tree-automaton $\mathcal{A} = (Q, \Gamma_\Box \times \Sigma_\Box^{m+1}, \delta, F)$,
Output: NTA $\mathcal{A}'$ s.t. $L(\mathcal{A}') = \{\langle t_p, t_1, \ldots, t_m\rangle \mid \exists t_{m+1} : \langle t_p, t_1, \ldots, t_{m+1}\rangle \in L(\mathcal{A})\}$

procedure PROJECT($\mathcal{A}$)
  $\Delta \leftarrow \{((\gamma, \sigma_1, \ldots, \sigma_m), p) \mid \exists t \in T_{\Sigma_\Box} : \delta(\gamma \otimes \sigma_1 \cdots \otimes \sigma_m \otimes t) = p\}$
  for $q_1, \ldots, q_k \in Q$ **do**
    for $\bar{a} = (\gamma, \sigma_1, \ldots, \sigma_m) \in (\Gamma_\Box \times (\Sigma_\Box)^m)$ **do**
      for $0 \neq k \in \text{rk}(\gamma, \sigma_1, \ldots, \sigma_m)$ **do**
        $\Delta \leftarrow \Delta \cup \{(q_1, \ldots, q_k, \bar{a}, p) \mid \exists \sigma : \delta(q_1, \ldots, q_k, (\bar{a}, \sigma)) = p\}$
      end for
    end for
  end for
  let $q_\Box \notin Q$ be a fresh state
  $q \leftarrow \delta(\Box^{m+1})$
  $\Delta \leftarrow \Delta \cup \{(\Box^{m+1}, q_\Box)\}$
  for $\bar{a} \in (\Gamma_\Box \times \Sigma_\Box^m), k \in \text{rk}(\bar{a}), \ell \leq k$ **do**
    for $q_1, \ldots, q_\ell \in Q$ **do**
      $\Delta \leftarrow \Delta \cup \{(q_1, \ldots, q_\ell, q_\Box^{(k-\ell)}, \bar{a}, p) \mid (q_1, \ldots, q_l, \bar{a}, p) \in \Delta\}$
    end for
    for $\bar{q}_1, \ldots, \bar{q}_{k-\ell+1}$ with $\bar{q}_1 \bar{q}_2 \ldots \bar{q}_{k-\ell+1} \in Q^\ell$ **do**
      $\Delta \leftarrow \Delta \cup \{(\bar{q}_1, q_\Box, \bar{q}_2, \ldots, q_\Box, \bar{q}_{k-\ell+1}, \bar{a}, p) \mid \delta(\bar{q}_1, q, \bar{q}_2, \ldots, q, \bar{q}_{k-\ell+1}, \bar{a}) = p\}$
    end for
  end for
  $\mathcal{A}' \leftarrow (Q \cup \{q_\Box\}, \Gamma_\Box \times (\Sigma_\Box)^m, \Delta, F)$
  return $\mathcal{A}'$
end procedure

Algorithm 6 Determinisation

Input: Non-deterministic tree-Automaton $\mathcal{A} = (S, \Sigma, \Delta, T)$
Output: Deterministic tree-automaton $\mathcal{A}'$ with $L(\mathcal{A}) = L(\mathcal{A}')$

```
 1: procedure DETERMINISE(A)
 2:     Q ← ∅
 3:     Q' ← {s ∈ S | ∃a ∈ Σ : (a, p) ∈ Δ} ⊆ P(S)
 4:     while Q ≠ Q' do
 5:         Q_last ← Q
 6:         Q ← Q'
 7:         for a ∈ Σ, k ∈ rk(a) do
 8:             for (S_1, ..., S_k) ∈ Q^k \ Q^k_last do
 9:                 Q' ← Q' ∪ {s∈S | ∃s̄∈S_1 × ... × S_k : (s̄, a, s) ∈ Δ}
10:             end for
11:         end for
12:     end while
13:     for a ∈ Σ, k ∈ rk(a), S_1, ..., S_k ∈ Q do
14:         δ(S_1, ..., S_k, a) := {s∈S | ∃s̄∈ S_1 × ... × S_k : (s̄, a, s)∈Δ}
15:     end for
16:     F ← Q ∩ {V ⊆ S | V ∩ T ≠ ∅}
17:     return (Q, Σ, δ, S_0, F)
18: end procedure
```

Lemma 5.3. *Let* $\mathcal{A} = (Q, \Gamma_\square \times (\Sigma_\square)^{m+1}, q_0, \delta, F)$ *be a deterministic tree-automaton. The set* $S := \{q \in Q \mid \exists t \in T_\Sigma : \delta(\varepsilon \otimes \underbrace{\varepsilon \otimes \cdots \otimes \varepsilon}_{r \text{ times}} \otimes t) = q\}$ *can be computed in time*

$$\mathcal{O}(|\Sigma| \cdot \mathrm{rk}(\Sigma) \cdot |S|^{\mathrm{rk}(\Sigma)}).$$

5.2 A Presentation Aware Runtime Analysis

The main ingredient to the runtime analysis of Algorithm 1 is the marriage of the Ehrenfeucht-Fraïssé relations (EF-relations) on the presented class of structures and the Myhill-Nerode congruences on the languages which form the presentation. Ehrenfeucht-Fraïssé relations were introduced by Fraïssé in his seminal work [44] as a purely combinatorial characterisation of elementary equivalence. His ideas were later popularised by the appealing game-theoretic presentation given by Ehrenfeucht in [35]. Even the possibility to bound the complexity of certain logical theories using EF-relations was already present in these early works. This technique was later systematically studied by Ferrante and Rackoff (see [39]). They used EF-relations to give upper bounds on the complexity of first-order theories like Presburger Arithmetic or the theory of one-to-one functions.

Klaetke used in [75] the ideas of Ferrante and Rackoff to bound the size of

the automata for linear arithmetic $(\mathbb{R}, +, <)$. Eisinger picked up the techniques and showed in [36] similar bounds for a certain automata based presentation of mixed integer and mixed real addition, respectively (we remark here that his way of presenting the structures by automata differs slightly from our definition of an automatic presentation). Durand-Gasselin and Habermehl recently showed that if a refinement of the EF-relations for a structure $\mathfrak{A}$ is compatible with an automatic presentation of $\mathfrak{A}$ in the sense that these relations are congruences on the encodings of the elements (with respect to concatenation) then the runtime of the standard algorithm for solving the theory of an automatic structure can be bounded in terms of the index these relations. In this section we build upon their work and generalise their result to classes with a uniform tree-automatic presentation. Therefore it is necessary to develop a suitable notion of EF-congruences for our purposes. Besides switching from automatic presentations to uniform tree-automatic presentations, there are a few subtle differences to the definition in [33] in order to make the technique applicable for more presentations.

Let Γ be an advice alphabet and Σ be an input alphabet. In the following we write $\hat{\Sigma}_m$ for $\Gamma_\Box \times (\Sigma_\Box)^m$

Definition 5.3. Let $\mathfrak{c} = (\mathcal{A}, (\mathcal{A}_R)_{R\in\tau})$ be a padding of a good parametrised tree-automatic presentation of a class $\mathcal{C} \subseteq \mathrm{Str}(\tau)$. An Ehrenfeucht-Fraïssé congruence (EF-congruence) for $\mathfrak{c}$ is a collection of equivalence relations $(E^r_m)_{r,m\in\mathbb{N}}$, where $E^r_m \subseteq T_{\hat{\Sigma}_m} \times T_{\hat{\Sigma}_m}$ and for all $r, m \in \mathbb{N}$:

1. The set $T_{\{\Box^{m+1}\}}$ forms a single equivalence-class in E^r_m.
2. If t is a padding of t' then $tE^r_m t'$.
3. All trees $t \in T_{\hat{\Sigma}_m}$ that are not a padding of a convolution of a tuple $(\alpha, t_1, \ldots, t_m) \in T_\Gamma \times (T_\Sigma)^m \setminus T_{\{\Box^{m+1}\}}$ form a single equivalence class in E^r_m.
4. The relation E^r_m separates the trees in $T_{\hat{\Sigma}_m}$ that are a padded convolution of a tuple $(\alpha, t_1, \ldots, t_m)$ such that $(t_1, \ldots, t_m)$ represents a tuple of elements in $\mathcal{S}(\mathfrak{c}[\alpha])$ from those trees in $T_{\hat{\Sigma}_m}$ that are not the convolution of such a tuple.
5. If $t_1, \ldots, t_m \in \mathcal{S}(\mathfrak{c}[\alpha])$, $t'_1, \ldots, t'_m \in \mathcal{S}(\mathfrak{c}[\beta])$, and $\langle \alpha, \bar{t}\rangle E^0_m \langle \beta, \bar{t}'\rangle$ then $(t_1, \ldots, t_m)$ and $(t'_1, \ldots, t'_m)$ satisfy the same atomic formulas in $\mathcal{S}(\mathfrak{c}[\alpha])$ and $\mathcal{S}(\mathfrak{c}[\beta])$, respectively.

6. If $sE_m^{r+1}s'$ for some $s, s' \in T_{\hat{\Sigma}_m}$ then for all $t \in T_{\Sigma_\square}$ there exists a $t' \in T_{\Sigma_\square}$ such that $\langle s, t\rangle E_{m+1}^r \langle s', t'\rangle$.

7. The relation E_m^r respects contexts, i.e. if $tE_m^r t'$ for some $t, t' \in T_{\hat{\Sigma}_m}$ then for all $\hat{\Sigma}_m$-contexts c the trees $c \circ t$ and $c \circ t'$ are also related by E_m^r.

For a function $f : \mathbb{N} \to \mathbb{N}$ we say that an EF-congruence $(E_m^r)_{r,m\in\mathbb{N}}$ is $f(r+m)$ bounded if the index of E_m^r is bounded by $f(r+m)$ for all $r, m \in \mathbb{N}$.

The EF-congruence $(E_m^r)_{r,m\in\mathbb{N}}$ for a presentation $\mathfrak{c}$ refines the indistinguishably relations $(\equiv_r)_{r\in\mathbb{N}}$ on the presented class $\mathcal{C}$. This can be shown using standard game theoretic arguments. We give the proof here for the sake of completeness.

Lemma 5.4. *Let $\mathfrak{c}$ be a uniform tree-automatic presentation of a class $\mathcal{C}$ and $(E_m^r)_{r,m\in\mathbb{N}}$ an EF-congruence with respect to $\mathfrak{c}$. Then for all $\alpha, \alpha' \in P^{\mathfrak{c}}$ and $t_1, t_1' \ldots, t_m, t_m'$ with $t_1, \ldots, t_m \in \mathcal{S}(\mathfrak{c}[\alpha])$ and $t_1', \ldots, t_m' \in \mathcal{S}(\mathfrak{c}[\alpha'])$ the following is true:*

$$\langle \alpha, t_1, \ldots, t_m\rangle E_m^r \langle \alpha', t_1', \ldots, t_m'\rangle$$

implies

$$(\mathcal{S}(\mathfrak{c}[\alpha]), t_1, \ldots, t_m) \equiv_r (\mathcal{S}(\mathfrak{c}[\alpha']), t_1', \ldots, t_m').$$

Proof. We prove the claim by induction on r. For $r = 0$ the claim follows directly from Item 5 of Definition 5.3.

Now suppose the claim has been established for all $m \in \mathbb{N}$ and some $r \in \mathbb{N}$ and consider α, α' and $t_1, t_1' \ldots, t_m, t_m'$ such that the trees t_i and t_i' are elements of $\mathcal{S}(\mathfrak{c}[\alpha])$ and $\mathcal{S}(\mathfrak{c}[\alpha'])$, respectively, for all $1 \leq i \leq m$. We only need to consider tuples with $\langle \alpha, t_1, \ldots, t_m\rangle E_m^{r+1} \langle \alpha', t_1', \ldots, t_m'\rangle$. In this case we show that Duplicator has a winning strategy in the $r+1$ round EF-game on $\mathfrak{A} := (\mathcal{S}(\mathfrak{c}[\alpha]), t_1, \ldots, t_m)$ and $\mathfrak{B} := (\mathcal{S}(\mathfrak{c}[\alpha']), t_1', \ldots, t_m')$. W.l.o.g. assume that Spoiler chooses an element $t_{m+1} \in \mathfrak{A}$. Then because of Item 6 of Definition 5.3 Duplicator can choose a $t_{m+1}' \in \mathfrak{B}$ with $\langle \alpha, t_1, \ldots, t_{m+1}\rangle E_m^r \langle \alpha', t_1', \ldots, t_{m+1}'\rangle$ and, by the induction hypothesis, there is a winning strategy for Duplicator in the r-round EF-game on $(\mathfrak{A}, t_{m+1})$ and $(\mathfrak{B}, t_{m+1}')$, which he can follow from this point onwards. □

As mentioned before, an EF-congruence with respect to some parametrised tree-automatic presentation connects the Myhill-Nerode-congruences of the

languages involved in the presentation with the EF-relations on the presented class. We want to show that the runtime of Algorithm 1 largely depends on how well these relations play along with each other.

Lemma 5.5. *Let Σ be a ranked alphabet, $\sim$ an equivalence relation on T_Σ, and $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$, $\mathcal{A}_2 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ tree-automata. Suppose $t \sim t'$ implies $\delta_i^*(t) = \delta_i^*(t')$ for all $i \in \{1,2\}$ and for all $t, t' \in T_\Sigma$. Then the number of reachable states from the initial state in $\mathcal{A}_1 \times \mathcal{A}_2$ is bounded by the index of $\sim$.*

Proof. Let n be the index of $\sim$ and $Q \subseteq Q_1 \times Q_2$ be the reachable states in $\mathcal{A}_1 \times \mathcal{A}_2$. Suppose $|Q| > n$. For every $q = (q_1, q_2) \in Q$ there exists a tree t_q such that $\delta_i(t_q) = q_i$ for $i \in \{1,2\}$. As $|Q| > n$ there are $q \neq q' \in Q$ with $t_q \sim t_{q'}$. But then $q = (\delta_1^*(t_q), \delta_2^*(t_q)) \overset{t_q \sim t_{q'}}{=} (\delta_1^*(t_{q'}), \delta_2^*(t_{q'})) = q'$. Contradiction! □

Theorem 5.1. *Let $\mathfrak{c} = (\mathcal{A}, (\mathcal{A}_R)_{R \in \tau})$ be a padding of a good parametrised tree-automatic presentation of a class of τ-structures. Suppose there is an $f(r+m)$ bounded EF-congruence $(E_m^r)_{r,m \in \mathbb{N}}$ for $\mathfrak{c}$. Then for every $\varphi(x_1, \ldots, x_m) \in \mathrm{FO}$ of quantifier rank r Algorithm 7 computes the automaton $\mathcal{A}_\varphi$ in time $\mathcal{O}(|\varphi|(|\mathfrak{c}|^{m+r} \cdot f(m+r))^c)$ for some constant c.*

Proof. We prove the claim by induction over the structure of φ. Actually we prove an extended claim, namely that the procedure computes the automaton $\mathcal{A}_\varphi$ in the given time and $\mathcal{A}_\varphi$ has the property $\delta^*_{\mathcal{A}_\varphi}(t) = \delta^*_{\mathcal{A}_\varphi}(t')$ for all $t, t' \in T_{\hat{\Sigma}_m}$ with $tE_m^r t'$.

Case $\varphi = R(x_{i_1}, \ldots, x_{i_k})$**:** Obviously $|\mathcal{A}_\varphi| \leq |\mathfrak{c}|^m$ and therefore there is a fixed polynomial p such that $\mathcal{A}_\varphi$ is constructed in time $p(|\mathfrak{c}|^m)$. Further, by construction, the automaton $\mathcal{A}_\varphi$ is minimal. Let s, s' be two trees from $T_{\hat{\Sigma}_m}$ with $sE_m^0 s'$. If $s, s' \in T_{\{\square^{r+1}\}}$ then for all contexts c we have that $c \circ s \in L(\mathcal{A}_\varphi) \Leftrightarrow c \circ s' \in L(\mathcal{A}_\varphi)$. In order to see this recall that $\mathfrak{c}$ is a padded presentation. Hence from the Myhill-Nerode Theorem for tree-languages (cf. [19]) we can infer that $\delta^*_{\mathcal{A}_\varphi}(s) = \delta^*_{\mathcal{A}_\varphi}(s')$ Then by Property 7 also $(c \circ s) E_m^0 (c \circ s')$ for all $\hat{\Sigma}_m$-contexts. If $c \circ s$ is not a padded convolution of a tuple $(\alpha, \bar{t})$ with $\bar{t} \in \mathcal{S}(\mathfrak{c}[\alpha])$ then because of the first two properties of E_m^0 the same holds for $c \circ s'$. Hence $c \circ s \notin L(\mathcal{A}_\varphi)$ and $c \circ s \notin L(\mathcal{A}_\varphi)$. Otherwise $c \circ s = \langle \alpha, \bar{t} \rangle$ and $c \circ s' = \langle \beta, \bar{t}' \rangle$ and Property 5 yields $\langle \alpha, \bar{t} \rangle \in L(\mathcal{A}_\varphi) \Leftrightarrow \langle \beta, \bar{t}' \rangle \in L(\mathcal{A}_\varphi)$. Again we obtain from Myhill-Nerode Theorem for tree-languages that $\delta^*_{\mathcal{A}_\varphi}(s) = \delta^*_{\mathcal{A}_\varphi}(s')$.

Case $\varphi = \psi(x_1, \ldots, x_m) \wedge \theta(x_1, \ldots, x_m)$**:** Let $\mathcal{A}_\psi$ and $\mathcal{A}_\theta$ be the automata constructed by COMPOSE in the recursion step. By the induction hypothesis, we know that all pairs of tuples t, t' that are related by E_m^r the computation of $\mathcal{A}_\psi$ and $\mathcal{A}_\theta$ reach the same state. Lemma 5.5 tells us that the number of reachable states in $\mathcal{A}_\varphi \times \mathcal{A}_\psi$ is bounded by $f(m+r)$. The automata $\mathcal{A}_\psi$ and $\mathcal{A}_\theta$ are computed in at most

$$d|\psi|(|\mathfrak{c}|^{m+r} \cdot f(m+r))^c + d|\theta|(|\mathfrak{c}|^{m+r} \cdot f(m+r))^c$$

many steps and, according to Lemma 5.1, the computation of $\mathcal{A}_\varphi$ takes at most

$$d'|\hat{\Sigma}_m| \operatorname{rk}(\hat{\Sigma}_m) f(m+r)^{\operatorname{rk}(\hat{\Sigma}_m)}.$$

But $\operatorname{rk}(\hat{\Sigma}_m) = \operatorname{rk}(\Gamma \times \Sigma)$ is constant, $|\mathfrak{c}|^{m+r}$ is an upper bound for $|\hat{\Sigma}_m|$ an $\operatorname{rk}(\hat{\Sigma}_m) \leq c$. Hence the overall runtime is bounded by

$$d(|\psi| + |\theta| + 1)(|\mathfrak{c}|^{m+r} \cdot f(m+r))^c = d|\varphi|(|\mathfrak{c}|^{m+r} \cdot f(m+r))^c.$$

The property $tE_m^r t' \Rightarrow \delta^*_{\mathcal{A}_\varphi}(t) = \delta^*_{\mathcal{A}_\varphi}(t')$ follows directly from the induction hypothesis and the fact that $\delta^*(t) = (\delta^*_{\mathcal{A}_\psi}(t), \delta^*_{\mathcal{A}_\theta}(t))$.

Case $\varphi = \neg\psi(x_1, \ldots, x_m)$**:** By the induction hypothesis the automaton $\mathcal{A}_\psi$ is constructed in time $d|\psi|(|\mathfrak{c}|^{m+r} \cdot f(m+r))^c$. The automaton $\mathcal{A}_D$ is the minimal automaton that recognises exactly the words of the form $\langle \alpha, t_1, \ldots, t_m \rangle$, where $\alpha \in P^{\mathfrak{c}}$ and $t_1, \ldots, t_m$ are elements of $\mathcal{S}(\mathfrak{c}[\alpha])$. Using the properties 3, 4, and 7 of Definition 5.3, we see that for all $t, t' \in T_{\hat{\Sigma}_m}$ with $tE_m^r t'$ and all $\hat{\Sigma}_m$-contexts c it is the case that $c \circ t \in L(\mathcal{A}_D) \Leftrightarrow c \circ t' \in L(\mathcal{A}_D)$. Therefore we can once again apply the lemmata 5.5 and 5.1 to establish that also $\mathcal{A}_\varphi$ is constructed in the right amount of time and has the proclaimed property.

Case $\varphi = \exists x_{m+1} \psi(x_1, \ldots, x_m, x_{m+1})$**:** Let $\mathcal{A}_\psi$ be the automaton that is constructed in the recursion step. Then $\mathcal{A}_\varphi$ is essentially the reachable part of the power-set automaton of the projection automaton derived from $\mathcal{A}_\psi$ under the projection $(\gamma, \sigma_1, \ldots, \sigma_{m+1}) \mapsto (\gamma, \sigma_1, \ldots, \sigma_m)$. Now suppose $sE_m^{r+1}s'$ for some $s, s' \in T_{\hat{\Sigma}_m}$. If $s, s' \in T_{\{\square^{m+1}\}}$ then $\delta^*_{\mathcal{A}_\varphi}(s) = \{q_\square\} = \delta^*_{\mathcal{A}_\varphi}(s')$. Otherwise consider the trees $\operatorname{cut}(s), \operatorname{cut}(s')$. Because of Property 2 and the transitivity of E_m^{r+1} we see that $\operatorname{cut}(s) E_m^{r+1} \operatorname{cut}(s')$ and by the construction of $\mathcal{A}_\varphi$ we have $\delta^*(\operatorname{cut}(s)) = \delta^*(s)$ and $\delta^*(\operatorname{cut}(s')) = \delta^*(s')$. Further, $q \in \delta^*_{\mathcal{A}_\varphi}(\operatorname{cut}(s))$ if and only if there is a $t \in T_{\Sigma_\square}$ such that $\delta^*_{\mathcal{A}_\psi}(\langle \operatorname{cut}(s), t \rangle) = q$. But then, by Property

6 of Definition 5.3, there is also a $t' \in T_{\Sigma_\square}$ with $\langle \mathrm{cut}(s), t\rangle E^r_m \langle \mathrm{cut}(s'), t'\rangle$. By the induction hypothesis $\delta^*_{\mathcal{A}_\psi}(\langle \mathrm{cut}(s'), t'\rangle) = q$ and thus $q \in \delta^*_{\mathcal{A}_\varphi}(\mathrm{cut}(s'))$. This shows that $sE^r_m s'$ implies $\delta^*_{\mathcal{A}_\varphi}(s) = \delta^*_{\mathcal{A}_\varphi}(s')$. Consequently the number of reachable states in the aforementioned power set automaton is bounded by $f(m+r)$. We can now apply the induction hypothesis, Lemma 5.3, and Lemma 5.2 to conclude that the algorithm takes at most $d|\varphi|(\mathfrak{c}^{m+r} f(m+r))^c$ many steps to compute $\mathcal{A}_\varphi$. □

In the following section we will be concerned with classes of finite structures that arise as the closure under direct products. In the previous chapter we have already seen how to construct a presentation of the closure under direct products from a presentation of the original class (see Lemma 4.4). We close this section by showing that also the EF-congruences can (with a certain blow up of the index) be lifted from the original presentation to the presentation of the direct product closure given in the previous section.

In order to ease the process of analysing the complexity of these presentations, we introduce some notations. Let Σ be an alphabet with $\# \notin \Sigma$. A tree $t \in T_{\Sigma_\square}$ is **well-formed** if $v \in \mathrm{dom}_t$ the following holds: $t(v) = \square$ implies $t(vw) = \square$ for all w with $vw \in \mathrm{dom}_t$ and if $v = ui$ then $t(uj) = \square$ for all $j \geq i$ with $ui \in \mathrm{dom}_t$. The n-context-tree $t^\#_n$ is the tree with domain $\mathrm{dom}(t^\#_n) = \{0^k \mid k < n\} \cup \{0^k 1 \mid k+1 < n\}$ and labelling

$$t^\#_n(w) = \begin{cases} \# & ;\text{if } w \in \{0\}^{<n} \\ c_{i+1} & ;\text{if } w = 0^i 1, \text{ with } 0 \leq i < n-1 \\ c_n & ;\text{if } w = 0^{n-1}. \end{cases}$$

With $T^{\#,n}_\Sigma$ we denote the set of all trees that are obtained from $t^\#_n$ by replacing all contexts with trees from T_Σ, that is $T^{\#,n}_\Sigma = \{t^\#_n[c_1/t_1, \ldots, c_n/t_n] \mid t_1, \ldots, t_n \in T_\Sigma\}$. Finally let $T^\#_\Sigma$ be the union of all sets $T^{\#,n}_\Sigma$ with $n \geq 1$. For natural numbers ℓ, m, n we write $m =_\ell n$ if $m = n$ or $m, n \geq \ell$.

Theorem 5.2. *Let $\mathfrak{c}$ be a good tree-automatic presentation of a class $\mathcal{C}$ with associated $f(r+m)$ bounded EF-congruences $(E^r_m)_{r,m\in\mathbb{N}}$. Then there is a parametrised tree-automatic presentation of $\mathcal{C}^\times$ with associated*

$$2^{\mathcal{O}((r+m)f(r+m)\log(f(r+m)))}$$

bounded EF-congruences.

Proof. Let $\mathfrak{c}^\times$ be the presentation of $\mathcal{C}^\times$ that is derived from $\mathfrak{c}$ by the construction from Lemma 4.2. Recall that if P is the set of advice trees for the presentation $\mathfrak{c}$ and $\alpha_1, \ldots, \alpha_n \in P$, then the structure $\mathcal{S}(\mathfrak{c}[\alpha_1]) \times \cdots \times \mathcal{S}(\mathfrak{c}[\alpha_n])$ is presented by the advice $t_n^\#[c_1/\alpha_1, \ldots, c_n/\alpha_n]$ and an element $(t_1, \ldots, t_n) \in \mathcal{S}(\mathfrak{c}[\alpha_1]) \times \cdots \times \mathcal{S}(\mathfrak{c}[\alpha_n])$ is represented by the tree $t_n^\#[c_1/t_1, \ldots, c_n/t_n]$, where $\#$ is a newly introduced letter.

For all $r, m \in \mathbb{N}$ we define a relation $\sim_m^r$ on $T_{(\Gamma_\square \cup \{\#\}) \times (\Sigma_\square \cup \{\#\})^m}$, where $t \sim_m^r t'$ if, and only if, one of the following conditions is true:

i) There are no n, n' such t and t' are the convolution of well-formed trees $\alpha \in T_{\Gamma_\square}^{\#,n}, t_1, \ldots, t_m \in T_{\Sigma_\square}^{\#,n}$ and $\alpha' \in T_{\Gamma_\square}^{\#,n'}, t'_1, \ldots, t'_m \in T_{\Sigma_\square}^{\#,n'}$, respectively.

ii) There are n, n' such t and t' are the convolution of well-formed trees $\alpha \in T_{\Gamma_\square}^{\#,n}, t_1, \ldots, t_m \in T_{\Sigma_\square}^{\#,n}$ and $\alpha' \in T_{\Gamma_\square}^{\#,n'}, t'_1, \ldots, t'_m \in T_{\Sigma_\square}^{\#,n'}$, respectively. That is

$$t = \langle t_n^\#[c_1/\alpha_1, \ldots, c_n/\alpha_n], t_n^\# c_1/t_{1,1}, \ldots, c_n/t_{1,n}], \ldots \\ \ldots, t_n^\#[c_1/t_{m,1}, \ldots, c_n/t_{m,n}]\rangle$$

and

$$t' = \langle t_{n'}^\#[c_1/\alpha'_1, \ldots, c_{n'}/\alpha'_{n'}], t_{n'}^\#[c_1/t'_{1,1}, \ldots, c_{n'}/t'_{1,n'}], \ldots \\ \ldots, t_{n'}^\#[c_1/t'_{m,1}, \ldots, c_{n'}/t'_{m,n'}]\rangle.$$

Then $t \sim_m^r t'$ if for all E_m^r equivalence classes κ

$$|\{i \mid 1 \leq i \leq n, [\langle \alpha_i, t_{1,i}, \ldots, t_{m,i}\rangle]_{E_m^r} = \kappa\}| \\ =_{f(r+m)^r} |\{i \mid 1 \leq i \leq n,' [\langle \alpha'_i, t'_{1,i}, \ldots, t_{m,i}\rangle']_{E_m^r} = \kappa\}|.$$

One easily checks that $\sim_m^r$ is an equivalence relation with index bounded by

$$(f(r+m)^{r+m} + 1)^{f(r+m)} + 1 \in 2^{\mathcal{O}((r+m) f(r+m) \log f(r+m))}$$

for all $r, m \in \mathbb{N}$. What is left is to verify is that $(\sim_m^r)_{r,m \in \mathbb{N}}$ is indeed an EF-congruence of $\mathfrak{c}^\times$. Therefore we check that the collection $(\sim_m^r)_{r,m \in \mathbb{N}}$ has the Properties 3 - 7 described in Definition 5.3. This is done in the lemmata below.

Lemma 5.6. *$T_{\{\square^{m+1}\}}$ forms a single equivalence class in $\sim_m^r$.*

Proof. All trees $t, t' \in T_{\{\square^{m+1}\}}$ fulfil the prerequirement for Condition ii) for $n = n' = 1$. For such trees the requirement of Condition ii) reduces to $tE_m^r t'$. Hence $t \sim_m^r t'$. On the other hand, if $t'' \notin T_{\{\square^{m+1}\}}$ then in order for $t \sim_m^r t''$ to hold there must be an n'' such that t'' is the convolution of well-formed trees $\alpha'' \in T_{\Gamma_\square}^{\#,n''}, t''_1, \ldots, t''_m \in T_{\Sigma_\square}^{\#,n''}$. But if $n'' = 1$ then $t \not\sim_m^r t''$ because $t \not E_m^r t''$ and if $n'' > 1$ then there must be a E_m^r-class κ such that the counting property of Condition ii) is violated (observe that $f(r+m) \geq 2$). Hence $t \not\sim_m^r t''$. □

Lemma 5.7. *If t is a padding of t' then $t \sim_m^r t'$.*

Proof. Let t be a padding of t'. If t' is not a padding of a convolution of well-formed trees from $T_{\Gamma_\square}^{\#,n}$ and $T_{\Sigma_\square}^{\#,n}$ then t is also not of this form and therefore $t \sim_m^r t'$ by Condition i). Finally, if

$$t' = \langle t_n^\#[c_1/\alpha'_1, \ldots, c_n/\alpha'_n], t_n^\#[c_1/t'_{1,1}, \ldots, c_n/t'_{1,n}], \ldots \\ \ldots, t_n^\#[c_1/t'_{m,1}, \ldots, c_n/t'_{m,n}]\rangle.$$

then

$$t' = \langle t_n^\#[c_1/\alpha_1, \ldots, c_n/\alpha_n], t_n^\#[c_1/t_{1,1}, \ldots, c_n/t_{1,n}], \ldots \\ \ldots, t_n^\#[c_1/t_{m,1}, \ldots, c_n/t_{m,n}]\rangle,$$

where every α_i is a padding of α'_i and every $t_{i,j}$ is a padding of $t'_{i,j}$. Then $\langle \alpha_i, t_{1,i}, \ldots, t_{m,i}\rangle E_m^r \langle \alpha'_i, t'_{1,i}, \ldots, t'_{m,i}\rangle$ for all $i \leq n$. Hence we get $t \sim_m^r t'$ by Condition ii).

□

Lemma 5.8. *All trees $t \notin T_{\{\square^{m+1}\}}$ that are not a padding of a convolution of a tuple $(\alpha, t_1, \ldots, t_m)$ form a single equivalence class in $\sim_m^r$.*

Proof. Condition i) ensures that all such trees are in a single equivalence class.

□

Lemma 5.9. *The relation $\sim_m^r$ separates the trees that are the convolution of a tuple $(\alpha, t_1, \ldots, t_n)$ such that $(t_1, \ldots, t_m)$ represents a tuple of elements in $\mathcal{S}(\mathfrak{c}^\times[\alpha])$ from those trees that are not the convolution of such a tuple.*

Proof. Suppose $t = \langle \alpha, t_1, \ldots, t_m \rangle$ is a padded convolution of a tuple with $\alpha \in P^{\mathfrak{c}^\times}$ and $(t_1, \ldots, t_m) \in \mathcal{S}(\mathfrak{c}^\times[\alpha])$ and suppose t' is not the convolution of such a tuple. If t' is not a padded convolution, then none of the two conditions holds for t and t' and they are not equivalent. Otherwise there are $n, n' \geq 1$ such that

$$t = \langle t_n^\#[c_1/\alpha_1, \ldots, c_n/\alpha_n], t_n^\#[c_1/t_{1,1}, \ldots, c_n/t_{1,n}], \ldots, t_n^\#[c_1/t_{m,1}, \ldots, c_n/t_{m,n}] \rangle$$

and

$$t' = \langle t_{n'}^\#[c_1/\alpha'_1, \ldots, c_{n'}/\alpha'_{n'}], t_{n'}^\#[c_1/t'_{1,1}, \ldots, c_{n'}/t'_{1,n'}], \ldots, t_{n'}^\#[c_1/t'_{m,1}, \ldots, c_{n'}/t'_{m,n'}] \rangle.$$

From our assumption about t and t' we know that $\alpha_i \in P^{\mathfrak{c}}$ and $t_{1,i}, \ldots t_{m,i} \in \mathcal{S}(\mathfrak{c}[\alpha_i])$ for all $1 \leq i \leq n$ and there is a $1 \leq j \leq n'$ with $\alpha'_j \notin P^{\mathfrak{c}}$ or $\alpha'_j \in P^{\mathfrak{c}}$ but $t'_{\ell,j} \notin \mathcal{S}(\mathfrak{c}[\alpha_i])$ for some $1 \leq \ell \leq m$.

But then $\langle \alpha_i, t_{1,i}, \ldots, t_{m,i} \rangle \not\mathrel{E}^r_m \langle \alpha'_j, t'_{1,j}, \ldots, t'_{m,j} \rangle$, since the relation E^r_m fulfils Property 4 of Definition 5.3. Hence t and t' do not fulfil condition ii) and therefore $t \not\sim^r_m t'$. □

Lemma 5.10. *If $t_1, \ldots, t_m \in \mathcal{S}(\mathfrak{c}[\alpha])$, $t'_1, \ldots, t'_m \in \mathcal{S}(\mathfrak{c}[\beta])$, and $\langle \alpha, \bar{t} \rangle \sim^0_m \langle \beta, \bar{t}' \rangle$ then $(t_1, \ldots, t_m)$ and $(t'_1, \ldots, t'_m)$ satisfy the same atomic formulas in $\mathcal{S}(\mathfrak{c}[\alpha])$ and $\mathcal{S}(\mathfrak{c}[\beta])$, respectively.*

Proof. Suppose

- $\alpha = t_n^\#[c_1/\alpha_1, \ldots, c_n/\alpha_n], \beta = t_k^\#[c_1/\beta_1, \ldots, c_k/\beta_k] \in P^{\mathfrak{c}^\times}$,
- $t_i = t_n^\#[c_1/t_{i,1}, \ldots, c_n/t_{i,n}] \in \mathcal{S}(\mathfrak{c}^\times[\alpha])$ for $i \in \{1, \ldots, m\}$, and
- $t'_i = t_k^\#[c_1/t'_{i,1}, \ldots, c_k/t'_{i,k}] \in \mathcal{S}(\mathfrak{c}^\times[\beta])$ for $i \in \{1, \ldots, m\}$.

We show that if $(t_1, \ldots, t_m)$ and $(t_1, \ldots, t'_m)$ do not fulfil the same atomic propositions in $\mathcal{S}(\mathfrak{c}^\times[\alpha])$ and $\mathcal{S}(\mathfrak{c}^\times[\beta])$, respectively, then they are not $\sim^0_m$-equivalent. Consider an arbitrary atomic formula $Rx_{i_1} \ldots x_{i_r}$ and suppose $\mathcal{S}(\mathfrak{c}^\times[\alpha]) \models Rt_{i_1} \ldots, t_{i_r}$ and $\mathcal{S}(\mathfrak{c}^\times[\alpha']) \not\models Rt'_{i_1} \ldots, t'_{i_r}$. Then by definition $\mathcal{S}(\mathfrak{c}[\alpha_j]) \models Rt_{j,i_1} \ldots t_{j,i_r}$ for all $1 \leq j \leq n$ but $\mathcal{S}(\mathfrak{c}[\beta_\ell]) \not\models Rt'_{\ell,i_1} \ldots t'_{\ell,i_r}$ for some $1 \leq \ell \leq k$. Consequently $\langle \alpha_j, t_{j,1}, \ldots, t_{j,m} \rangle \not\mathrel{E}^0_m \langle \beta_\ell, t_{\ell,1}, \ldots, t_{\ell,m} \rangle$ for all $1 \leq j \leq n$ and therefore $\langle \alpha, t_1, \ldots, t_m \rangle \not\sim^0_m \langle \beta, t'_1, \ldots, t'_m \rangle$. □

Lemma 5.11. *If $s \sim_m^{r+1} s'$ then for all $t \in T_{(\Sigma\cup\{\#\})_\Box}$ there exists a $t' \in T_{(\Sigma\cup\{\#\})_\Box}$ such that $\langle s,t\rangle \sim_{m+1}^{r} \langle s',t'\rangle$.*

Proof. Let s, s' be two trees from $T_{(\Gamma\cup\{\#\})_\Box\times(\Sigma\cup\{\#\})_\Box^m}$ such that $s \sim_m^{r+1} s'$ and $t \in T_{(\Sigma\cup\{\#\})_\Box}$. We distinguish two cases.
If s and s' are not convolutions of well-formed trees then $\langle s,t\rangle$ as well as $\langle s',t\rangle$ are not convolutions of well-formed trees for every tree t. Hence Property 6 is fulfilled for such trees.
In the other case s and s' must be convolutions of well-formed trees, that is

$$s = \langle t_n^{\#}[c_1/\alpha_1,\cdots,c_n/\alpha_n], t_n^{\#}[c_1/t_{1,1},\cdots,c_n/t_{1,n}],\ldots,t_n^{\#}[c_1/t_{m,1},\cdots,c_n/t_{m,n}]\rangle$$

and

$$s' = \langle t_k^{\#}[c_1/\alpha'_1,\cdots,c_k/\alpha'_k], t_k^{\#}[c_1/t'_{1,1},\cdots,c_k/t'_{1,k}],\ldots,t_k^{\#}[c_1/t'_{m,1},\cdots,c_k/t'_{m,k}]\rangle$$

for some $n,k \geq 1$ and trees $\alpha_i, \alpha'_j \in T_\Gamma$ and $t_{i,j}, t'_{s,t} \in T_\Sigma$. Let t_{m+1} be an arbitrary tree from $T_{(\Sigma\cup\{\#\})_\Box}$. If $t_{m+1} \notin T_{\Sigma_\Box}^{\#,n}$ take some tree t'_{m+1} that is not in $T_{\Sigma_\Box}^{\#,k}$. Then $\langle s,t_{m+1}\rangle \sim_m^r \langle s',' t'_{m+1}\rangle$ because of Condition i). Otherwise $t_{m+1} = t_n^{\#}[c_1/t_{m+1,1},\ldots c_n/t_{m+1,n}]$. For every E_m^r equivalence class κ let

$$\kappa(s) = \{i \in \{1,\ldots,n\} \mid [\langle \alpha_i, s_{i,1},\ldots,s_{i,m}\rangle]_{E_m^r} = \kappa\}.$$

Let $X_1^\kappa,\ldots,X_{\ell_\kappa}^\kappa$ be the partition of $\kappa(s)$ with respect to the E_{m+1}^r equivalence classes of $\{\langle \alpha, t_{1,i},\ldots,t_{m+1,i}\rangle \mid i \in \kappa(t)\}$. Because $s \sim_m^{r+1} s'$ it is ensured that

$$|\kappa(s)| =_{f(m+r+1)^{r+1}} |\kappa(s')|$$

and therefore we can find a partition $Y_1^\kappa,\ldots,Y_{\ell_\kappa}^\kappa$ of $\kappa(s')$ with

$$|X_i^\kappa| =_{f(r+m+1)^r} |Y_i^\kappa|$$

(if $|\kappa(s)| < f(m+r+1)^{r+1}$ partition $\kappa(s')$ according to some bijection between $\kappa(s)$ and $\kappa(s')$. Otherwise, because $\ell_\kappa < f(m+r+1)$ there is at least one X_i^κ with $|X_i^\kappa| \geq f(m+r+1)$ which also ensures that we can find such a partition).
By construction, $\langle \alpha, t_{1,i},\ldots,t_{m,i}\rangle E_m^{r+1} \langle \alpha', t'_{1,j},\ldots,t'_{m,j}\rangle$ whenever $i \in X_k^\kappa$ and $j \in Y_k^\kappa$. Thus $\langle \alpha, t_{1,i},\ldots,t_{m+1,i}\rangle E_{m+1}^r \langle \alpha', t'_{1,j},\ldots,t'_{m+1,j}\rangle$ for some appropriate $t'_{m+1,j}$. Now choose $t'_{m+1} = t_k^{\#}[c_1/t'_{m+1,1},\ldots,c_k/t'_{m+1,k}]$. By construction $\langle s,t_{m+1}\rangle \sim_{m+1}^r \langle s',t'_{m+1}\rangle$ due to Condition ii). □

In order to show that Property 7 is fulfilled, it is convenient to define a kind of convolution for contexts. For $i \in \{1, \dots, n\}$ let c_i be an Σ_i-context such such that $x(c_1) = x(c_2) = \cdots = x(c_n)$. Then $\langle c_1, \dots, c_n \rangle_c$ is the $((\Sigma_1)_\Box \times \cdots \times (\Sigma_n)_\Box)$-context with $\mathrm{dom}(\langle c_1, \dots, c_n \rangle_c) = \bigcup_{1 \leq i \leq n} \mathrm{dom}_(c_i)$ and

$$\langle c_1, \dots, c_n \rangle_c(w) = \begin{cases} (\sigma_1, \dots, \sigma_n) & \text{if } w \neq x(c_1) \\ & \text{where } \sigma_i = c_i(w) \text{ if } w \in \mathrm{dom}(c_i) \\ & \text{and } \sigma_i = \Box \text{ otherwise} \\ x & \text{otherwise} \end{cases}$$

Lemma 5.12. *The relations $(\sim^r_m)_{r,m \in \mathbb{N}}$ respect contexts.*

Proof. Suppose $s \sim^r_m s'$ and let c be a $((\Gamma \cup \{\#\})_\Box \times (\Sigma \cup \{\#\})^m_\Box)$-context. We can assume that s and s' are equivalent due to Condition ii) and that

$$c = \langle t^\#_n[c_1/\alpha_1, \dots, c_n/\alpha_n], t^\#_n[c_1/t_{1,1}, \dots, c_n/t_{1,n}], \dots, t^\#_n[c_1/t_{m,1}, \dots, c_n/t_{m,n}] \rangle_c$$

for some $n \geq 1$ and $\langle \alpha_i, t_{1,i}, \dots t_{m,i} \rangle_c$ is a $(\Gamma_\Box \times \Sigma^m_\Box)$-context for exactly one $1 \leq i \leq n$ (because in any other case $c \circ t$ and $c \circ t'$ equivalent by Condition i)). Fix this i and let $c' := \langle \alpha_i, t_{1,i}, \dots t_{m,i} \rangle$.

There two cases that we need to consider. First if s, s' are elements of $T^{\#,1}_{\Gamma_\Box} \otimes (T^{\#,1}_{\Sigma_\Box})^{\otimes m}$ $(= T_{\Gamma_\Box} \otimes (T_{\Sigma_\Box})^{\otimes m})$. Then the requirement of Condition ii) reduces to $sE^r_m s'$. But then $c' \circ tE^r_m c' \circ s'$ and hence $c \circ s \sim^r_m c \circ s'$.

Otherwise we can even assume that

$$c = \langle t^\#_n[c_1/\alpha_1, \dots, c_n/x], t^\#_n[c_1/t_{1,1}, \dots, c_n/x], \dots, t^\#_n[c_1/t_{m,1}, \dots, c_n/x] \rangle_c$$

(again otherwise we would get equivalence by Condition i)). But then

$$\begin{aligned} c \circ s = \langle & t^\#_{n+k-1}[c_1/\alpha_1, \dots, c_{n-1}/\alpha_{n-1}, c_n/\beta_1, \dots, c_{n+k-1}/\beta_k], \\ & t^\#_{n+k-1}[c_1/t_{1,1}, \dots, c_{n-1}/t_{1,n-1}, c_n/s_{1,1}, \dots, c_{n+k-1}/s_{1,k}], \\ & \vdots \\ & t^\#_{n+k-1}[c_1/t_{m,1}, \dots, c_{n-1}/t_{m,n-1}, c_n/s_{m,1}, \dots, c_{n+k-1}/s_{m,k}] \rangle \end{aligned}$$

and

$$\begin{aligned} c \circ s' = \langle & t^{\#}_{n+k-1}[c_1/\alpha_1, \ldots, c_{n-1}/\alpha_{n-1}, c_n/\beta'_1, \ldots, c_{n+k-1}/\beta'_{k'}], \\ & t^{\#}_{n+k-1}[c_1/t_{1,1}, \ldots, c_{n-1}/t_{1,n-1}, c_n/s'_{1,1}, \ldots, c_{n+k-1}/s'_{1,k'}], \\ & \vdots \\ & t^{\#}_{n+k-1}[c_1/t_{m,1}, \ldots, c_{n-1}/t_{m,n-1}, c_n/s'_{m,1}, \ldots, c_{n+k-1}/s'_{m,k'}]\rangle \end{aligned}$$

Using that s and s' are equivalent by Condition ii), it is easy to see that also $c \circ t$ and $c \circ t'$ are equivalent. □

The preceding lemmata show that $(\sim^r_m)_{r,m\in\mathbb{N}}$ is an EF-congruence for $\mathfrak{c}^\times$, which completes the proof of Theorem 5.2. □

Another important class of operations under which uniform tree-automatic presentations are closed are parametrised first-order interpretations. Also in this case the complexity of the EF-congruence grows rather tamely under these operations.

Lemma 5.13. *Let $\mathfrak{c}$ be a good tree-automatic presentation of a class $\mathcal{C}$ of τ-structures and $\mathcal{I}$ be a parametrised τ-to-σ-interpretation of width ℓ that interprets for every $\mathfrak{A} \in \mathcal{C}$ a structure $\mathcal{I}(\mathfrak{A})$. Further let c be the maximal quantifier rank of any of the formulas in $\mathcal{I}$. If there is an $f(r+m)$ bounded EF-congruence for $\mathfrak{c}$ then there is a uniform tree-automatic presentation $\mathcal{I}^{\mathfrak{c}}$ of the class $\mathcal{I}^{\mathcal{C}} = \{\mathcal{I}^{\mathfrak{A}}(a) \mid \mathfrak{A} \in \mathcal{C}, a \in A\}$ with $g(r+m) := f((\ell+c)(r+m)+c)$ bounded EF-congruence.*

5.3 FPT Model Checking With Elementary Parameter Dependence

The runtime analysis from Section 5.1 not only enables us to show that first-order model checking is fixed parameter tractable on several classes of finite structures, but also gives us elementary bounds on the parameter dependence. In the following we write $\exp_k(x)$ for the k-fold tower of twos function applied to x, that is

$$\begin{aligned} \exp_0(x) &= x \text{ and} \\ \exp_{k+1}(x) &= 2^{\exp_k(x)}. \end{aligned}$$

Theorem 5.3. *Let $\mathfrak{c}$ be a padded good tree-automatic presentation such that Algorithm 7 computes in time $T(|\varphi|)$ from $\mathfrak{c}$ the corresponding automaton $\mathcal{A}_\varphi$. Suppose for a class of finite structures $\mathcal{C}$ there is a function $f : \operatorname{code}(\mathcal{C}) \to \Sigma^*$ that computes in time $F(|w|)$ for every $w \in \operatorname{code}(\mathfrak{A})$ with $\mathfrak{A} \in \mathcal{C}$ a tree α with $\mathfrak{A} \cong \mathcal{S}(\mathfrak{c}[\alpha])$. Then* FO *model checking on $\mathcal{C}$ is decidable in time*

$$\mathcal{O}(T(|\varphi|) \cdot |f(w)| + F(|w|)).$$

Proof. The runtime is achieved by the straight forward method of checking whether $\mathcal{A}_\varphi$ accepts $f(w)$. □

Boolean Algebras Our simplest application of Theorem 5.2 and Theorem 5.3 is for the class of all finite Boolean algebras. It is well known that every finite Boolean algebra is isomorphic to a finite direct power of the two element Boolean algebra. Especially, every finite Boolean algebra contains exactly 2^n elements for some $n \geq 1$ and every finite Boolean algebra is uniquely determined by the number of elements. Because of this simple structure it is natural to consider succinct encodings of Boolean algebras as inputs. In the following we will assume that a Boolean algebra is given by the number of atoms, encoded in unary. In other words, a finite Boolean algebra $\mathfrak{B} = (B, \cap, \cup, \bar{}, \mathbf{0}, \mathbf{1})$ is encoded by the string $1^{\log |B|}$.

Theorem 5.4. *First-order model checking is fixed parameter tractable on the class of all finite Boolean algebras. Given a Boolean algebra $\mathfrak{B}$ and an* FO *sentence φ one can decide in time*

$$\exp_2(\operatorname{poly}(|\varphi|)) \log |\mathfrak{B}|$$

whether $\mathfrak{B} \models \varphi$.

Proof. The single element class that contains only the two element Boolean algebra $\mathfrak{B}_2 = (\{\mathbf{0}, \mathbf{1}\}, \cap, \cup, \bar{}, \mathbf{0}, \mathbf{1})$ has the trivial automatic presentation $\mathfrak{c}$ over the advice alphabet $\Gamma = \{a\}$ and the alphabet $\Sigma = \{0, 1\}$ with $\operatorname{rk}(a) = \operatorname{rk}(0) = \operatorname{rk} 1 = \{0\}$. The advice a (the tree of height 0 where the root is labelled with a) represents $\mathfrak{B}_2$ and the elements $\mathbf{0}$ and $\mathbf{1}$ are represented by 0 and 1, respectively. One checks that the relations $(E^r_m)_{r,m \in \mathbb{N}}$ where

$$\begin{aligned} tE^r_m t' :\Leftrightarrow \quad & \exists t_1, \ldots, t_m \in \{0,1\}(\operatorname{cut}(t) = \langle a, t_1, \ldots, t_m \rangle = \operatorname{cut}(t')) \\ & \vee \forall t_1, \ldots, t_m \in \{0,1\}(\operatorname{cut}(t), \operatorname{cut}(t') \neq \langle a, t_1, \ldots, t_m \rangle) \end{aligned}$$

are an EF-congruence with respect to $\mathfrak{c}$ and the index of E_m^r is bounded by $f(r+m) = 2^{r+m} + 2$ for all $r, m \in \mathbb{N}$.

As mentioned before, every finite Boolean algebra is a finite direct product of $\mathfrak{B}_2$ and hence $\mathfrak{c}^\times$ is a uniform presentation of the class of all finite Boolean algebras. According to Theorem 5.2, $\mathfrak{c}^\times$ has an EF-congruence bounded by $f'(r+m) \in 2^{\mathcal{O}((r+m+1)\left(2^{r+m}+2\right)\log(2^{r+m}+2))} \subseteq 2^{2^{\mathrm{poly}(|\varphi|)}}$. Using Theorem 5.3, we conclude that for a sentence φ of quantifier-rank r Algorithm 1 constructs the corresponding automaton $\mathcal{A}_\varphi$ in time

$$\mathcal{O}\left(|\varphi|\left(|\mathfrak{c}^\times|^{m+r} \cdot 2^{2^{\mathrm{poly}(|\varphi|)}}\right)^c\right) \subseteq 2^{2^{\mathrm{poly}(|\varphi|)}}$$

(because $|\mathfrak{c}^\times|$ is constant). Note that the Boolean algebra with n atoms is represented by the tree $t_n^\#[c_1/a, \ldots, c_n/a]$ in $\mathfrak{c}^\times$. We can therefore transform the encoding of the Boolean algebra into the tree-representation in linear time. Finally the claim follows from Theorem 5.3. □

With respect to the height of the tower of twos in the parameter dependence this result is probably optimal, as stated by the following theorem.

Theorem 5.5. *There is no algorithm that solves the model checking problem for finite Boolean algebras in time*

$$2^{\mathrm{poly}(|\varphi|)} \cdot \log|\mathfrak{B}|$$

unless $\bigcup_{c\in\mathbb{N}} \mathrm{STA}(*, 2^{cn}, n) = \mathrm{EXP}$.

Proof. It is known that the theory of all finite Boolean algebras is complete for the class $\bigcup_{c\in\mathbb{N}} \mathrm{STA}(*, 2^{cn}, n)$. Further, using Lemma 5.4 and the computations of Theorem 5.4, we see that there is a constant c such that if $\mathfrak{B}$ and $\mathfrak{B}'$ are two Boolean algebras with at least 2^{r^c} many atoms then $\mathfrak{B} \equiv_r \mathfrak{B}'$. To check that a sentence φ of quantifier rank r belongs to the theory of finite Boolean algebras it is sufficient to check whether every finite Boolean algebra with at most 2^{r^c} many atoms models φ. If we could perform model-checking in time $\mathcal{O}(2^{\mathrm{poly}\,|\varphi|} \cdot \log|\mathfrak{B}|)$ we could hence solve the theory of finite Boolean algebras in time $\mathcal{O}\left(2^{\mathrm{poly}(|\varphi|)} \cdot \sum_{i=1}^{2^{r^c}} i\right) \subseteq 2^{\mathrm{poly}(|\varphi|)}$, which implies $\bigcup_{c\in\mathbb{N}} \mathrm{STA}(*, 2^{cn}, n) = \mathrm{EXP}$.

□

Remark 5.1. Needless to say than an analogue of Theorem 5.4 also holds if the Boolean algebra is encoded traditionally by the multiplication tables of the operators. Obviously one can compute the succinct encoding from the traditional encoding efficiently by simply counting the number of atoms.

However, one could also argue that our encoding for the Boolean algebras is not even optimal. Indeed a finite Boolean algebra $\mathfrak{B}$ can be encoded by a word of length $\lceil \log \log |\mathfrak{B}| \rceil$ when we encode the number of atoms by its binary expansion. In this case our algorithm would not have a polynomial runtime in the size of the encoding of the structure because the advice would be of exponential size. However we could slot in a kernelisation procedure ahead. As we already explained in the proof of Theorem 5.5, there is a fixed polynomial p such that all finite Boolean algebras with at least $2^{p(k)}$ atoms are indistinguishable by a first-order Formula of quantifier rank at most k. In turn we can compute for a given finite Boolean Algebra $\mathfrak{B}$ and a natural number k an advice α of size $\mathcal{O}\left(2^{2^{p(k)}}\right)$ such that $\mathcal{S}(\mathfrak{c}[\alpha]) \equiv_k \mathfrak{B}$ (where $\mathfrak{c}$ is the presentation of the finite Boolean algebras constructed in Theorem 5.4). Because we are more interested in the application of automata based presentations than on encoding issues we will not work out the details here.

Finite Groups Probably a bit more interesting is the class of all finite groups. In [42], Grohe posed the question on which classes of finite groups first-order model checking is fixed parameter tractable. In order to tackle this question we propose a structural parameter on finite groups. The Remak-Krull-Shimidt Theorem [64] states that a factorization of $\mathfrak{G} = \mathfrak{G}_1 \otimes \mathfrak{G}_2 \otimes \cdots \otimes \mathfrak{G}_n$ into indecomposable subgroups $\mathfrak{G}_i$ is unique up to permutation and isomorphism of the occurring subgroups for any finite group $\mathfrak{G}$. Therefore the size of the largest non-abelian subgroup in such a factorisation is uniquely determined. This leads to the following parameter.

Definition 5.4. Let $\mathfrak{G}$ be a finite group. The *non-abelian decomposition width* of $\mathfrak{G}$ is

$$\mathrm{dw}(\mathfrak{G}) = \max(\{|\mathfrak{G}'| \mid \mathfrak{G}' \text{ is non-abelian, indecomposable, and } \mathfrak{G} \cong \mathfrak{G}' \oplus \mathfrak{G}''\})$$

the size of a maximal non-abelian indecomposable factor of $\mathfrak{G}$.

Note that the finite abelian groups are exactly the groups with non-abelian decomposition width one. As for the case of Boolean algebras, finite abelian groups have a quite simple structure. By the classification of finitely generated abelian groups every finite abelian group $\mathfrak{G}$ is isomorphic to a finite sum of finite cyclic groups. That is $\mathfrak{G} \cong \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_k}$ for some $k \geq 1$ and $n_1, \ldots, n_k \geq 1$. Hence, a finite abelian group can be encoded by a sequence of natural numbers $(n_1, \ldots, n_k)$. Bova and Martin have independently shown in [18] that first-order model-checking is FPT on the class of all finite abelian groups. Their algorithm uses a quantifier elimination procedure. However, their analysis of the algorithm only yields a non-elementary parameter dependence. We will show that the automata based approach yields an algorithm with elementary parameter dependence.

Theorem 5.6. FO-*model-checking is* FPT *on the class of all finite abelian groups. More precisely one can decide, given a finite abelian group* $\mathfrak{G}$ *and a formula* $\varphi \in \mathrm{FO}$*, in time*

$$\mathcal{O}\left(\exp_4(\mathrm{poly}(|\varphi|)) \cdot \log|\mathfrak{G}|\right)$$

whether $\mathfrak{G} \models \varphi$.

Proof. Durand-Gasselin and Habermehl gave in an automatic presentation $\mathfrak{d}$ of Presburger arithmetic and proved that there is a $f(m+r) = \exp_3(c(m+r))$ bounded EF-congruence with respect to $\mathfrak{d}$ for some $c \in \mathbb{N}$ [33, Lemma 15].

We construct a uniform presentation of all finite cyclic groups from $\mathfrak{d}$ by a parametrised first-order interpretation $\mathcal{I} = (\delta(n,x), \varphi_\circ(n,x,y,z))$. It is a well known fact that such an interpretation exists. For instance, we might choose the formulas in $\mathcal{I}$ such that

$$\begin{aligned} \delta(n,x) &\equiv \exists c(c+c \neq c \wedge x+c=n) \\ \varphi_\circ(n,x,z) &\equiv (x+y<n \wedge x+y=z) \vee (x+y>n \wedge x+y=n+z). \end{aligned}$$

Then $\mathcal{I}^{(\mathbb{N},+)}(n) \cong \mathbb{Z}_n$ for all $n \in \mathbb{N}$ and therefore $\mathcal{I}^{\mathfrak{d}}$ is a uniform presentation of the class of all finite cyclic groups. By Lemma 5.13 there is a constant c' such that $\mathcal{I}^{\mathfrak{d}}$ has a $g(r+m) = \exp_3(c'(r+m))$ bounded EF-congruence. Further $(\mathcal{I}^{\mathfrak{d}})^\times$ is a uniform presentation of the class of all finite abelian groups and Theorem 5.2 tells us that it has a $(g(r+m)^r)^{g(r+m)} \in \exp_4(\mathrm{poly}(|\varphi|))$ bounded EF-congruence. Note that in $(\mathcal{I}^{\mathfrak{d}})^\times$ a group $\mathfrak{G} \cong \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_k}$ is

represented by the tree $t_k^{\#}[c_1/\operatorname{bin}^R(n_1), \ldots, c_k/\operatorname{bin}^R(n_k)]$. Of course this tree can trivially be computed in linear time from the encoding $(n_1, \ldots, n_k)$ of G. By applying Theorem 5.3 we conclude that our algorithm solves the model-checking problem for finite abelian groups in time $\mathcal{O}(\exp_4(\text{poly}(|\varphi|)) \cdot \log|G|)$. □

Remark 5.2. Although the encoding of an abelian group by the orders of its cyclic factors makes it trivial to compute the tree-presentation because it makes the relevant structural properties of the group explicit, it is still true that an analog of Theorem 5.6 holds if the group is encoded by its multiplication table. Indeed Algorithm 7 provides a simple procedure to compute the cyclic factors of the group in linear time. To see this, note that if g is an element

Algorithm 7 Decomposing a Finite Abelian Group into Cyclic Factors

Input: Finite abelian group $\mathfrak{G}$
Output: String $\operatorname{bin}(n_1)\# \cdots \#\operatorname{bin}(n_k)$ such that $\mathfrak{G} \cong \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_k}$

```
procedure DECOMPOSE(𝔊)
    Compute g with |g| maximal in 𝔊
    if ⟨g⟩ = 𝔊 then
        return bin(|g|)
    else
        w ← DECOMPOSE(𝔊/⟨g⟩)
        return bin(|g|)#w
    end if
end procedure
```

of maximal order in a finite abelian group $\mathfrak{G}$ then $\mathfrak{G} \cong \langle g\rangle \oplus \mathfrak{G}/\langle g\rangle$. The Algorithm 7 therefore computes a representant of a decomposition of $\mathfrak{G}$ into cyclic factors. The computation of an element with maximal order can be done in time $\mathcal{O}(|\mathfrak{G}|^2)$ by computing the order of every element. The group $\mathfrak{G}/\langle g\rangle$ can also be computed in time $\mathcal{O}(|\mathfrak{G}|^2)$ by computing the multiplication table on the cosets of $\langle g\rangle$. Finally the procedure DECOMPOSE($\mathfrak{G}$) is called at most $\log_2(|\mathfrak{G}|)$ times because $|\mathfrak{G}/\langle g\rangle| = |\mathfrak{G}|/|g|$. Together this gives a running time of $\mathcal{O}(|\mathfrak{G}|^2 \cdot \log(|\mathfrak{G}|))$, which is linear in the size of the multiplication table.

Finally, we turn out attation to encoding issues. As it was the case for Boolean algebras, there is an encoding of finite abelian groups, which in some cases allows for a considerably more succinct presentation. More precisely an

abelian group $\mathfrak{G} \cong (\mathbb{Z}_{n_1})^{k_1} \times \cdots \times (\mathbb{Z}_{n_\ell})^{k_\ell}$ can be encoded by the tuple of pairs $((n_1, k_1), \ldots, (n_\ell, k_\ell))$. Again, using this encoding we would not directly obtain an FPT-algorithm from our method. However, using the same argument as for the Boolean algebras, for some fixed polynomial p we can turncate the second components of each pair to $\exp_3(p(r))$ in a preprocessing step, where r is the quantifier rank of the formula under consideration. Again we will leave the details of this approach to the reader.

We extend our ideas from abelian groups to groups of bounded non-abelian decomposition width.

Theorem 5.7. *First-order model checking is* FPT *on the class of all finite groups with bounded non-abelian decomposition width. More precisely there exists a constant c such that we can decide in time*

$$\mathcal{O}(\exp_4(\mathrm{poly}(|\varphi|)) \cdot \log|\mathfrak{G}| + |\mathfrak{G}|^c)$$

whether $\mathfrak{G} \models \varphi$.

Proof. First we build a trivial presentation for the groups of order at most d. Let $\mathfrak{G}_1, \ldots, \mathfrak{G}_n$ be an enumeration of the non-abelian groups of size at most d (up to isomorphism). The advice alphabet is $\Gamma = \{g_1, \ldots, g_n\}$. The input alphabet Σ is extended by new letters $a_1, \ldots, a_d$ also with $\mathrm{rk}(a_1) = \cdots = \mathrm{rk}(a_d) = \{0\}$. For every $1 \leq i \leq n$ we choose a bijection $\pi_i : \{a_1, \ldots, a_{|\mathfrak{G}_i|}\} \to \mathfrak{G}_i$ and construct the automata that recognise the languages $\{\langle g_i, a_j\rangle \mid 1 \leq i \leq n, j \leq |\mathfrak{G}_i|\}$ and

$$\{\langle g_i, a_x, a_y, a_z\rangle \mid 1 \leq i \leq n, 1 \leq x, y, z \leq |\mathfrak{G}_i|, \pi_i(a_x) \circ_{\mathfrak{G}_i} \pi_i(y) = \pi_i(a_z)\}.$$

Note that the trivial EF-congruence for $\mathfrak{d}$ is $g(m+r) = G(d)d^{r+m}$ bounded, where $G(d)$ is the number of groups of size at most d.

Let $\mathfrak{c}$ be the uniform presentation of the cyclic groups as described previously. We build automata that recognize the alphabet-disjoint union of the languages in $\mathfrak{d}$ and corresponding languages from $\mathfrak{c}$ and obtain a presentation $\mathfrak{e}$ of all cyclic groups and groups of order at most d. It is not hard to see that this presentation is also $\exp_3(\mathrm{poly}(|\varphi|))$ bounded. Basically the union of the EF-congruences for $\mathfrak{d}$ and $\mathfrak{c}$ (where the "is not a tuple of the presentation" equivalence class of $\mathfrak{d}$ is merged with the "is not a convolution"' equivalence class of $\mathfrak{c}$) is an EF-congruence for $\mathfrak{e}$. Then $\mathfrak{e}^\times$ is a presentation of the class of all finite groups

with bounded abelian decomposition width at most d. By Theorem 5.2, $\mathfrak{e}^\times$ is $\exp_4(\text{poly}(|\varphi|))$ bounded.

A decomposition of $\mathfrak{G} = \mathfrak{G}_1 \oplus \cdots \oplus \mathfrak{G}_k \oplus \mathbb{Z}_{n_1} \oplus \cdots \oplus \mathbb{Z}_{n_\ell}$ with non-abelian indecomposable factors $\mathfrak{G}_1, \ldots, \mathfrak{G}_k$ can be computed in polynomial time [71]. From the decomposition we can compute in linear time an advice that represents $\mathfrak{G}$. Note that such an advice has logarithmic size in $|\mathfrak{G}|$. Applying Theorem 5.3 completes the proof. □

Graphs of bounded Tree-Depth and MSO Model Checking Algorithmic meta-theorems for MSO are particularly interesting because MSO is capable of defining many NP-complete problems such as 3-colourability. The most famous result of this kind is probably the theorem of Courcelle that every MSO-definable query can be decided in linear time on the class of all graphs with treewidth at most c for any given constant $c \in \mathbb{N}$ [26]. Because trees have treewidth one, it is immediately clear that the parameter dependence in Courcelle's Theorem must be non-elementary.

Tree-depth is another parameter on graphs that has recently drawn quite some attention. Tree-depth is a more restrictive parameter than treewidth. Indeed, every class of graphs of bounded tree-depth has also bounded treewidth but there are classes of graphs of bounded treewidth that have unbounded tree-depth. It was shown by Gajarský and Hliněný that, in terms of the parameter dependence, MSO-model-checking can be performed significantly faster on graphs of bounded tree-depth [49]. Their algorithm relies on kernelisation to perform fast MSO-model-checking on trees of bounded depth. However, transferring their arguments into our framework reveals that no specialised algorithm is needed to achieve this runtime.

In the following we need to make a distinction between trees that serve as an input to a tree automaton and an unordered rooted tree in the graph theoretic sense. A finite unordered labelled tree-structure $\mathfrak{T}$ is a tuple $(V, E, P_1, \ldots, P_n, r)$ where

- V is a finite set of nodes,
- $E \subseteq \binom{V}{2}$ such that (V, E) is connected and cycle free,
- $P_i \subseteq V$ for all $1 \leq i \leq n$, and
- $r \in V$ is the root of the tree.

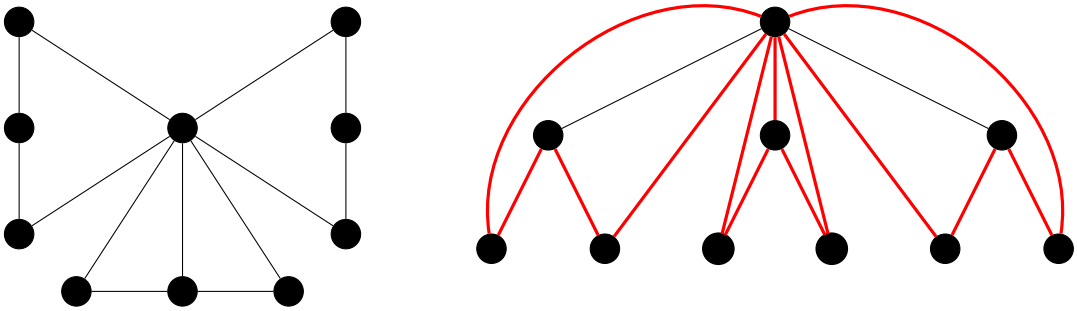


Figure 5.1: Decomposition of a Simple Graph

There are standard techniques to encode a finite unordered tree-structures of unbounded degree by trees of bounded degree.

Definition 5.5. For a finite unordered tree-structure $\mathfrak{T} = (V, E, P_1, \ldots, P_n, r)$ the set of **tilts** of $\mathfrak{T}$, $\mathrm{tilt}(\mathfrak{T}) \subseteq T_{\mathcal{P}(\{1,\ldots,n\})}$, is inductively defined by the following rules.

- if $\mathfrak{T} = (\{v\}, P_1, \ldots, P_n, v)$ then $\mathrm{tilt}(\mathfrak{T}) = \{t\}$, where $\mathrm{dom}(t) = \{\varepsilon\}$ and $t(\varepsilon) = \{i \mid v \in P_i\}$
- if $\mathfrak{T} = (V, P_1, \ldots, P_n, r)$ is of depth $h > 1$ then $t \in \mathrm{tilt}(\mathfrak{T})$ if, and only if, there is an enumeration $\mathfrak{T}_0, \ldots, \mathfrak{T}_k$ of the subtrees induced by the children of the root r of $\mathfrak{T}$ such that there are trees $t_0, \ldots, t_k$ with
  - t_i is a tilt of $\mathfrak{T}_i$,
  - $\mathrm{dom}(t) = \bigcup_{0 \le i \le k} \{1^i 0\} \mathrm{dom}(t_i)$,
  - $t(w) = \begin{cases} \{i \mid w \in P_i\} & w = \varepsilon \\ X & w \in \{1\}^i, 1 \le i \le k \\ t_i(w') & w = 1^i 0 w', 0 \le i \le k \end{cases}$

Note that if t is a tilt of a tree-structure $\mathfrak{T}$ and $v \in \mathrm{dom}(t)$ with $t(v) \neq X$ then v corresponds to a node of depth $|v|_0 + 1$ in $\mathfrak{T}$.

Lemma 5.14. *Let $h \in \mathbb{N}$ be some fixed number. Then the class $\mathcal{C}_h$ of all power set structures of graphs of tree-depth at most h is regularly tree-automatic.*

Proof. The advice set consists of all tilts of unordered labelled tree-structures $(V, E, P_1, \dots, P_{h-1})$ of depth at most $h+1$ such that every node of depth ℓ appears only in sets P_i with $i+1 < \ell$. This is obviously a regular set. Such a tree α presents (the isomorphism type of) the graph $\mathfrak{G} = (V, E)$ with $V = \mathrm{dom}(\alpha) \cap (\{0,1\}^*\{0\})$ and $E = \{\{v, w\} \mid v \preceq w \text{ and } |v|_0 \in \alpha(w)\}$. If α is a tilt of an optimal decomposition of $\mathfrak{G}$ then the subtrees induced by the nodes in $\mathrm{dom}_\alpha \cap \{1\}^*\{0\}$ correspond to the connected components of $\mathfrak{G}$. Building a good tree-automatic presentation $\mathfrak{c} = (\mathcal{A}, \mathcal{A}_E, \mathcal{A}_\subseteq)$ is then straight forward. The automaton $\mathcal{A}$ is chosen such that

$$L(\mathcal{A}[\alpha]) = \{t \in T_{\{0,1,X\}} \mid \mathrm{dom}(t) = \mathrm{dom}(\alpha) \wedge \\ \forall w \in \mathrm{dom}(\alpha) : \alpha(w) = X \to t(w) = X\}.$$

A tree $t \in L(\mathcal{A}[\alpha])$ represents the set $\{v \in \mathrm{dom}(\alpha) \mid t(v) = 1\}$. Then the relation $\subseteq$ is trivially regular and the relation E can also be recognised with the advice α, because the prefix relation is regular on the domain of a tree and $|w|_0 \leq h$ for every $w \in \mathrm{dom}(t)$ and every $t \in L(\mathcal{A}[\alpha])$, so an automaton can check whether w is the first ancestor with $|w|_0 = i$ of a node v with $i \in t(v)$. □

Figure 5.1 shows a decomposition of a simple graph $\mathfrak{G}$ and Figure 5.3 shows an advice for $\mathfrak{G}$.

Theorem 5.8. *The* MSO *model checking problem for graphs of tree-depth at most h is fixed parameter tractable. Given an* MSO *sentence φ and a graph $\mathfrak{G}$ of tree-depth at most h one can decide in time*

$$\mathcal{O}\left(\exp_{(h+2)}(\mathrm{poly}(|\varphi|)) \cdot \mathrm{poly}(|\mathfrak{G}|)\right)$$

whether $\mathfrak{G} \models \varphi$.

Proof. Fix some $h \in \mathbb{N}$ and let $\mathfrak{c}$ be the presentation constructed in Lemma 5.14. Since for every advice α of $\mathfrak{c}$ all elements of $\mathcal{S}(\mathfrak{c}[\alpha])$ have the same domain as α, we don't need a padding symbol to represent tuples but can interpret m-tuples simply as $\mathcal{P}(\{1, \dots, h-1, h, \dots, h+m-1\})$-labelled trees. It is a simple exercise to check that Theorem 5.1 is still valid in this case and that we need only the properties 3 - 7 of Definition 5.3.

We define the EF-congruence on the basis of equivalence relations $(\sim^h_{r,k})_{r,k \in \mathbb{N}}$ on $(P_1, \dots, P_k)$-labelled tree-structures of depth h:

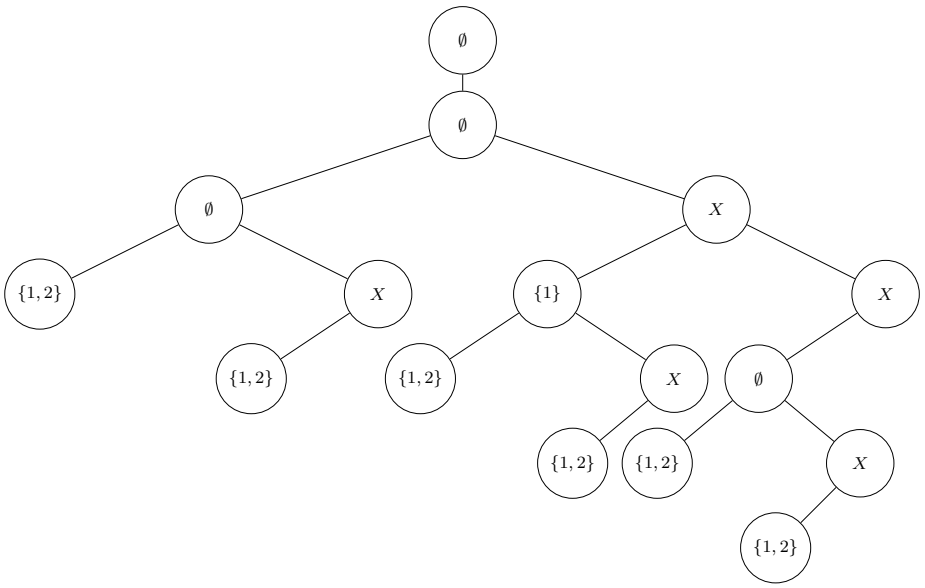


Figure 5.2: An advice for the Graph in Figure 5.1

- For tree-structures $\mathfrak{S}, \mathfrak{T}$ of depth 1 we define $\mathfrak{S} \sim^1_{r,k} \mathfrak{T} :\Leftrightarrow \mathfrak{S} \cong \mathfrak{T}$.

- Let $\mathfrak{S}, \mathfrak{T}$ be trees of depth $h+1$ and let $\mathfrak{S}^i_1, \ldots, \mathfrak{S}^i_{n_i}$ be the subtrees of depth i rooted in a child node of the root in $\mathfrak{S}$ for all $i \leq h$ and let $\mathfrak{T}^i_1, \ldots, \mathfrak{T}^i_{n'_i}$ be the corresponding trees with respect to $\mathfrak{T}$. Then $\mathfrak{S}$ and $\mathfrak{T}$ are $\sim^{h+1}_{r,k}$-equivalent if, and only if, the roots of $\mathfrak{T}$ and $\mathfrak{S}$ share the same labels and for all $i < h$ and all $\sim^i_{r,k}$-equivalence classes κ

$$|\{j \in \mathbb{N} \mid j \leq n_i, \mathfrak{S}^i_j \in \kappa\}| =_{\mathrm{index}(\sim^i_{r,k})^r+1} |\{j \in \mathbb{N} \mid j \leq n'_i, \mathfrak{T}^i_j \in \kappa\}|$$

The proof of [49, Theorem 3.1] can be easily adapted to show that no FO-formula with r quantifiers can distinguish between two power set structures of two $\sim^h_{r,(r+k)}$-equivalent $(P_1, \ldots, P_k)$-labelled tree-structures of depth h.

Moreover, a straightforward induction shows that whenever two such tree-structures $\mathfrak{S}, \mathfrak{T}$ of depth h are $\sim^h_{0,k}$-equivalent then the following two observations hold for every path $v_0 v_1 \ldots v_n$ in $\mathfrak{S}$ starting from the root:

1. There is a path $w_0 w_1 \dots w_n$ in $\mathfrak{T}$ starting from the root of $\mathfrak{T}$ such that for all $0 \leq i \leq n$ the nodes v_i and w_i share the same labels, that is $v_i \in P_j^{\mathfrak{S}} \Leftrightarrow w_i \in P_j^{\mathfrak{T}}$ for all $1 \leq j \leq k$.

2. If for some subsets $I \subseteq \{1, \dots, n\}$, $J \subseteq \{1, \dots, k\}$ the nodes v_i with $i \in I$ are unique in the sense that for every path $v_0' v_1' \dots v_n'$ with $v_i \in P_j^{\mathfrak{S}} \Leftrightarrow v_i' \in P_j^{\mathfrak{S}}$ for all $i \in I$ and $j \in J$ implies $v_i = v_i'$ for all $1 \leq i \leq n$ then there is also a unique path $w_0 w_1 \dots w_n$ $\mathfrak{T}$ with $w_i \in P_j^{\mathfrak{T}} \Leftrightarrow v_i \in P_j^{\mathfrak{S}}$ for all $i \in I, j \in J$.

Let $\sim_{r,k}^{\leq h} := \bigcup_{1 \leq i \leq h} \sim_{r,k}^{i}$. We define an EF-congruence for the presentation in Lemma 5.14 from $\sim_{r,k}^{\leq h}$. Let h be fixed.

In a first step, we partition the set of all $(\mathcal{P}(\{1, \dots, h+m-1\}) \uplus \{X\})$-labelled trees into $2h+1$ classes $T_1^m, \dots, T_h^m, Q_1^m, \dots, Q_h^m, F$.

- A tree t is in T_i^m if, and only if, t is a tilt of a tree of depth i.
- A tree t is in Q_i^m if, and only if, t is not a tilt of a tree of depth i but $t[\varepsilon \to \emptyset]$ is a tilt of a tree of depth i (this is exactly the case if $t = t'[\varepsilon \to X]$ for some tilt t' of a tree of depth i).
- All other trees are in F.

The EF-congruence is then defined by

$$\begin{aligned} t E_m^r t' :\Leftrightarrow & \exists 0 \leq i \leq h : (t \in T_i^m \wedge t' \in T_i^m \wedge \\ & \exists \mathfrak{S}, \mathfrak{S}' : t \in \mathrm{tilt}(\mathfrak{S}) \wedge t' \in \mathrm{tilt}(\mathfrak{S}') \wedge \mathfrak{S} \sim_{r,(r+m+k)}^{i} \mathfrak{S}') \\ & \vee \exists 0 \leq i \leq h : (t \in Q_i^m \wedge t' \in Q_i^m \wedge \\ & \exists \mathfrak{S}, \mathfrak{S}' : t[\varepsilon \to \emptyset] \in \mathrm{tilt}(\mathfrak{S}) \wedge t'[\varepsilon \to \emptyset] \in \mathrm{tilt}(\mathfrak{S}'') \wedge \mathfrak{S} \sim_{r,(r+m+k)}^{i} \mathfrak{S}') \\ & \vee t, t' \in F \end{aligned}$$

From the definition and the remarks about convolutions it is immediately clear that $(E_m^r)_{r,m \in \mathbb{N}}$ fulfils Property 3 of Definition 5.3.

For Property 4 let us consider under which circumstances a tree t does not present a graph of tree-depth at most h. First of all t might not be a tilt of a tree-structure of depth at most $h+1$. In this case $t \in F$ or $t \in Q_i$ for some $i \leq h+1$. In this case E_m^r seperates t from all trees that represent a graph

from $\mathcal{C}_h$. Otherwise t might be the tilt of a tree-structure $\mathfrak{T}$ of depth at most $h+1$ but there is a note $v \in \mathfrak{T}$ of depth i with $v \in P_i^{\mathfrak{T}}$ and $i+1 \geq j$. But then by Observation 1 every E_m^r-equivalent tree-structure contains also a node of depth j which is contained in P_i and therefore does also not present a graph from $\mathcal{C}_h$.

We use Observation 2 to show that Property 5 is fulfilled. Let s and t be $(\mathcal{P}(\{1,\ldots,h+m-1\}) \cup \{X\})$-labelled trees that present Structures in $\mathfrak{c}$ with $sE_m^r t$. Let $\mathfrak{S}, \mathfrak{T}$ be the tree-structures with $s \in \text{tilt}(\mathfrak{S})$ and $t \in \text{tilt}(\mathfrak{T})$, let $(\mathfrak{G}_s, V_1, \ldots, V_m)$ be the tuple presented by s, and $(\mathfrak{G}_t, W_1, \ldots, W_m)$ be the tuple presented by t. If $\mathfrak{G}_s \models E(V_i, V_j)$ for some $i, j \leq m$ then V_i and V_j are singletons and therefore there are unique nodes v_i, v_j with $v_i \in P_{h+i-1}^{\mathfrak{S}}$ and $v_j \in P_{h+j-1}^{\mathfrak{S}}$. Further v_i and v_j are ordered by the ancestor-relationship. Without loss generality assume that v_i is an ancestor of v_j and let d be the depth of v_i in $\mathfrak{S}$. Then $v_j \in P_{d-1}^{\mathfrak{S}}$. By Observation 2 there must be unique nodes w_i, w_j with $w_i \in P_{h+i-1}^{\mathfrak{T}}$ and $w_j \in P_{h+j-1}^{\mathfrak{T}}$. Further w_i has depth d, is an ancestor of w_j, and $w_j \in P_{d-1}^{\mathfrak{T}}$. Hence $\mathfrak{G}_t \models E(W_i, W_j)$. If $\mathfrak{G}_s \not\models V_i \subseteq V_j$ then there is node $v \in \text{dom}_s$ such that $i \in s(v)$ but $j \notin s(v)$. Using similar arguments as in the previous case we can follow that there is also a $w \in \text{dom}_t$ with $i \in s(v)$ and $j \notin s(v)$. Hence $\mathfrak{G}_t \not\models W_i \subseteq W_j$. The case of $\mathfrak{G}_s \not\models V_i = V_j$ is analogous.

In order to establish Property 6 suppose $sE_m^{r+1}t$. Let s' be any tree that can be derived from s by adding the label $(h+m)$ to some nodes $w \in \text{dom}(s) \cap \{0,1\}^*\{0\}$. We distinguish three cases.

Case $s, t \in F$: then $t' \in F$ and we can extend the labelling of t in an arbitrary way to obtain an E_{m+1}^r-equivalent t'.

Case $s, t \in T_i^m$ for some $1 \leq i \leq h$: then there is a $(P_1, \ldots, P_{h+m-1})$-labelled tree-structures $\mathfrak{S}, \mathfrak{T}$ of depth i with $s \in \text{tilt}(\mathfrak{S})$ and $t \in \text{tilt}(\mathfrak{T})$. Further there is a set $X_{\mathfrak{S}} \subseteq \mathfrak{S}$ such that s' is a tilt of $(\mathfrak{S}, X_{\mathfrak{S}})$. Because $\mathfrak{S} \sim^h_{(r+1),((r+1)+h+m)} \mathfrak{T}$ there must be a set $X_{\mathfrak{T}} \subseteq \mathfrak{T}$ such that $(\mathfrak{S}, X_{\mathfrak{S}}) \sim^h_{r,(r+h+(m+1))} (\mathfrak{T}, X_{\mathfrak{T}})$. Finally choose the extension t' of the labelling of t such that $t' \in \text{tilt}((\mathfrak{T}, X_{\mathfrak{T}}))$. Then $t' E_{m+1}^r s'$.

Case $s, t \in Q_i^m$ for some $1 \leq i \leq h$: the case follows analogously to the previous one by considering $s[\varepsilon \to \emptyset]$ and $t[\varepsilon \to \emptyset]$.

At last, we see that Property 7 holds. Indeed, if $t \in F$ then $(c \circ t) \in F$ for every context c. For the case $s, t \in T_i^m$ for some $1 \leq i \leq h$ one can distinguish

two cases based on the structure of the context c.

Case $x(c) \in \{0,1\}^*\{1\} \cup (\{1\}^*\{0\})^{>(h-i)}$: then $(c \circ s)$ and $(c \circ t)$ do not present trees of depth at most h and hence $s, t \in F$.

Case $x(c) \in (\{1\}^*\{0\})^{\leq(h-i)}$: there are three subcases that might occur.

It might be that $(c \circ t) \in F$ and $(c \circ s) \in F$ (because c is a "template" of a tree of depth larger than h or c contains an inconsistent labelling). in this case equivalence is guaranteed by definition.

It is also possible that $(c \circ t) \in T_j^m$ and $(c \circ s) \in T_j^m$ for some $i \leq j \leq h$. Then let $\mathfrak{S}, \mathfrak{T}$ be trees of depth j such that $(c \circ t) \in \mathrm{tilt}(\mathfrak{S})$ and $(c \circ s) \in \mathrm{tilt}(\mathfrak{T})$. By induction over $j - i$ one shows that $\mathfrak{S} \sim^j_{r,r+h+m} \mathfrak{T}$. For $j - i = 0$ this is the case by definition. For $j - i = k + 1$ let $\mathfrak{S}_1, \ldots, \mathfrak{S}_\ell$ and $\mathfrak{T}_1, \ldots, \mathfrak{T}_\ell$ be the subtrees of $\mathfrak{S}$ and $\mathfrak{T}$ that are rooted in the children of the roots $\mathfrak{S}$ and $\mathfrak{T}$, respectively. Without loss of generality assume that $\mathfrak{S}_1$ and $\mathfrak{T}_1$ are the subtrees which resulted from adding s and t into the context c. Then by the induction hypothesis $\mathfrak{S}_1 \sim^h_{r,r+h+m} \mathfrak{T}_1$ and also $\mathfrak{S}_n \cong \mathfrak{T}_n$ for all $1 < n \leq \ell$. But then for all $n < j$ and all $\sim^n_{r,r+h+m}$-equivalence classes τ the number of τ-children of the root in $\mathfrak{S}$ is equal to the number in $\mathfrak{T}$, hence $\mathfrak{S} \sim^j_{r,r+h+m} \mathfrak{T}$ and therefore $(c \circ s) E^r_m (c \circ t)$.

The last case that might happen is $(c \circ t) \in Q_j^m$ and $(c \circ s) \in Q_j^m$ for some $i \leq j \leq h$. In this case we might again argue analogously to the previous cases by considering $(c \circ t)[\varepsilon \to \emptyset]$ and $(c \circ s)[\varepsilon \to \emptyset]$.

Next, let us estimate the index of E^r_m. By the definition of E^r_m,

$$\mathrm{index}(E^r_m) \leq 1 + 2 \sum_{i=0}^{h+1} \mathrm{index}(\sim^i_{r,r+m+h+1})$$

An inductive analysis of $\mathrm{index}(\sim^i_{r,r+m+h+1})$ (see [49, Lemma 3.1 c)]) shows

$$\mathrm{index}(\sim^i_{r,r+m+h+1}) \in \exp_{(i+1)}(\mathrm{poly}(r + m + h + 1)).$$

Applying this to the above estimation yields

$$\mathrm{index}(E^r_m) \in \exp_{(h+2)}(\mathrm{poly}(r + m)).$$

In order to fulfil the prerequisites of Theorem 5.3 we can apply textbook methods to compute the decomposition of a graph of fixed tree-depth (see for instance [90]). From the decomposition the construction of an advice for the presentation in Lemma 5.14 can be performed efficiently. □

Corollary 5.1. *Let $h \in \mathbb{N}$ be fixed and let $\mathcal{C}_h$ be the class of finite graphs with tree-depth at most h. Suppose for a class of finite structures $\mathcal{C} \in \mathrm{USI}(\mathcal{C}_h)$ there is an* FO*-interpretation $\mathcal{I}$ such that one can compute from every $\mathfrak{A} \in \mathcal{C}$ in polynomial time a graph $\mathfrak{G} \in \mathcal{C}_h$ with $\mathcal{I}(\mathfrak{G}) \cong \mathfrak{A}$. Then first order model checking is fixed parameter tractable on $\mathcal{C}$ For every sentence $\varphi \in \mathrm{FO}$ and $\mathfrak{A} \in \mathcal{C}$ one can decide in time*

$$\mathcal{O}\left(\exp_{(h+2)}(\mathrm{poly}(|\varphi|)) \cdot \mathrm{poly}(|\mathfrak{A}|)\right)$$

whether $\mathfrak{A} \models \varphi$.

Note that in the extended version [48] of the above mentioned paper, Gajarský and Hliněný show a similar result that bounds the size of an automaton for certain presentations of structures that are uniformly MSO-interpretable in a class of trees of bounded depth. However, they only prove the existence of small automata.

5.4 Discussion

In this chapter we investigated the application of uniformly tree-automatic presentations in finite model theory. We observed that once a presentable class of structures allows an efficient transformation from (an encoding of) the structure to a corresponding advice one naturally obtains fixed parameter tractability of the model checking problem in the size of the formula for first-order (or monadic second-order) logic on this class. This in turn is absolutely in line with Courcelle's approach to obtain algorithmic meta-theorems for graphs of bounded treewidth or bounded cliquewidth. Indeed, as we have pointed out, the work of Courcelle is actually a special case, were the class that is presented is simple enough to represent even the class of power set structures. It seems very conceivable that the use of set-interpretations opens this technique for much broader classes of graphs (while simultaneously restricting the logic, of course).

However, in these first investigations we were more concerned with the question of how efficient this approach really is. Indeed, the classes that we considered here are rather simple instances and should be understood as proof of concept examples. In fact, all these classes are even uniformly word automatic. The reason why we gave tree-automatic presentations is that this allowed us

to express Theorem 5.2 without having to restrict the classes to finite structures. Additionally the analysis of the EF-congruences would probably be a bit more tedious for the word automatic presentations. The catch is that although we used straightforward presentations for the respective classes, our analysis shows that the worst-case runtime of this generic approach is either optimal or at least matches or even beats all other known approaches. Our findings should therefore also be considered as a praise to this, nowadays simple looking, automata based approach to solve the MSO-theory of trees. This is also reflected in the fact that it goes along so well with the composition operators we considered.

A natural next step would be to consider under which circumstances the existence a uniform (tree-)automatic presentation leads to fixed parameter tractability of the respective model checking problem. All that we need in order to apply Theorem 5.3 is that we can compute from the structure an corresponding advice efficiently. Recall that FPT-algorithms for computing tree decompositions are a major cornerstone of Courcelle's Theorem [15]. Therefore the following question arises: has every uniformly automatic class $\mathcal{C}$ of finite structures a presentation $\mathfrak{c}$ (that possibly presents a class that contains $\mathcal{C}$) such that we can efficiently compute from a given structure $\mathfrak{A} \in \mathcal{C}$ an advice α with $\mathcal{S}(\mathfrak{c}[\alpha]) \cong \mathfrak{A}$? If we could answer this question positively then this would lead to a wealth of new classes where first-order model checking is fixed parameter tractable.

6 Extending Fixed-Point Logic by Interpretations

In the first part of this thesis we used interpretations to reduce algorithmic problems, like solving the theory of a given structure or model checking to deciding certain properties of finite automata. This is the classical application for interpretations. In the second part of this thesis, however, we will take a different attitude towards interpretations. Namely we consider interpretations as the description of the behaviour of an abstract machine. Roughly speaking, we regard a structure to be the representation of a machine state and the interpretation to describe the transition from one state to another. By applying this transition over and over again we obtain a computation. On the one hand this is a model of computation but on the other hand, since all ingredients are formulated in purely logical terms, it is also a logic. Therefore it is natural to investigate this idea in the context of descriptive complexity.

6.1 Is PTIME a Logical Concept?

Descriptive complexity theory is the branch of finite model theory that tries to characterise complexity classes by the type of logic that is needed to express exactly the languages inside this complexity class. The initial result of this field is Fagin's Theorem that the properties of finite structures that are recognisable in non-deterministic polynomial time are exactly the properties expressible in existential second-order logic (see [38]).

Starting from this result, other characterisations of several complexity classes above NP, like PSPACE, EXPTIME, or elementary time, have been found. For a survey see [53, 66]. Below NP the situation is more delicate. The most interesting question, especially in the light of the P versus NP Problem, is whether there is a logic for polynomial time (and if so can we separate it from existential second-order logic?). If we assume that a linear order is present, then fixed-point logic (FP) is the logic we are looking for. This was discovered independently by Immerman and Vardi [65, 99]. Without the presence of a

linear order, however, FP is relatively far from capturing PTIME. For instance FP cannot even express simple properties that involve counting: the class of all structures where the universe has even size is not definable in fixed-point logic. Immerman therefore suggested to extend FP with a mechanism to count. The formal definition of fixed-point logic with counting (FPC), based on inflationary fixed points, was given by Grädel and Otto in [55]. Although Immerman himself, together with Cai and Fürer, proved that FPC does not capture polynomial time, it turned out to be a very strong logic that captures a large fragment of polynomial time. Even today there are only a few known problems in PTIME that are not expressible in FPC. We give a quick overview over some of the most important ones. A survey on the expressive power of fixed-point logic (with counting) can be found in [84].

The Cai, Fürer, Immerman Construction The first example of a query that is decidable in polynomial time but not definable in FPC was given by Cai, Fürer, and Immerman [22] and is today known as the CFI-query. It involves a clever construction of a certain class of graphs, the so-called CFI-graphs. We give a short description of the construction here.

Let $G = (V, E, \leq)$ be a finite undirected graph equipped with a linear order and let $f : V \to \{0, 1\}$ be a function. The **CFI-graph**

$$\mathrm{CFI}(G, f) = \left(V^{\mathrm{CFI}(G,f)}, O, I, E^{\mathrm{CFI}(G,f)}, \preceq\right)$$

over G is the graph where the vertex set is partitioned into the outer nodes $O = (E \times \{0, 1\})$, which consist of two nodes for each Edge of G, and the inner nodes $I = \{(v, X) \in V \times \mathcal{P}(vE) \mid |X| \equiv_2 f(v)\}$, which consist of $2^{|vE|-1}$ many nodes for every vertex v of G. The inner nodes correspond to the even sized subsets of vE if $f(v) = 0$ and to the odd sized subsets of vE if $f(v) = 1$. The edges of $\mathrm{CFI}(G, f)$ are

$$\{\{(v, X), (e, a)\} \mid e = \{v, w\} \text{ and } (a = 1 \text{ iff } w \in X)\},$$

that is each inner node (v, X) is connected to exactly one of the two outer nodes that correspond to an edge e of G for all e with $v \in e$ and the set X determines which of the two is chosen. The gadget of a node $v \in V$ is the subgraph of $\mathrm{CFI}(G, f)$ that is induced by the set

$$\{(e, a) \in O \mid v \in e\} \cup \{(u, X) \in I \mid u = v\}.$$

The gadget of v is called even if $f(v) = 0$ and odd otherwise. The relation $\preceq$ is the preorder on the inner nodes that is induced by the order on G, that is $(v, X) \prec (w, Y)$ if and only if $v \leq w$.

It might seem at first glance that there are many CFI-graphs over a given graph G. But it turns out, and this is one of the beauties of the CFI-construction, that if G is connected and every vertex has at least degree 3 then there are up to isomorphism exactly two CFI-graphs over G. Let $\mathcal{C}$ be the class of all finite connected undirected graphs where every vertex has degree at least 3. It can be shown that for every $G \in \mathcal{C}$ two CFI-graphs $\mathrm{CFI}(G, f)$ and $\mathrm{CFI}(G, g)$ are isomorphic if, and only if

$$|\{v \in V \mid f(v) = 1\}| \equiv_2 |\{v \in V \mid g(v) = 1\}|$$

or equivalently if the number of odd gadgets has the same parity in $\mathrm{CFI}(G, f)$ and $\mathrm{CFI}(G, g)$. If the number of odd gadgets is even we refer to this graph as the even CFI-graph $\mathrm{CFI}_0(G)$ and otherwise as the odd CFI-graph $\mathrm{CFI}_1(G)$.

The CFI-query is to determine whether a given graph G is isomorphic to an even CFI-graph $\mathrm{CFI}_0(H)$ for some graph $H \in \mathcal{C}$. With the above properties in mind, it is not hard to see that the CFI-query is solvable in polynomial time. First, it can easily be checked that G has the structure of some CFI-graph and the outer nodes that correspond to the same edge can be identified. To each such pair of outer nodes one arbitrarily assigns a label 0 and a label 1, which renders every gadget of G either an even gadget or an odd gadget. Now one can simply count the number of odd gadgets and decide in this way whether G is an even or an odd CFI-graph.

However Cai et al. could show that this is not expressible in FPC. They showed that if a graph $G \in \mathcal{C}$ has no k-separators, then $\mathrm{C}^k_{\infty\omega}$, the k-variable fragment of infinitary logic with counting, cannot distinguish between $\mathrm{CFI}_0(G)$ and $\mathrm{CFI}_1(G)$. Note that for every FPC-sentence φ there is a k and a $C^k_{\infty\omega}$-sentence ψ that is equivalent to φ on the class of all finite structures. Further, they could show that for every k there are graphs of degree three such that the CFI-construction yields indistinguishable structures. This leads to two major shortcomings of FPC with respect to expressiveness.

Definition 6.1. Let $\tau = \{R_1, \ldots, R_k\}$ be a vocabulary. A **τ-structure of colour class size** q (or **q-bounded structure**) is a $\tau \uplus \{\preceq\}$-structure $\mathfrak{H} = (H, R_1^{\mathfrak{H}}, \ldots, R_k^{\mathfrak{H}}, \preceq)$ where $\preceq$ is a total preorder on H of width at most q.

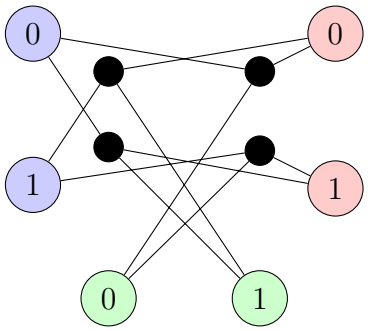


Figure 6.1: An even CFI-gadget obtained from a vertex of degree 3

Observe that for graphs of bounded degree the CFI construction yields graphs of bounded degree and bounded colour class size.

Theorem 6.1 (Cai, Fürer, Immerman [22]). *There are constants c, d such that* FPC *fails to capture* PTIME *on the class of all finite c-bounded graphs of degree at most d.*

An intuitive reason why there is no k such that $\mathrm{C}^k_{\infty\omega}$ can distinguish between even and odd CFI-graphs is that the structure of the automorphism groups of the CFI-graphs allows it to "hide" the odd gadget in the k-pebble bijection game (a model comparison game for $\mathrm{C}^k_{\infty\omega}$ similar to EF-games for first-order logic). Indeed, the property of having an odd gadget is not inherent in the gadgets when they are considered separately but is only enforced by the complete interconnection between the gadgets. One might ask whether the structure of the automorphism group is the only reason why FPC fails to capture PTIME on this class of graphs. In fact, Dawar conjectured that fixed-point logic captures PTIME on every finitely axiomatisable class of rigid structures [28].

Multipedes The conjecture of Dawar was refuted by Blass, Gurevich and Shelah [12, 61] through a class of structures which they call Multipedes. This leads us to the next example that separates FPC from PTIME. We describe their construction here because it will play a role in the following chapter.

A **Multipede** (in [12] called 3-Multipede with shoes) is a structure

$$\mathfrak{M} = (A, F, S, P, E, T, <)$$

such that

1. F and S from a partition of A. The elements of F are called feet and the elements of S are called segments.

2. The relation $<$ is a linear order on the segments S,

3. The relation $E \subseteq F \times S$ connects feet to segments in a way that every foot f is connected to exactly one segment (which we denote $s(f)$) and every segment has exactly two feet.

4. The relation T is a set of hyperedges $T \subseteq \binom{S}{3} \cup \binom{F}{3}$ (called foot and segment edges) such that
   - if x is a segment edge of T then there are exactly four foot edges a in T with $s(a) = \{s(f) \mid f \in a\} = x$,
   - if a is a foot edge of T then $|s(a)| = 3$ (i.e. no two feet in a foot edge belong to the same segment),
   - if a and b are foot edges of T with $s(a) = s(b)$ then a and b are either identical or $|a \cap b| = 1$, and
   - if a is a foot edge of T then $s(a)$ is a segment edge of T.

5. $P \subseteq F$ is the set of "feet with shoes" where for every segment at most one of its feet wears a shoe.

Note that the relation $a \sim b :\Leftrightarrow |a \cap b| = 1$ is an equivalence relation that partitions the possible foot edges for every segment edge into two equivalence classes of size four. The aforementioned properties therefore ensure that for every foot edge in T exactly one of these equivalence classes is contained in T.

A Multipede is **odd** if for every set X of segments the intersection with every segment edge in T has odd cardinality. The Multipede query is to determine, given the disjoint union of two multipedes $\mathfrak{M}$, $\mathfrak{M}'$, if $\mathfrak{M}$ and $\mathfrak{M}'$ are isomorphic.

Blass, Gurevich, and Shelah show that every odd Multipede is rigid and that for every k there is an odd Multipede $\mathfrak{M}$ such that $\mathrm{C}^k_{\infty\omega}$ cannot distinguish between $\mathfrak{M}$ and an Multipede $\mathfrak{M}'$ that differs from $\mathfrak{M}$ only by the fact that for one segment the feet have switched shoes. However, the isomorphism problem for Multipedes is solvable in polynomial time. This is shown by reducing the problem of finding a suitable bijection between the feet of the two Multipedes

to finding a solution of a system of linear equations over $\mathbb{Z}_2$ (note that the bijection between the segments is already prescribed by the linear orders on them).

Theorem 6.2 (Blass, Gurevich, Shelah [12])**.** *The Multipede query is not definable in* FPC*.*

Using the order on the segments of a Multipede one easily defines a total preorder of width two. Hence, Multipedes can be seen as 2-bounded structures.

Corollary 6.1. FPC *fails to capture* PTIME *on the class of all* 2*-bounded structures.*

Linear Algebra So far our examples, as beautiful as they are from a mathematical point of view, are rather artificial. It would therefore be desirable to find problems of practical relevance that separate PTIME from FPC. The reduction of the isomorphism problem for Multipedes to the solvability of a linear equation system suggests that problems from linear algebra might be a source for such examples. This is in fact the case. Atserias, Bulatov and Dawar [6] have shown that several kinds of *constraint satisfaction problems*, the solvability of linear equation systems being one of them, are not expressible in FPC.

Where FPC **captures** PTIME Besides identifying separating problems, people where also interested in what can actually be expressed in FPC. This direction of research branches further into two parts. First, one tries to find problems where there is no obvious description of the problem in FPC but there are still clever ways to express the problem. Second, one tries to find larger and larger subclasses of structures where FPC is actually capable of expressing all PTIME properties. On the first branch let us mention two results. The first one is due to Otto [85], namely that FPC is capable of expressing every bisimulation invariant PTIME property of graphs (more precisely a fragment, the higher order μ-calculus, captures this fragment of polynomial time). Another inspiring result, due to Anderson [5], is that FPC can solve linear programs. An application is that the size of maximum matching is definable in FPC.

For the second branch the most general result known today is Grohe's Theorem that FPC captures PTIME on every class of graphs that excludes a fixed

minor [59]. Excluding a specific minor is a property that applies to a variety of classes such as planar graphs or graphs of bounded treewidth. For a more elaborate survey on recent positive results concerning FPC we refer to [29].

In the light of this history, FPC is currently the benchmark for any logic that wants to be a candidate for a logic capturing PTIME. In the following we go a step further and review two logics (which turn out to be of equal expressiveness) that are at the time of this writing the most promising candidates for a logic capturing PTIME.

6.2 Interpretation Logic

We continue by formalising the ideas that we broached in the introduction. The study of the resulting formalism, which we will call interpretation logic, was suggested by Kaiser and later defined by Schalthöfer [94].

Definition 6.2. Let τ be a signature. A program of **interpretation logic** (IL) over τ is a tuple $P = (\mathcal{I}_{\text{init}}, \mathcal{I}_{\text{step}}, \varphi_{\text{halt}}, \varphi_{\text{eval}})$, where $\mathcal{I}_{\text{init}}$ is an FO + H τ-to-σ interpretation, $\mathcal{I}_{\text{step}}$ is an FO + H σ-to-σ interpretation, and φ_{halt} as well as φ_{eval} are sentences of FO[σ] for some finite relational signature σ.

A program of interpretation logic over τ defines a query on the class of all τ-structures.

Definition 6.3. Let $P = (\mathcal{I}_{\text{init}}, \mathcal{I}_{\text{step}}, \varphi_{\text{halt}}, \varphi_{\text{eval}})$ be a program of interpretation logic over the signature τ and let $\mathfrak{A}$ be a τ-structure. The **run** of P on $\mathfrak{A}$ is the sequence of structures $(\mathfrak{B}_i)_{i\in\mathbb{N}}$ with

$$\mathfrak{B}_0 = \mathcal{I}_{\text{init}}(\mathfrak{A}) \text{ and}$$
$$\mathfrak{B}_{i+1} = \mathcal{I}_{\text{step}}(\mathfrak{B}_i)$$

for all $i \in \mathbb{N}$.

If there is no $i \in \mathbb{N}$ such that $\mathfrak{B}_i \models \varphi_{\text{halt}}$ we say that P diverges on $\mathfrak{A}$. Otherwise P is evaluated to true in $\mathfrak{A}$ if $\mathfrak{B}_i \models \varphi_{\text{eval}}$ and P is evaluated to false on $\mathfrak{A}$ if $\mathfrak{B}_i \not\models \varphi_{\text{eval}}$ for the minimal i such that $\mathfrak{B}_i \models \varphi_{\text{halt}}$.

An IL-program P over the signature τ **recognises** the class

$$\{\mathfrak{A} \in \text{finStr}(\tau) \mid P \text{ is evaluated to true in } \mathfrak{A}\}.$$

We say that P **decides** the class $\mathcal{C} \subseteq \text{finStr}(\tau)$ if P recognises $\mathcal{C}$ and P never diverges on a finite τ-structure.

Example 6.1. The following program recognises the class of all finite sets whose size is a power of two.

- $\mathcal{I}_{\text{init}} = (\varphi_U(x_0, x_1), \varepsilon_i(\overline{x}, \overline{y}), \varphi_A(\overline{x}), \varphi_E(\overline{x}, \overline{y}))$ with

$$\begin{aligned} \varphi_U(x_0, x_1) &\equiv (x_0 = x_0) \\ \varepsilon_i(x_0, x_1, y_0, y_1) &\equiv (\{x_0, x_1\} = \{y_0, y_1\}) \\ \varphi_A(x_0, x_1) &\equiv (x_0 = x_1) \\ \varphi_E(x_0, x_1, y_0, y_1) &\equiv (y_0 = y_1) \wedge y_0 \in \{x_0, x_1\} \end{aligned}$$

- $\mathcal{I}_{\text{step}} = (\psi_U(x_0, x_1), \varepsilon_s(\overline{x}, \overline{y}), \psi_A(\overline{x}), \psi_E(\overline{x}, \overline{y}))$ with

$$\begin{aligned} \psi_U(x_0, x_1) &\equiv (Ax_0 \wedge x_0 = x_1) \vee (\neg Ax_0 \wedge \neg Ax_1 \wedge x_0E \cap x_1E = \emptyset) \\ \varepsilon_s(x_0, x_1, y_0, y_1) &\equiv (x_0E \cup x_1E = y_0E \cup y_1E) \\ \psi_A(x_0, x_1) &\equiv (Ax_0 \wedge x_0 = x_1) \\ \psi_E(x_0, x_1, y_0, y_1) &\equiv y_0 \in x_0E \cup x_1E \end{aligned}$$

- $\varphi_{\text{halt}} \equiv \exists x, y : A = xE \cup yE$
- $\varphi_{\text{eval}} \equiv \exists x, y : (A = xE \cup yE) \wedge (xE \cap yE = \emptyset)$

Indeed, the structure $\mathcal{I}^n_{\text{step}}(\mathcal{I}_{\text{init}}(\mathfrak{A}))$ is isomorphic to the structure whose universe consists of the disjoint union of A and $\binom{A}{2^{n+1}}$. The edge relation $E \subseteq \binom{A}{2^{n+1}} \times A$ encodes the membership relation, that is $(X, a) \in E$ if, and only if, $a \in X$. The program halts on every finite structure after $n = \lceil \log A \rceil - 1$ iterations of $\mathcal{I}_{\text{step}}$. Then $|A|$ is a power of twos if there are two disjoint sets $X, Y \in \binom{A}{2^n}$ with $X \cup Y = A$. Hence, P decides the query on the class of all finite structures.

There is a direct correspondence between programs of interpretation logic and Turing machines.

Theorem 6.3. *Let τ be a finite signature and $\mathcal{C}$ be a class of finite τ-structures. Then $\mathcal{C}$ is recognisable by an* IL*-program if, and only if, it is recursively enumerable. Further $\mathcal{C}$ is decidable if, and only if, there is an* IL*-program that decides $\mathcal{C}$ on the class of all finite τ-structures.*

Proof Sketch. The direction from IL-programs to Turing machines is rather obvious. The iteration process that builds up the run of an IL-program can be simulated by a Turing machine until at some point the formula φ_{halt} is true.

On the other side, from a Turing machine that decides or recognises the class $\mathcal{C}$ we can construct an IL-program that first builds up every possible code of $\mathfrak{A}$ and then simultaneously simulates M on all these inputs step by step. The formula φ_{halt} checks if one of these computations has reached a halting configuration and φ_{eval} checks if one of these configurations is accepting. □

Using the analogy that structures represent the state of a machine and interpretations describe the computation steps, we can bound the resources that a machine is allowed to use during the computation.

Definition 6.4. Let $f, g : \mathbb{N} \to \mathbb{N}$ be strictly increasing functions. The logic $\text{IL}[f, g]$ consists of all IL-programs such that if $(\mathfrak{B}_i)_{i\in\mathbb{N}}$ is the computation on a finite structure $\mathfrak{A}$ then there is an $n \leq f(|A|)$ such that $\mathfrak{B}_n \models \varphi_{\text{halt}}$ and $|B_i| \leq g(|A|)$ for all $1 \leq i \leq n$.

We define the polynomial time fragment of IL in terms of Definition 6.4.

Definition 6.5. Polynomial time interpretation logic (PIL) is the logic

$$\text{IL}[\text{poly}, \text{poly}] = \bigcup_{p,q\in\text{poly}} \text{IL}[p, q].$$

As usual we will also write $\mathcal{C} \in \text{PIL}$ for a class of finite structures to indicate that there is a PIL-program which decides $\mathcal{C}$.

We observe that Example 6.1 does not describe a PIL-program. While the number of iterations is bounded logarithmically in the size of every finite input structure, the universe of the interpreted structures becomes exponentially large during the computation. The next example gives a PIL-program which also demonstrates the use of the Härtig quantifier.

Example 6.2. The following PIL-program P decides the class of all finite structures with an even number of elements.

- $\mathcal{I}_{\text{init}} = (\varphi_U(x_0, x_1), \varepsilon_i(\overline{x}, \overline{y}), \varphi_A(\overline{x}), \varphi_L(\overline{x}), \varphi_P(\overline{x}), \varphi_S(\overline{x}, \overline{y}))$ with

$$\begin{aligned}
\varphi_U(x_0, x_1) &= \textbf{true} \\
\varepsilon_i(x_0, x_1, y_0, y_1) &\equiv (x_0 = y_0 \wedge x_1 = y_1) \vee (x_0 \neq x_1 \wedge y_0 \neq y_1) \\
\varphi_A(x_0, x_1) &\equiv (x_0 = x_1) \\
\varphi_L(x_0, x_1) &\equiv (x_0 \neq x_1) \\
\varphi_P(x_0, x_1) &\equiv \textbf{false} \\
\varphi_S(x_0, x_1, y_0, y_1) &\equiv \textbf{false}
\end{aligned}$$

- $\mathcal{I}_{\text{step}} = (\psi_U(x_0, x_1), \varepsilon_s(\overline{x}, \overline{y}), \psi_A(\overline{x}), \psi_L(\overline{x}), \psi_P(\overline{x}), \psi_S(\overline{x}, \overline{y}))$ with

$$\begin{aligned}
\psi_U(x_0, x_1) &\equiv (x_0 = x_1) \vee (Lx_0 \wedge Ax_1) \\
\varepsilon_s(x_0, x_1, y_0, y_1) &\equiv (x_0 = y_0 \wedge x_1 = y_1) \vee (Lx_0 \wedge Ly_0 \wedge Ax_1 \wedge Ay_1) \\
\psi_A(x_0, x_1) &\equiv (Ax_0) \\
\psi_L(x_0, x_1) &\equiv (Lx_0) \\
\psi_P(x_0, x_1) &\equiv (x_0 = x_1 \wedge Px_0) \vee \\
&\qquad (Lx_0 \wedge Ax_1 \wedge \exists z(Lz \wedge \neg \exists z' Szz') \wedge \neg Pz) \\
\psi_S(x_0, x_1, y_0, y_1) &\equiv (x_0 = x_1 \wedge y_0 = y_1 \wedge Sx_0y_0) \vee \\
&\qquad (Lx_1 \wedge \neg \exists z(Sx_1z) \wedge Ly_0 \wedge Ay_1)
\end{aligned}$$

- $\varphi_{\text{halt}} \equiv Hxy(Ax)(Ly)$
- $\varphi_{\text{eval}} \equiv \exists x(Lx \wedge \neg \exists y(Sxy) \wedge Px)$

Every structure $\mathcal{I}^n_{\text{step}}(\mathcal{I}_{\text{init}}(\mathfrak{A}))$ is isomorphic to a structure that consists of the universe of $\mathfrak{A}$ and a path of length $n+1$ where every other node of the path belongs to the set P. In each interpretation step the path is prolonged by one node. Using the Härtig quantifier, the program halts as soon as the number of nodes on the path is equal to the size of the universe of $\mathfrak{A}$. To determine whether this number is even it suffices to check if the last node of the path is in P.

Note that every class of finite structures $\mathcal{C} \in \text{PIL}$ is also in PTIME: for every first-order interpretation φ there is a fixed polynomial p such that $\mathcal{I}(\mathfrak{A})$ can be computed in time $p(|A|)$ for every matching finite structure $\mathfrak{A}$. The same is true for the evaluation of first order formulae (which can be done even in

logarithmic space). The other direction yields a much more delicate question. The simulation of a polynomially time bounded Turing machine, like it was suggested in the proof of Theorem 6.3, does not lead to a PIL-program. In fact, the program cannot choose an arbitrary code of the structure to simulate the Turing machine, because every automorphism of the input structure naturally induces an automorphism of the interpreted structure. The program therefore simulates the machine on all possible codes simultaneously which exceeds the polynomial bound for the size of the structures that may appear in a run of the program.

This, of course, does not necessarily mean that there is no other way to capture all PTIME properties by PIL-programs, but it reflects the fundamental difficulty which makes the quest for a logic for PTIME so intriguing.

6.3 Choiceless Polynomial Time

Another prominent candidate for a logic for PTIME is choiceless polynomial time with counting (CPT). It is defined as BGS logic with additional explicitly given resource bounds. The logics CPT and BGS where introduced by Blass, Gurevich, and Shelah [11]. However, the definition that we are about to give deviates from the original one and rather sticks to the presentation given by Rossman [92].

Definition 6.6. Let A be a set. Then $\mathrm{HF}(A)$ denotes the smallest set such that $A \subseteq \mathrm{HF}(A)$ and if B is a finite subset of $\mathrm{HF}(A)$ then also $B \in \mathrm{HF}(A)$. The elements of A are called **atoms** and the elements of $\mathrm{HF}(A) \setminus A$ are called the **hereditarily finite sets** over A.

Note that $\mathrm{HF}(A)$ contains the natural numbers as von Neumann ordinals independent of the particular set A.

We also need the notion of the hereditarily finite expansion of a structure.

Definition 6.7. Let τ be a signature and $\mathfrak{A}$ a τ-structure. Then $\mathrm{HF}(\mathfrak{A})$ is the $\tau \cup \{\mathrm{In}, \mathrm{Atoms}, \mathrm{Empty}, \mathrm{Pair}, \mathrm{Union}, \mathrm{TheUnique}, \mathrm{Card}\}$ Structure with universe $\mathrm{HF}(A)$, where the symbols in τ are interpreted as in $\mathfrak{A}$ and

- $(x, y) \in \mathrm{In} \Leftrightarrow x \in y$,
- $\mathrm{Empty} = \emptyset$,
- $\mathrm{Atoms} = A$,
- $\mathrm{Pair}(x, y) = \{x, y\}$,

- $\mathrm{Union}(x) = \bigcup_{y \in x} y$,
- $\mathrm{TheUnique}(x) = \begin{cases} y & x = \{y\} \\ \emptyset & \text{else} \end{cases}$
- $\mathrm{Card}(x) = \begin{cases} |x| & x \text{ is a set} \\ 0 & \text{else} \end{cases}$

The signature $\tau \cup \{\mathrm{In}, \mathrm{Atoms}, \mathrm{Empty}, \mathrm{Pair}, \mathrm{Union}, \mathrm{TheUnique}, \mathrm{Card}\}$ is denoted by τ^{HF}.

Like first-order logic, BGS consists of the two syntactical objects terms and formulas, which are defined by mutual induction.

Definition 6.8. The BGS terms and formulas over the signature τ are the smallest sets such that

- every first-order term over τ^{HF} is a BGS term,
- every quantifier-free first-order formula over τ^{HF} and BGS terms is a BGS formula,
- if s and t are BGS terms, v is a variable that does not occur free in t, and φ is a BGS formula, then $\{s(v) : v \in t : \varphi(v)\}$ is a BGS term (called comprehension term). Note that s, t, and φ might contain free variables besides v.

The evaluation of BGS terms and formulas built up from the first two items coincides with the semantics for first order logic. The comprehension is evaluated as

$$\{s(v, \overline{x}) : v \in t(\overline{x}) : \varphi(v, \overline{x})\}^{\mathrm{HF}(\mathfrak{A}, \overline{a})} = \\ \{s^{\mathrm{HF}(\mathfrak{A})}(v, \overline{a}) \mid v \in t^{\mathrm{HF}(\mathfrak{A})}(\overline{a}) \wedge \mathrm{HF}(\mathfrak{A}) \models \varphi(v, \overline{a})\}.$$

Finally we can define BGS programs.

Definition 6.9. A BGS **program** is a triple $P = (\pi(x), \varphi_h(x), \varphi_o(x))$ that consists of a term π, and two formulas φ_h, φ_o with a single free variable each. The evaluation $P^{\mathfrak{A}}$ is defined as follows. Let $(a_i)_{i \in \mathbb{N}}$ be the sequence with $a_0 = \emptyset$ and $a_{i+1} = \pi^{\mathfrak{A}}(a_i)$ for all $i > 0$. If there is no i such that $\mathfrak{A} \models \varphi_h(a_i)$ then $P^{\mathfrak{A}}$ is undefined and we say that the program diverges on $\mathfrak{A}$. Otherwise, let i be the smallest number such that $\mathfrak{A} \models \varphi_h(a_i)$. Then $P^{\mathfrak{A}}$ is true if and only if $\mathfrak{A} \models \varphi_o(a_i)$ and otherwise it is false.

The logic BGS without any restrictions is a much too powerful logic to be used in the context of descriptive complexity theory. In fact, every recursively enumerable property of finite structures is expressible in BGS. To obtain a candidate for a logic for PTIME we need to impose certain restrictions on the resources that are available to a BGS program. The resource bound that leads to the logic CPT is the number of objects that are created throughout the execution of the program. These objects are called the **active elements**. Roughly an element is active if it is contained in the transitive closure of some set that is created while the term π is iterated.

Definition 6.10. Let $t(x_1, \ldots, x_n)$ be a BGS term and $a_1, \ldots, a_n$ elements of $\mathrm{HF}(\mathfrak{A})$. The set $\langle t(a_1, \ldots, a_n)\rangle^{\mathfrak{A}}$ is inductively defined by

- $\langle a_i\rangle = \{a_i\}$
- $\langle R(t_1, \ldots, t_m)\rangle = \bigcup_{1\leq i\leq m}\langle t_i\rangle$ for m-ary relation symbols
- $\langle f(t_1, \ldots, t_m)\rangle = \{f(t_1, \ldots, t_m)^{\mathfrak{A}}\} \cup \bigcup_{1\leq i\leq m}\langle t_i\rangle$ for function symbols
- $\langle\{s(v) : v \in t : \varphi(v)\}\rangle = \langle t\rangle \cup \bigcup_{v\in t^{\mathfrak{A}}}(\langle s(v)\rangle \cup \varphi(v))$

For a program $P = (\pi, \varphi_h, \varphi_o)$ the active elements are

$$\langle P\rangle^{\mathfrak{A}} = \begin{cases} \bigcup_{i\in\mathbb{N}}\langle\pi(a_i)\rangle \cup \langle\varphi_h(a_i)\rangle & \text{if } P \text{ diverges on } \mathfrak{A} \\ \bigcup_{1\leq i\leq \ell}(\langle\pi(a_i)\rangle \cup \langle\varphi_h(a_i)\rangle) \cup \langle\varphi_o(a_\ell)\rangle & \text{otherwise} \end{cases}$$

where $a_0 = \emptyset$, $a_{i+1} = \pi^{\mathfrak{A}}(a_i)$ for $i \geq 0$, and ℓ is the smallest number such that $\mathfrak{A} \models \varphi_h(a_\ell)$ (is such an ℓ exists).

For a function $f : \mathbb{N} \to \mathbb{N}$, we denote by $\mathrm{BGS}[f]$ the class of all BGS programs P such that P never diverges on $\mathfrak{A}$ and $\langle P\rangle^{\mathfrak{A}} \leq f(|A|)$ for all finite structures $\mathfrak{A}$.

Definition 6.11.

$$\mathrm{CPT} = \mathrm{BGS}(\mathrm{poly}) = \bigcup_{p\in\mathrm{poly}} \mathrm{BGS}[p]$$

Again, we will take the freedom to write $\mathcal{C} \in \mathrm{CPT}$ for a class of finite structures $\mathcal{C}$ whenever we want to express that there exists a CPT-program that decides $\mathcal{C}$.

It turns out that CPT and PIL have the same expressive power.

Theorem 6.4 (Schalthöfer [94])**.** PIL ≡ CPT

Given that CPT and PIL use quite different mechanics, the results suggest that these logics capture a natural level of expressibility. The big open question is how much of PTIME is contained in this level.

Open Problem 6.1. *Does* PIL *(or equivalently* CPT*) capture* PTIME*?*

We do not actually belief that this is the case, but answering this question in any direction would probably signify a great progression in our understanding.

Although our original motivation was to study a logic that is based on iterated interpretation, Theorem 6.4 justifies that for the positive results in the following chapter we use the logic CPT as our underlying framework. The reason for this is that CPT is simply the more established logic with this expressiveness. Since we will refrain from writing down formulae explicitly in our proofs, this will not make much of a difference anyway.

Remark 6.1. The reader should be aware of the fact that we defined PIL and CPT directly with an ability to count (in the case of PIL indirectly through the Härtig quantifier). In the literature there is usually a distinction between the versions with and without counting. We renounced from this convention because in this introduction we only wanted to present the actual candidates for capturing PTIME. However, one should note that we do not make use of counting for the canonisation procedure in the following chapter.

7 Canonising Structures of Bounded Colour Class Size

There are several examples which show that CPT is indeed more powerful than FPC. First, because CPT is constrained to use only polynomially many resources from outside, it is sensible to artificial enlargement of the structure by padding elements. Hence, any computable query on small enough definable substructures is expressible in CPT, a property that FPC does not possess. Even more interesting, Dawar, Richerby, and Rossman [30] showed that the CFI-query is definable by a CPT-program.

In this chapter we contribute to the exploration of which other shortcomings of FPC are no longer present in CPT. Since we will only consider finite structures in this chapter we will pick up the convention that whenever we use the term structure we implicitly mean finite structure. One of the major restrictions imposed on FPC by the CFI-construction was the failure to capture PTIME on structures of bounded colour class size. This is quite a strong result because structures with colour class size bounded by some constant q are almost linearly ordered. Although it is not obvious that a canonical extension of the total preorder to a linear order (i.e. a canonisation) can be computed in polynomial time. The goal of this chapter is to define in CPT a canonisation procedure for an important subclass of bounded colour class size structures. This leads to the following theorems.

Theorem 7.1. 2-*bounded structures can be canonised in* CPT.

Accordingly, CPT captures PTIME on 2-bounded structures. As we already mentioned Multipedes are 2-bounded structures. Hence our result answers the question of Blass, Gurevich, and Shelah.

Corollary 7.1. *The Multipedes query is definable in* CPT.

We further generalise Theorem 7.1 to structures of bounded colour class size where the automorphism group of every colour class is abelian.

Theorem 7.2. q-*bounded structures with abelian colours can be canonised in* CPT.

7.1 A Generic Canonisation Algorithm for Structures of Bounded Colour Class Size

We begin with the outline of a generic procedure to define from a given input structure of bounded colour class size an isomorphic copy over an ordered universe (a **canonical copy** or **canonisation**). The idea is to canonise larger and larger parts of the structure along the linear order on the colour classes. Of course it is possible to canonise every colour class individually due to their constant size, but we also need to take the relations between different colour classes into account. Since the relations in the signature have a fixed arity we can partition the relations with respect to their "colour type" and canonise these parts individually. However, these substructures will have non-disjoint universes in general and fixing an ordering on the universe of one substructure clearly constrains the possible choices of orderings on the other substructures.

It is therefore necessary to maintain a data structure that reflects these constraints. The information that we want to maintain is the set of isomorphisms between the canonised part of the input structure and its partial canonisation. When we proceed to choose a canonisation for the next substructure, we first check whether, with this choice, some of the old isomorphisms extends to the new larger part of the structure. If so, we can update our data structure and proceed.

If we want to implement this approach in CPT the crucial step is to find a way of representing the sets of isomorphisms in a succinct and CPT-definable way.

Definition 7.1. Let $\mathfrak{H} = (H, R_1^{\mathfrak{H}}, \dots, R_k^{\mathfrak{H}}, \preceq)$ be a q-bounded structure. We write $H = H_1 \preceq \cdots \preceq H_n$ for the canonical enumeration of the colour classes and denote by $q_i := |H_i| \leq q$ the size of the i-th colour class H_i. We let $H_i^<$ be the ordered set $\{(i,0), \dots, (i, q_i - 1)\}$ and write $\mathcal{O}(H_i)$ to denote the set of bijections between H_i and $H_i^<$, that is $\mathcal{O}(H_i) = \{\pi : H_i \to H_i^<, \pi \text{ is a bijection}\}$.

For our procedure it is crucial that the vocabulary is **fixed** when we speak of a class of q-bounded structures. In particular, the arity of all relations which appear in a q-bounded structure is also bounded by a constant, say r.

For a q-bounded structure with n colour classes let $\mathcal{P} = \mathcal{P}(n, r)$ denote the set of all non-empty subsets $I \subseteq \{1, \dots, n\}$ of size $\leq r$. We can define the set $\mathcal{P}$ together with a linear order in CPT (since r is fixed).

For $I \in \mathcal{P}$ we set $H_I = \bigcup_{i \in I} H_i$ and denote by $\mathfrak{H}_I \subseteq \mathfrak{H}$ the substructure of $\mathfrak{H}$ which is **induced** on the set H_I. Since r bounds the arity of relations in τ we have $\mathfrak{H} = \bigcup_{I \in \mathcal{P}} \mathfrak{H}_I$. We set

$$\mathcal{O}(H) = \mathcal{O}(H_1) \times \cdots \times \mathcal{O}(H_n)$$

and

$$\mathcal{O}(H_I) = \mathcal{O}(H_{i_1}) \times \cdots \times \mathcal{O}(H_{i_\ell})$$

for $I = \{i_1 < i_2 < \cdots < i_\ell\} \in \mathcal{P}$. Given $C \subseteq \mathcal{O}(H_I)$ the **extension** of C to $\mathcal{O}(H)$ is the set

$$\mathrm{ext}(C) = \{(\sigma_1, \ldots, \sigma_n) \in \mathcal{O}(H) : (\sigma_{i_1}, \ldots, \sigma_{i_\ell}) \in C\}.$$

Every $\sigma = (\sigma_1, \ldots, \sigma_n) \in \mathcal{O}(H)$ can be identified with a bijection between H and the lexicographically ordered vertex set

$$H^< = \{(i,j) : 1 \leq i \leq n, 0 \leq j < q_i\}.$$

The preorder $\preceq$ on H translates to the preorder $\sigma(\preceq)$ on $H^<$ in the natural way, i.e.

$$(i,j)\sigma(\preceq)(i',j') \Leftrightarrow i \leq i'.$$

Specifically, $\sigma \in \mathcal{O}(H)$ defines an isomorphism between the input structure $\mathfrak{H}$ and the structure $\sigma(\mathfrak{H}) = (H^<, \sigma(R_1^{\mathfrak{H}}), \ldots, \sigma(R_k^{\mathfrak{H}}), \sigma(\preceq))$. Of course we can apply $\sigma \in \mathcal{O}(H)$ also to substructures of $\mathfrak{H}$. In particular for $I \in \mathcal{P}$, every $\sigma \in \mathcal{O}(H_I)$ defines an isomorphism between $\mathfrak{H}_I$ and the ordered structure $\sigma(\mathfrak{H}_I) = (H_I^<, \sigma(R_1^{\mathfrak{H}_I}), \ldots, \sigma(R_k^{\mathfrak{H}_I}), \sigma(\preceq^{\mathfrak{H}_I}))$ where $H_I^< = \{(i,j) \in H^< : i \in I\}$. Our aim is to construct for a q-bounded structure $\mathfrak{H}$ a copy of the form $\sigma(\mathfrak{H})$ in an isomorphism invariant way.

In general, two different $\sigma, \tau \in \mathcal{O}(H)$ will result in two different canonisations, that is $\sigma(\mathfrak{H}) \neq \tau(\mathfrak{H})$. Since the structures $\sigma(\mathfrak{H})$ and $\tau(\mathfrak{H})$ are defined over an ordered universe we can distinguish them logically in CPT. Moreover, it is easy to see that $\sigma(\mathfrak{H}) = \tau(\mathfrak{H})$ holds if, and only if, $\tau^{-1}\sigma \in \mathrm{Aut}(\mathfrak{H})$. This reflects the well known connection between the set of all isomorphisms between two structures and their automorphism groups:

Remark 7.1. For $\sigma \in \mathcal{O}(H)$ we have

$$\{\tau \in \mathcal{O}(H) : \tau(\mathfrak{H}) = \sigma(\mathfrak{H})\} = \sigma\mathrm{Aut}(\mathfrak{H}) = \mathrm{Aut}(\sigma(\mathfrak{H}))\sigma.$$

Let $I_1 < I_2 < \cdots < I_m$ be the enumeration of $\mathcal{P}$ according to the CPT-definable linear order $<$ on $\mathcal{P}$. We denote by $\mathfrak{H}[1 \cdots s] \subseteq \mathfrak{H}$ the (not necessarily induced) substructure of $\mathfrak{H}$ which consists of the first s components of $\mathfrak{H}$, that is $\mathfrak{H}[1 \cdots s] = \mathfrak{H}_{I_1} \cup \cdots \cup \mathfrak{H}_{I_s}$.

Definition 7.2. For $0 \leq s \leq m$ an **s-canonisation** of $\mathfrak{H}$ is a canonisation of $\mathfrak{H}[1 \cdots s]$, that is a structure $\sigma(\mathfrak{H}[1 \cdots s]) = \sigma(\mathfrak{H}_{I_1}) \cup \cdots \cup \sigma(\mathfrak{H}_{I_s})$ for some $\sigma \in \mathcal{O}(H)$.

A non-empty set $C \subseteq \mathcal{O}(H)$ **witnesses** an s-canonisation of $\mathfrak{H}$ if for all $\sigma, \tau \in C$ we have $\tau(\mathfrak{H}_{I_j}) = \sigma(\mathfrak{H}_{I_j})$ for $j = 1, \ldots, s$.

Since $\mathfrak{H} = \bigcup_{I \in \mathcal{P}} \mathfrak{H}_I$, an m-canonisation of $\mathfrak{H}$ is also a canonisation of $\mathfrak{H}$. We are prepared to describe our generic canonisation procedure for q-bounded structures. The idea is that we iteratively construct, for all $0 \leq s < m$, $(s+1)$-canonisations $\mathfrak{H}^<_{s+1}$ of $\mathfrak{H}$ by extending the previously computed s-canonisation $\mathfrak{H}^<_s$. In order to ensure consistence of our choices, we additionally maintain a set $C_{s+1} \subseteq \mathcal{O}(H)$ which witnesses this $(s+1)$-canonisation of $\mathfrak{H}$. A single iteration step is performed as follows: we start by constructing all ordered versions of the component $\mathfrak{H}_{I_{s+1}}$ and check which of them are consistent with the set C_s. The first part is easy. Because the size of the component $\mathfrak{H}_{I_{s+1}}$ only depends on q and τ, we can explicitly describe all ordered versions of $\mathfrak{H}_{I_{s+1}}$ by a CPT-formula. Secondly, to obtain the $(s+1)$-canonisation $\mathfrak{H}^<_{s+1}$ of $\mathfrak{H}$ we add the lexicographically smallest compatible canonisation of $\mathfrak{H}_{I_{s+1}}$ to the s-canonisation $\mathfrak{H}^<_s$. Finally, we need to update our presentation of the set C_s of witnesses for $\mathfrak{H}^<_s$ by describing the new constraints imposed by fixing the canonisation of $\mathfrak{H}_{I_{s+1}}$.

In order to describe this procedure more formally, let us assume that we have preselected for each colour class H_i a set of linear orderings $\sigma_i\Gamma_i \subseteq \mathcal{O}(H_i)$ where $\Gamma_i \leq \mathrm{sym}(H_i)$ and $\sigma_i \in \mathcal{O}(H_i)$. The group $\Gamma = \Gamma_1 \times \cdots \times \Gamma_n$ acts on $\mathcal{O}(H)$ in the obvious way and for $\sigma = (\sigma_1, \ldots, \sigma_n) \in \mathcal{O}(H)$ we have $\sigma\Gamma = \tau\Gamma$ for every $\tau \in \sigma\Gamma$. For an index set $I = \{i_1, \ldots, i_\ell\} \in \mathcal{P}$ we write Γ_I to denote the group $\Gamma_I = \Gamma_{i_1} \times \cdots \times \Gamma_{i_\ell}$ and $(\sigma\Gamma)_I$ to denote the set $(\sigma\Gamma)_I = \sigma_{i_1}\Gamma_{i_1} \times \cdots \times \sigma_{i_\ell}\Gamma_{i_\ell} \subseteq \mathcal{O}(H_I)$. The **extension** of a set of partial orderings $C \subseteq (\sigma\Gamma)_I$ to $\sigma\Gamma$ is the set

$$\mathrm{ext}(C) = \{(\sigma_1, \ldots, \sigma_n) \in \sigma\Gamma : (\sigma_{i_1}, \ldots, \sigma_{i_\ell}) \in C\} \subseteq \sigma\Gamma.$$

The canonisation procedure for q-bounded structures is given below.

Algorithm 8 CPT-canonisation of q-bounded structures

Input: A q-bounded structure $\mathfrak{H}$ and sets $\sigma_i\Gamma_i \subseteq \mathcal{O}(H_i)$ for $\Gamma_i \leq \mathrm{sym}(H_i)$, $\sigma_i \in \mathcal{O}(H_i)$

Output: A canonisation $\mathfrak{H}^<$ of $\mathfrak{H}$

$C_0 := \sigma\Gamma$ and $\mathfrak{H}_0^< := \emptyset$
for $s = 1, \ldots, m$ **do**
  Set $I := I_s$ and define $\Delta := \mathrm{Aut}(\mathfrak{H}_I) \cap \Gamma_I$ and $D := \{\tau\Delta : \tau \in (\sigma\Gamma)_I\}$
  Define enumeration $D = \{d_1, \ldots, d_m\}$ of D ▷ possible by Remark 7.1
  Set $C_s := \emptyset \subseteq \sigma\Gamma$ and $\mathfrak{H}_s^< := \emptyset$
  $i := 1$
  while $C_{s-1} \cap \mathrm{ext}(d_i) = \emptyset$ **do**
    $i := i + 1$
  end while
  Set $C_s := C_{s-1} \cap \mathrm{ext}(d_i)$ and $\mathfrak{H}_s^< := \mathfrak{H}_{s-1}^< \cup \tau'(\mathfrak{H}_I)$ for some (all) $\tau' \in d_i$
end for
  return The canonisation $\mathfrak{H}^< := \mathfrak{H}_m^<$ of $\mathfrak{H}$

The only obstacle which prevents us from implementing this procedure in CPT is that we need to find a suitable presentation of the sets C_s. Clearly, it is not possible to store them explicitly as their size is exponential in the size of the input structure. In the following sections we identify classes of q-bounded structures where such a presentation can be based on linear algebra. In the following definition we describe which requirements must be fulfilled by the presentations of the sets C_s in order to implement Algorithm 8.

Definition 7.3. Assume that the sets $\sigma_i\Gamma_i \subseteq \mathcal{O}(H_i)$ are given explicitly. A CPT-definable representation of sets $\tau\Delta$ with $\Delta \leq \Gamma$ and $\tau \in \sigma\Gamma$ is **suitable to guarantee** CPT**-definable canonisation** of q-bounded structures if:

(i) **Consistency**. Given a representation of $\tau\Delta$, it is CPT-definable whether $\tau\Delta \neq \emptyset$.

(ii) **Intersection**. Given two representations of sets $\tau_1\Delta_1$ and $\tau_2\Delta_2$, a representation of the set $\tau_1\Delta_1 \cap \tau_2\Delta_2$ is CPT-definable.

(iii) **Representation of basic sets**. Given a set $\tau\Delta$ with $\tau \in (\sigma\Gamma)_I$ and $\Delta \leq \Gamma_I$ for $I \in \mathcal{P}$, a representation for its extension $\mathrm{ext}(\tau\Delta) \subseteq \sigma\Gamma$ can be defined in CPT.

7.2 The Case of 2-Bounded Structures

Before we present our procedure in its most general form, we take a look at an important special case. We show that 2-bounded structures can be canonised in CPT. In this case we can store the sets of witnessing isomorphisms as solutions of linear equation systems over a finite ring (here over $\mathbb{Z}_2$). According to Definition 7.3, the necessary operations which have to be CPT-definable are a **consistency check** which corresponds to deciding the solvability of a linear equation system, the **intersection operation** which corresponds to combining the equations of two linear systems, and the **representation of basic sets** which corresponds to constructing a linear equation systems over a small set of variables.

While the last two operations are easy to implement, the consistency check is the hard part. Note that it is not known whether the solvability of linear equation systems over finite rings can be defined in CPT. We only know that it is not definable in FPC. We show that CPT can at least define the solvability of special linear equation systems which we call **cyclic linear equation systems** (CES). Luckily it turns out that CESs suffice to capture the structure of witnessing isomorphisms. In this section, we consider CESs over $\mathbb{Z}_2$. The general case of CESs over arbitrary finite rings is handled in Section 7.3.

Definition 7.4. For a set V of variables of size at most we define cyclic constraint over $\mathbb{Z}_2$ as the set of equations $\{v + v' = 1 \mid v \neq v' \in V\}$. A **cyclic linear equation systems** (CES) over $\mathbb{Z}_2$ is a triple $(V, S, \preceq)$ where V is a set of variables, $\preceq$ is a preorder of width two on $V = V_0 \preceq V_1 \preceq \cdots \preceq V_n$, and S is a linear equation system which contains for every V_i a cyclic constraint C_i.

Let us make some simple observations. Without loss of generality we can assume that $|V_i| = 2$ for all $i = 1, \ldots, n$. Then $C_i = \{v + v' = 1, v' + v = 1\}$ where $V_i = \{v, v'\}$. Moreover, because we work over $\mathbb{Z}_2$, we can consider linear terms t to be subsets $t \subseteq V$. Also, we can assume that the term t in every linear equation $e = (t, z) \in S$ is **simplified** meaning that is does not contain two variables from the same block V_i (otherwise, use C_i to simplify the equation).

We assign to each simplified linear term t its **signature** $\mathrm{sgn}(t) \subseteq \{1, \ldots, n\}$, which is the set of indices i such that e contains a variable $v \in V_i$, that is $\mathrm{sgn}(t) := \{i : t \cap V_i \neq \emptyset\}$. In this way we obtain a CPT-definable preorder $\preceq$ on S which is

$$(t, z) \preceq (s, z') \text{ if, and only if, } \mathrm{sgn}(t) < \mathrm{sgn}(s) \text{ or } (\mathrm{sgn}(t) = \mathrm{sgn}(s) \text{ and } z \leq z').$$

We write $S = S_0 \preceq S_1 \preceq \cdots \preceq S_m$ and say that S_i is the i-th block of incomparable equations. Let $e = (s, z)$ and $f = (t, z)$ be equations in S_i and consider the equation $e + f = (s + t, 0)$ which results from adding equation f to equation e. Since $\mathrm{sgn}(s) = \mathrm{sgn}(t)$ we can use the cyclic constraints to simplify the linear term $s + t$ to a constant. Hence, $e + f$ is either equivalent to the trivial equation $0 = 0$ or to the inconsistent equation $1 = 0$, and this only depends on the parity of the number of components $i \in \mathrm{sgn}(s) = \mathrm{sgn}(t)$ for which s and t disagree. We conclude that we can partition linear terms of a fixed signature into two equivalence classes T and T' such that each block S_i of incomparable equations only contains linear terms from one of these classes (or is trivially inconsistent).

The problem that we face when we want to implement an algorithm like Gaussian elimination on CESs in CPT is that we will never be able to pick a specific equation from a block S_i. We will therefore have to find a way to perform all of these choices simultaneously while avoiding the combinatorial explosion that would occur when we actually perform every possible choice individually. The crucial ingredient of our CPT-procedure for solving CESs over $\mathbb{Z}_2$ is the notion of a **hyperterm** which generalises a data structure used in the CPT-procedure of Dawar, Richerby and Rossman for deciding the CFI-query [30]. Intuitively, a hyperterm is a succinct encoding of an equivalence class of linear terms as described above. Most importantly, the syntactic structure of a hyperterm, seen as an object in $\mathrm{HF}(V)$, has a very direct connection with its intended semantics.

Definition 7.5. Let A denote the set of all assignments $\alpha : V \to \mathbb{Z}_2$ which satisfy the equations $v + v' = 1$ for all $V_i = \{v, v'\}$. We inductively define

1. **hyperterms** $T \in \mathrm{HF}(V)$ with associated **dual hyperterms** $S = \tilde{T}$ such that $\tilde{S} = T$,

2. for assignments $\alpha \in A$ the **value** $T[\alpha] \in \mathbb{Z}_2$ such that $T[\alpha] + \tilde{T}[\alpha] = 1$, and

3. the **parity** $p(V_i, T) = p(V_i, \tilde{T}) \in \mathbb{Z}_2$ of variable block V_i in the hyperterms T, $\tilde{T}$.

- $T = 0$ and $S = 1$ are hyperterms with associated dual hyperterms $\tilde{0} = 1$ and $\tilde{1} = 0$. We set $p(V_i, T) = p(V_i, S) = 0$ for all $1 \leq i \leq n$ and $T[\alpha] = 0$ and $S[\alpha] = 1$ for all assignments $\alpha \in A$.

Moreover, for $V_i = \{v, v'\}$, $T = v$ and $S = v'$ are hyperterms, where $\tilde{T} = S$ and $\tilde{S} = T$. We set $p(V_j, T) = p(V_j, S) = 1$ if, and only if, $j = i$. Finally, $T[\alpha] = \alpha(v)$ and $S[\alpha] = \alpha(v')$.

- Let Q, R be hyperterms. Then $T = Q \oplus R := \{(Q, R), (\tilde{Q}, \tilde{R})\}$ is a hyperterm with the associated dual hyperterm $\tilde{T} = \{(\tilde{Q}, R), (Q, \tilde{R})\}$. We set $p(V_i, T) = p(V_i, Q) + p(V_i, R)$ and $T[\alpha] := Q[\alpha] + R[\alpha]$ for $\alpha \in A$.

We need to make sure that the number of active elements does not grow too fast by constructing new hyperterms. Therefore we analyse the complexity of hyperterms as objects in $\mathrm{HF}(V)$. Assume that we have already constructed the hyperterms Q and R together with the corresponding dual hyperterms $\tilde{Q}$ and $\tilde{R}$. In order to construct the new hyperterms $T = Q \oplus R$ and $\tilde{T}$ in CPT we only need to construct the objects (Q, R), $(\tilde{Q}, R)$, $(Q, \tilde{R})$ and $(\tilde{Q}, \tilde{R})$. This means $|\langle T \rangle| = |\langle Q, \tilde{Q}, R, \tilde{R} \rangle| + c$ for a constant c. Therefore, it is safe to construct hyperterms $T \in \mathrm{HF}(V)$ using the $\oplus$-operation for a polynomially bounded number of steps.

We now turn our attention to the connection between the syntax and the semantics of hyperterms. For every $1 \leq i \leq n$ let $\pi_i : V \to V$ denote the transposition of the two variables in $V_i = \{v, v'\}$ extended to a permutation acting on $\mathrm{HF}(V)$. Since hyperterms are objects in $\mathrm{HF}(V)$ we can apply π_i to hyperterms. Interestingly the **syntactic switch** of the two V_i-variables in a hyperterm T either preserves T or maps T to its dual $\tilde{T}$. On the other hand, applying π_i to assignments $\alpha \in A$, that is $\pi_i(\alpha)(w) = \alpha(\pi_i(\alpha))$, can be considered as the corresponding **semantic switch** of the variables in V_i. The next lemma shows that hyperterms are compatible with respect to these syntactic and semantic switches, and moreover, a hyperterm is only influenced by a (semantic or syntactic) switch of variables in V_i, if the block V_i occurs in T with odd parity. We omit the proof which is a straightforward induction on the structure of hyperterms.

Lemma 7.1. *For every hyperterm T, every $1 \leq i \leq n$, and every $\alpha \in A$ we have*

1. $\pi_i(T) = \begin{cases} T, & \text{if } p(V_i, T) = 0, \\ \tilde{T}, & \text{if } p(V_i, T) = 1. \end{cases}$
2. $\pi_i(T)[\alpha] = T[\pi_i(\alpha)]$.

As we want to use hyperterms as succinct encodings of linear terms, our next aim is to define, given a (simplified) linear term t, an equivalent hyperterm T_t in CPT. Let t be a (simplified) linear term. If t consists of a single variable, then t already is a hyperterm. Otherwise there exists a maximal $i \in \mathrm{sgn}(t)$ such that $t = s + v$ for $v \in V_i$ and a linear term $s \subsetneq t$. Then we recursively set $T_t = T_s \oplus v$. Since the set of variable blocks V_i is linearly ordered, the mapping $t \mapsto T_t$ is clearly definable in CPT. The next lemma shows that starting from two equivalent terms we end up with the same hyperterm. Remember that we want to use hyperterms to present a block of equivalent equations.

Lemma 7.2. *Let s and t be linear terms with $\mathrm{sgn}(s) = \mathrm{sgn}(t)$ Then $T_s = T_t$ if s and t disagree on an even number of variable blocks. Otherwise $T_s = \tilde{T}_t$. Moreover, for all $\alpha \in A$ we have $T_t[\alpha] = t[\alpha]$.*

Again we omit the proof, as it goes through by a direct induction. Recall that every block of incomparable equations S_i only contains equations (s, z_i) and (t, z_i) such that s and t disagree on an even number of variable blocks (otherwise, the system is trivially inconsistent). Hence, we can construct in parallel for each equation $(t, z_i) \in S_i$ the hyperterm T_t and end up with a single hyperterm T_i for each block S_i. The pair (T, z) consisting of a hyperterm T and a constant $z \in \mathbb{Z}_2$ is called a **hyperequation**. Since for all $\alpha \in A$ we have that $T_t[\alpha] = t[\alpha]$, the given linear equation system is solvable, if and only if, there exists a solution $\alpha \in A$ of the system of hyperequations $S^* := \{(T_1, z_1), \ldots, (T_m, z_m)\}$. Note that the preorder $\preceq$ on the blocks S_i induces a linear order on the hyperequations of the system S^*. Our aim is to use the method of Gaussian elimination for the system S^* to decide the solvability of the original system S. We state that the usual properties one needs for the correctness of the Gaussian elimination method are still valid for systems of hyperequations.

Lemma 7.3. *Let S^* be a system of hyperequations, and let $(T, z), (T', z') \in S^*$. Then the system S^* and the system $(S^* \setminus \{(T, z)\}) \cup \{(T \oplus T', z + z')\}$ have the same set of solutions.*

Definition 7.6. Let S^* be a system of hyperequations. A hyperequation $(T, z) \in S^*$ **contains** the variable block V_i if $p(V_i, T) = 1$. If $p(V_i, T) = 0$ for all $i = 1, \ldots, n$ we say that (T, z) is **atomic**.

We say that S^* is in **row echelon form** if each non-atomic hyperequation (T, z) contains a variable block V_i which is not contained in any other hyperequation in S^*.

If two hyperequations $(T, z), (T', z') \in S^*$ contain the variable block V_i, then V_i is not contained in the hyperequation $(T \oplus T', z + z')$. Hence, according to Lemma 7.3 we may apply the method of Gaussian elimination to obtain, given S^*, an equivalent system of hyperequations in row echelon form. Because S^* is linearly ordered, we can perform this transformation in CPT. The next lemma tells us that the solvability of systems of hyperequations S^* in row echelon form reduces to checking the atomic hyperequations for consistency.

Lemma 7.4. *Let S^* be a system of hyperequations in row echelon form. Then S^* has a solution $\alpha \in A$ if, and only if, each atomic hyperequation $(T, z) \in S^*$ is consistent.*

Proof. Let $(T, z) \in S^*$ be an atomic hyperequation. By Lemma 7.1 we know that for all $\alpha, \beta \in A$ it holds $T[\alpha] = T[\beta]$. Hence, if $T[\alpha] \neq z$ for some (all) $\alpha \in A$, then clearly the system S^* has no solution.

Assume on the other hand, that every atomic hyperequation $(T, z) \in S^*$ is consistent and let $\alpha \in A$ be an assignment that violates a minimal number of hyperequations. Suppose α is not a solution to S^*. Then let $(T, z) \in S^*$ be such that $T[\alpha] \neq z$. Because of our assumption, T is not atomic and, since S^* is in row echelon form, contains a variable block V_i which is not contained in any other hyperequation in S^*. Let $\alpha' \in A$ be the assignment $\alpha' = \pi_i(\alpha)$. With Lemma 7.1 we compute $T[\pi_i(\alpha)] = \tilde{T}[\alpha] = 1 + T[\alpha] = z$ and $T'[\pi_i(\alpha)] = T'[\alpha]$ for all $(T', z') \in S^* \backslash \{(T, z)\}$. Hence, we have found an assignment that violates less equations than α, contradicting our initial choice of α. □

What is left open is to show that CPT can indeed define the consistency of atomic hyperequations $(T, z) \in S^*$. From Lemma 7.1 we know that $T[\alpha]$ is constant for all $\alpha \in A$. For the sake of explanation, assume we could fix some $\alpha \in A$. Let T' be the hyperterm which results from T by syntactically substituting all occurrences of a variable $v \in V$ by its value $\alpha(v) \in \mathbb{Z}_2$. Then T' is an object in $\mathrm{HF}(\mathbb{Z}_2)$, and clearly, the value of T' is $T[\alpha]$.

Lemma 7.5. *Let T' be a hyperterm with $T' \in \mathrm{HF}(\mathbb{Z}_2)$. Then we can define the value of T' in* CPT.

Proof. As long as T' is different from 0 or 1, we recursively substitute in T' all occurrences of hyperterms $\{(0,0),(1,1)\}$ by 0 and of $\{(0,1),(1,0)\}$ by 1. □

It remains to show how we can obtain from an atomic hyperterm $T \in \mathrm{HF}(V)$ an equivalent hyperterm $T' \in \mathrm{HF}(\mathbb{Z}_2)$. To start let us set $T' := T$. Now we iteratively choose a variable block V_i such that T' still *syntactically* contains variables from the block $V_i = \{v, v'\}$. Restricted to this block, each assignment $\alpha \in A$ might either be the mapping $\beta : (v, v') \mapsto (0,1)$ or $\gamma : (v, v') \mapsto (1,0)$. Since we cannot choose one of the two assignments β or γ in CPT, we try both of them in parallel. Let $T'[V_i \mapsto \beta]$ and $T'[V_i \mapsto \gamma]$ denote the hyperterms which result from T' by syntactically substituting all occurrences of the variables v and v' according to the assignments β and γ, respectively. The crucial observation is that $T'[V_i \mapsto \beta] = T'[V_i \mapsto \gamma]$. To see this, note that $\pi_i(\gamma) = \beta$ and $\pi_i(T') = T'$. Moreover, we have $T'[V_i \mapsto \beta] = \pi_i(T')[V_i \mapsto \pi_i(\beta)]$ which implies that

$$T'[V_i \mapsto \beta] = \pi_i(T')[V_i \mapsto \pi_i(\beta)] = \pi_i(T')[V_i \mapsto \gamma] = T'[V_i \mapsto \gamma].$$

Hence, we can continue the process with $T' := T'[V_i \mapsto \beta] = T'[V_i \mapsto \gamma]$ and finally we obtain the equivalent hyperterm $T' \in \mathrm{HF}(\mathbb{Z}_2)$ in CPT and we conclude:

Theorem 7.3. *The solvability of a* CES *over* $\mathbb{Z}_2$ *can be defined in* CPT.

We are now prepared to show that we can canonise 2-bounded τ-structures $\mathfrak{H} = (H, R_1^{\mathfrak{H}}, \ldots, R_k^{\mathfrak{H}}, \preceq)$ in CPT. As in Section 7.1 we write $H = H_1 \preceq \cdots \preceq H_n$ to denote the ordered partition of H into colour classes H_i of size $q_i \leq 2$. Without loss of generality we assume that $q_i = 2$ for all $i = 1, \ldots, n$.

Using the notation from Section 7.1 we set $\Gamma_i = \mathrm{sym}(H_i)$ and $\sigma_i\Gamma_i = \mathcal{O}(H_i)$. For a colour class $H_i = \{h, h'\}$ this means $\Gamma_i = \{\mathrm{id}_{H_i}, (h\,h')\}$ and $\mathcal{O}(H_i) = \{(h,h') \mapsto ((i,0),(i,1)), (h,h') \mapsto ((i,1),(i,0))\}$.

To obtain a CPT-definable canonisation procedure it remains to specify CPT-definable representations of sets $\tau\Delta$ where $\tau \in \sigma_1\Gamma_1 \times \cdots \times \sigma_n\Gamma_n = \sigma\Gamma$ and $\Delta \leq \Gamma_1 \times \cdots \times \Gamma_n = \Gamma$ which satisfy the requirements summarised in Definition 7.3.

To this end we identify each $\tau = (\tau_1, \ldots, \tau_n) \in \sigma\Gamma$ with a vector $\vec{\tau} \in \mathbb{Z}_2^H$ by setting $\vec{\tau}(h) = j$ where $\tau_i(h) = (i,j)$ for $h \in H_i$. Then for each colour class $H_i = \{h, h'\}$ we have $\vec{\tau}(h) + \vec{\tau}(h') = 1$.

Moreover, we identify each $\gamma = (\gamma_1, \ldots, \gamma_n) \in \Gamma$ with the vector $\vec{\gamma} \in \mathbb{Z}_2^H$ which is defined uniformly on each colour class $H_i = \{h, h'\}$ as $\vec{\gamma}(H_i) = 1$ if $\gamma_i = (h\, h')$ and $\vec{\gamma}(H_i) = 0$ if $\gamma_i = \mathrm{id}_{H_i}$. In this encoding, the usual vector addition of two H-vectors $\vec{\gamma_1}, \vec{\gamma_2} \in \mathbb{Z}_2^H$ with $\gamma_1, \gamma_2 \in \Gamma$ is compatible with the group operation of Γ, i.e. we have $\vec{\gamma_1} + \vec{\gamma_2} = \vec{\gamma}$ where $\gamma = \gamma_1 \circ \gamma_2$. Also for any $\tau_1, \tau_2 \in \sigma\Gamma$ and $\gamma \in \Gamma$ we have

$$\vec{\tau_1} + \vec{\tau_2} = \vec{\delta} \text{ where } \delta = \tau_1^{-1} \circ \tau_2 = \tau_2^{-1} \circ \tau_1, \text{ and,}$$
$$\vec{\tau_1} + \vec{\gamma} = \vec{\rho} \text{ where } \rho = \tau_1 \circ \gamma.$$

This means that the action of Γ on elements in $\sigma\Gamma$ corresponds to the vector addition of the associated vectors in $\mathbb{Z}_2^H$. In particular this shows that the encoding of a set $\tau\Delta$ with $\tau \in \sigma\Gamma$ and $\Delta \leq \Gamma$ is an affine subspace in $\mathbb{Z}_2^H$ (where the associated linear subspace is the set of vectors which represent Δ). Now, since each such affine subspace only contains vectors $\vec{\tau}$ with $\vec{\tau}(h) + \vec{\tau}(h') = 1$ for each colour class $H_i = \{h, h'\}$ and because we have a preorder on the index set H of width two, every such affine space can be represented as the solution space of a CES over $\mathbb{Z}_2$. Let us check that this representation satisfies the requirements from Definition 7.3:

1. **Consistency**. Given a CES over $\mathbb{Z}_2$, its solvability is CPT-definable by Theorem 7.3.

2. **Intersection**. Given two CESs over $\mathbb{Z}_2$, we can combine in CPT the sets of linear equations to obtain a CES whose solution space is the intersection of the solution spaces of the two given CESs.

3. **Representation of basic sets**. Assume we have given for some $I \in \mathcal{P}$ a set of the form $\tau\Delta$ where $\tau \in (\sigma\Gamma)_I$ and $\Delta \leq \Gamma_I$.

   Then we can define in CPT a CES which represents $\mathrm{ext}(\tau\Delta) \subseteq \sigma\Gamma$ just by taking the union of all CESs whose solution space represents $\tau\Delta$. Here we only need to consider CESs which contain (besides the cyclic constraints) only equations with variables in H_I (which is a set of constant size).

Theorem 7.1. 2-*bounded structures can be canonised in* CPT.

7.3 Cyclic Linear Equation Systems

In this section we generalise our notion of cyclic linear equation systems (CESs) over $\mathbb{Z}_2$ to finite rings $\mathbb{Z}_d$ where $d = p^k$ is a prime power. Moreover, we extend our techniques from Section 7.2 to develop a CPT-procedure to express the solvability of CESs over finite rings $\mathbb{Z}_d$.

Definition 7.7. A **cyclic constraint** over $\mathbb{Z}_d$ for a set of variables V is a consistent set of equations such that for all $v, v' \in V$ there is an equation $v - v' = z \in C$ for some $z \in \mathbb{Z}_d$.

Definition 7.8. A **cyclic linear equation systems** (CES) over $\mathbb{Z}_d$, where $d = p^k$ is a prime power, is a triple $(V, S, \preceq)$ where V is a set of variables over $\mathbb{Z}_d$, $\preceq$ is a preorder on $V = V_0 \preceq V_1 \preceq \cdots \preceq V_n$ and S is a linear equation system which contains for every block V_i a cyclic constraint C_i.

In the definition we did not require that $\preceq$ is of bounded width. However, given the cyclic constraint $C_i \subseteq S$ we can assume that $|V_i| = d$ for all $1 \leq i \leq n$. To see this, consider a constraint C_i for the variable block V_i. For every variable $v \in V_i$ and $z \in \mathbb{Z}_m$ we add the linear term $v+z$ as a (syntactically) new variable. Let us denote the resulting set of variables by V_i^*. We define an equivalence relation $\sim$ on V_i^* as follows: $v + z \sim w + z'$ if, and only if, $v - w = z' - z \in C_i$. It is easy to verify that the consistency of C_i implies that $\sim$ is an equivalence relation on V_i^*.

We next define a cyclic constraint C_i^* on the set $V_i^*/\sim$ which contains for every pair $[v + z] \neq [w + z']$ the constraint $[v + z] - [w + z'] = c + z - z'$ where $c \in Z_d$ is chosen such that C_i contains the constraint $v - w = c$. Again it is straightforward to show that C_i^* is well-defined.

To $\alpha : V_i \to \mathbb{Z}_d$ with $\alpha \models C_i$ we associate an assignment $\alpha^* : (V_i^*/\sim) \to \mathbb{Z}_d$ which is given as $\alpha^*([v + z]) = \alpha(v) + z$. Then α^* is well-defined, $\alpha^* \models C_i^*$ and $\alpha^*([v + 0]) = \alpha(v)$. In the other direction, to every $\alpha^* : (V_i^*/\sim) \to \mathbb{Z}_d$ with $\alpha^* \models C_i^*$ we can associate $\alpha : V_i \to \mathbb{Z}_d$ with $\alpha(v) = \alpha^*([v + 0])$ such that $\alpha \models C_i$. We conclude that there exists a one-to-one correspondence between assignments $\alpha : V_i \to \mathbb{Z}_d$ with $\alpha \models C_i$ and assignments $\alpha^* : (V_i^*/\sim) \to \mathbb{Z}_d$ with $\alpha^* \models C_i^*$ which are related via $\alpha(v) = \alpha^*([v + 0])$.

Finally we observe that the number of $\sim$-equivalence classes is precisely d. Indeed for each $v \in V_i$ and $z, z' \in \mathbb{Z}_d$ with $z \neq z'$ we have that $[v+z] \neq [v+z']$ and $[v + z] - [v + z'] = z - z' \in C_i^*$. By substituting each occurrence of a

variable $v \in V_i$ in the original linear equation system by the corresponding $\sim$-equivalence class $[v+0] \in (V_i/\sim)$, and by replacing each cyclic constraint C_i by C_i^* we obtain an equivalent CES with $|V_i| = d$ for all $i = 1, \ldots, n$.

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 $v^{+z} - v = z$. We observe that there exist precisely d different assignments $\alpha : V_i \to \mathbb{Z}_d$ with $\alpha \models C_i$ and each of these is determined by fixing the value of a single variable $v \in V_i$. This allows a generalisation of the notion of hyperterms which we used in Section 7.2 for solving CESs over $\mathbb{Z}_2$.

Definition 7.9. Let A denote the set of assignments which satisfy all cyclic constraints C_i, that means $A := \{\alpha : V \to \mathbb{Z}_d : \alpha \models C_i \text{ for } i = 1, \ldots, n\}$. We inductively define

(i) **hyperterms** T together with their associated **shifted hyperterms** T^{+z} for $z \in \mathbb{Z}_d$ such that $T^{+(z_1+z_2)} = (T^{+z_1})^{+z_2}$ for $z_1, z_2 \in \mathbb{Z}_d$, and $T^{+d} = T$, and

(ii) for an assignment $\alpha \in A$ the **value** $T[\alpha] \in \mathbb{Z}_d$ such that $T^{+z}[\alpha] - T[\alpha] = z$, and

(iii) the **coefficient** $c(V_i, T) = c(V_i, T^{+z}) \in \mathbb{Z}_d$ of variable block V_i in the hyperterms $T, T^{+1}, \ldots, T^{+(d-1)}$.

- For every $z \in \mathbb{Z}_d$ we define the hyperterm $T = z$ and set $T^{+y} = z + y$ for $y \in \mathbb{Z}_d$. We let $c(V_i, T) = c(V_i, T^{+y}) = 0$ for each variable block V_i and all $y \in \mathbb{Z}_d$ and let $T[\alpha] = z$ and $T^{+y}[\alpha] = z + y$ for all assignments $\alpha \in A$ and $y \in \mathbb{Z}_d$.

  Moreover, for $v \in V_i$, $T = v$ is a hyperterm where $T^{+y} = v^{+y}$ for $y \in \mathbb{Z}_d$. We set $c(V_j, T) = c(V_j, T^{+y}) = 1$ for $y \in \mathbb{Z}_d$ if $j = i$ and $c(V_j, T) = c(V_j, T^{+y}) = 0$ otherwise. Finally, we let $T[\alpha] = \alpha(v)$. Then $T^{+y}[\alpha] = \alpha(v^{+y}) = \alpha(v) + y$.

- Let Q, R be hyperterms. Then $T = Q \oplus R := \{(Q^{+z_1}, R^{+z_2}) : z_1 + z_2 = 0\}$ is a hyperterm with shifted hyperterm $T^{+y} = \{(Q^{+z_1}, R^{+z_2}) : z_1 + z_2 = y\}$ for $y \in \mathbb{Z}_d$. We set $c(V_i, T) = c(V_i, T^{+y}) = c(V_i, Q) + c(V_i, R)$, $T[\alpha] := Q[\alpha] + R[\alpha]$ for $\alpha \in A$.

- Let Q be a hyperterm, $z \in \mathbb{Z}_d$. Then $T = z \odot Q := Q \oplus \cdots \oplus Q$ (apply the $\oplus$-operation z-times to Q) is a hyperterm. The definitions of $T^{+y}, c(V_i, T)$ and $T[\alpha]$ follow from the definition of $\oplus$.

The tight correspondence between the syntactic structure and the intended semantics for hyperterms generalises from $\mathbb{Z}_2$ to $\mathbb{Z}_d$.

Definition 7.10. For $\alpha \in A$, $1 \leq i \leq n$ and $z \in \mathbb{Z}_d$ we define the assignment $\alpha^{i:+z} \in A$ which results from a z-**shift** of variable block V_i as

$$\alpha^{i:+z}(v) := \begin{cases} \alpha(v) + z, & v \in V_i, \\ \alpha(v), & \text{else.} \end{cases}$$

Moreover we let $\pi^{i:+z} : V_i \to V_i$ be the **cyclic** z-**shift** on the set V_i which is defined as $\pi^{i:+z}(v) := v^{+z}$ for $v \in V_i$ lifted to a permutation acting on $\mathrm{HF}(V)$.

The following lemma is a generalisation of Lemma 7.1.

Lemma 7.6. *Let $1 \leq i \leq n, z \in \mathbb{Z}_d$, and let T be a hyperterm and let $c = c(V_i, T) \in \mathbb{Z}_d$ be the coefficient of variable block V_i in T.*

(a) Then $\pi^{i:+z}(T) = T^{+c \cdot z}$. In particular if $c = 0$ then $\pi^{i:+z}(T) = T$.

(b) For any assignment $\alpha \in A$ we have $T[\alpha^{i:+z}] = \pi^{i:+z}(T)[\alpha]$.

Proof. We first prove (a) by an induction on the structure of hyperterms.

- Let $T = x$ be a hyperterm for $x \in \mathbb{Z}_d$. Then $c = 0$ and $\pi^{i:+z}(T) = x$.
- Let $T = w$ for $w \in V_j$, $j \neq i$. Then $c = 0$ and $\pi^{i:+z}(T) = w$.
- Let $T = v$ for $v \in V_i$. Then $\pi^{i:+z}(T) = v^{+z}$, $c = 1$, and $T^{+z} = v^{+z}$.
- Let Q, R be hyperterms and let $T = Q \oplus R$. Then $c = c_q + c_r$ where $c_q = c(V_i, Q)$ and $c_r = c(V_i, R)$. Moreover,

$$\begin{aligned} \pi^{i:+z}(T) &= \pi^{i:+z}(\{(Q^{+y_1}, R^{+y_2}) : y_1 + y_2 = 0\}) \\ &\overset{\text{(IH)}}{=} \{(Q^{+y_1 + z \cdot c_q}, R^{+y_2 + z \cdot c_r}) : y_1 + y_2 = 0\} (= Q^{+z \cdot c_q} \oplus R^{+z \cdot c_r}) \\ &= \{(Q^{+y_1}, R^{+y_2}) : y_1 + y_2 = z c_q + z c_r = zc\} = T^{+z \cdot c}. \end{aligned}$$

  In particular this shows that $\pi^{i:+z}(Q \oplus R) = \pi^{i:+z}(Q) \oplus \pi^{i:+z}(R)$.

- Let $T = y \odot Q$ for $y \in \mathbb{Z}_d$ and a hyperterm Q. Then $c = y \cdot c_q$ where $c_q = c(V_i, Q)$. We proceed by induction on y. 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 above and the induction hypothesis we know that

$$\begin{aligned}\pi^{i:+z}(T) &= \pi^{i:+z}((y-1) \odot Q \oplus Q) = \pi^{i:+z}((y-1) \odot Q) \oplus \pi^{i:+z}(Q) \\ \text{(IH)} &= \big((y-1) \odot Q\big)^{+(y-1)\cdot c_q \cdot z} \oplus Q^{+c_q \cdot z} \\ &= \{(\big((y-1) \odot Q\big)^{+y_1}, Q^{+y_2}) : y_1 + y_2 = (y-1) \cdot c_q + c_q \cdot z = z \cdot c\} \\ &= T^{+z \cdot c}.\end{aligned}$$

  In particular we have that $\pi^{i:+z}(y \odot Q) = y \odot \pi^{i:+z}(Q)$.

Using similar arguments, it is easy to show (b). □

Definition 7.11. Two linear terms s, t are **equivalent** ($s \equiv t$) if $s(\alpha) = t(\alpha)$ for all $\alpha \in A$. Two linear equations $e = (s, z_s)$, $f = (t, z_t)$ are **equivalent** if $s - t \equiv z_s - z_t$.

Lemma 7.7. *Let $t = z_t \cdot v_t$ and $s = z_s \cdot v_s$ be two atomic linear terms with $v_t, v_s \in V_i$. If $t \equiv s$ then $z_t = z_s$.*

Proof. Obvious, since we can find a (unique) $z \in \mathbb{Z}_d$ such that $v_t \equiv v_s + z$. □

A linear term t over V can be decomposed into an ordered set of linear subterms which only contain variables from the blocks V_i. More specifically, we let t_{*i} denote the linear subterm of t with variables in V_i, that is $t_{*i} := \{z \cdot v \in t : v \in V_i\}$. Then $t = \bigcup_{i=1}^{n} t_{*i}$.

For CESs over $\mathbb{Z}_2$ and a variable block $V_i = \{v, v'\}$ it was sufficient to consider the cases where $t_{*i} = \{v\}$ and $t_{*i} = \{v'\}$, because the only other possible (non-trivial) linear subterm $t_{*i} = \{v, v'\}$ could be reduced to the constant 1 using the cyclic constraint $v + v' = 1$. Clearly, for CESs over $\mathbb{Z}_d$ the set of possible linear subterms for a variable block V_i is can be more complicated. However, as we show next we can still assume that these linear subterms are within a certain set of nearly atomic linear terms.

Lemma 7.8. *Given a linear term t in which only variables from V_i occur, we can define in* CPT *a minimal constant $y \in \mathbb{Z}_d$, a coefficient $z \in \mathbb{Z}_d$ and a set $W \subseteq V_i$ such that every linear term $s = z \cdot v + y$ 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 every 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 $z_v \cdot v + y_v \equiv z_w \cdot w + y_w$ and thus $z_v \cdot v \equiv z_w \cdot w$. By Lemma 7.7 we have $z_v = z_w$. We fix the minimal $y \in \mathbb{Z}_d$ and $z \in \mathbb{Z}_d$ 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. □

The previous lemma allows to adapt the notion of signatures for CESs over $\mathbb{Z}_d$. We assign to each linear term t its **signature** $\mathrm{sgn}(t) \in (\mathbb{Z}_d \times \mathbb{Z}_d)^n$ as the sequence of pairs (y_i, z_i) as in Lemma 7.8 such that $t_{*i} \equiv z_i \cdot v + y_i$ for suitable $v \in V_i$. In this way we obtain a CPT-definable preorder $\preceq$ on S which is

$$(t, z) \preceq (s, z') \text{ if, and only if, } \mathrm{sgn}(t) < \mathrm{sgn}(s) \text{ or } (\mathrm{sgn}(t) = \mathrm{sgn}(s) \text{ and } z \leq z').$$

We write $S = S_0 \preceq S_1 \preceq \cdots \preceq S_m$ and say that S_i is the i-th block of incomparable equations. Let $(t, z), (s, z) \in S_i$. We claim that either $t \equiv s$ or the linear system is inconsistent. To see this, first note that $t - s = 0$ is a consequence of the linear system, but of course, knowing that $t[\alpha] = s[\alpha]$ for some $\alpha \in A$ does not mean that $t[\alpha] = s[\alpha]$ for all $\alpha \in A$. However, since $\mathrm{sgn}(t) = \mathrm{sgn}(s)$ it is easy to see that the linear term $t - s$ is equivalent to a constant in $\mathbb{Z}_d$. Thus either we have that $t - s \equiv 0$ or the given CES is trivially inconsistent. Hence from now on we assume that for each pair of linear equations $(s, z), (t, z) \in S_i$ the respective linear terms s and t are equivalent.

Our next aim is to translate linear terms t into equivalent hyperterms T_t such that equivalent terms s, t with $\mathrm{sgn}(s) = \mathrm{sgn}(t)$ are mapped to the same hyperterm $T_s = T_t$.

Lemma 7.9. *Let $s = z \cdot v$ and $t = z \cdot w$ be equivalent linear terms with $z \in \mathbb{Z}_d$ and $v, w \in V_i$. Then $z \odot w = z \odot v$.*

Proof. Let $v - w \equiv k \in \mathbb{Z}_d$. Then $z \cdot k = 0$ and $z \odot \pi^{i:+k}(w) = z \odot v$. From the proof of Lemma 7.6 we know that $z \odot w = (z \odot w)^{+z \cdot k} = \pi^{i:+k}(z \odot w) = z \odot \pi^{i:+k}(w)$ which shows the claim. □

We are prepared to describe the CPT-definable translation $t \mapsto T_t$. Given a linear term t we proceed as follows:

- First of all, for every subterm t_{*i} we apply Lemma 7.8 to define $y_i, z_i \in \mathbb{Z}_d$ and $W_i \subseteq V_i$ such that $t_{*i} \equiv z_i \cdot w + y_i$ for all $w \in W_i$. Note that for two different $w, w' \in W_i$ we have $z_i \cdot w \equiv z_i \cdot w'$. Using Lemma 7.9 we conclude that $z_i \odot w = z_i \odot w'$. Thus we can set $T_i := (z_i \odot w)^{+y}$ for some (any) $w \in W_i$ to obtain a hyperterm which is equivalent to t_{*i}.
- Finally we obtain a hyperterm T_t equivalent to t by $T_t := T_1 \oplus \cdots \oplus T_n$.

Lemma 7.10. *Let s, t be a pair of linear terms with $\operatorname{sgn}(s) = \operatorname{sgn}(t)$. If $t - s \equiv \delta$ for an appropriate constant $\delta \in \mathbb{Z}_d$, then $T_t = T_s^{+\delta}$.*

Proof. Let $\operatorname{sgn}(s) = \operatorname{sgn}(t) = ((y_1, z_1), \ldots, (y_n, z_n))$ and let $I = \{i : (y_i, z_i) \neq (0, 0)\}$. For $i \in I$ 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 $t - s \equiv \sum_{i \in I} \delta_i =: \delta$.

Let T_i and S_i be the hyperterms associated to t_{*i} and to s_{*i}, respectively. Using Lemma 7.8 we conclude that $T_i = S_i^{+\delta_i}$. Since by definition we have that $T_t = S_1^{+\delta_1} \oplus \cdots \oplus S_n^{+\delta_n}$ and $T_s = S_1 \oplus \cdots \oplus S_n$ it is easy to show that $T_t = T_s^{+\delta}$. □

Lemma 7.10 shows that for all pairs of equations $(s, z), (t, z) \in S_i$ the translation defined above yields a **unique** hyperterm $T_i := T_t = T_s$. Hence the given linear equation system is solvable, if and only if, there exists a solution (in A) for the system of hyperequations $S^* := \{(T_1, z_1), \ldots, (T_m, z_m)\}$. We observe that the preorder $\preceq$ on the blocks S_i induces a linear order on S^*. Again, we can apply elementary operations to this system.

Lemma 7.11. *Let S^* be a system of hyperequations, and let $(T, z), (T', z') \in S^*$. Then the system S^* and the system $(S^* \setminus \{(T, z)\}) \cup \{(T \oplus T', z + z')\}$ have the same solutions in A.*

Another important difference to the case of CESs over $\mathbb{Z}_2$ is the fact that $\mathbb{Z}_d$ is a finite **ring** rather than a finite **field**. Consequently, an equivalent system in row echelon form does not longer exist. Instead we use the **Hermite normal form** which turns out to be the appropriate generalisation over finite rings $\mathbb{Z}_d$.

To this end, we associate the $m \times n$-matrix $M[S^*] : \{1, \ldots, m\} \times \{1, \ldots, n\} \to \mathbb{Z}_d$ with the system S^* of hyperequations defined as $M[S^*](i, j) := c(V_j, T_i)$. Note that $M[S^*]$ is a matrix over two **ordered** index sets. We say that $M[S^*]$ is in **Hermite normal form** if for appropriate permutation matrices $Q : \{1, \ldots, m\} \times \{1, \ldots, m\} \to \{0, 1\}$ and $P : \{1, \ldots, n\} \times \{1, \ldots, n\} \to \{0, 1\}$ we have that

$$Q \cdot M[S^*] \cdot P = \begin{pmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1n} \\ 0 & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & a_{kk} & \cdots & a_{kn} \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

where $k \leq \min\{m, n\}$ and $a_{11} \mid a_{22} \mid a_{33} \mid \cdots \mid a_{kk}$ and such that for all $1 \leq i \leq k$ and $1 \leq j \leq n$ it holds that $a_{ii} \mid a_{ij}$.

Definition 7.12. Let S^* be a system of hyperequations. We say that $(T, z) \in S^*$ **contains** a variable block V_i if $c(V_i, T) \neq 0$. We say that the hyperequation (T, z) is **atomic** if it does not contain any of the variable blocks V_i. The system S^* is in **Hermite normal form** if $M[S^*]$ is in Hermite normal form.

A very important structural property of the ring $\mathbb{Z}_d$ is the fact that divisibility is a preorder (which is not longer true if d is composed of distinct primes). Using this property it is easy to see that every matrix $M : \{1, \ldots, n\} \times \{1, \ldots, m\} \to \mathbb{Z}_d$ can be transformed via elementary row operations into a matrix in Hermite normal form (iteratively select rows with elements which are minimal with respect to divisibility and eliminate the respective column). Moreover, as we have given a linear order on the set of hyperequations S^*, Lemma 7.11 shows that we can define from S^* in CPT an equivalent system of hyperequations in Hermite normal form. The solvability of systems of hyperequations in Hermite normal form can be characterised as follows.

Lemma 7.12. *Let S^* be a system of hyperequations in Hermite normal form. Then S^* is consistent 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 $(T, z) \in S^*$, let $\ell \geq 1$ be minimal such that $p^\ell \cdot c(V_i, T) = 0$ for all variable blocks V_i (e.g. if $p^\ell = 0 \in \mathbb{Z}_d$ then one non-zero coefficient $c(V_i, T)$ is a unit in $\mathbb{Z}_d$). Then the atomic hyperequation $(p^\ell \odot T, p^\ell \cdot z)$ is consistent.

In order to prove Lemma 7.12 we make use of following result which can easily be obtained by a structural induction on hyperterms.

Lemma 7.13. *For each hyperterm T there exists an equivalent linear term t of the form $z_1 \cdot v_1 + z_2 \cdot v_2 + \cdots + z_n \cdot v_n + y$ where $z_i = c(V_i, T)$, $y \in \mathbb{Z}_d$ and $v_i \in V_i$ for $i = 1, \ldots, n$.*

Proof of Lemma 7.12. Clearly, if S^* is consistent then the conditions (i) and (ii) hold.

For the other direction, we use that S^* is in Hermite normal form. Let $S' \subseteq S^*$ be the subset of non-atomic hyperequations and let $(T_1, z_1), \ldots, (T_k, z_k)$ be an enumeration of S' such that for the $k \times n$-coefficient matrix M associated to S' which is defined as $M(j, i) = a_{ji} = c(V_i, T_j)$, there is a permutation matrix $P : \{1, \ldots, n\} \times \{1, \ldots, n\} \to \{0, 1\}$ such that

$$M \cdot P = \begin{pmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1n} \\ 0 & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & a_{kk} & \cdots & a_{kn}. \end{pmatrix},$$

where $a_{11} \mid a_{22} \mid a_{33} \mid \cdots \mid a_{kk}$ and for all $1 \leq j \leq k$ and $1 \leq i \leq n$ it holds that $a_{jj} \mid a_{ji}$. We use Lemma 7.13 to obtain for every hyperterm T_j an equivalent linear term $\sum_{i=1}^{n} a_{ji} \cdot v_i + y_j$ for appropriate $v_i \in V_i$ and $y_j \in Z_d$.

Let $1 \leq j \leq k$. By condition (ii) we know that $(p^\ell \odot T_j, p^\ell \cdot z_j)$ is consistent where p^ℓ 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 $u \in \mathbb{Z}_d^*$.

The system of hyperequations S' is consistent if, and only if, the system of linear equations $(M \cdot P) \cdot \bar{x} = b$ is consistent where $\bar{x} = (v_1, \ldots, v_n)$ and

$$M \cdot P = \begin{pmatrix} a_{11} & \cdots & \cdots & \cdots & a_{1n} \\ 0 & \ddots & \vdots & \cdots & \vdots \\ 0 & 0 & a_{kk} & \cdots & a_{kn}. \end{pmatrix}, b = \begin{pmatrix} z_1 - y_1 \\ \vdots \\ z_k - y_k. \end{pmatrix},$$

so it suffices to verify this. Let $\alpha \in A$ be such that for a maximal $j \leq k$ all equations with index $j' > j$ are satisfied under the assignment α. Now we have $j = 0$, because $a_{jj} \mid a_{ji}$ for all $i = 1, \ldots, n$ and $a_{jj} \mid (z_j - y_j)$ thus $a_{jj} \mid \sum_{i>j} a_{ji} \cdot \alpha(v_{ji}) - (z_j - y_j)$. □

This means that for a system of hyperequations in Hermite normal form, deciding its solvability reduces to checking the consistency of a set of atomic hyperequations. Using Lemma 7.6 this can be done as in the case of CESs over $\mathbb{Z}_2$.

Lemma 7.14. *Let T' be a hyperterm with $T' \in \mathrm{HF}(\mathbb{Z}_d)$. Then the value of T' can be defined in* CPT.

Proof. We recursively substitute in T' all occurrences of hyperterms $\{(z_1, z_2) : z_1, z_2 \in \mathbb{Z}_d, z_1 + z_2 = z\}$ by $z \in Z_d$ until T' is a constant in $\mathbb{Z}_d$. □

Lemma 7.15. *Let $T \in \mathrm{HF}(V)$ be an atomic hyperterm. Then we can define in* CPT *an equivalent hyperterm $T' \in \mathrm{HF}(\mathbb{Z}_2)$.*

Proof. We start with $T' := T$ and choose a variable block V_i such that T' **syntactically** contains variables in V_i. Let β, γ be two (partial) assignments $\beta, \gamma : V_i \to \mathbb{Z}_d$ which satisfy the cyclic constraint C_i. Then $\gamma = \beta^{i:+z}$ for an appropriate $z \in \mathbb{Z}_d$. Let $T'[V_i \mapsto \beta]$ and $T'[V_i \mapsto \gamma]$ denote the hyperterms which result from T' by syntactically substituting all occurrences of the variables v by their values $\beta(v)$ and $\gamma(v)$ under the assignments β and γ, respectively. Again, the crucial observation is that $T'[V_i \mapsto \beta] = T'[V_i \mapsto \gamma]$. First of all, note that $T[V_i \to \beta] = \pi^{i:-z}(T)[V_i \to \gamma]$ (in the first case, $v \in V_i$ is substituted by $\beta(v)$ and in the second case, $v \in V_i$ is first mapped to v^{-z} which is then substituted by $\gamma(v^{-z}) = \beta(v^{-z}) + z = \beta(v)$). Since Lemma 7.6 shows that $T = \pi^{i:-z}(T)$, the claim follows. Hence, we can continue the process with $T' := T'[V_i \mapsto \beta]$ for some (all) $\beta : V_i \to \mathbb{Z}_d$ which satisfy the cyclic constraint C_i. In this way we finally obtain the equivalent hyperterm $T' \in \mathrm{HF}(\mathbb{Z}_2)$ in CPT. □

Altogether we have shown:

Theorem 7.4. *The solvability of* CESs *over $\mathbb{Z}_d$ can be defined in* CPT.

7.4 A CPT Procedure for Canonising q-Bounded Structures with Abelian Colours

In this section we use the CPT-procedure for solving CESs over finite rings $\mathbb{Z}_d$ to show that q-bounded structures with **abelian symmetries** can be canonised in CPT.

Definition 7.13. Let $\mathcal{K}$ be a class of q-bounded τ-structures. We say that $\mathcal{K}$ has **(CPT-definable) abelian symmetries** if there exist CPT-programs which define, given any $\mathfrak{H} \in \mathcal{K}$, on each colour class $H_i \subseteq H$ a **transitive abelian** group $\Gamma_i \leq \mathrm{sym}(H_i)$ together with two linear orderings on $\{\sigma\Gamma_i : \sigma \in \mathcal{O}(H_i)\}$ and on Γ_i.

First of all, let us motivate this definition by giving an example of a class of q-bounded structures with abelian symmetries (up to a CPT-definable reduction). Recall that for $i = 1, \ldots, n$ we denote by $\mathfrak{H}_i$ the substructure of $\mathfrak{H}$ induced on the colour class H_i.

Definition 7.14. A class $\mathcal{K}$ of q-bounded structures has **abelian colours** if for all structures $\mathfrak{H} \in \mathcal{K}$ and for all colour classes H_i of $\mathfrak{H}$ the automorphism group $\mathrm{Aut}(\mathfrak{H}_i)$ is abelian.

Theorem 7.5. *Let $\mathcal{K}$ be a class with abelian colours. Then there exists a* CPT*-program which defines, given $\mathfrak{H} \in \mathcal{K}$, a refinement $\preceq_r^{\mathfrak{H}}$ of the preorder $\preceq^{\mathfrak{H}}$ on H such that the class $\mathcal{K}'$ of all structures $\mathfrak{H}' = \mathfrak{H}[\preceq^{\mathfrak{H}} \setminus \preceq_r^{\mathfrak{H}}]$ has abelian symmetries.*

Since for the translation $\mathfrak{H} \in \mathcal{K} \mapsto \mathfrak{H}' \in \mathcal{K}'$ we only refine the preorder on the universe H, it is easy to obtain a canonisation of the structure $\mathfrak{H} \in \mathcal{K}$ from a canonisation of the structure $\mathfrak{H}'$. In other words, if we can canonise in CPT classes of q-bounded structures with abelian symmetries, then we can also canonise classes of q-bounded structures $\mathfrak{H}$ where $\mathrm{Aut}(\mathfrak{H}_i)$ is abelian for all colour classes H_i.

To prove Theorem 7.5, let $\mathcal{K}$ be a class with abelian colours, and let $\mathfrak{H} \in \mathcal{K}$ with colour classes $H = H_1 \preceq \cdots \preceq H_n$. In order to construct the required groups $\Gamma_i \leq \mathrm{sym}(H_i)$, a good starting point is to set $\Gamma_i := \mathrm{Aut}(\mathfrak{H}_i)$. Then Γ_i is an abelian group which clearly is CPT-definable as its size is constant.

However, it is not clear whether we can define in CPT, as required by Definition 7.13, a linear order on Γ_i and on $\{\sigma\Gamma_i : \sigma \in \mathcal{O}(H_i)\}$. Moreover, Γ_i might not be transitive. Let us discuss the first two points first.

Lemma 7.16. *If $\Gamma_i := \mathrm{Aut}(\mathfrak{H}_i)$ is abelian, then a linear order on Γ_i and on $\{\sigma\Gamma_i : \sigma \in \mathcal{O}(H_i)\}$ is* CPT*-definable.*

Proof. A linear order on $\{\sigma\Gamma_i : \sigma \in \mathcal{O}(H_i)\}$ is CPT-definable, because two different sets $\tau\Gamma_i \neq \sigma\Gamma_i$ with $\tau, \sigma \in \mathcal{O}(H_i)$ induce two different *ordered* structures $\tau(\mathfrak{H}_i) \neq \sigma(\mathfrak{H}_i)$.

To see that we can define a linear order on Γ_i, we use that Γ_i is abelian. First we fix some set $\sigma\Gamma_i$ for $\sigma \in \mathcal{O}(H_i)$. Every ordering $\tau \in \sigma\Gamma_i$ induces an isomorphism $\varphi_\tau : \Gamma_i \to (\Gamma_i)^\tau, \gamma \mapsto \gamma^\tau$. But $(\Gamma_i)^\tau$ acts on an ordered set and hence we can define a linear order on $(\Gamma_i)^\tau$. If we could fix an isomorphism φ_τ we could then transfer the order to Γ_i. We will show that for all $\tau, \rho \in \sigma\Gamma_i$ we have that $\varphi_\tau = \varphi_\rho$, which is sufficient to define the linear order as described above. Indeed if $\tau = \sigma\gamma_\tau \in \sigma\Gamma$ then we have

$$\tau\gamma\tau^{-1} = \sigma\gamma_\tau\,\gamma\,\gamma_\tau^{-1}\sigma^{-1} \overset{\Gamma_i \text{ abelian}}{=} \sigma\gamma\sigma^{-1}.$$

□

Secondly, assume that the action of Γ_i on H_i is not transitive. Again, for each colour class H_i, we can fix a set $\sigma\Gamma_i \subseteq \mathcal{O}(H_i)$ with $\sigma \in \mathcal{O}(H_i)$. Let $X \subseteq H_i$ be a Γ_i-orbit. For every $\tau, \rho \in \sigma\Gamma_i$ we have $\tau(X) = \rho(X) \subseteq \{0, \ldots, |H_i| - 1\}$. Hence we can define for every colour class H_i in CPT a linear order $\leq_i$ on the set of Γ_i-orbits by setting $X \leq_i Y$ for two Γ_i-orbits $X, Y \subseteq H_i$ if $\tau(X) \leq \tau(Y)$ for some (all) $\tau \in \sigma\Gamma_i$. As a result we obtain $\preceq_r^{\mathfrak{H}}$ as the (CPT-definable) refinement of $\preceq^{\mathfrak{H}}$ where for $x, y \in H_i$ we only let $x \preceq_r^{\mathfrak{H}} y$ if $\Gamma(x) \leq_i \Gamma(y)$.

It remains to show that the class $\mathcal{K}'$ of structures $\mathfrak{H}' = \mathfrak{H}[\preceq^{\mathfrak{H}} \setminus \preceq_r^{\mathfrak{H}}]$ allows CPT-definable abelian symmetries. Let Δ_j denote the restriction of Γ_i to the orbit X_j. Alternatively, Δ_j might be viewed as the group Γ_i/Λ_j^i, where Λ_j^i denotes the pointwise stabiliser of X_j in Γ_i, i.e. $\Lambda_j^i = \{\gamma \in \Gamma_i : \gamma(x) = x \text{ for } x \in X_j\}$.

Clearly, Δ_j is an abelian group and from the CPT-definable linear order on Γ_i we get a CPT-definable linear order on Δ_j. Furthermore, let $\sigma, \tau \in \mathcal{O}(X_j)$ such that $\sigma\Delta_j \neq \tau\Delta_j$. If we fix sets $\bar{\sigma}\Gamma_i$ and $\bar{\tau}\Gamma_i$ for $\bar{\sigma}, \bar{\tau} \in \mathcal{O}(H_i)$ such that the restriction of $\bar{\sigma}\Gamma_i$ and $\bar{\tau}\Gamma_i$ to the orbit X_j coincides with $\sigma\Delta_j$ and $\tau\Delta_j$

respectively, then $\bar{\sigma}\Gamma_i \neq \bar{\tau}\Gamma_i$. This shows that the CPT-definable linear order on $\{\sigma\Gamma_i : \sigma \in \mathcal{O}(H_i)\}$ induces a CPT-definable linear order on $\{\sigma\Delta_j : \sigma \in \mathcal{O}(X_j)\}$ for each Γ_i-orbit X_j. Altogether this yields Theorem 7.5.

We proceed by showing that every class $\mathcal{K}$ of q-bounded structures with abelian symmetries allows CPT-definable canonisation. Let $\mathfrak{H} \in \mathcal{K}$ with colour classes $H = H_1 \preceq \cdots \preceq H_n$ and let $\Gamma_i \leq \mathrm{sym}(H_i)$ denote the associated abelian transitive groups. If we want to use our generic CPT-canonisation procedure from Section 7.1, it suffices to find CPT-definable representations of sets $\tau\Delta$ where $\Delta \leq \Gamma_1 \times \cdots \times \Gamma_n$ and $\tau \in \mathcal{O}(H)$ which satisfy the requirements of Definition 7.3.

Analogously to the case of 2-bounded structures in Section 7.2 we encode the sets $\tau\Delta$ by CESs over $\mathbb{Z}_d$. The crucial step is to find such representations for the basic sets $\sigma\Delta \subseteq \mathcal{O}(H_i)$ with $\Delta \leq \Gamma_i$ and $\sigma \in \mathcal{O}(H_i)$ for each colour class H_i.

Lemma 7.17. *Given a set $B \subseteq \mathrm{HF}(H)$ with $|B| \leq q$ and an abelian transitive group $\Gamma \leq \mathrm{sym}(B)$ which can be written as the direct sum of k cyclic subgroups of prime-power order, i.e. $\Gamma = \langle\delta_1\rangle \oplus \cdots \oplus \langle\delta_k\rangle$ for $\delta_1, \ldots, \delta_k \in \Gamma$ where $|\delta_i| = d_i = p_i^{\ell_i}$ is a prime-power, and given a set $\sigma\Gamma \subseteq \mathcal{O}(B)$ for $\sigma \in \mathcal{O}(B)$*

- *we can define in* CPT *sets $W_1, \ldots, W_k \subseteq \mathrm{HF}(B)$ together with a linear ordering $W_1 < W_2 < \cdots < W_k$ such that $|W_i| = d_i$, and*
- *if we set $L_i := \mathbb{Z}_{d_i}^{W_i}$ and let $e_i \in L_i$ denote the L_i-unit vector which is $e_i(w) = 1$ for all $w \in W_i$, then we can define in* CPT *an embedding $\varphi : \sigma\Gamma \to L_1 \times \cdots \times L_k$ which respects the action of Γ on $\sigma\Gamma$ in the following way. For all $\tau \in \sigma\Gamma$ and $\gamma = \ell_1 \cdot \delta_1 \oplus \cdots \oplus \ell_k \cdot \delta_k \in \Gamma$ we have that*

$$\varphi(\tau \circ \gamma) = \varphi(\tau) + (\ell_1 \cdot e_1, \cdots, \ell_k \cdot e_k).$$

Proof. First of all, as we have given a set $\sigma\Gamma \subseteq \mathcal{O}(B)$ we can define, as in the proof of Lemma 7.16, a linear order on Γ. Hence it is possible to fix appropriate $\delta_1, \ldots, \delta_k \in \Gamma$ which yield the decomposition of Γ . We proceed recursively by induction on k.

If $k = 1$, then $\Gamma = \langle\delta\rangle$ is a cyclic group of prime-power order which acts transitively on $W := B$. The proof of Lemma 7.16 shows that we can define the mapping $\Gamma \to \Gamma^\sigma, \gamma \mapsto \gamma^\sigma$ in CPT. We choose $\rho \in \mathrm{sym}(\{0, \ldots, |B| - 1\})$

such that $(\delta^\sigma)^\rho = \delta^{\rho\sigma} = (0\,1\,2\cdots|B|-1) \in \mathrm{sym}(\{0,\dots,|B|-1\})$. We let $L = \mathbb{Z}_d^W$ and denote by $e \in L$ the L-unit vector $e(w) = 1$ for all $w \in W$.

Then the mapping $\varphi : \sigma\Gamma \to L, \tau \mapsto \varphi(\tau)$ where $\varphi(\tau)[w] := \rho\tau(w)$ for $w \in W$ is CPT-definable. We show that $\varphi(\tau \circ \delta) = \varphi(\tau) + e$ for for all $\tau = \sigma\delta^s \in \sigma\Gamma$. To verify this let $w \in W$. Then $(\varphi(\tau) + e)[w] = \varphi(\tau)[w] + 1 = \delta^{\rho\sigma}\rho\sigma\delta^s(w) = \rho\sigma\delta^{s+1}(w)$.

Let $k > 1$. Then $\Gamma = \Delta \oplus \Lambda$ where $\Delta = (\delta_1)$ and $\Lambda = (\delta_2) \oplus \cdots \oplus (\delta_k)$. Let $\bar{X} := \{X_0, \dots, X_{t-1}\}$ be the set of Λ-orbits and let $\bar{Y} := \{Y_0, \dots, Y_{s-1}\}$ be the set of Δ-orbits. Then Δ and Γ act transitively on $\bar{X}$ and on $\bar{Y}$, respectively. Note that $t = d_1$ and $s = \Pi_{j=2}^k d_j$.

As a first step we partition $\{0,\dots,|B|-1\}$ into t segments $S_0 = \{0,\dots,s-1\}$, $S_1 = \{s,\dots,2s-1\},\dots,S_{t-1} = \{(t-1)s,\dots,ts-1\}$ which are modulo s just t disjoint copies of the segment $\{0,\dots,s-1\}$. Moreover, we fix $\rho \in \mathrm{sym}(\{0,\dots,|B|-1\})$ such that for every $\sigma\gamma \in \sigma\Gamma$ the following holds:

(i) for every Λ-orbit $X_i \in \bar{X}$ we have $\rho\sigma\gamma(X_i) = S_j$ for some $j = 0,\dots,t-1$, and

(ii) for every Δ-orbit $Y_i \in \bar{Y}$ we have

$$\rho\sigma\gamma(Y_i) = \{0+r, s+r, 2s+r, \dots, (t-1)s+r\}$$

for some $0 \le r < s$. Hence, for a Δ-orbit Y_i and $y \in Y_i$ we have $\rho\sigma\gamma(y) = r \mod s$.

To show the existence of such $\rho \in \mathrm{sym}(\{0,\dots,|B|-1\})$ note that we can find for $\sigma \in \mathcal{O}(B)$ an appropriate $\rho \in \mathrm{sym}(\{0,\dots,|B|-1\})$ such that $\sigma \in \sigma\Gamma$ satisfies the properties (i) and (ii) for this particular choice of ρ. But then it is easy to check that for the same ρ indeed each $\sigma\gamma \in \sigma\Gamma$ satisfies the properties (i) and (ii).

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 $\eta : \rho\sigma\Gamma \to \mathcal{O}(\bar{X}) \times \mathcal{O}(\bar{Y}), \rho\sigma\gamma \mapsto \eta(\rho\sigma\gamma)$ if we let $\eta(\rho\sigma\gamma) \in \mathcal{O}(\bar{X}) \times \mathcal{O}(\bar{Y})$ be the linear order which assigns to every Λ-orbit $X \in \bar{X}$ the position j for $0 \le j \le t-1$ such that $\rho\sigma\gamma(X) = S_j$ and to each Δ-orbit $Y \in \bar{Y}$ the position $0 \le r \le s-1$ such that $\rho\sigma\gamma(Y) = \{0+r, s+r, \dots, (t-1)s+r\}$. For all $\delta \oplus \lambda \in \Delta \oplus \Lambda$ and $\tau \in \rho\sigma\Gamma$ we have $\eta(\tau \circ (\delta \oplus \lambda)) = (\eta(\tau) \circ \delta, \eta(\tau) \circ \lambda)$. In particular we have that $\eta(\rho\sigma\Gamma) = \nu_X\Delta \times \nu_Y\Lambda$ 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

$$\varphi^X : \nu_X \Delta \to L_1 \text{ and } \varphi^Y : \nu_Y \Lambda \to L_2 \times \cdots \times L_k,$$

with the appropriate properties stated above. Now we put everything together to obtain a CPT-definable embedding $\varphi : \sigma\Gamma \to L_1 \times \cdots \times L_k$ via

$$\varphi(\sigma\gamma) = \varphi^X(\eta(\tau\sigma\gamma) \restriction \bar{X}) \times \varphi^Y(\eta(\tau\sigma\gamma) \restriction \bar{Y}).$$

It is easy to check that the mapping φ has the desired properties. □

We fix for all $i = 1, \ldots, n$ a set $\sigma_i \Gamma_i$ with $\sigma_i \in \mathcal{O}(H_i)$ and let $\sigma\Gamma = \sigma_1\Gamma_1 \times \cdots \times \sigma_n\Gamma_n$. By Lemma 7.17 we can write $\Gamma_i = \langle \delta_1^i \rangle \oplus \cdots \oplus \langle \delta_{k_i}^i \rangle$ where $|\delta_j^i| = d_j^i$ is a prime-power and define for each $i = 1, \ldots, n$ in CPT

- sets $W_1^i < W_2^i < \cdots < W_{k_i}^i$ of size $|W_j^i| = d_j^i$ and for $L_j^i := \mathbb{Z}_{d_j^i}^{W_j^i}$ embeddings
$$\varphi^i : \sigma_i \Gamma_i \to L_1^i \times \cdots \times L_{k_i}^i,$$
- such that for the L_j^i-unit vectors $e_j^i \in L_j^i$, each $\gamma = \ell_1 \cdot \delta_1^i \oplus \cdots \oplus \ell_{k_i} \cdot \delta_{k_i}^i \in \Gamma_i$ and each $\tau \in \sigma_i \Gamma_i$ it holds that
$$\varphi^i(\tau \circ \gamma) = \varphi^i(\tau) + (\ell_1 \cdot e_1^i, \ldots, \ell_{k_i} \cdot e_{k_i}^i).$$

If we let $L = L_1^1 \times \cdots \times L_{k_1}^1 \times \cdots \times L_1^n \times \cdots \times L_{k_n}^n$, then we can combine the mappings φ^i to obtain a CPT-definable mapping $\varphi : \sigma\Gamma \to L, (\tau_1, \ldots, \tau_n) \mapsto (\varphi^1(\tau_1), \ldots, \varphi^n(\tau_n))$.

Since

$$\Gamma = \Gamma_1 \times \cdots \times \Gamma_n = \langle \delta_1^1 \rangle \oplus \cdots \oplus \langle \delta_{k_1}^1 \rangle \times \cdots \times \langle \delta_1^n \rangle \oplus \cdots \oplus \langle \delta_{k_n}^n \rangle$$

we also obtain a definable group embedding $\psi : \Gamma \to L$ as the homomorphic extension of $\psi(\delta_j^i) = e_j^i$ for $i = 1, \ldots, n$ and $j = 1, \ldots, k_i$. For all $\tau \in \sigma\Gamma$ and $\gamma \in \Gamma$ we have

$$\varphi(\tau \circ \gamma) = \varphi(\tau) + \psi(\gamma).$$

Let us now consider for some set $\sigma_i\Gamma_i$ the image under φ restricted to one component L^i_j, i.e. the set $(\varphi(\sigma_i\Gamma_i) \restriction L^i_j) \subseteq L^i_j$. If we denote by $E^i_j := \{\ell \cdot e^i_j : 0 \leq \ell \leq d^i_j - 1\} \subseteq L^i_j$, then $O^i_j := (\varphi(\sigma_i\Gamma_i) \restriction L^i_j) = (\varphi(\sigma_i) \restriction L^i_j) + E^i_j$. This means that for two vectors $x, y \in O^i_j$ it holds that $x - y \in E^i_j$. This in turn implies that for all vectors $x, y \in O^i_j$ and indices $w, w' \in W^i_j$ we have $x(w) - x(w') = y(w) - y(w')$. Consequently we can define a cyclic constraint C^i_j on the set W^i_j such that O^i_j precisely corresponds to the set of assignments $\alpha : W^i_j \to \{0, \ldots, d^i_j - 1\}$ with $\alpha \models C^i_j$.

Let $P := \{p_1, \ldots, p_s\}$ be the set of all primes p_i such that Γ contains elements of order p_i. For $p \in P$ let $\Gamma^p_i \leq \Gamma_i$ denote the subgroup of Γ_i which consists of all elements $\gamma \in \Gamma_i$ whose order is a power of p. Then $\Gamma_i = \Gamma^{p_1}_i \oplus \cdots \oplus \Gamma^{p_s}_i$. In particular we have $\psi(\Gamma_i) = \psi(\Gamma^{p_1}_i) + \cdots + \psi(\Gamma^{p_s}_i)$.

Similarly, for any subgroup $\Delta \leq \Gamma$ and prime $p \in P$ we let $\Delta^p \leq \Delta$ denote the subgroup of Δ which consists of all elements $\delta \in \Delta$ whose order is a power of p. Then $\Delta = \Delta^{p_1} \oplus \cdots \oplus \Delta^{p_s}$ and $\Delta^p \leq \Gamma^p_1 \times \Gamma^p_2 \times \cdots \times \Gamma^p_n =: \Gamma^p$.

Of course we also obtain a corresponding decomposition of L. For $p \in P$ we let $L[p] = \{(v^1_1, \ldots, v^1_{k_1}, \ldots, v^n_1, \ldots, v^n_{k_n}) \in L :$ if $v^i_j \neq 0$ then d^i_j is a p-power$\}$. Then $\psi(\Gamma^p) \leq L[p]$ and $L = L[p_1] \oplus \cdots \oplus L[p_s]$. For $\tau \in \mathcal{O}(H)$ and $\Delta \leq \Gamma$ we have

$$\varphi(\tau\Delta) = \varphi(\tau(\Delta^{p_1} \oplus \cdots \oplus \Delta^{p_s})) = \underbrace{\varphi(\tau)}_{\in L = L[p_1] \oplus \cdots \oplus L[p_s]} + \underbrace{\psi(\Delta^{p_1})}_{\leq L[p_1]} + \cdots + \underbrace{\psi(\Delta^{p_s})}_{\leq L[p_s]}.$$

If we write $\varphi(\tau)^{L[p]}$ to denote the projection of $\varphi(\tau) \in L = L[p_1] \oplus \cdots \oplus L[p_s]$ onto the component $L[p]$ then we obtain

$$\varphi(\tau\Delta) = \big(\underbrace{\varphi(\tau)^{L[p_1]} + \psi(\Delta^{p_1})}_{\subseteq L[p_1]}, \ldots, \underbrace{\varphi(\tau)^{L[p_s]} + \psi(\Delta^{p_s})}_{\subseteq L[p_s]}\big) \subseteq L[p_1] \oplus \cdots \oplus L[p_s] = L.$$

Hence, in order to represent $\varphi(\tau\Delta)$ it suffices to represent the individual components $\varphi(\tau)^{L[p]} + \psi(\Delta^p) \subseteq L[p]$ by CESs. To this end, let us fix the set of variables as $W[p] := \biguplus\{W^i_j : d^i_j$ is a p-power$\}$ which is the index set of vectors in $L[p]$. Moreover, let $d := p^\ell = \max\{d^i_j : d^i_j$ is a p-power$\}$. As a second technical preparation we note that vectors in $L[p]$ may have entries in different rings $\mathbb{Z}_{d_1}, \mathbb{Z}_{d_2}$ for $d_1 = p^{\ell_1} \neq p^{\ell_2} = d_2$. By the choice of d we know that for every such $d' = d^i_j = p^k$ 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 $\mathbb{Z}^W_{d'}$ with vectors

in $\mathbb{Z}_d^W$. Of course, this embedding is not surjective. However, we can add for each set $L_j^i = Z_{d_j^i}^W$ which we lifted via a mapping $\iota : L_j^i \to \mathbb{Z}_d^W$ the 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 the cyclic constraints in C have to be lifted in a similar fashion.

With this preparation we identify $L[p]$ with a subspace of $\mathbb{Z}_d^{W[p]}$ and we show that we can represent $\varphi(\tau)^{L[p]} + \psi(\Delta^p) \subseteq L[p]$ as a CES 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 relevant sets $W_j^i \subseteq W[p]$ then the set of $L[p]$-vectors which satisfy the cyclic constraints in $C[p]$ is $\varphi(\tau)^{L[p]} + \psi(\Gamma^p)$. The question remains whether we can add an appropriate set of linear equations to obtain a CES which represents $\varphi(\tau)^{L[p]} + \psi(\Delta^p) \subseteq \varphi(\sigma)^{L[p]} + \psi(\Gamma^p)$.

To answer this question we analyse the algebraic structure of the set $\varphi(\tau)^{L[p]} + \psi(\Delta^p)$ from a general point of view. Assume for some set W and some prime power $d = p^\ell$ we have given 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 Δ. 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 $B := Q \cdot A \cdot R$ 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 $M_B \cdot \bar{x} = 0$ has $\langle B \rangle$ as its solution space. We claim that for $M_A := M_B \cdot Q$, the linear equation system $M_A \cdot \bar{x} = 0$ has $\langle A \rangle$ as its solution space. To verify this it suffices to check that $Q \cdot \langle A \rangle = \langle B \rangle$. Then for every $\bar{w} \in \mathbb{Z}_d^W$ we have that $M_A \cdot \bar{w} = 0$ if, and only if, $Q \cdot \bar{w} \in \langle B \rangle$ if, and only if, $\bar{w} \in \langle A \rangle$.

Finally, to capture the algebraic structure of the set $\varphi(\tau)^{L[p]} + \psi(\Delta^p)$ we show how we can represent $\bar{w} + \Delta \subseteq \mathbb{Z}_d^W$ for a given vector $\bar{w} \in \mathbb{Z}_d^W$ 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 $M \cdot \bar{x} = 0$ has solution space Δ. Then for $\bar{w} + \Delta$ we just take as an appropriate linear equation system $M \cdot \bar{x} = M \cdot \bar{w}$ which has $\bar{w} + \Delta$ as solution space. In particular note that for any $\bar{v} \in \bar{w} + \Delta$ we have that $M \cdot \bar{w} = M \cdot \bar{v}$.

We conclude that for $p \in P$ the set $\varphi(\tau)^{L[p]} + \psi(\Delta^p)$ can be represented as a CES $\mathcal{S}_p$ over $\mathbb{Z}_d$ with variable set $W[p]$. Putting everything together we

obtain an encoding of sets $\tau\Delta$ with $\Delta \leq \Gamma$ and $\tau \in \sigma\Gamma$ as a sequence of CESs $(\mathcal{S}_{p_1}, \ldots, \mathcal{S}_{p_s})$ where $\mathcal{S}_p$ represents $\varphi(\tau)^{L[p]} + \psi(\Delta^p)$. We claim that this encoding is suitable with respect to Definition 7.3 and thus yields a CPT-definable canonisation procedure for a class of q-bounded structures with abelian symmetries.

(i) **Consistency**. To decide whether $(\mathcal{S}_{p_1}, \ldots, \mathcal{S}_{p_s})$ represents a non-empty set $\tau\Delta$ we just need to check whether each of the CESs $\mathcal{S}_p$ is consistent. This is CPT-definable by Theorem 7.4.

(ii) **Intersection**. Assume we have given two representations of sets $\tau_1\Delta_1$ and $\tau_2\Delta_2$ as sequences of CESs $(\mathcal{S}_{p_1}, \ldots, \mathcal{S}_{p_s})$ and $(\mathcal{T}_{p_1}, \ldots, \mathcal{T}_{p_s})$ such that the solutions of the CES $\mathcal{S}_p$ are $\varphi(\tau_1)^{L[p]} + \psi(\Delta_1^p)$ and the solutions of $\mathcal{T}_p$ are $\varphi(\tau_2)^{L[p]} + \psi(\Delta_2^p)$.

Then $\varphi(\tau_1\Delta_1 \cap \tau_2\Delta_2) = \varphi(\tau_1\Delta_1) \cap \varphi(\tau_2\Delta_2)$ is represented by the sequence of CESs $(\mathcal{S}_{p_1} \cup \mathcal{T}_{p_1}, \ldots, \mathcal{S}_{p_s} \cup \mathcal{T}_{p_s})$ where $\mathcal{S}_p \cup \mathcal{T}_p$ is the CES that results from combining the linear equations of $\mathcal{S}_p$ and $\mathcal{T}_p$. This CES clearly is CPT-definable.

(iii) **Representation of basic sets**. Given a set of the form $\rho\Delta$ with $\rho \in (\sigma\Gamma)_I$ and $\Delta \leq \Gamma_I$ for $I \in \mathcal{P}$, we find a representation for the extension $\mathrm{ext}(\rho\Delta)$ of $\rho\Delta$ to $\sigma\Gamma$ as follows. First we simply represent $\varphi(\rho\Delta)$ as a sequence of CESs $(\mathcal{S}_{p_1}, \ldots, \mathcal{S}_{p_s})$ over the variable sets $W[p] \cap \bigcup\{W_i^j : i \in I\}$. Since these sets are of constant size, we can clearly define such a sequence of CESs in CPT. To represent $\varphi(\mathrm{ext}(\rho\Delta))$ we just have to extend these systems by adding the cyclic constraints in $C[p]$ for the other components $W_j^i \subseteq W[p]$.

Theorem 7.6. *On every class $\mathcal{K}$ of q-bounded structures with abelian symmetries there exists a* CPT*-definable canonisation procedure.*

7.5 Discussion

Although structures with bounded abelian colours are an important class with respect to the constructions which appear in finite model theory there is still quite some room for improvement. The most obvious question is whether we can drop the restrictions to the automorphism group. A possibility to proceed

would be to consider more general automorphism groups on the colour classes, such as nilpotent or even polycyclic groups.

Open Problem 7.1. *Does* CPT *capture* PTIME *on structures of bounded colour class size?*

Looking at the technical part of our construction, another obvious way to proceed would be to consider more general linear equation systems.

Open Problem 7.2. *Is the solvability of linear equation systems over finite fields definable in* CPT*?*

Solving this question would build a bridge to another candidate for capturing polynomial time, namely Rank Logic. In its original version Rank Logic is an extension of FPC by operators to compute the rank of an interpretable linear equation system for every finite field $\mathbb{F}_p$ of prime order. Very recently, Grädel and Pakusa showed that Rank Logic fails to capture PTIME [56]. They use a generalised version of the CFI-construction, which yields CFI-like graphs where automorphism groups can be isomorphic to $(\mathbb{Z}_n)^k$ for arbitrary numbers n (instead of $\mathbb{Z}_2^k$ in the original construction). Note that in order to solve this generalised CFI-query one only needs to solve a definable CES. Hence, our result implies that this query is solvable in CPT. On the other hand there is also a stronger version of Rank Logic that has been considered in the literature, where a uniform rank operator is used. In this case the field is also given by an interpretation. So far the relationship between this logic and CPT is completely open. At first glance, it seems like these two logic capture somewhat different fragments of PTIME. Considering problems from linear algebra in CPT might help to understand their relationship better.

8 Conclusion

We want to use this last chapter to briefly sum up the possible directions for future research that emerge from the results presented in this thesis.

In the Chapter 3 we considered extensions of automatic structures as a powerful characterisation of set-interpretations. Our main question is how rich the closure under set interpretations of the class of all structures with decidable MSO-theory is. While we could make progress for ω-automatic presentations, many fundamental questions remain unsolved. One way to make further progress would be to tackle the question whether the field of reals is ω-tree-automatic.

However, we belief that we should also try to make the investigation more independent of the underlying structure. Possible ways to go could be the description of regularity by monads in category theory due to Bojańczyk [16] or to work along a hierarchy of structures with decidable MSO-theory, like the Caucal hierarchy. Towards this approach some first steps have been made by Colcombet and Löding [25].

In Chapter 4 uniformly automatic classes were introduced and in Chapter 5 applications in finite model theory were studied. While similar notions building on MSO-interpretations have mainly been investigated in the context of algorithmic meta-theorems, this concept might also give new impulses to the automatic structure community. On the other side, switching from MSO-interpretations to set-interpretation might open the automata theoretic approach to algorithmic meta-theorems to broader classes of structures.

For this purpose we need to clarify under which circumstances the existence a uniform (tree-)automatic presentation leads to fixed parameter tractability of the respective model checking problem. As we explained earlier in this thesis, the only obstacle is that we need to be able to compute from the structure a corresponding advice efficiently. Therefore the following question arises: if we have fixed uniform (tree-)automatic presentation $\mathfrak{c}$ of a class of finite structures $\mathcal{C}$, can we compute from a given $\mathfrak{A} \in \mathcal{C}$ an advice α with $\mathfrak{A} \cong \mathcal{S}(\mathfrak{C}[\alpha])$? In any case, it would be highly desirable to obtain a deeper understanding of the

structural properties of uniformly automatic classes. Additionally, we would like to see whether one can draw connections between uniform automaticity and structural graph theory. Is there a graph parameter that is in a similar fashion connected to set-interpretations as cliquewidth is connected to MSO-interpretations?

In Chapter 7 we considered the logic CPT. We have seen that this logic takes us a large step further to capturing PTIME. However, we still do not know whether this logic captures PTIME on structures of bounded colour class size. Similar, we do not know if one can define the solvability of linear equation systems in CPT.

On the other side, we do not belief that CPT actually captures polynomial time. Hence, there is also a need to develop techniques that would allow us to separate logics like CPT from PTIME. Although this task seems to be technically very challenging, recent advances like the separation of Rank Logic from PTIME [56] give hope that this question might also be solved in the not so far future.

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 E. Csuhaj-Varjú, M. Dietzfelbinger, and Z. Ésik, editors, *Mathematical Foundations of Computer Science 2014*, number 8634 in Lecture Notes in Computer Science, pages 50–62. Springer Berlin Heidelberg, Jan. 2014.

[2] F. Abu Zaid, E. Grädel, and S. Jaax. Bisimulation Safe Fixed Point Logic. In *Advances in Modal Logic 10, invited and contributed papers from the tenth conference on "Advances in Modal Logic," held in Groningen, The Netherlands, August 5-8, 2014*, pages 1–15, 2014.

[3] F. Abu Zaid, E. Grädel, and L. Kaiser. The Field of Reals is not omega-Automatic. In C. Dürr and T. Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*, volume 14 of *LIPIcs*, pages 577–588. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[4] F. Abu Zaid, E. Grädel, L. Kaiser, and W. Pakusa. Model-Theoretic Properties of omega-Automatic Structures. *Theory of Computing Systems, Special Issue dedicated to STACS 2012*, 2013. published online 17 October 2013.

[5] M. Anderson, A. Dawar, and B. Holm. Maximum Matching and Linear Programming in Fixed-Point Logic with Counting. In *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '13, pages 173–182, Washington, DC, USA, 2013. IEEE Computer Society.

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

[7] R. Baer. Abelian groups without elements of finite order. *Duke Math. J.*, 3(1):68–122, 1937.

[8] R. M. Baer and E. H. Spanier. Referenced automata and metaregular families. *Journal of Computer and System Sciences*, 3(4):423–446, Nov. 1969.

[9] V. Bárány, L. Kaiser, and A. Rabinovich. Cardinality quantifiers in mlo over trees. pages 117–131, 2009.

[10] J. R. Büchi. On a Decision Method in Restricted Second Order Arithmetic. In S. M. Lane and D. Siefkes, editors, *The Collected Works of J. Richard Büchi*, pages 425–435. Springer New York, Jan. 1990.

[11] A. Blass, Y. Gurevich, and S. Shelah. Choiceless Polynomial Time. *Ann. Pure Appl. Logic*, 100(1-3):141–187, 1999.

[12] A. Blass, Y. Gurevich, and S. Shelah. On polynomial time computation over unordered structures. *The Journal of Symbolic Logic*, 67(03):1093–1125, 2002.

[13] A. Blumensath. *Automatic Structures*. 1999. Published: Diploma thesis, RWTH-Aachen University.

[14] A. Blumensath and E. Grädel. Automatic Structures. *IN PROC. 15TH IEEE SYMP. ON LOGIC IN COMPUTER SCIENCE*, 2000:51–62, 1999.

[15] H. L. Bodlaender. A Linear Time Algorithm for Finding Tree-decompositions of Small Treewidth. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 226–234, New York, NY, USA, 1993. ACM.

[16] M. Bojańczyk. Recognisable Languages over Monads. In I. Potapov, editor, *Developments in Language Theory*, number 9168 in Lecture Notes in Computer Science, pages 1–13. Springer International Publishing, July 2015. DOI: 10.1007/978-3-319-21500-6_1.

[17] G. Boole. *An investigation of the laws of thought, on which are founded the mathematical theories of logic and probabilities*. Dover, dover (reprint) edition, 1854.

[18] S. Bova and B. Martin. First-Order Queries on Finite Abelian Groups. In S. Kreutzer, editor, *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, volume 41 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41–59, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[19] W. S. Brainerd. The Minimalization of Tree Automata. *Information and Control*, 13(5):484–491, 1968.

[20] V. Bárány. A Hierarchy of Automatic ω -Words having a Decidable MSO Theory. *RAIRO - Theoretical Informatics and Applications*, 42(3):417–450, July 2008.

[21] V. Bárány, E. Grädel, and S. Rubin. Automata-Based Presentations of Infinite Structures. In J. Esparza, C. Michaux, and C. Steinhorn, editors, *Finite and Algorithmic Model Theory*, pages 1–76. Mar. 2011.

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

[23] D. Caucal. On Infinite Terms Having a Decidable Monadic Theory. In K. Diks and W. Rytter, editors, *Mathematical Foundations of Computer Science 2002*, number 2420 in Lecture Notes in Computer Science, pages 165–176. Springer Berlin Heidelberg, Jan. 2002.

[24] C. Choffrut and S. Grigorieff. Uniformization of Rational Relations. In *Jewels Are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 59–71, London, UK, UK, 1999. Springer-Verlag.

[25] T. Colcombet and C. Löding. Transforming structures by set interpretations. *CoRR*, abs/cs/0703039, 2007.

[26] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, Mar. 1990.

[27] B. Courcelle, J. A. Makowsky, and U. Rotics. Linear Time Solvable Optimization Problems on Graphs of Bounded Clique-Width. *Theory Comput. Systems*, 33(2):125–150, Apr. 2000.

[28] A. Dawar. *Feasible Computation Through Model Theory*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 1993.

[29] A. Dawar. The Nature and Power of Fixed-point Logic with Counting. *ACM SIGLOG News*, 2(1):8–21, Jan. 2015.

[30] 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, Mar. 2008.

[31] A. De Morgan. *Formal logic : or, The calculus of inference, necessary and probable / by Augustus de Morgan*. Taylor and Walton, London, 1847.

[32] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer New York, New York, NY, 1999.

[33] A. Durand-Gasselin and P. Habermehl. Ehrenfeucht-Fraïssé goes elementarily automatic for structures of bounded degree. In C. Dürr and T. Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*, volume 14 of *LIPIcs*, pages 242–253. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[34] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical logic (2. ed.)*. Undergraduate texts in mathematics. Springer, 1994.

[35] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49(2):129–141, 1961.

[36] J. Eisinger. Upper Bounds on the Automata Size for Integer and Mixed Real and Integer Linear Arithmetic (Extended Abstract). In M. Kaminski and S. Martini, editors, *Computer Science Logic*, number 5213 in Lecture Notes in Computer Science, pages 431–445. Springer Berlin Heidelberg, Sept. 2008. DOI: 10.1007/978-3-540-87531-4_31.

[37] D. B. A. Epstein, M. S. Paterson, J. W. Cannon, D. F. Holt, S. V. Levy, and W. P. Thurston. *Word Processing in Groups.* A. K. Peters, Ltd., Natick, MA, USA, 1992.

[38] R. Fagin. Generalized first-order spectra, and polynomial. time recognizable sets. *SIAM-AMS Proceedings*, 7:43–73, 1974.

[39] J. Ferrante and C. W. Rackoff. *The Computational Complexity of Logical Theories*, volume 718 of *Lecture Notes in Mathematics.* Springer Berlin Heidelberg, 1979.

[40] O. Finkel and Stevo Todorcevic. Automatic Ordinals. *International Journal of Unconventional Computing*, 9(1-2):61–70, 2013. To appear in a Special Issue on New Worlds of Computation 2011 of the International Journal of Unconventional Computing.

[41] O. Finkel and S. Todorčević. A hierarchy of tree-automatic structures. *J. Symbolic Logic*, 77(1):350–368, Mar. 2012.

[42] J. Flum, E. Grädel, and T. Wolke, editors. *Logic and Automata: History and Perspectives.* Amsterdam University Press, 2008.

[43] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[44] R. Fraïssé. Sur l'extension aux relations de quelques propriétés des ordres. *Annales scientifiques de l'École Normale Supérieure*, 71(4):363–388, 1954.

[45] G. Frege. *Begriffsschrift, eine der Arithmetischen Nachgebildete : Formelsprache des reinen Denkens.* Verlag von Louis Nebert, 1879.

[46] G. A. Freiman. Inverse additive number theory. XI: Long arithmetic progressions in sets with small sumsets. *Acta Arith.*, 137(4):325–331, 2009.

[47] M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. In *17th Annual IEEE Symposium on Logic in Computer Science, 2002. Proceedings*, pages 215–224, 2002.

[48] J. Gajarský and P. Hlinený. Kernelizing MSO Properties of Trees of Fixed Height, and Some Consequences. *Logical Methods in Computer Science*, 11(1), 2015.

[49] J. Gajarsky and P. Hlineny. Faster Deciding MSO Properties of Trees of Fixed Height, and Some Consequences. In D. D'Souza, T. Kavitha, and J. Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*, volume 18 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 112–123, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[50] K. Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatsh. f. Mathematik und Physik*, 37(1):349–360, Dec. 1930.

[51] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatsh. f. Mathematik und Physik*, 38(1):173–198, Dec. 1931.

[52] K. I. G. Gottfried Wilhelm von Leibniz. *Die philosophischen Schriften von Gottfried Wilhelm Leibniz*. Weidmann, 1875.

[53] E. Grädel. Finite Model Theory and Descriptive Complexity. In *Finite Model Theory and Its Applications*, pages 125–230. Springer, 2007.

[54] E. Grädel, P. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and Its Applications (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[55] E. Grädel and M. Otto. Inductive Definability with Counting on Finite Structures. In *IN PROC. OF COMPUTER SCIENCE LOGIC 92*, pages 231–247. Springer-Verlag, 1993.

[56] E. Grädel and W. Pakusa. Rank Logic is Dead, Long Live Rank Logic! In S. Kreutzer, editor, *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, volume 41 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 390–404, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[57] M. Grohe. Complete problems for fixed-point logics. *Journal of Symbolic Logic*, 60(02):517–527, June 1995.

[58] M. Grohe. Logic, Graphs, and Algorithms. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(091), 2007.

[59] M. Grohe. Fixed-Point Definability and Polynomial Time on Graphs with Excluded Minors. In *2010 25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 179–188, July 2010.

[60] M. Grohe, S. Kreutzer, and S. Siebertz. Deciding First-order Properties of Nowhere Dense Graphs. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 89–98, New York, NY, USA, 2014. ACM.

[61] Y. Gurevich and S. Shelah. On finite rigid structures. *The Journal of Symbolic Logic*, 61(02):549–562, 1996.

[62] T. L. Heath and Euclid. *The Thirteen Books of Euclid's Elements, Books 1 and 2*. Dover Publications, Incorporated, 1956.

[63] G. Hjorth, B. Khoussainov, A. Montalbán, and A. Nies. From Automatic Structures to Borel Structures. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science (LICS 2008)*, pages 431–441, Pittsburgh, PA, USA, June 2008. IEEE Computer Society Press.

[64] T. W. Hungerford. *Algebra*, volume 73 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 1980.

[65] N. Immerman. Relational Queries Computable in Polynomial Time (Extended Abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 147–152, New York, NY, USA, 1982. ACM.

[66] N. Immerman. *Descriptive Complexity*. Springer New York, New York, NY, 1999.

[67] A. J.J. Jenkinson. Aristotle: Prior Analytics – Translated by A. J. Jenkinson. In *The Works of Aristotle translated into English*. Clarendon Press, Oxford, 1928. Originalarbeit: Analytica priora.

[68] L. Kaiser, S. Rubin, and V. Bárány. Cardinality and counting quantifiers on omega-automatic structures. In S. Albers and P. Weil, editors, *25th International Symposium on Theoretical Aspects of Computer Science*, volume 1 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 385–396, Dagstuhl, Germany, 2008. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[69] A. Kartzow. The field of the Reals and the Random Graph are not Finite-Word Ordinal-Automatic. *CoRR*, abs/1410.5197, 2014.

[70] A. Kartzow and P. Schlicht. Structures without Scattered-Automatic Presentation. In P. Bonizzoni, V. Brattka, and B. Löwe, editors, *The Nature of Computation. Logic, Algorithms, Applications*, number 7921 in Lecture Notes in Computer Science, pages 273–283. Springer Berlin Heidelberg, July 2013. DOI: 10.1007/978-3-642-39053-1_32.

[71] N. Kayal and T. Nezhmetdinov. Factoring Groups Efficiently. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5555 of *Lecture Notes in Computer Science*. Springer Verlag, 2009.

[72] B. Khoussainov and A. Nerode. Automatic presentations of structures. In D. Leivant, editor, *Logic and Computational Complexity*, number 960 in Lecture Notes in Computer Science, pages 367–392. Springer Berlin Heidelberg, Jan. 1995.

[73] B. Khoussainov and S. Rubin. Automatic Structures: Richness and Limitations. In *PROCEEDINGS OF THE 19TH IEEE SYMPOSIUM ON LOGIC IN COMPUTER SCIENCE*, pages 44–53. IEEE Computer Society, 2004.

[74] B. Khoussainov, S. Rubin, and F. Stephan. Definability and Regularity in Automatic Structures. In *STACS 2004*, pages 440–451. Springer, Berlin, Heidelberg, Mar. 2004. DOI: 10.1007/978-3-540-24749-4_39.

[75] F. Klaedtke. Ehrenfeucht–Fraïssé goes automatic for real addition. *Information and Computation*, 208(11):1283–1295, Nov. 2010.

[76] A. Kruckman, S. Rubin, J. Sheridan, and B. Zax. A Myhill-Nerode theorem for automata with advice. *Electronic Proceedings in Theoretical Computer Science*, 96:238–246, Oct. 2012. arXiv: 1210.2462.

[77] D. Kuske. Is Ramsey's Theorem omega-automatic? In J.-Y. Marion and T. Schwentick, editors, *27th International Symposium on Theoretical Aspects of Computer Science*, volume 5 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 537–548, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[78] D. Kuske, J. Liu, and M. Lohrey. The Isomorphism Problem On Classes of Automatic Structures. *arXiv:1001.2086 [cs]*, Jan. 2010. arXiv: 1001.2086.

[79] D. Kuske and M. Lohrey. First-order and counting theories of ω-automatic structures. *IN FOSSACS*, pages 322–336, 2006.

[80] D. Kuske and M. Lohrey. Automatic structures of bounded degree revisited. *J. Symbolic Logic*, 76(4):1352–1380, Dec. 2011.

[81] A. Langer, F. Reidl, P. Rossmanith, and S. Sikdar. Evaluation of an MSO-Solver. In D. A. Bader and P. Mutzel, editors, *ALENEX*, pages 55–63. SIAM / Omnipress, 2012.

[82] A. Nies. Describing groups. *Bulletin of Symbolic Logic*, 13(3):305–339, Sept. 2007.

[83] D. C. Oppen. A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger Arithmetic. *Journal of Computer and System Sciences*, 16(3):323–332, June 1978.

[84] M. Otto. *Bounded variable logics and counting – A study in finite models*, volume 9. Springer-Verlag, 1997. IX+183 pages.

[85] M. Otto. Bisimulation-invariant PTIME and higher-dimensional μ-calculus. *Theoretical Computer Science*, 224(1–2):237–265, Aug. 1999.

[86] D. Perrin and J.-E. Pin. *Infinite words : automata, semigroups, logic and games*. Pure and applied mathematics. Academic, London, San Diego (Calif.), 2004.

[87] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Bull. Amer. Math. Soc.*, 74(5):1025–1029, Sept. 1968.

[88] M. O. Rabin and D. Scott. Finite Automata and Their Decision Problems. *IBM J. Res. Dev.*, 3(2):114–125, Apr. 1959.

[89] F. P. Ramsey. On a Problem of Formal Logic. *Proceedings of the London Mathematical Society*, s2-30(1):264–286, Jan. 1930.

[90] F. Reidl, P. Rossmanith, F. S. Villaamil, and S. Sikdar. A Faster Parameterized Algorithm for Treedepth. In J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias, editors, *Automata, Languages, and Programming*, number 8572 in Lecture Notes in Computer Science, pages 931–942. Springer Berlin Heidelberg, July 2014. DOI: 10.1007/978-3-662-43948-7_77.

[91] F. Reinhardt. *Automatic Structures with Parameters*. 2013. Published: Diploma thesis, RWTH-Aachen University.

[92] B. Rossman. Choiceless Computation and Symmetry. In A. Blass, N. Dershowitz, and W. Reisig, editors, *Fields of Logic and Computation: Essays Dedicated to Yuri Gurevich on the Occasion of His 70th Birthday*, pages 565–580. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[93] S. Rubin. Automata presenting structures: A survey of the finite string case. *Bull. Symbolic Logic*, 14(2):169–209, June 2008.

[94] S. Schalthöfer. *Computing on Abstract Structures with Logical Interpretations*. 2013. Published: Master thesis, RWTH Aachen University.

[95] P. Schlicht and F. Stephan. Automata on Ordinals and Linear Orders. In B. Löwe, D. Normann, I. Soskov, and A. Soskova, editors, *Models of Computation in Context*, number 6735 in Lecture Notes in Computer Science, pages 252–259. Springer Berlin Heidelberg, June 2011. DOI: 10.1007/978-3-642-21875-0_27.

[96] D. Seese. Linear Time Computable Problems and First-Order Descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.

[97] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*, pages 24–84. Springer Vienna, Vienna, 1998.

[98] T. Tsankov. The additive group of the rationals does not have an automatic presentation. *Journal of Symbolic Logic*, 76(4):1341–1351, Dec. 2011.

[99] M. Y. Vardi. The Complexity of Relational Query Languages (Extended Abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 137–146, New York, NY, USA, 1982. ACM.

[100] A. N. Whitehead and B. A. W. Russell. *Principia mathematica; 2nd ed.* Cambridge Univ. Press, Cambridge, 1927.

[101] T. Wilke. An algebraic theory for regular languages of finite and infinite words. *International Journal of Algebra and Computation*, 03(04):447–489, Dec. 1993.