Rheinisch-Westfälische Technische Hochschule Aachen
Lehr- und Forschungsgebiet Mathematische Grundlagen der Informatik

Diploma Thesis

# Automatic Structures

Achim Blumensath

October 1999

Hiermit versichere ich, daß ich die Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, den 25. Oktober 1999

# Contents


**1 Introduction** **1**

**2 Formal Languages and Logic** **3**
- 2.1 Formal Languages . . . . . . 3
- 2.2 Logic . . . . . . 6

**3 Automatic Presentations and Queries** **9**
- 3.1 Automatic Presentations . . . . . . 9
- 3.2 First-Order Queries . . . . . . 12
- 3.3 Extensions of First-Order Logic . . . . . . 16
- 3.4 Complexity of Queries . . . . . . 18

**4 Complete Structures** **29**
- 4.1 Word Languages . . . . . . 29
- 4.2 $\omega$-Languages . . . . . . 34
- 4.3 Tree Languages . . . . . . 37
- 4.4 $\omega$-Tree Languages . . . . . . 40

**5 Classes of Automatic Structures** **43**
- 5.1 Growth Rates and Length Sequences . . . . . . 43
- 5.2 Applications and Examples . . . . . . 48
- 5.3 Composition of Structures . . . . . . 52
- 5.4 The Class Hierarchy . . . . . . 57

**6 Model Theory** **61**
- 6.1 Compactness . . . . . . 61
- 6.2 Axiomatisation of $\mathrm{Th}(\mathfrak{N}_p)$ . . . . . . 62
- 6.3 Non-Standard Models . . . . . . 66

**7 Unary Presentations** **69**
- 7.1 Complete Structure . . . . . . 69
- 7.2 Structures with Unary Presentation . . . . . . 72
- 7.3 Complexity . . . . . . 81
- 7.4 Decidability . . . . . . 83

**8 Other Types of Presentations** **87**
- 8.1 Weak Presentations . . . . . . 87
- 8.2 Star-free and Locally Threshold Testable Presentations . . . . . . 91

**9 Conclusion** **95**

**Bibliography** **97**

**Index** **99**

# Chapter 1

# Introduction

Starting with the famous characterisation of NPTIME by Fagin in 1974, finite model theory has grown into a field of its own with many applications to computer science, especially in complexity theory where it turned out that there is a close correspondence between complexity classes and certain logics. But also the investigation of query languages in database theory and the design of model-checking algorithms for automatic verification was strongly influenced by finite model theory.

In recent years the need of a theory covering not only finite but also infinite structures became apparent in those fields. For instance, currently model-checking of systems with infinite state space can be performed only in some very restricted cases which do not cover most real-world problems. Another example are geometrical databases where—for operations like intersection—it is more convenient to treat geometric shapes as (infinite) sets of points instead of using parametrised basic shapes.

Of course, only restricted classes of structures are meaningful for such an approach. In order to be able to process an infinite structure by algorithmic means it must possess a finite encoding, and the operations being performed must be recursive.

In this thesis we will investigate several classes of possibly infinite structures meeting those requirements. The general idea is to use finite automata to present a given structure. Each element of the structure is encoded by one or several words. The language of all valid encodings is required to be regular, and for each relation, including equality, an automaton is constructed which accepts a tuple of words iff the corresponding tuple of elements is in the relation. Instead of normal finite automata one can also use automata over $\omega$-words, trees, etc., leading to several different classes of automatic structures. In each case, such structures can be encoded by a list of automata and processed using well-known automata constructions—which, in particular, include boolean operations and projection so that we are able to evaluate first-order formulae.

These concepts were introduced by Cannon and Thurston [ECH$^+$92] in group theory—where they, e.g., solved word problems using automata—, and subsequently generalised to arbitrary structures by Khoussainov and Nerode [KN95]. This thesis will extend the results of the later focusing on model theoretic issues. Automatic groups will hardly be mentioned.

One fundamental result is that each of the investigated classes contains a complete structure, i.e., a structure $\mathfrak{C}$ such that any structure $\mathfrak{A}$ is a member of the given class if and only if there is an interpretation of $\mathfrak{A}$ in $\mathfrak{C}$.

The outline of this thesis is as follows. We start in Chapter 3 with the definition of automatic presentations using languages of, respectively, finite words, $\omega$-words, trees, and $\omega$-trees. We prove some of their basic properties such as closure under first-order interpretations, and study decidability and complexity of queries on automatic structures. We show that first-order queries are effectively computable and that their results are again automatic, while slightly stronger logics already become undecidable, and we present some restricted cases in which the complexity is acceptable.

The fundamental characterisation of automatic structures in terms of first-order interpretations which makes many methods from logic available to us is given in the following chapter. For each class we present a structure $\mathfrak{C}$ such that some structure $\mathfrak{A}$ belongs to the class if and only if there is a first-order interpretation of $\mathfrak{A}$ in $\mathfrak{C}$.

In Chapter 5 we take a closer look at the classes of structures defined so far, determine their hierarchy, and investigate the closure under Feferman-Vaught like products. In order to prove that some structure is not automatic we develop methods based on the calculation of bounds on the length of the encoding of elements.

Chapter 6 is devoted to purely logical questions. It is shown that the Compactness Theorem fails if the class of models is restricted to automatic ones, and an axiomatisation is given for the structure $(\mathbb{N}, +, |_p)$ which plays an important role for the characterisation of automatic structures. We also construct a non-standard model of this axiom system.

In the final chapters we consider restricted types of presentations. Chapter 7 deals with the case of presentations over a unary alphabet which yields an interesting subclass of automatic structures with many pleasant theoretical properties and complexity results which are low enough for practical applications.

The last chapter investigates another way to encode the input which turns out to yield a much weaker class, and the restriction to star-free and locally threshold testable languages.

I would like to thank Erich Grädel for his guidance while I wrote this thesis, and Eric Rosen for his valuable comments.

# Chapter 2

# Formal Languages and Logic

## 2.1 Formal Languages

**Regular languages.** We assume that the reader is familiar with the fundamental notions of formal language theory. For an introduction see [HU79, Eil74, RS97], readers with a background in logic are referred to [EF95, Chapter 5]. An overview of $\omega$-languages is given in [Tho90]. We use the following conventions regarding automata. A *finite automaton* is a tuple $\mathfrak{A} = (Q, \Sigma, \Delta, q_0, F)$ with set of states $Q$, input alphabet $\Sigma$, initial state $q_0$, set of final states $F$, and transition relation $\Delta \subseteq Q \times \Sigma \times Q$. A *finite $\omega$-automaton* is a tuple $\mathfrak{A} = (Q, \Sigma, \Delta, q_0, \mathscr{F})$ with set of states $Q$, input alphabet $\Sigma$, initial state $q_0$, transition relation $\Delta \subseteq Q \times \Sigma \times Q$, and Muller acceptance condition $\mathscr{F} \subseteq \mathscr{P}(Q)$, where some $\omega$-word is accepted iff the set of states appearing infinitely often in some run is a member of $\mathscr{F}$. We call an automaton *deterministic* iff for every $q \in Q$ and $a \in \Sigma$ there is at most one $q' \in Q$ such that $(q, a, q') \in \Delta$.

For $L, W \subseteq \Sigma^*$ we denote the *left-* and *right-quotient* by

$$W^{-1}L := \{\, x \mid \exists y \in W : yx \in L \,\},$$
$$LW^{-1} := \{\, x \mid \exists y \in W : xy \in L \,\}.$$

**Definition 2.1.** Let $L \subseteq \Sigma^*$. The *Nerode-congruence* $\sim_L$ is defined by

$$x \sim_L y \;:\text{iff}\; x^{-1}L = y^{-1}L.$$

Clearly, $\sim_L$ is a right-congruence, i.e., $x \sim_L y \Longrightarrow xz \sim_L yz$. By the Myhill-Nerode Theorem, $\sim_L$ is of finite index if and only if $L$ is regular. In this case the index is equal to the number of states of the minimal deterministic automaton for $L$.

Recall that the class of regular languages is closed under

(i) boolean operations: union, intersection, and complement,

(ii) concatenation and star,

(iii) homomorphisms and inverse homomorphisms, and

(iv) left- and right-quotients.

An important tool to show non-regularity which will frequently be used in the following is the

**Pumping Lemma.** *Let $L \subseteq \Sigma^*$ be regular. There exists a constant $m$ such that for all words $uvw \in \Sigma^*$ with $|v| \geq m$ there exists a factorisation $v_0v_1v_2$ of $v$ with $v_1 \neq \varepsilon$ such that*

$$uvw \in L \text{ iff } uv_0v_1^kv_2w \in L \textit{ for all } k \in \mathbb{N}.$$

When investigating $\omega$-languages one frequently uses topological techniques. $\Sigma^\omega$ is equipped with the product topology where $\Sigma$ is taken as discrete space. In this topology open sets are of the form $W\Sigma^\omega$ for some $W \subseteq \Sigma^*$. All regular $\omega$-languages are contained in $\mathcal{B}(G_\delta)$, the boolean closure of the second level of the Borel hierarchy, i.e., every regular language can be written as a boolean combination of countable intersections $\bigcap_i W_i\Sigma^\omega$ with $W_0, W_1, \ldots \subseteq \Sigma^*$.

**Definition 2.2.** Let $\Sigma$ be a finite alphabet and fix a linear ordering $<$ of $\Sigma$. The *lexicographic ordering* $\prec_\mathrm{l}$ and the *alphabetic ordering* $\prec_\mathrm{a}$ induced by $<$ are defined as

$$\begin{aligned} x \prec_\mathrm{l} y \ :\text{iff}\ \ & y = xy', \text{ or } x = zax' \text{ and } y = zby' \text{ for some} \\ & z, x', y' \in \Sigma^*, \text{ and } a, b \in \Sigma \text{ with } a < b, \end{aligned}$$

and

$$x \prec_\mathrm{a} y \ :\text{iff}\ \ |x| < |y| \text{ or } |x| = |y| \text{ and } x \prec_\mathrm{l} y.$$

**Convolution.** The operation of convolution plays a central role in the following. Ordinary finite automata take single words as their input. When representing relations of arity greater than one by automata one needs a model with several inputs. In order to avoid having to define a new type of automaton we introduce an operation which encodes several words into one word in such a way that the automaton reading the new word has access to the original ones.

**Definition 2.3.** Let $\Sigma$ be a finite alphabet with $\Box \notin \Sigma$. The *convolution* of $x_0, \ldots, x_{n-1} \in \Sigma^*$ with $x_i = x_{i0} \cdots x_{il_i}$ is defined as

$$x_0 \otimes \cdots \otimes x_{n-1} := \begin{bmatrix} x'_{00} \\ \vdots \\ x'_{(n-1)0} \end{bmatrix} \cdots \begin{bmatrix} x'_{0l} \\ \vdots \\ x'_{(n-1)l} \end{bmatrix} \in (\Sigma \cup \{\Box\})^n$$

where

$$x'_{ij} := \begin{cases} x_{ij} & \text{if } j \leq l_i, \\ \Box & \text{otherwise,} \end{cases} \qquad l := \max\{l_1, \ldots, l_n\}.$$

For $L$, $L' \subseteq \Sigma^*$ we define

$$\begin{aligned} L \otimes L' &:= \{\, x \otimes y \mid x \in L, y \in L' \,\}, \\ L^{\otimes n} &:= L \otimes \cdots \otimes L \qquad (n \text{ times}). \end{aligned}$$

*Remark.* Regular languages are closed under convolution.

For notational convenience we introduce the following functions to translate between product and convolution. Let $R \subseteq (\Sigma^*)^n$ and $L \subseteq (\Sigma^*)^{\otimes n}$.

$$\begin{aligned} \mathrm{fold}(R) &:= \{\, x_0 \otimes \cdots \otimes x_{n-1} \mid (x_0, \ldots, x_{n-1}) \in R \,\}, \\ \mathrm{unfold}(L) &:= \{\, (x_0, \ldots, x_{n-1}) \mid x_0 \otimes \cdots \otimes x_{n-1} \in L \,\}. \end{aligned}$$

**Trees.** We recall some basic definitions regarding tree languages (see [GS97], [Tho90]).

**Definition 2.4.** Let $\Sigma$ be a finite alphabet. A *finite binary tree* over $\Sigma$ is a mapping $t : \mathrm{dom}(t) \to \Sigma$ where $\mathrm{dom}(t) \subseteq \{0,1\}^*$ is finite and satisfies the following closure condition: $wi \in \mathrm{dom}(t)$ for some $w \in \{0,1\}^*$ and $i \in \{0,1\}$ implies $w \in \mathrm{dom}(t)$ and $wj \in \mathrm{dom}(t)$ for all $j < i$.

A *binary $\omega$-tree* over $\Sigma$ is a mapping $t : \mathrm{dom}(t) \to \Sigma$ with $\mathrm{dom}(t) = \{0,1\}^*$. The set of all finite trees is denoted by $T_\Sigma$, the set of all $\omega$-trees by $T^\omega_\Sigma$.

To avoid cumbersome definitions we use the following notation in this section. Let $t \in T_\Sigma$. By $t_a$ we denote the $\omega$-tree defined as

$$t_a(x) := \begin{cases} t(x) & \text{if } x \in \mathrm{dom}(t), \\ a & \text{otherwise.} \end{cases}$$

The notion of convolution readily generalises to trees.

**Definition 2.5.** The *convolution* of finite or infinite trees $t_0, \dots, t_{n-1}$ over $\Sigma$ is defined as

$$(t_0 \otimes \cdots \otimes t_{n-1})(x) := ((t_0)_\Box(x), \dots, (t_{n-1})_\Box(x)) \in T_{(\Sigma\cup\{\Box\})^n}$$

where $\mathrm{dom}(t_0 \otimes \cdots \otimes t_{n-1}) := \mathrm{dom}(t_0) \cup \cdots \cup \mathrm{dom}(t_{n-1})$.

A (*bottom-up*) *tree automaton* is a tuple $\mathfrak{A} = (Q, \Sigma, \Delta, F)$ with set of states $Q$, input alphabet $\Sigma$, set of final states $F$, and transition relation

$$\Delta \subseteq Q \times \Sigma \times (Q \cup \{\Box\}) \times (Q \cup \{\Box\}).$$

A *run* of $\mathfrak{A}$ on some input tree $t \in T_\Sigma$ is a tree $\varrho \in T_Q$ satisfying the following conditions:

(i) $\mathrm{dom}(t) = \mathrm{dom}(\varrho)$,

(ii) $\varrho(\varepsilon) \in F$, and

(iii) $(\varrho(x), t(x), \varrho_\Box(x0), \varrho_\Box(x1)) \in \Delta$ for all $x \in \mathrm{dom}(t)$.

A (*top-down*) *$\omega$-tree automaton* is a tuple $\mathfrak{A} = (Q, \Sigma, \Delta, Q_0, \mathscr{F})$ with set of states $Q$, input alphabet $\Sigma$, set of initial states $Q_0$, Muller acceptance condition $\mathscr{F}$, and transition relation $\Delta \subseteq Q \times \Sigma \times Q \times Q$. A *run* of $\mathfrak{A}$ on some input tree $t \in T^\omega_\Sigma$ is a tree $\varrho \in T^\omega_Q$ satisfying the following conditions:

(i) $\varrho(\varepsilon) \in Q_0$,

(ii) each path through $\varrho$ satisfies the Muller-condition $\mathscr{F}$, and

(iii) $(\varrho(x), t(x), \varrho(x0), \varrho(x1)) \in \Delta$ for all $x \in \mathrm{dom}(t)$.

The tree language $T(\mathfrak{A})$ recognised by some ($\omega$-)tree automaton $\mathfrak{A}$ is the set of trees, respectively $\omega$-trees $t$ for which there is a run of $\mathfrak{A}$ on $t$.

## 2.2 Logic

For an introduction to mathematical logic, see for example [EFT94]. We recall some basic notions.

A *signature* $\tau$ is a set of relation and function symbols each of which is equipped with an arity. Constants are regarded as functions of arity 0. FO$[\tau]$ is the set of all first-order formulae using only relation and functions symbols from $\tau$ (and equality). A *$\tau$-structure* $\mathfrak{A} = (A, R_0^{\mathfrak{A}}, \dots, f_0^{\mathfrak{A}}, \dots)$ consists of a set $A$, called the universe of $\mathfrak{A}$, and of one relation $R^{\mathfrak{A}}$ for each relation symbol $R$ in $\tau$ and one function $f^{\mathfrak{A}}$ for each function symbol $f$ in $\tau$. For $\varphi(\overline{x}) \in$ FO we define

$$\varphi^{\mathfrak{A}} := \{\, \overline{a} \in A^r \mid \mathfrak{A} \models \varphi(\overline{a}) \,\}.$$

First-order formulae are classified according to their quantifier-prefix. The class $\Sigma_k$ contains all formulae whose prenex normal form has $k$ alternations between existential and universal quantifiers and starts with an existential quantifier. Similarly, the prenex normal form of an $\Pi_k$-formula begins with a universal quantifier, and $\Delta_k$ denotes the class $\Sigma_k \cap \Pi_k$.

Besides FO$[\tau]$ we consider several other logics in the following (see [EF95]). MSO and SO are monadic second-order and second-order logic which permit quantification over sets and relations of arbitrary arity, respectively. FO$(\exists^\omega)$ extends FO by the quantifier "there are infinitely many," whereas FO(DTC) introduces the deterministic transitive closure operator DTC.

Let $\mathfrak{A}$ be a structure and $\beta$ an assignment, i.e., a mapping of variables to elements of $\mathfrak{A}$. We define for $\varphi \in$ FO(DTC)

$$(\mathfrak{A}, \beta) \models [\mathrm{DTC}_{\overline{x},\overline{y}}\ \varphi(\overline{x}, \overline{y}, \overline{z})](\overline{a}, \overline{b})$$

iff there are $\overline{c}_0, \dots, \overline{c}_n$ with $n \geq 1$ such that $\overline{c}_0 = \overline{a}$, $\overline{c}_n = \overline{b}$ and, for all $i < n$, $\overline{c}_{i+1}$ is the *unique* tuple with

$$\big(\mathfrak{A}, \beta[\overline{x}/\overline{c}_i, \overline{y}/\overline{c}_{i+1}]\big) \models \varphi.$$

Finally, FO(#) is the extension of first-order logic by variables of a second sort ranging over cardinal numbers up the the cardinality of the universe and the cardinality operator # which is defined as

$$\big(\#_x \varphi(x)\big)^{(\mathfrak{A},\beta)} := \big|\{\, a \in A \;\big|\; (\mathfrak{A}, \beta[x/a]) \models \varphi \,\}\big|.$$

**Definition 2.6.** Let $\mathfrak{L}$ be a logic, $\sigma = \{R_0, \dots, R_r\}$ a relational signature where the arity of $R_j$ is $r_j$, $\mathfrak{A}$ a $\sigma$-structure, and $\mathfrak{B}$ a $\tau$-structure. A *$k$-dimensional $\mathfrak{L}$-interpretation* of $\mathfrak{A}$ in $\mathfrak{B}$ is a tuple

$$\mathcal{I} = \big(h, \delta(\overline{x}), \varepsilon(\overline{x}, \overline{y}), \varphi_{R_0}(\overline{x}_0, \dots, \overline{x}_{r_0-1}), \dots, \varphi_{R_r}(\overline{x}_0, \dots, \overline{x}_{r_r-1})\big)$$

satisfying the following conditions:

(i) $\delta$, $\varepsilon$, $\varphi_{R_0}, \dots, \varphi_{R_r} \in \mathfrak{L}$ and each each tuple $\overline{x}$ consists of $k$ variables,

(ii) $h : \delta^{\mathfrak{B}} \to \mathfrak{A}$ is surjective,

(iii) $\mathfrak{B} \models \varepsilon(\overline{b}_0, \overline{b}_1)$ iff $h(\overline{b}_0) = h(\overline{b}_1)$ for all $\overline{b}_0$, $\overline{b}_1$ in $\delta^{\mathfrak{B}}$, and

(iv) $\mathfrak{B} \models \varphi_{R_j}(\overline{b}_0, \dots, \overline{b}_{r_j-1})$ iff $(h(\overline{b}_0), \dots, h(\overline{b}_{r_j-1})) \in R_j^{\mathfrak{A}}$ for all $\overline{b}_0, \dots, \overline{b}_{r_j-1}$ in $\delta^{\mathfrak{B}}$.

Thus, an interpretation $\mathcal{I}$ of $\mathfrak{A}$ in $\mathfrak{B}$ defines an isomorphic copy of $\mathfrak{A}$ in $\mathfrak{B}$. If there is some $\mathfrak{L}$-interpretation of $\mathfrak{A}$ in $\mathfrak{B}$ we write $\mathfrak{A} \leq_{\mathfrak{L}} \mathfrak{B}$. If both $\mathfrak{A} \leq_{\mathfrak{L}} \mathfrak{B}$ and $\mathfrak{B} \leq_{\mathfrak{L}} \mathfrak{A}$ we say $\mathfrak{A}$ and $\mathfrak{B}$ are *mutually interpretable* and write $\mathfrak{A} \rightleftharpoons_{\mathfrak{L}} \mathfrak{B}$.

*Example.* A standard example is the interpretation of the rationals $(\mathbb{Q}, +, \cdot)$ in the integers $(\mathbb{Z}, +, \cdot)$. Fractions $p/q$ are represented by the pair $(p, q)$. All pairs with non-zero second component encode a rational number. Therefore the universe is defined by

$$\delta(x_0, x_1) := x_1 \neq 0.$$

Two pairs $(p, q)$ and $(p', q')$ are equal if $p/q = p'/q'$. Thus we set

$$\varepsilon(x_0, x_1, y_0, y_1) := x_0 \cdot y_1 = y_0 \cdot x_1.$$

Addition and multiplication can be defined the usual way.

$$\varphi_+(\overline{x}, \overline{y}, \overline{z}) := \varepsilon(z_0, z_1,\ x_0 \cdot y_1 + y_0 \cdot x_1, x_1 \cdot y_1),$$
$$\varphi_\cdot(\overline{x}, \overline{y}, \overline{z}) := \varepsilon(z_0, z_1,\ x_0 \cdot y_0, x_1 \cdot y_1).$$

A stronger notion than interpretability is given by the definition of a *reduct*. $\mathfrak{A}$ is an $\mathfrak{L}$-reduct of $\mathfrak{B}$ if both have the same universe and each relation of $\mathfrak{A}$ is $\mathfrak{L}$-definable in $\mathfrak{B}$. $\mathfrak{A}$ and $\mathfrak{B}$ are *definitional* $\mathfrak{L}$*-equivalent*, $\mathfrak{A} =_{\mathfrak{L}} \mathfrak{B}$, if both, $\mathfrak{A}$ is an $\mathfrak{L}$-reduct of $\mathfrak{B}$ and vice versa.

The following result shows that when dealing with infinite structures one easily crosses the boundary to undecidability.

**Proposition 2.7.** *The* FO(DTC)*-theory of* $(\mathbb{N}, s)$ *is undecidable where* $s$ *is the successor function.*

*Proof.* We show how to define addition and multiplication in $(\mathbb{N}, s)$. Hence, using FO(DTC)-formulae it is possible to interpret Arithmetic in $(\mathbb{N}, s)$ whose theory is undecidable.

$$z = x + y \ :\text{iff}\ [\mathrm{DTC}_{uv,u'v'}\ u' = su \wedge v' = sv](0y, xz)$$
$$z = x \cdot y \ :\text{iff}\ [\mathrm{DTC}_{uv,u'v'}\ u' = su \wedge v' = v + x](00, yz)$$

□

In particular, in any class of structures containing $(\mathbb{N}, s)$ there are structures with undecidable FO(DTC)-theory. Thus, if one is interested in logics with recursion, i.e., transitive closure or fixed point logics, one should look at classes with very simple infinite structures or structures with dense orderings. All but one of the classes we consider in the following contain $(\mathbb{N}, s)$.

# Chapter 3

# Automatic Presentations and Queries

## 3.1 Automatic Presentations

The idea of representing possibly infinite structures by finite automata can be made precise as follows. We encode the elements of the structure by words over some alphabet. In order to determine whether a tuple $(a_0, \dots, a_{n-1})$ belongs to some relation $R$ we take the tuple $(w_0, \dots, w_{n-1})$ of words encoding $(a_0, \dots, a_{n-1})$ and test whether the convolution $w_0 \otimes \dots \otimes w_{n-1}$ is accepted by the automaton representing $R$.

**Definition 3.1.** Let $\tau = \{R_0, \dots, R_r\}$ be a finite relational signature, $r_j$ the arity of $R_j$, and let $\mathfrak{A} = (A, R_0^{\mathfrak{A}}, \dots, R_r^{\mathfrak{A}})$ be a $\tau$-structure.

$$\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$$

is an *automatic presentation* of $\mathfrak{A}$ if the following conditions are satisfied:

(i) $L_\delta \subseteq \Sigma^*$, $L_\varepsilon \subseteq (\Sigma^*)^{\otimes 2}$, and $L_{R_j} \subseteq (\Sigma^*)^{\otimes r_j}$ for $j \leq r$, are regular languages.

(ii) $\nu : L_\delta \to A$ is surjective and

$$\begin{aligned} x_0 \otimes x_1 \in L_\varepsilon \quad & \text{iff} \quad \nu(x_0) = \nu(x_1), \\ x_0 \otimes \dots \otimes x_{r_j - 1} \in L_{R_j} \quad & \text{iff} \quad \big(\nu(x_0), \dots, \nu(x_{r_j-1})\big) \in R_j^{\mathfrak{A}} \end{aligned}$$

for all $j \leq r$.

Note the similarity between the definitions of an automatic presentation and an interpretation. We will see in Chapter 4 that basically an automatic presentation is an interpretation in a fixed structure.

If regular languages of $\omega$-words, trees, or $\omega$-trees are used instead of word languages we speak of $\omega$-automatic, tree-automatic, and $\omega$-tree-automatic presentations, respectively. The classes of $\tau$-structures possessing a presentation of one of the above defined types is denoted by AutStr[$\tau$], $\omega$-AutStr[$\tau$], TAutStr[$\tau$], and $\omega$-TAutStr[$\tau$], respectively. Furthermore we use abbreviations like [T]AutStr[$\tau$] meaning AutStr[$\tau$] or TAutStr[$\tau$].

If the signature $\tau$ contains functions, an automatic presentation of some $\tau$-structure $\mathfrak{A}$ is a presentation of its relational variant where each function is replaced by its graph.

*Example.* (1) An important example of a structure with an automatic presentation is Presburger Arithmetic $(\mathbb{N}, +)$. Each number $n \in \mathbb{N}$ is encoded the standard way as binary number without leading zeros, but in reversed order, i.e., with the least significant digit first. A presentation is $\mathfrak{d} = (\nu, \{0,1\}, L_\delta, L_\varepsilon, L_+)$ with

$$\nu(b_0 \cdots b_l) := \sum_{i \le l} b_i 2^i, \qquad L_\varepsilon := \{[\begin{smallmatrix}0\\0\end{smallmatrix}], [\begin{smallmatrix}1\\1\end{smallmatrix}]\}^*,$$
$$L_\delta := \{0,1\}^*1 \cup \{0\}, \qquad L_+ := L(\mathfrak{A}_+).$$

$\mathfrak{A}_+$ is an automaton which compares its input digit by digit and remembers the carry at every step. Formally, $\mathfrak{A} := \big(\{0,1\}, \{0,1,\square\}^3, \Delta, 0, \{0\}\big)$ with

$$\Delta := \{\, (i, (a,b,c), j) \mid a + b + i = 2j + c \text{ (counting } \square \text{ as } 0) \,\}.$$

(2) Natural candidates for structures with automatic presentation are those consisting of words. (But note that the free monoid—with at least two generators—does *not* have such a presentation as we will see in Section 5.1.) Let $\Sigma$ be some alphabet and consider the structure $(\Sigma^*, (D_a)_{a\in\Sigma}, \le)$ where

$$D_a xy \;:\text{iff}\;\; x = uav \text{ for some } u, v \in \Sigma^* \text{ with } |u| = |y|\,,$$
$$x \le y \;:\text{iff}\;\; |x| \le |y|\,.$$

It can be presented as $\mathfrak{d} = (\mathrm{id}, \Sigma, \Sigma^*, L_\varepsilon, (L_a)_{a\in\Sigma}, L_\le)$ with

$$L_\varepsilon := \{\, [\begin{smallmatrix}a\\a\end{smallmatrix}] \mid a \in \Sigma \,\}^*,$$
$$L_a := \{\, [\begin{smallmatrix}b\\c\end{smallmatrix}] \mid b, c \in \Sigma \,\}^* \, [\begin{smallmatrix}a\\\square\end{smallmatrix}] \, \{\, [\begin{smallmatrix}b\\\square\end{smallmatrix}] \mid b \in \Sigma \,\}^*,$$
$$L_\le := \{\, [\begin{smallmatrix}a\\b\end{smallmatrix}] \mid a, b \in \Sigma \,\}^* \{\, [\begin{smallmatrix}\square\\b\end{smallmatrix}] \mid b \in \Sigma \,\}^*.$$

**Definition 3.2.** Let $\mathfrak{A} \in \mathrm{AutStr}$ be a structure with automatic presentation $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$. Denote by $\lambda^{\mathfrak{d}} : A \to \mathbb{N}$ the function mapping each element of $\mathfrak{A}$ to the length of its shortest encoding.

$$\lambda^{\mathfrak{d}}(a) := \min\{\, |x| \mid \nu(x) = a \,\}$$

Let us start with some basic observations about automatic presentations. First, a binary alphabet is always sufficient.

**Lemma 3.3.** *Let $\mathfrak{A} \in [\omega\text{-}][\mathrm{T}]\mathrm{AutStr}$. Then $\mathfrak{A}$ has a presentation over a binary alphabet.*

*Proof.* Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ be a presentation of $\mathfrak{A}$. If $|\Sigma| = 1$ we can simply add some symbol to $\Sigma$. Otherwise, let $\Sigma = \{a_0, \dots, a_{n-1}\}$. Consider the family of homomorphisms $h_m : (\Sigma^m)^* \to (\{0,1\}^m)^*$ defined by

$$h_m(a_{i_0}, \dots, a_{i_{m-1}}) := (\mathrm{bin}(i_0), \dots, \mathrm{bin}(i_{m-1}))$$

where $\mathrm{bin}(i)$ is the binary encoding of $i$ of fixed length $\lceil \log_2 n \rceil$. As all words $\mathrm{bin}(i_k)$, $k < m$, in the definition above have the same length

$$\mathfrak{d}' := (\nu \circ h^{-1}, \{0,1\}, h_1(L_\delta), h_2(L_\varepsilon), h_{r_0}(L_{R_0}), \dots, h_{r_r}(L_{R_r}))$$

is a presentation of $\mathfrak{A}$ of the required form. $\square$

The next result turns out to be vital in many circumstances—especially when applying the Pumping Lemma as it guarantees that all pumped words encode different elements. The case of AutStr is due to Khoussainov and Nerode [KN95].

**Theorem 3.4.** *Every* $\mathfrak{A} \in$ [T]AutStr *has an injective automatic presentation.*

*Proof.* (AutStr) Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ be a presentation of $\mathfrak{A} \in$ AutStr. Fix an ordering of $\Sigma$ and consider the alphabetical ordering $\preceq$ of $\Sigma^*$ induced by it. This ordering is recognisable by an automaton. In order to define an injective presentation we pick from each set $\nu^{-1}(a)$ the least word with respect to $\preceq$ and obtain the injective presentation

$$\mathfrak{d}' = (\nu, \Sigma, L'_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$$

where the language

$$L'_\delta := \{\, x \in L_\delta \mid \forall y \in L_\delta : x \otimes y \in L_\varepsilon \to x \preceq y \,\}$$

is regular.

(TAutStr) We have to define a well-ordering on the set of finite trees which is recognisable by an automaton. Then the rest of the proof is identical to the case above. Thus we set $t_0 \prec t_1$ if either

(i) $\mathrm{dom}(t_0) \neq \mathrm{dom}(t_1)$ and the leftmost position in the symmetric difference of $\mathrm{dom}(t_0)$ and $\mathrm{dom}(t_1)$ belongs to $\mathrm{dom}(t_1)$ or

(ii) $\mathrm{dom}(t_0) = \mathrm{dom}(t_1)$ and at the leftmost position $x$ where $t_0$ and $t_1$ differ we have $t_0(x) < t_1(x)$.

This relation can be recognised by an automaton as follows. It guesses which case applies and the position of the difference, and checks that to the left of this position both trees are identical. □

In the case of $\omega$-AutStr all we can do at the moment is to classify the sets of $\omega$-words encoding the same element.

**Lemma 3.5.** *Let* $\mathfrak{d}$ *be an* $\omega$*-automatic presentation of* $\mathfrak{A}$ *and let* $a \in A$*. The set of all* $\omega$*-words encoding* $a$ *belongs to* $\mathcal{B}(G_\delta)$*, the boolean closure of the second level of the Borel hierarchy.*

*Proof.* Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$. Take any $\omega$-word $w$ encoding $a$. The function $\iota_w : \Sigma^\omega \to (\Sigma^\omega)^{\otimes 2} = (\Sigma \times \Sigma)^\omega$ defined by $\iota_w(x) := x \otimes w$ is continuous. As every regular $\omega$-language is in $\mathcal{B}(G_\delta)$, and since the inverse of a continuous function leaves levels of the Borel hierarchy invariant, we obtain $\nu^{-1}(a) = \iota_w^{-1}(L_\varepsilon) \in \mathcal{B}(G_\delta)$. □

We end this section with some simple remarks about how to construct automatic structures from other ones.

**Lemma 3.6.** *Every automatic presentation of a structure* $\mathfrak{A} \in$ [T]AutStr *can effectively be extended to a presentation of* $(\mathfrak{A}, \leq) \in$ [T]AutStr *for some well-ordering* $\leq$*.*

*Proof.* (AutStr) Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ be an injective presentation of $\mathfrak{A}$. Define

$$a \leq b \;:\text{iff}\; \nu^{-1}(a) \preceq \nu^{-1}(b)$$

where $\preceq$ is some fixed alphabetical ordering of $\Sigma^*$.

(TAutStr) Take the well-ordering defined in the proof of Theorem 3.4. □

**Lemma 3.7.** (i) *If* $\mathfrak{A} \in$ [T]AutStr *then* $(\mathfrak{A}, \overline{a}) \in$ [T]AutStr *for any tuple* $\overline{a}$ *of finitely many elements of* $\mathfrak{A}$.

(ii) *Let* $\mathfrak{A} \in \omega$-AutStr *with presentation* $\mathfrak{d}$. *If there is some ultimately periodic* $\omega$*-word encoding* $a \in A$ *then* $(\mathfrak{A}, a) \in \omega$-AutStr.

*Proof.* (i) Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ be an injective presentation of $\mathfrak{A}$. For each $a \in A$ one can construct an automaton which recognises the single word $\nu^{-1}(a)$.

(ii) Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ and $a$ be encoded by $uv^\omega$. Then the presentation of $a$ is

$$L_a := \{\, w \in \Sigma^\omega \mid w \otimes uv^\omega \in L_\varepsilon \,\} = \pi_1\big(L_\varepsilon \cap (\pi_2)^{-1}(uv^\omega)\big),$$

where $\pi_i$ is the projection on the $i^{\text{th}}$ component. □

**Proposition 3.8.** [T]AutStr *is closed under finite variations of some relation.*

*Proof.* Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ be an injective presentation of some $\mathfrak{A} = (A, R_0, \dots, R_r) \in$ [T]AutStr. We have to show that $\mathfrak{A}' = (A, R'_0, \dots, R'_n)$ is also in [T]AutStr where $R'_j$ and $R_j$ differ only in finitely many tuples. Construct

$$\mathfrak{d}' = (\nu, \Sigma, L_\delta, L_\varepsilon, L'_{R_0}, \dots, L'_{R_r})$$

with $L'_{R_j} := L_{R_j} \setminus X_j^- \cup X_j^+$ where

$$X_j^- := \nu^{-1}(R_j \setminus R'_j) \qquad \text{and} \qquad X_j^+ := \nu^{-1}(R'_j \setminus R_j)$$

are finite sets. Therefore $L'_{R_j}$ is also regular. □

## 3.2 First-Order Queries

After having defined automatic presentations the question arises what can be done with them. The most fundamental operation on structures is the evaluation of a query, i.e., we are given a formula $\varphi(\overline{x})$ and ask which elements $\overline{a}$ of the structure $\mathfrak{A}$ satisfy $\varphi$. Formally, we want to compute $\varphi^{\mathfrak{A}}$ from $\mathfrak{A}$ and $\varphi$. In case of automatic structures this operation is not only effective but—due to the extensive closure properties of regular languages—the encoding of the resulting set is also regular.

For ease of notation we use regular expressions instead of automata constructions in the definition below and in most other places. But in an actual implementation one will usually work with automata which are easier to handle algorithmically.

**Definition 3.9.** Let $\tau = \{R_0, \dots, R_r\}$ be a finite relational signature, $r_j$ the arity of $R_j$, and $\mathfrak{A}$ a $\tau$-structure with presentation

$$\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r}).$$

We define the function $\eta_n^{\mathfrak{d}} : \mathrm{FO}[\tau] \to \mathscr{P}(L_\delta^{\otimes n})$ which maps formulae $\varphi$ all of whose variables are among $\{x_0, \dots, x_{n-1}\}$ to a presentation of the set

$$\{\, (a_0, \dots, a_{n-1}) \mid \mathfrak{A} \models \varphi(\overline{a}) \,\}.$$

From this set an encoding of $\varphi^{\mathfrak{A}}$ can be obtained by removing the components of those variables which do not appear free in $\varphi$. The corresponding function is denoted $\eta^{\mathfrak{d}}$ (without the index $n$).

To select and permute the components of a word we define the auxiliary mapping

$$\sigma^{m,n}_{(i_0,k_0)\dots(i_{l-1},k_{l-1})} : \mathscr{P}\big((\Sigma^*)^{\otimes m}\big) \to \mathscr{P}\big((\Sigma^*)^{\otimes n}\big)$$

which takes a language $L$ to the set

$$\{\, w_0 \otimes \dots \otimes w_{n-1} \mid \exists u_0 \otimes \dots \otimes u_{m-1} \in L \colon u_{i_j} = w_{k_j} \text{ for all } j < l \,\},$$

i.e., component $i_j$ is moved to position $k_j$. $\sigma^{m,n}_{(i_0,k_0)\dots(i_{l-1},k_{l-1})}$ preserves regularity since it can be defined as

$$\sigma^{m,n}_{(i_0,k_0)\dots(i_{l-1},k_{l-1})}(L) := (\Sigma^*)^{\otimes n} \cap \big[\big(\pi^{n,l}_{k_0\dots k_{l-1}}\big)^{-1} \circ \big(\pi^{m,l}_{i_0\dots i_{l-1}}\big)\big]\big(L(\square^m)^*\big)$$

where $\pi^{n,l}_{i_0\dots i_{l-1}} : \mathscr{P}((\Sigma^n)^*) \to \mathscr{P}((\Sigma^l)^*)$ denotes the projection with

$$\pi^{n,l}_{i_0\dots i_{l-1}}\big((a_0, \dots, a_{n-1})\big) = (a_{i_0}, \dots, a_{i_{l-1}}).$$

In the above definition we had to add the factor $(\square^m)^*$ because the other—unspecified—components may be longer that those from $L$.

Using this function, $\eta^{\mathfrak{d}}$ can be defined in terms of $\eta_n^{\mathfrak{d}}$ by

$$\eta^{\mathfrak{d}}(\varphi) := \sigma^{n,k}_{(i_0,0)\dots(i_{k-1},k-1)}\big(\eta_n^{\mathfrak{d}}(\varphi)\big)$$

if the free variables of $\varphi$ are $x_{i_0}, \dots, x_{i_{k-1}}$.

Finally, $\eta_n^{\mathfrak{d}}$ is defined per induction on $\varphi$. For atoms we simply return the corresponding language of the presentation after moving the components into the right position.

$$\begin{aligned}
\eta_n^{\mathfrak{d}}(R_j x_{i_0} \dots x_{i_{r_j-1}}) &:= L_\delta^{\otimes n} \cap \sigma^{r_j,n}_{(0,i_0)\dots(r_j-1,i_{r_j-1})}\big(L_{R_j}\big), \\
\eta_n^{\mathfrak{d}}(x_i = x_j) &:= L_\delta^{\otimes n} \cap \sigma^{2,n}_{(0,i)(1,j)}\big(L_\varepsilon\big).
\end{aligned}$$

Boolean connectives are handled by the corresponding set operations.

$$\begin{aligned}
\eta_n^{\mathfrak{d}}(\neg\varphi) &:= L_\delta^{\otimes n} \setminus \eta_n^{\mathfrak{d}}(\varphi), \\
\eta_n^{\mathfrak{d}}(\varphi \vee \psi) &:= \eta_n^{\mathfrak{d}}(\varphi) \cup \eta_n^{\mathfrak{d}}(\psi).
\end{aligned}$$

Finally, for the existential quantifier we erase the component of the variable in question.

$$\eta_n^{\mathfrak{d}}(\exists x_i \varphi) := L_\delta^{\otimes n} \cap \sigma^{n,n}_{(0,0)\dots(i-1,i-1)(i+1,i+1)\dots(n-1,n-1)}\big(\eta_n^{\mathfrak{d}}(\varphi)\big).$$

Of course, we have to show that the above construction is correct.

**Proposition 3.10.** *Let* $\mathfrak{A} \in [\omega\text{-}][\mathrm{T}]\mathrm{AutStr}$ *have the automatic presentation* $\mathfrak{d}$. *For all formulae* $\varphi \in \mathrm{FO}$ *it holds that* $\nu\big(\eta^{\mathfrak{d}}(\varphi)\big) = \varphi^{\mathfrak{A}}$.

*Proof.* Per induction on the structure of $\varphi$ prove that

$$\nu(\eta^{\mathfrak{d}}_n(\varphi)) = \{\, \overline{a} \in A^n \mid (\mathfrak{A}, \overline{a}) \models \varphi \,\}$$

where $n$ is chosen large enough such that the indices of all variables $x_i$ appearing in $\varphi$ are below $n$. As an example we prove the case of $\varphi = Rx_ix_j$.

($\subseteq$) Let

$$w_0 \otimes \cdots \otimes w_{n-1} \in \eta^{\mathfrak{d}}_n(Rx_ix_j) = L_\delta^{\otimes n} \cap \sigma^{2,n}_{(0,i)(1,j)}(L_R).$$

Then $w_0, \dots, w_{n-1} \in L_\delta$, and $w_i \otimes w_j \in L_R$. Thus, $\big(\nu(w_i), \nu(w_j)\big) \in R$ and

$$(\mathfrak{A}, \nu(w_0) \dots \nu(w_{n-1})) \models Rx_ix_j.$$

($\supseteq$) If on the other hand $(\mathfrak{A}, \overline{a}) \models Rx_ix_j$ for some $\overline{a} \in A^n$ with encodings $w_0, \dots, w_{n-1} \in L_\delta$, then $(a_i, a_j) \in R$ and thus $w_i \otimes w_j \in L_R$. Hence,

$$w_0 \otimes \cdots \otimes w_{n-1} \in L_\delta^{\otimes n} \cap \sigma^{2,n}_{(0,i)(1,j)}(L_R) = \eta^{\mathfrak{d}}_n(Rx_ix_j).$$

□

In the case of word and tree languages we are able to do a bit more.

**Proposition 3.11.** *For* $\mathfrak{A} \in [\mathrm{T}]\mathrm{AutStr}$ *the function* $\eta$ *can be extended to formulae of* $\mathrm{FO}(\exists^\omega)$.

*Proof.* Let $\mathfrak{d}$ be an injective presentation of $\mathfrak{A}$. Define

$$\eta^{\mathfrak{d}}_n(\exists^\omega x_i\varphi) := L_\delta^{\otimes n} \cap \sigma^{n,n}_{(0,0)\dots(i-1,i-1)(i+1,i+1)\dots(n-1,n-1)}\big(\eta^{\mathfrak{d}}_n(\varphi)W_k^{-1}\big).$$

where $k$ is the index of the Nerode-congruence of the language $\eta^{\mathfrak{d}}_n(\varphi)$ and

$$W_k := \varepsilon^{\otimes i-1} \otimes \Sigma^k \otimes \varepsilon^{\otimes n-i}.$$

We give the induction step in the proof that

$$\nu\big(\eta^{\mathfrak{d}}_n(\exists^\omega x_{n-1}\varphi)\big) = \{\, \overline{a} \in A^n \mid \text{there are infinitely many } a \in A \text{ such that } \mathfrak{A} \models \varphi(a_0, \dots, a_{n-2}, a) \,\}.$$

($\supseteq$) Fix values $a_0, \dots, a_{n-2}$. If there are infinitely many values $a_{n-1}$ for $x_{n-1}$ satisfying $\varphi$ there exists such an element $a_{n-1} \in A$ with $\lambda^{\mathfrak{d}}(a_{n-1}) \geq k + \max\{\lambda^{\mathfrak{d}}(a_0), \dots, \lambda^{\mathfrak{d}}(a_{n-2})\}$. Thus

$$(a_0, \dots, a_{n-2}, a_{n-1}) \in \nu\big[\eta^{\mathfrak{d}}_n(\varphi) \cap (\Sigma^*)^{\otimes n}(\varepsilon^{\otimes n-1} \otimes \Sigma^k)\big].$$

Let $x$ be the prefix of $\nu^{-1}(a_{n-1})$ of length $\lambda^{\mathfrak{d}}(a_{n-1}) - k$. Then

$$\nu^{-1}(a_0) \otimes \cdots \otimes \nu^{-1}(a_{n-2}) \otimes x \in \eta^{\mathfrak{d}}_n(\varphi)\big(\varepsilon^{\otimes n-1} \otimes \Sigma^k\big)^{-1},$$

which implies that $(a_0, \dots, a_{n-2}, a) \in \nu\big(\eta^{\mathfrak{d}}_n(\exists^\omega x_{n-1}\varphi)\big)$ for all $a \in A$.

($\subseteq$) If on the other hand there are elements $(a_0, \dots, a_{n-1}) \in \nu\big(\eta_n^{\mathfrak{d}}(\exists^\omega x_{n-1}\varphi)\big)$ then there is some $a_{n-1} \in A$ with

$$\nu^{-1}(a_0) \otimes \cdots \otimes \nu^{-1}(a_{n-2}) \otimes \nu^{-1}(a_{n-1}) \in \eta_n^{\mathfrak{d}}(\varphi) \cap (\Sigma^*)^{\otimes n}(\varepsilon^{\otimes n-1} \otimes \Sigma^k).$$

When applying the Pumping Lemma to the suffix of length $k$ of this word we get infinitely many words of the form $\nu^{-1}(a_0)\otimes\cdots\otimes\nu^{-1}(a_{n-2})\otimes x$ which differ only in $x$ as the suffix does not contain any symbols from the first $n-1$ arguments. Since the presentation is injective each $x$ encodes a different element and thus there are infinitely many $a_{n-1} \in A$ with $(a_0, \dots, a_{n-2}, a_{n-1}) \in \varphi^{\mathfrak{A}}$. □

As the definition of $\eta^{\mathfrak{d}}$ is effective we obtain the following

**Corollary 3.12.**

(i) *The* FO($\exists^\omega$)*-theory of any structure in* [T]AutStr *is decidable.*

(ii) *The* FO*-theory of any structure in* $\omega$-[T]AutStr *is decidable.*

Its importance lies in the fact that it yields one of the two methods known to the author to prove that a structure is not automatic. If the first-order theory of some structure $\mathfrak{A}$ is undecidable then $\mathfrak{A}$ cannot be automatic.

*Example.* As the first-order theory of Arithmetic $(\mathbb{N}, +, \cdot)$ is undecidable it does not have an automatic presentation, i.e., $(\mathbb{N}, +, \cdot) \notin [\omega\text{-}]$[T]AutStr.

A second important consequence of Proposition 3.10 is the following result which yields a notion of reduction of one automatic structure to another.

**Proposition 3.13.**

(i) [T]AutStr *is closed under (*$k$*-dimensional)* FO($\exists^\omega$)*-interpretations.*

(ii) $\omega$-[T]AutStr *is closed under (*$k$*-dimensional)* FO*-interpretations.*

*Proof.* We just give the proof for AutStr. Let $\mathcal{I} = (h, \delta, \varepsilon, \varphi_{R_0}, \dots, \varphi_{R_r})$ be a $k$-dimensional FO($\exists^\omega$)-interpretation of $\mathfrak{A}$ in $\mathfrak{B}$. Let

$$\mathfrak{d}^{\mathfrak{B}} = (\nu^{\mathfrak{B}}, \Sigma^{\mathfrak{B}}, L_\delta^{\mathfrak{B}}, L_\varepsilon^{\mathfrak{B}}, L_{S_0}^{\mathfrak{B}}, \dots, L_{S_s}^{\mathfrak{B}})$$

be a presentation of $\mathfrak{B}$. We construct an automatic presentation $\mathfrak{d}^{\mathfrak{A}}$ of $\mathfrak{A}$. Set

$$\mathfrak{d}^{\mathfrak{A}} := (\nu^{\mathfrak{A}}, \Sigma^{\mathfrak{A}}, L_\delta^{\mathfrak{A}}, L_\varepsilon^{\mathfrak{A}}, L_{R_0}^{\mathfrak{A}}, \dots, L_{R_r}^{\mathfrak{A}})$$

where

$$\begin{aligned}
\Sigma^{\mathfrak{A}} &:= \big(\Sigma^{\mathfrak{B}} \cup \{\Box\}\big)^k, \\
\nu^{\mathfrak{A}}(x) &:= h\big[\nu^{\mathfrak{B}}\big(\pi_0(x)\big), \dots, \nu^{\mathfrak{B}}\big(\pi_{k-1}(x)\big)\big], \\
L_\delta^{\mathfrak{A}} &:= (L_\delta^{\mathfrak{B}})^{\otimes k} \cap \eta_k^{\mathfrak{d}^{\mathfrak{B}}}(\delta), \\
L_\varepsilon^{\mathfrak{A}} &:= (L_\delta^{\mathfrak{A}})^{\otimes 2} \cap \eta_{2k}^{\mathfrak{d}^{\mathfrak{B}}}(\varepsilon), \\
L_{R_j}^{\mathfrak{A}} &:= (L_\delta^{\mathfrak{A}})^{\otimes r_j} \cap \eta_{r_j k}^{\mathfrak{d}^{\mathfrak{B}}}(\varphi_{R_j}).
\end{aligned}$$

□

Some immediate consequences are summarised in

**Corollary 3.14.** [$\omega$-][T]AutStr *is closed under*

(i) *expansions by definable relations,*

(ii) *factorisations by definable congruences,*

(iii) *substructures with definable universe, and*

(iv) *finite powers.*

Before getting ones hopes too high, here is a warning that even some of the simplest model theoretic constructions do not work for automatic structures.

**Lemma 3.15.** *There is a structure $\mathfrak{A}$ such that every reduct of $\mathfrak{A}$ has an automatic presentation but $\mathfrak{A}$ itself is not automatic.*

*Proof.* Consider $\mathfrak{A} := (\mathbb{N}, +, {}^2)$, the natural numbers with addition and squaring function. Since multiplication is definable in $\mathfrak{A}$ its first-order theory is undecidable and therefore $\mathfrak{A} \notin$ [$\omega$-][T]AutStr. What about the reducts? $(\mathbb{N})$ obviously has an automatic presentation, and we already know that $(\mathbb{N}, +) \in$ AutStr. A presentation of $(\mathbb{N}, {}^2)$ can be constructed as follows. Let

$$M := \mathbb{N} \setminus \{\, k^2 \mid k \in \mathbb{N} \}$$

be the set of non-squares. Every natural number $n \in \mathbb{N} \setminus \{0, 1\}$ can uniquely be written as $n = m^{2^k}$ for some $m \in M$ and $k \in \mathbb{N}$. Hence, we can encode $n$ by $(m, k)$. The squaring functions acts as $(m, k) \mapsto (m, k+1)$ on this encoding. Therefore we set $\mathfrak{d} := (\nu, \{0, 1, a, b\}, L_\delta, L_\varepsilon, L_2)$ where

$$\begin{aligned} \nu(0) &:= 0, & L_\delta &:= \{0, 1\} \cup a^* b^*, \\ \nu(1) &:= 1, & L_\varepsilon &:= \left\{ \left[\begin{smallmatrix} c \\ c \end{smallmatrix}\right] \mid c \in \{0, 1, a, b\} \right\}^*, \\ \nu(a^m b^k) &:= l_m^{2^k}, & L_2 &:= \left\{ \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right] \right\} \cup \left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right]^* \left[\begin{smallmatrix} b \\ b \end{smallmatrix}\right]^* \left[\begin{smallmatrix} \Box \\ b \end{smallmatrix}\right], \end{aligned}$$

and $l_0, l_1, \ldots$ is an enumeration of $M$. □

**Lemma 3.16.** [$\omega$-][T]AutStr *is not closed under arbitrary substructures.*

*Proof.* Consider $\mathfrak{A} := (\mathbb{N}, <, P)$ with $P := 2\mathbb{N}$. This structure is clearly automatic. Let $X \subseteq \mathbb{N}$ be any non-recursive set, and construct the substructure $\mathfrak{B} \subseteq \mathfrak{A}$ with universe

$$B := \{\, 2n \mid n \in X \,\} \cup \{\, 2n + 1 \mid n \notin X \,\}.$$

Then $\mathfrak{B} = (B, <|_B, P|_B) \cong (\mathbb{N}, <, X)$. But $\mathrm{Th}(\mathfrak{B})$ cannot be decidable for, otherwise, $X$ would be recursive. □

## 3.3 Extensions of First-Order Logic

We have seen that automatic structures are quite well behaved with regard to first-order logic. What about stronger logics? Possible applications of automatic structures include automatic verification where the most important problem is Reachability, and databases where one usually wants to have some sort of recursion. Thus it is natural to consider transitive closure and fixed-point extensions of first-order logic. Unfortunately, even slightly stronger logics than FO, respectively $\mathrm{FO}(\exists^\omega)$, are already undecidable.

**Proposition 3.17.**

(i) $[\omega\text{-}][\mathrm{T}]\mathrm{AutStr}$ *contains structures with undecidable* $\mathrm{FO}(\mathrm{DTC}^1)$*-theory.*

(ii) $[\omega\text{-}][\mathrm{T}]\mathrm{AutStr}$ *is not closed under expansion by* $\mathrm{FO}(\mathrm{DTC}^1)$*-definable relations.*

*Proof.* This result follows immediately from Proposition 2.7 and the closure of $[\omega\text{-}][\mathrm{T}]\mathrm{AutStr}$ under finite powers. Nevertheless we give an explicit proof which strengthens the claim to formulae such that in all subformulae of the form $[\mathrm{DTC}_{x,y}\ \psi(x,y)](x,y)$ the only free variables appearing in $\psi$ are $x$ and $y$.

Presburger Arithmetic $(\mathbb{N},+)$ is automatic. We define multiplication using transitive closure.

$$x \cdot y = z \ :\text{iff}\ [\mathrm{DTC}_{xyz,x'y'z'}\ x' = x \wedge y' + 1 = y \wedge z' = z + x](xy0, x0z)$$

If automatic structures were closed under deterministic transitive closure there would be an automatic presentation of Arithmetic $(\mathbb{N},+,\cdot)$ in contradiction to the example above.

The expression above uses a 3-dimensional DTC-operator. We can replace it by a 1-dimensional one if we take the structure consisting of Presburger Arithmetic together with its third power, i.e., $(\mathbb{N} \cup \mathbb{N}^3, +, \pi_0, \pi_1, \pi_2)$ where $+ \subseteq \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ is the graph of addition and $\pi_i \subseteq \mathbb{N}^3 \times \mathbb{N}$ is the projection on the $i^{\text{th}}$ coordinate. □

**Proposition 3.18.** *For structures in* $[\omega\text{-}][\mathrm{T}]\mathrm{AutStr}$*,* Reachability *is undecidable.*

*Proof.* Let $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ be a Turing machine. We construct an automatic presentation of its configuration graph. A configuration $(q, w, p)$ is encoded by the word $w_0 q w_1$ with $w = w_0 w_1$ and $|w_0| = p$. The transition relation $\vdash_M$ is clearly recognisable by an automaton as it depends only on the finite region of the word around the position of the state symbol. If Reachability were decidable there would be an algorithm deciding the halting problem. W.l.o.g. assume $M$ has an unique accepting state $q_f$ and clears its tape before accepting. Then, given $M$ and an input $x$, we could construct the presentation of its configuration graph and check whether the accepting configuration is reachable from the starting configuration, i.e., whether $(q_f, \varepsilon, 0)$ is reachable from $(q_0, x, 0)$. □

**Proposition 3.19.**

(i) $[\omega\text{-}][\mathrm{T}]\mathrm{AutStr}$ *contains structures with undecidable* $\mathrm{FO}(\#)$*-theory.*

(ii) $[\omega\text{-}][\mathrm{T}]\mathrm{AutStr}$ *is not closed under expansion by* $\mathrm{FO}(\#)$*-definable relations.*

*Proof.* Consider the automatic structure $(\mathbb{N} \cup \mathbb{N}^2, +, \pi_0, \pi_1)$ where $+ \subseteq \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ is the graph of addition and $\pi_i \subseteq \mathbb{N}^2 \times \mathbb{N}$ is the projection on the $i^{\text{th}}$ coordinate. Multiplication can be defined as

$$x \cdot y = z \ :\text{iff}\ \#_v(v < z) = \#_v \exists u_1 u_2 (\pi_0 v u_1 \wedge \pi_1 v u_2 \wedge u_1 < x \wedge u_2 < y)$$

with the abbreviation

$$x < y \ :\text{iff}\ \exists z (x + z = y) \wedge x \neq y.$$

Therefore there is a FO(#)-interpretation of Arithmetic in this structure and the undecidability follows. □

## 3.4 Complexity of Queries

After having seen what can be done with automatic structures we now study the complexity of those operations. (For an introduction to complexity theory see [HU79, Pap94, Imm98].) We investigate the following fundamental problems.

The most basic one is the *model-checking* problem: Given a $\tau$-structure $\mathfrak{A}$, a formula $\varphi \in \mathrm{FO}[\tau]$, and a tuple of parameters $\overline{a}$ in $\mathfrak{A}$, decide whether $\mathfrak{A} \models \varphi(\overline{a})$ does or does not hold.

A generalisation is the *query-evaluation* problem: Given a presentation $\mathfrak{d}$ and a formula $\varphi$, compute $\eta^{\mathfrak{d}}(\varphi)$.

The complexity of both problems can be investigated under three points of view. First one can hold the formula fixed and ask how the complexity depends on the input structure. If the complexity is measured in this way we speak of *structure complexity*. On the other hand one can fix the structure and measure the dependency on the formula. This leads to the notion of *expression complexity*. Finally, one can look at the so called *combined complexity* where both parts may vary.

Of course, statements about complexity are only meaningful if the encoding of the input is specified. A presentation $\mathfrak{d}$ is given by a mapping $\nu$ and several regular languages. $\nu$ is a purely semantic object which is not part of the input of an algorithm. There are various ways to encode regular languages, but the representation which can be handled by algorithms most easily uses automata. Therefore in this section we assume that $\mathfrak{d}$ is given by a list of deterministic automata. Furthermore we only consider presentations using binary alphabets.

Deterministic automata are chosen because boolean operations on them can be performed in polynomial time whereas negation of nondeterministic automata may cause an exponential blowup. If the input is restricted to positive formulae the results below hold for presentations given by nondeterministic automata as well.

We use the following notations for the size of the input. For a presentation $\mathfrak{d}$, $|\mathfrak{d}|$ denotes the maximal size of the automata belonging to $\mathfrak{d}$, and we use $\lambda^{\mathfrak{d}}(\overline{a})$ as an abbreviation for the maximum of $\lambda^{\mathfrak{d}}(a_i)$ for all $i$.

Our first result is rather discouraging. A function is said to be *non-elementary* if it cannot be bounded from above by a function of the form

$$\left.2^{2^{\cdot^{\cdot^{2^n}}}}\right\}k$$

for fixed $k$.

**Proposition 3.20.** *The expression complexity of the model-checking problem is non-elementary.*

*Proof.* The claim follows immediately from the fact that $\mathfrak{N}_p := (\mathbb{N}, +, |_p)$ is automatic where

$$a \,|_p\, b \;:\text{iff}\;\; a \text{ is a power of } p \text{ and } a \mid b,$$

since the theory of $\mathfrak{N}_p$ has non-elementary complexity (see [Grä90]). □

Let us hope that in some restricted cases the complexity is less devastating. We begin by taking a closer look at the simulation of automata.

**Lemma 3.21.** *Given a deterministic automaton* $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$ *and a word* $w \in \Sigma^*$*, to check whether* $w \in L(\mathfrak{A})$ *is in*

$$\text{DTIME}\big[\mathcal{O}\big(|w|\,|Q|\log|Q|\big)\big] \qquad \textit{and} \qquad \text{DSPACE}\big[\mathcal{O}\big(\log|Q| + \log|w|\big)\big].$$

*Proof.* We use the following algorithm:

```
Input: δ, F, w
q  := q0
i  := 0
m := |w|
while i < m do
    a := w[i]
    q := δ(q, a)
    i  := i + 1
end
return q ∈ F
```

The space used consists of the current state, the input position, and the length of the input.

In order to minimise the time needed to access the current symbol of the word $w$ we slightly modify the above algorithm such that it copies $w$ to a separate work tape first. Then we can leave the head of that tape on the current symbol and do not need to go back and forth between the two parts of the input. Therefore the first and last line of the loop can be performed in constant time. The second line requires a lookup in $\delta$ which can be done by scanning $\delta$ until the state $q$ is found. This takes $\mathcal{O}(|\delta|\log|Q|) = \mathcal{O}(|Q|\log|Q|)$ steps. The loop is executed $|w|$ times.

The initialisations take time $O(|w|)$. To check whether $q \in F$ the algorithm scans the encoding of $F$ and looks for $q$. This needs time $\mathcal{O}\big(|F|\log|Q|\big) = \mathcal{O}\big(|Q|\log|Q|\big)$. Putting everything together, we obtain the desired bound. □

**Lemma 3.22.** *Given a nondeterministic automaton* $\mathfrak{A} = (Q, \Sigma, \Delta, q_0, F)$ *and a word* $w \in \Sigma^*$*, to check whether* $w \in L(\mathfrak{A})$ *is in*

$$\text{DTIME}\big[\mathcal{O}\big(|w|\,|\Delta|\,|Q|\log|Q|\big)\big] \qquad \textit{and} \qquad \text{DSPACE}\big[\mathcal{O}\big(|Q| + \log|w|\big)\big].$$

*Proof.* We use the following algorithm:

```
Input: Δ, F, w
P := {q0}
for i = 0, ..., |w| − 1 do
    a  := w[i]
    P' := ∅
    forall (q, a, q') ∈ Δ do
        if q ∈ P then   P' := P' ∪ {q'}
    P := P'
end
return P ∩ F ≠ ∅
```

The space used consists of the current set of states, the input position, and the length of the input.

If the sets are implemented using arrays of bits, the statements in the body of the loop use time $\mathcal{O}(|Q|)$ for erasing $P'$; $\mathcal{O}(|Q|\log|Q|)$ for testing the condition in the **if**-statement; and $\mathcal{O}(|Q|\log|Q|)$ for the updates of $P'$ and $P$.

To check whether there was a successful run takes time $\mathcal{O}(|Q|)$. Therefore, the overall time used is as given above. □

When considering the structure complexity of a problem, the automata of the presentation are fixed. Therefore we also look at the non-uniform version of the membership problem for regular languages.

**Lemma 3.23.** *Let $L \subseteq \{0,1\}^*$ be regular. The problem to determine, given a word $w \in \{0,1\}^*$, whether $w \in L$, is in* ALOGTIME.

*Proof.* Our alternating log-time algorithm is based on the characterisation of a regular language $L$ via its syntactic monoid $M(L)$. It is well known that a language $L$ is regular if and only if there is some finite monoid $M(L)$, a subset $P \subseteq M(L)$, and a homomorphism $\eta_L : \{0,1\}^* \to M(L)$ such that $L = \eta_L^{-1}(P)$.

Let $w = a_0 \cdots a_{n-1}$ and $e_i := \eta_L(a_i)$ for $i < n$. Thus

$$a_0 \cdots a_{n-1} \in L \text{ iff } e_0 \cdots e_{n-1} \in P.$$

The algorithm starts by guessing $e_0 \cdots e_{n-1}$ and verifies its guess by recursively determining the values of $e_0 \cdots e_{n/2-1}$ and $e_{n/2} \cdots e_{n-1}$.

> Input: $a_0 \cdots a_{n-1}$
> existentially guess $m \in P$
> **repeat** $\lceil \log n \rceil$ times
>   existentially guess $m_0$, $m_1 \in M(L)$
>   **if** $m \neq m_0 m_1$ **then return** false
>   universally choose $i \in \{0,1\}$
>   append $i$ to the index tape
>   $m := m_i$
> **end**
> read the symbol $a$ whose number is stored on the index tape
> **return** $\eta_L(a) = m$ □

So far, we only dealt with relational signatures as functions can easily be replaced by their graphs. But to do so we need to introduce additional quantifiers which is not possible if we want to investigate quantifier-free formulae. When studying quantifier-free formulae with functions we need an algorithm to compute the value of a function whose graph is given by some automaton.

**Lemma 3.24** (Epstein et al. [ECH+92]). *Given a tuple $\overline{w}$ of words over $\Sigma$, and an automaton $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$ recognising the graph of a function $f$, the calculation of $f(\overline{w})$ is in*

$$\text{DTIME}\big[\mathcal{O}\big(|Q|^2 \log|Q|\,(|Q| + |\overline{w}|)\big)\big] \textit{ and}$$
$$\text{DSPACE}\big[\mathcal{O}\big(|Q|\log|Q| + \log|\overline{w}|\big)\big].$$

*Proof.* The following algorithm simulates $\mathfrak{A}$ on input $w_0 \otimes \cdots \otimes w_{n-1} \otimes x$ where $x$ is the result that we want to calculate. For every position $i$ of the input, the set $Q_i$ of states which can be reached for various values of $x$ is determined. At the same time the sets $Q_i$ and $Q_{i+1}$ are connected by edges $E_i$ labelled by the symbol of $x$ by which the second state could be reached. When a final state is found, $x$ can be read off the graph.

We use the following function to compute $Q_{i+1}$ and $E_i$ from $Q_i$ and the input symbol $\overline{a}$.

```
Step(Q, ā)
    Q' := ∅
    E  := ∅
    forall q ∈ Q do
        forall c ∈ Σ do
            q' := δ(q, āc)
            if q' ∉ Q' then
                E  := E ∪ {(q, c, q')}
                Q' := Q' ∪ {q'}
            end
        end
    return (Q', E)
```

If $E$ is realised as an array containing, for every $q \in Q$, the values $q'$ and $c$ such that $(q', c, q) \in E$, this function needs space $\mathcal{O}\big(|Q| \log |Q|\big)$ and time

$$\mathcal{O}\big(|Q|\,(|Q| \log |Q| + |Q| \log |Q|)\big) = \mathcal{O}\big(|Q|^2 \log |Q|\big).$$

We use two slightly different algorithms for the time and space complexity bounds. The first one simply computes all set $Q_i$ and $E_i$ and determines $x$. The second one reuses space and keeps only one set $Q_i$ and $E_i$ in memory. Therefore it has to start the computation from the beginning in order to access old values of $E_i$ in the second part.

In the first version the function Step is invoked $|x|$ times, and the second part is executed in time $\mathcal{O}\big(|x|\,|Q| \log |Q|\big)$.

The space needed by the second version consists of storage for $Q$, $E$, and the counters $i$ and $k$. Hence, $\mathcal{O}(|Q| + |Q| \log |Q| + \log |x|)$ bits are used.

Since $\mathfrak{A}$ recognises a function the length of $x$ can be at most $|Q| + |\overline{w}|$ (see Proposition 5.1 for a detailed proof). This yields the given bounds. □

```
Input: 𝔄 = (Q, Σ, δ, q0, F), w̄
Q0 := {q0}
i := 0
while Qi ∩ F = ∅ do
    if i < |w̄| then
        ā := w̄[i]
    else
        ā := □̄
    (Qi+1, Ei) := Step(Qi, ā)
    i := i + 1
end
```

```
Input: 𝔄 = (Q, Σ, δ, q0, F), w̄
Q := {q0}
i := 0
while Q ∩ F = ∅ do
    if i < |w̄| then
        ā := w̄[i]
    else
        ā := □̄
    (Q, E) := Step(Q, ā)
    i := i + 1
end
```

```
let q ∈ Q_i ∩ F
while i > 0 do
    i := i − 1









    let (q′, c, q) ∈ E_i
    x[i] := c
    q := q′
end
return x
```

```
let q ∈ Q ∩ F
while i > 0 do
    i := i − 1
    Q := {q_0}
    for k = 0, ..., i − 1 do
        if k < |w̄| then
            ā := w̄[k]
        else
            ā := □̄
        (Q, E) := Step(Q, ā)
    end
    let (q′, c, q) ∈ E
    x[i] := c
    q := q′
end
return x
```

Obviously, the formula is responsible for the high complexity of the model-checking problem. So we consider restricted classes of formulae. It turns out that model-checking and query-evaluation for quantifier-free and existential formulae are still—to some extent—tractable.

**Proposition 3.25.** (i) *Let $\tau$ be a relational signature. Given the presentation $\mathfrak{d}$ of a structure $\mathfrak{A} \in \mathrm{AutStr}[\tau]$, a tuple $\overline{a}$ in $\mathfrak{A}$, and a quantifier-free formula $\varphi(\overline{x}) \in \mathrm{FO}[\tau]$, the model-checking problem for $(\mathfrak{A}, \overline{a}, \varphi)$ is in*

$$\mathrm{DTIME}\big[\mathcal{O}\big(|\varphi|\,\lambda^{\mathfrak{d}}(\overline{a})\,|\mathfrak{d}|\log|\mathfrak{d}|\big)\big] \text{ and}$$
$$\mathrm{DSPACE}\big[\mathcal{O}\big(\log|\varphi| + \log|\mathfrak{d}| + \log\lambda^{\mathfrak{d}}(\overline{a})\big)\big].$$

(ii) *The structure complexity of the model-checking problem for quantifier-free formulae is* LOGSPACE*-complete with regard to* FO*-reductions.*

(iii) *The expression complexity is* ALOGTIME*-complete with regard to deterministic log-time reductions.*

*Proof.* (i) In order to check whether $\mathfrak{A} \models \varphi(\overline{a})$ holds, we need to know the truth value of each atom appearing in $\varphi$. Then, all what remains is to evaluate a boolean formula which can be done in $\mathrm{DTIME}\big[\mathcal{O}\big(|\varphi|\big)\big]$ and $\mathrm{ATIME}\big[\mathcal{O}\big(\log|\varphi|\big)\big] \subseteq \mathrm{DSPACE}\big[\mathcal{O}\big(\log|\varphi|\big)\big]$ (see [Bus87]). The truth value of an atom $R\overline{x}$ can be calculated by simulating the corresponding automaton on those components of $\overline{a}$ which belong to the variables appearing in $\overline{x}$. According to the lemma above this can be done in time $\mathcal{O}\big(\lambda^{\mathfrak{d}}(\overline{a})\,|\mathfrak{d}|\log|\mathfrak{d}|\big)$ and space $\mathcal{O}\big(\log|\mathfrak{d}| + \log\lambda^{\mathfrak{d}}(\overline{a})\big)$.

For the time complexity bound we perform this simulation for every atom, store the outcome, and evaluate the formula. Since there are at most $|\varphi|$ atoms the claim follows.

To obtain the space bound we cannot store the value of each atom. Therefore we use the LOGSPACE-algorithm to evaluate $\varphi$ and, every time the value of an atom is needed, we simulate the run of the corresponding automaton on a separate set of tapes.

(ii) We reduce the LOGSPACE-complete problem DETREACH, of reachability by deterministic paths, (see e.g. [Imm98]) to the model-checking problem. Given a graph $\mathfrak{G} = (V, E, s, t)$ we construct the automaton $\mathfrak{A} = (V, \{0\}, \Delta, s, \{t\})$ with

$$\begin{aligned}\Delta := \{\, (u, 0, v) \mid u \neq t,\ (u, v) \in E \text{ and there is no } v' \neq v \text{ with} \\ (u, v') \in E \,\} \\ \cup \{(t, 0, t)\}.\end{aligned}$$

That is, we remove all edges originating at vertices with out-degree greater than 1 and add a loop at $t$. Then there is a deterministic path from $s$ to $t$ in $\mathfrak{G}$ iff $\mathfrak{A}$ accepts some word $0^n$ iff $0^{|V|} \in L(\mathfrak{A})$. Thus,

$$(V, E, s, t) \in \text{DETREACH} \quad \text{iff} \quad \mathfrak{B} \models P0^{|V|}$$

where $\mathfrak{B} = (B, P)$ is the structure presented by $(\nu, \{0\}, \{0\}^*, L(\mathfrak{A}))$.

A closer inspection reveals that the above transformation can be defined in first-order logic.

(iii) The third part follows immediately from Lemma 3.23 and the fact that the evaluation of boolean formulae is ALOGTIME-complete (see [Bus87]). ☐

It was remarked above that for quantifier-free formulae the question whether functions are allowed does make a difference.

**Proposition 3.26.** (i) *Let $\tau$ be a signature containing functions. Given the presentation $\mathfrak{d}$ of a structure $\mathfrak{A} \in \mathrm{AutStr}[\tau]$, a tuple $\overline{a}$ in $\mathfrak{A}$, and a quantifier-free formula $\varphi(\overline{x}) \in \mathrm{FO}[\tau]$, the model-checking problem for $(\mathfrak{A}, \overline{a}, \varphi)$ is in*

$$\text{DTIME}\big[\mathcal{O}\big(|\varphi|\,|\mathfrak{d}|^2 \log|\mathfrak{d}|\,\big(|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\overline{a})\big)\big)\big] \text{ and}$$
$$\text{DSPACE}\big[\mathcal{O}\big(|\varphi|\,\big(|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\overline{a})\big) + |\mathfrak{d}| \log|\mathfrak{d}|\big)\big].$$

(ii) *The structure complexity of the model-checking problem for quantifier-free formulae with functions is in* NLOGSPACE.

(iii) *The expression complexity is* PTIME*-complete with regard to $\leq_{\mathrm{m}}^{\log}$-reductions.*

*Proof.* (i) Our algorithm proceeds in two steps. First the values of all functions appearing in $\varphi$ are calculated starting with the innermost one. Then all functions can be replaced by their values and a formula containing only relations remains which can be evaluated as above.

We need to evaluate at most $|\varphi|$ functions. If they are nested the result can be of length $|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\overline{a})$. Thus, by Lemma 3.24, we need space

$$\mathcal{O}\big(|\mathfrak{d}| \log|\mathfrak{d}| + \log\big(|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\overline{a})\big)\big)$$

for the evaluation of a function, space

$$\mathcal{O}\big(|\varphi|\,\big(|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\overline{a})\big)\big)$$

to store the results, and space

$$\mathcal{O}\big(\log|\varphi| + \log|\mathfrak{d}| + \log\big(|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\overline{a})\big)\big)$$

for the final evaluation of $\varphi$. This yields the bound given above.

The evaluation of $|\varphi|$ functions takes time

$$\mathcal{O}\big(|\varphi|\,|\mathfrak{d}|^2 \log|\mathfrak{d}|\,(|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\overline{a}))\big),$$

the evaluation of $\varphi$ time

$$\mathcal{O}\big(|\varphi|\,\big(|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\overline{a})\big)\,|\mathfrak{d}|\log|\mathfrak{d}|\big).$$

(ii) It is sufficient to present a nondeterministic log-space algorithm for evaluating a single fixed atom containing functions. The algorithm simultaneously simulates the automata of the relation and of all functions on the given input. Components of the input corresponding to values of functions are guessed nondeterministically. Each simulation only needs counters for the current state and the input position which both use logarithmic space.

(iii) Let $M$ be a $p(n)$ time-bounded deterministic Turing Machine for some polynomial $p$. A configuration $(q, w, p)$ of $M$ can be coded as word $w_0 q w_1$ with $w = w_0 w_1$ and $|w_0| = p$. Using this encoding both the function $f$ mapping one configuration to its successor and the predicate $P$ for configurations containing accepting states can be recognised by automata. We assume that $f(c) = c$ for accepting configurations $c$. Let $q_0$ be the starting state of $M$. Then $M$ accepts some word $w$ if and only if the configuration $f^{p(|w|)}(q_0 w)$ is accepting if and only if $\mathfrak{A} \models P f^{p(|w|)} q_0 w$ where $\mathfrak{A} = (A, P, f)$ is automatic. Hence, the mapping taking $w$ to the pair $q_0 w$ and $P f^{p(|w|)} x$ is the desired reduction which can clearly be computed in logarithmic space. ☐

**Proposition 3.27.** (i) *Let $\tau$ be a fixed relational signature. Given the presentation $\mathfrak{d}$ of a structure $\mathfrak{A} \in \mathrm{AutStr}[\tau]$, a tuple $\overline{a}$ in $\mathfrak{A}$, and a formula $\varphi(\overline{x}) \in \Sigma_1[\tau]$, the model-checking problem for $(\mathfrak{A}, \overline{a}, \varphi)$ is in*

$$\textsc{Ntime}\big[\mathcal{O}\big(|\varphi|\,|\mathfrak{d}|\,\lambda^{\mathfrak{d}}(\overline{a}) + |\mathfrak{d}|^{\mathcal{O}(|\varphi|)}\big)\big] \text{ and}$$
$$\textsc{Nspace}\big[\mathcal{O}\big(|\varphi|\,(|\mathfrak{d}| + \log|\varphi|) + \log\lambda^{\mathfrak{d}}(\overline{a})\big)\big].$$

(ii) *The structure complexity of the model-checking problem for $\Sigma_1$-formulae is* NPTIME*-complete with regard to $\leq^{\mathrm{p}}_{\mathrm{T}}$-reductions.*

(iii) *The expression complexity is* PSPACE*-complete with regard to $\leq^{\log}_{\mathrm{m}}$-reductions.*

*Proof.* (i) As above we can run the corresponding automaton for every atom appearing in $\varphi$ on the encoding of $\overline{a}$. But now there are some elements of the input missing which we have to guess. Since we have to ensure that the guessed inputs are the same for all automata, the simulation is performed simultaneously.

```
Input: 𝔡, ā, φ = ∃y_0 ··· ∃y_{k−1} ψ(x̄, ȳ)
Let 𝔄_i = (Q_i, Σ, δ_i, 0, F_i), for i < n, be the automata belonging
   to the atoms of φ.
q̄ := (0, ..., 0)
m := λ^𝔡(ā)
for i = 0, ..., m − 1 do
    b̄ := ā[i]
    guess c̄ ∈ Σ^k
    for j = 0, ..., n − 1 do   q_j := δ_j(q_j, b̄c̄)
end
```

**repeat** at most $|Q_0 \times \cdots \times Q_{n-1}|$ times
  **guess** $\overline{c} \in \Sigma^k$
  **for** $j = 0, \ldots, n-1$ **do** $q_j := \delta_j(q_j, \Box \cdots \Box \overline{c})$
**end**
evaluate $\varphi$ with values determined by $\overline{q}$

The algorithm needs the following space:

- for each atom the number of the relation and the numbers of the variables: $\mathcal{O}(|\varphi| \log |\varphi|)$,
- $P$ and $P'$: $\mathcal{O}(|\varphi|\,|\mathfrak{d}|)$ (note that $\tau$ is fixed),
- $i$ and $m$: $\mathcal{O}(\log \lambda^{\mathfrak{d}}(\overline{a}))$, and
- $\overline{b}$ and $\overline{c}$: $\mathcal{O}(|\varphi|)$.

The initialisation can be performed in time $\mathcal{O}\big(|\varphi| + \lambda^{\mathfrak{d}}(\overline{a})\big)$. The **while**-loop is executed $\lambda^{\mathfrak{d}}(\overline{a})$ times. Its body requires $\mathcal{O}\big(|\varphi| + |\varphi|\,|\mathfrak{d}|\big) = \mathcal{O}\big(|\varphi|\,|\mathfrak{d}|\big)$ steps. The body of the **repeat**-loop uses time $\mathcal{O}\big(|\varphi|\,|\mathfrak{d}|\big)$. Therefore the total time is

$$\begin{aligned} &\mathcal{O}\big(|\varphi| + \lambda^{\mathfrak{d}}(\overline{a}) + \lambda^{\mathfrak{d}}(\overline{a})\,|\varphi|\,|\mathfrak{d}| + |\varphi|\,|\mathfrak{d}|\,|\mathfrak{d}|^{|\varphi|}\big) \\ &= \mathcal{O}\big(|\varphi|\,|\mathfrak{d}|\,\lambda^{\mathfrak{d}}(\overline{a}) + |\mathfrak{d}|^{\mathcal{O}(|\varphi|)}\big). \end{aligned}$$

(ii) We reduce the NPTIME-complete non-universality problem for nondeterministic automata over a unary alphabet (see [MS73, HRS76]), given such an automaton check whether it does not recognise the language $0^*$, to the given problem. This reduction is performed in two steps. First the automaton must be simplified and transformed into a deterministic one, then we construct an automatic structure and a formula $\varphi(x)$ such that $\varphi(a)$ holds for several values of $a$ if and only if the original automaton recognises $0^*$. As the model-checking has to be performed for more than one parameter this yields not a many-to-one but a Turing-reduction.

Let $\mathfrak{A} = (Q, \{0\}, \Delta, q_0, F)$ be a nondeterministic finite automaton over the alphabet $\{0\}$. We construct an automaton $\mathfrak{A}'$ such that there are at most two transitions outgoing at every state. This is done be replacing all transition form some given state by a binary tree of transitions with new states as internal nodes. Of course, this changes the language of the automaton. Since in $\mathfrak{A}$ every state has at most $|Q|$ successors, we can take trees of fixed height $k := \lceil \log |Q| \rceil$. Thus, $L(\mathfrak{A}') = h(L(\mathfrak{A}))$ where $h$ is the homomorphism taking 0 to $0^k$. Note that the size of $\mathfrak{A}'$ is polynomial in that of $\mathfrak{A}$.

$\mathfrak{A}'$ still is nondeterministic. To make it deterministic we add a second component to the labels of each transitions which is either 0 or 1. This yields an automaton $\mathfrak{A}''$ such that $\mathfrak{A}$ accepts the word $0^n$ iff there is some word $y \in \{0,1\}^{kn}$ such that $\mathfrak{A}''$ accepts $0^{kn} \otimes y$.

$\mathfrak{A}''$ can be used in a presentation. Let $\mathfrak{d} = (\nu, \{0,1\}, \{0,1\}^*, L(\mathfrak{A}''))$ be the presentation of some $\{R\}$-structure $\mathfrak{B}$. Then

$$\mathfrak{B} \models \exists y\, R0^{kn}y \quad \text{iff} \quad 0^{kn} \otimes y \in L(\mathfrak{A}'') \quad \text{iff} \quad 0^n \in L(\mathfrak{A}).$$

It follows that

$$L(\mathfrak{A}) = 0^* \quad \text{iff} \quad \mathfrak{B} \models \exists y\, R0^{kn}y \text{ for all } n < 2\,|Q|\,.$$

The part ($\Rightarrow$) is trivial. To show ($\Leftarrow$) let $n$ be the least number such that $0^n \notin L(\mathfrak{A})$. By assumption $n \geq 2\,|Q|$. But then we can apply the Pumping Lemma and find some $n' < n$ with $0^{n'} \notin L(\mathfrak{A})$. Contradiction.

(iii) Let $M$ be a $p(n)$ space-bounded Turing machine for some polynomial $p$. As above we encode configurations as words, but this time we append enough spaces to increase their length to $p(n)+1$. Let $L_\vdash := \{\, c_0 \otimes c_1 \mid c_0 \vdash c_1 \,\}$ be the transition relation of $M$. The run of $M$ on input $w$ is encoded as sequence of configurations separated by some marker $\#$. $L_\vdash$ can be used to check whether some word $x$ represents a run of $M$. Let $y$ be the suffix of $x$ obtained by removing the first configuration. The word $x \otimes y$ has the form

$$\begin{matrix} c_0 & \# & c_1 & \# & \multirow{2}{*}{$\dots$} & \# & c_{s-1} & \# & c_s \\ c_1 & \# & c_2 & \# & & \# & c_s & \# & \end{matrix} .$$

Thus $x$ encodes a valid run iff $x \otimes y \in L_T$ where

$$L_T := \Big(L_\vdash \begin{bmatrix}\#\\\#\end{bmatrix}\Big)^* (\Sigma^* \otimes \varepsilon).$$

Clearly, the language $L_F$ of all runs whose last configuration is accepting is regular. Finally, we need two additional relations. Both, the prefix relation $\preceq$ and the shift $s$ are regular where $s(ax) := x$ for $a \in \Sigma$ and $x \in \Sigma^*$. Therefore, the structure $\mathfrak{A} := (A, T, F, s, \preceq)$ is automatic, and it should be clear that

$$w \in L(M) \ \text{ iff } \ \mathfrak{A} \models \varphi_w\big(q_0 w \Box^{k-|w|}\#\big),$$

where $k := p(|w|)$ and

$$\varphi_w(x) := \exists y_0 \cdots \exists y_{k+1} \Big( \bigwedge_{i \leq k} s y_i y_{i+1} \wedge x \preceq y_0 \wedge T y_0 y_{k+1} \wedge F y_0 \Big).$$

$\varphi_w(x)$ states that there is an accepting run $y_0$ of $M$ starting with configuration $x$. $y_1, \dots, y_{k+1}$ are used to remove the first configuration from $y_0$, so we can use $T$ to check whether $y_0$ is valid.

Clearly, the mapping of $w$ to $\varphi_w$ and $q_0 w \Box^{k-|w|}\#$ can be computed in logarithmic space. □

**Proposition 3.28.** (i) *Let $\tau$ be a relational signature. Given the presentation $\mathfrak{d}$ of a structure $\mathfrak{A} \in \mathrm{AutStr}[\tau]$ and a quantifier-free formula $\varphi(\overline{x}) \in \mathrm{FO}[\tau]$, the language $\eta^{\mathfrak{d}}(\varphi)$ can be computed in time $\mathcal{O}\big(|\mathfrak{d}|^{\mathcal{O}(|\varphi|)}\big)$ and space $\mathcal{O}\big(|\varphi| \log |\mathfrak{d}|\big)$.*

*In particular, the structure complexity is in* LOGSPACE *and the expression complexity in* PSPACE.

(ii) *This result is optimal in the sense that there exist presentations $\mathfrak{d}$ and formulae $\varphi$ such that the output is of size $\mathcal{O}\big(|\mathfrak{d}|^{\mathcal{O}(|\varphi|)}\big)$.*

*Proof.* (i) Use the naïve algorithm:

> Input: $\mathfrak{d}$, $\varphi(x_0, \dots, x_{l-1})$
> Let $\mathfrak{A}_i = (Q_i, \Sigma, \delta_i, 0, F_i)$, for $i < n$, be the automata belonging
>   to the atoms of $\varphi$.
> **forall** $\overline{q} \in Q_0 \times \cdots \times Q_{n-1}$ **do**
>   **forall** $\overline{a} \in \Sigma^l$ **do**
>     **for** $j = 0, \dots, n-1$ **do** $q'_j := \delta_j(q_j, \overline{a})$
>     **output** "$\delta(\overline{q}, a) = \overline{q}'$"
>   **end**

> **forall** $\overline{q} \in Q_0 \times \cdots \times Q_{n-1}$ **do**
>   **if** $\varphi$ with values determined by $\overline{q}$ evaluates to true **then**
>     **output** "$\overline{q} \in F$"

The claim follows as $|Q_0 \times \cdots \times Q_{n-1}| = \mathcal{O}(|\mathfrak{d}|^{\mathcal{O}(|\varphi|)})$.

(ii) Let $\mathfrak{d}$ be a presentation of a structure with a single unary relation $R$ which is represented by the language

$$L := \{\, uuv \mid |u| = n \,\}.$$

Let $\mathfrak{A}$ be a minimal automaton recognising $L$. It has

$$2^{n+1} - 1 + 2^n - 1 + 1 = 3 \cdot 2^n - 1$$

states ($2^i$ states for $i \leq n$ to store the prefix of length $i$, $2^i$ states for $i < n$ to store the remaining suffix of length $i$, and one failure state). Define $\varphi$ as

$$\varphi(x_0, \ldots, x_{k-1}) := Rx_0 \wedge \cdots \wedge Rx_{k-1}.$$

Since the run of the resulting automaton on all components is independent it is easy to see that at least $(32^n - 2)^k + 1$ states are needed (the failure state can be shared). □

**Proposition 3.29.** *Let $\tau$ be a relational signature. Given the presentation $\mathfrak{d}$ of a structure $\mathfrak{A} \in \mathrm{AutStr}[\tau]$ and a formula $\varphi(\overline{x}) \in \Sigma_1[\tau]$, the language $\eta^{\mathfrak{d}}(\varphi)$ can be computed in time $\mathcal{O}\big(2^{|\mathfrak{d}|^{\mathcal{O}(|\varphi|)}}\big)$ and space $\mathcal{O}\big(|\mathfrak{d}|^{\mathcal{O}(|\varphi|)}\big)$.*

*In particular, the structure complexity is in* PSPACE *and the expression complexity in* EXPSPACE.

*Proof.* Analogous to above with the state-space $\mathscr{P}(Q_1 \times \cdots \times Q_n)$. □

The complexity results of this section are summarised in the following table.

| | | Structure-Complexity | Expression-Complexity |
|---|---|---|---|
| Model-Checking | $\Sigma_0$ | LOGSPACE-complete | ALOGTIME-complete |
| | $\Sigma_0$ + fun | NLOGSPACE | PTIME-complete |
| | $\Sigma_1$ | NPTIME-complete | PSPACE-complete |
| Query-Evaluation | $\Sigma_0$ | LOGSPACE | PSPACE |
| | $\Sigma_1$ | PSPACE | EXPSPACE |

# Chapter 4

# Complete Structures

We have seen that [$\omega$-][T]AutStr is closed under FO-interpretations. Those interpretations can be regarded as reductions in the sense of complexity theory. A natural question is whether [$\omega$-][T]AutStr contains any *complete* structures with regard to this reduction, i.e., structures $\mathfrak{A}$ such that all other structures in [$\omega$-][T]AutStr can be interpreted in $\mathfrak{A}$. The following theorem gives an affirmative answer. (The structures $\mathfrak{N}_p$, $\mathfrak{R}_p$, $\mathfrak{P}_p$, and $\mathfrak{P}_p^\omega$ are defined below.)

**Theorem 4.1.** *Let $\mathfrak{A}$ be a $\tau$-structure.*

(i) $\mathfrak{A} \in \mathrm{AutStr}[\tau]$ iff $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{N}_p$ *for some/all* $p \geq 2$.

(ii) $\mathfrak{A} \in \omega\text{-}\mathrm{AutStr}[\tau]$ iff $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{R}_p$ *for some/all* $p \geq 2$.

(iii) $\mathfrak{A} \in \mathrm{TAutStr}[\tau]$ iff $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{P}_p$ *for some/all* $p \geq 2$.

(iv) $\mathfrak{A} \in \omega\text{-}\mathrm{TAutStr}[\tau]$ iff $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{P}_p^\omega$ *for some/all* $p \geq 2$.

The proof will take the rest of this chapter. We will show for each type of language (finite words, trees, etc.) that there are structures $\mathfrak{A}$ with presentations of this type whose universe consists of (an encoding of) $\Sigma^*$ for some alphabet $\Sigma$ such that a subset of $\mathfrak{A}$ is FO-definable if and only if its encoding is regular.

## 4.1 Word Languages

Logical definability of regular languages of finite words was investigated already in the 60's by Büchi, Trakhtenbrot and others. We present one classical result (see [BHMV94] for an overview). The structures we are looking at are

$$\mathfrak{N}_p := (\mathbb{N}, +, |_p) \qquad \text{and} \qquad \mathfrak{W}(\Sigma) := (\Sigma^*, (\sigma_a)_{a \in \Sigma}, \preceq, \mathrm{el}),$$

where $+$ is addition, $p \in \mathbb{N} \setminus \{0, 1\}$, and

$$\begin{aligned} x \,|_p\, y \quad &:\text{iff } \exists n, k \in \mathbb{N}\colon x = p^n \text{ and } y = kx, \\ \sigma_a(x) \quad &:= xa, \\ x \preceq y \quad &:\text{iff } \exists z : xz = y, \\ \mathrm{el}(x, y) \quad &:\text{iff } |x| = |y|. \end{aligned}$$

First we show that both are equivalent. Thus we can choose whichever fits out momentary needs. While $\mathfrak{N}_p$ is more convenient to work with, $\mathfrak{W}(\Sigma)$ is much closer to formal languages thereby simplifying some proofs. Actually, in this section we will only be concerned with $\mathfrak{N}_p$, but in the case of $\omega$-languages an adapted version of $\mathfrak{W}(\Sigma)$ will save a lot of work.

**Proposition 4.2** (cf. [Grä90]). $\mathfrak{N}_{|\Sigma|} \rightleftharpoons_{\mathrm{FO}} \mathfrak{W}(\Sigma)$.

*Proof.* W.l.o.g. assume $\Sigma = \mathbb{Z}_p := \{0, \ldots, p-1\}$ for some $p > 1$.

($\mathfrak{W}(\mathbb{Z}_p) \leq_{\mathrm{FO}} \mathfrak{N}_p$) Let $\mathrm{val}_p(w)$ denote the value of the word $w \in \mathbb{Z}_p^*$ viewed as a $p$-adic number with the least significant digit *first*. We cannot just map every word $w \in \mathbb{Z}_p^*$ to $\mathrm{val}_p(w)$, for $w$ may end with zeros which would be discarded. Therefore we encode words $w \in \mathbb{Z}_p^*$ by the number $\mathrm{val}_p(w1)$.

We introduce some abbreviations. In order to access the digits of a number we define

$$\mathrm{dig}_k(x, y) := \exists s \exists t (x = s + k \cdot y + t \ \wedge \ t < y \ \wedge \ p \cdot y \mid_p s)$$

which says that the digit of $x$ at position $y$ is $k$. Powers of $p$ can be defined by $P_p x := x \mid_p x$. The last digit of $x$ is characterised by

$$\mathrm{end}(x, z) := P_p z \ \wedge \ z \leq x < 2 \cdot z.$$

The desired interpretation of $\mathfrak{W}(\Sigma)$ in $\mathfrak{N}_p$ is

$$\begin{aligned}
\delta(x) \quad &:= \exists z \, \mathrm{end}(x, z), \\
\varepsilon(x, y) \quad &:= x = y, \\
\varphi_{\sigma_a}(x, y) &:= \exists z (\mathrm{end}(x, z) \wedge y = p \cdot z + a \cdot z + (x - z)), \\
\varphi_{\preceq}(x, y) \ &:= \exists z \Big[\mathrm{end}(x, z) \wedge \\
&\qquad \forall z' \Big(z' < z \rightarrow \bigwedge_{k<p} (\mathrm{dig}_k(x, z') \leftrightarrow \mathrm{dig}_k(y, z'))\Big)\Big], \\
\varphi_{\mathrm{el}}(x, y) \ &:= \exists z (\mathrm{end}(x, z) \wedge \mathrm{end}(y, z)).
\end{aligned}$$

($\mathfrak{N}_p \leq_{\mathrm{FO}} \mathfrak{W}(\mathbb{Z}_p)$) Here, every word $w$ can simply be seen as $p$-adic encoding of the number $\mathrm{val}_p(w)$. Again, we define some abbreviations. The length of words can be compared with $|x| \leq |y|$ :iff $\exists z (\mathrm{el}(x, z) \wedge z \preceq y)$. The digit of $\mathrm{val}_p(x)$ at position $|y|$ is

$$\mathrm{dig}_k(x, y) := \exists z (|z| = |y| \wedge \sigma_k z \preceq x).$$

In case $k = 0$ we have to consider the case $|y| \geq |x|$ as well.

$$\mathrm{dig}_0(x, y) := \exists z (|z| = |y| \wedge \sigma_k z \preceq x) \vee |y| \geq |x|$$

The universe of the interpretation consists of all words. Two words are equal if they have the same digits.

$$\begin{aligned}
\delta(x) \quad &:= \mathrm{true}, \\
\varepsilon(x, y) &:= \forall z \bigwedge_{k<p} (\mathrm{dig}_k(x, z) \leftrightarrow \mathrm{dig}_k(y, z)).
\end{aligned}$$

$x \mid_p y$ holds iff $x = 0 \cdots 010 \cdots$ and $y = 0 \cdots 0y'$.

$$\begin{aligned}
\varphi_{|_p}(x, y) := \exists z [&\mathrm{dig}_1(x, z) \wedge \forall z' (|z'| \neq |z| \rightarrow \mathrm{dig}_0(x, z')) \\
&\wedge \forall z' (|z'| < |z| \rightarrow \mathrm{dig}_0(y, z'))].
\end{aligned}$$

Addition is slightly more involved. Let

$$A := \{\, (a, b, c, d, d') \mid a + b + d = d'p + c,\ a, b, c \in \mathbb{Z}_p,\ d, d' \in \{0, 1\} \,\}$$

be the set of digits valid for addition. $\varphi_+(x, y, z)$ says that there is some word $u$ encoding the carry such that at all positions the digits of $x$, $y$, $z$, and $u$ are in $A$.

$$\begin{aligned}\varphi_+(x, y, z) := \exists u \big(&\forall v (\mathrm{dig}_0(u, v) \lor \mathrm{dig}_1(u, v)) \\ &\land \forall v \bigvee_{(a,b,c,d,d') \in A} (\mathrm{dig}_a(x, v) \land \mathrm{dig}_b(y, v) \land \mathrm{dig}_c(z, v) \\ &\qquad\qquad \land \mathrm{dig}_d(u, v) \land \mathrm{dig}_{d'}(u, \sigma_0 v))\big).\end{aligned}$$

□

As the universe of $\mathfrak{W}(\Sigma)$ is $\Sigma^*$ one can ask which languages are definable in $\mathfrak{W}(\Sigma)$. We want to use $\mathfrak{N}_p$ instead, so we have to use some sort of encoding. Since numbers may have arbitrarily many leading zeros we can take 0 as the blank symbol □ used by the convolution.

The following result was first proved by Büchi in 1960 where it is stated in a different but equivalent way using weak monadic second-order logic. In the form below it was first proved by Bruyère. A detailed overview is given in [BHMV94].

**Theorem 4.3.** *$R \subseteq \mathbb{N}^n$ is* FO*-definable in $\mathfrak{N}_p$ if and only if* $\mathrm{fold}(\mathrm{val}_p^{-1}(R))$ *is regular.*

*Proof.* (⇒) We construct an automatic presentation of $\mathfrak{N}_p$ using the $p$-adic encoding. Let $\mathfrak{d} := (\mathrm{val}_p, \mathbb{Z}_p, L_\delta, L_\varepsilon, L(\mathfrak{A}_+), L_{|_p})$ where

$$\begin{aligned} L_\delta &:= \mathbb{Z}_p^*, \\ L_\varepsilon &:= \{ \left[\begin{smallmatrix} i \\ i \end{smallmatrix}\right] \mid i \in \mathbb{Z}_p \}^*, \\ L_{|_p} &:= \left[\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right]^* \{ \left[\begin{smallmatrix} 1 \\ i \end{smallmatrix}\right] \mid i \in \mathbb{Z}_p \} \{ \left[\begin{smallmatrix} 0 \\ i \end{smallmatrix}\right] \mid i \in \mathbb{Z}_p \}^*, \end{aligned}$$

and $\mathfrak{A}_+ := (\{0, 1\}, \mathbb{Z}_p^3, \Delta, 0, \{0\})$ just needs to keep track of the carry.

$$\Delta := \{\, (i, (a, b, c), j) \mid pj + c = a + b + i \,\}$$

(⇐) Let $\mathfrak{A} = (Q, \mathbb{Z}_p^n, \Delta, q_0, F)$ be an automaton recognising $\mathrm{fold}(\mathrm{val}_p^{-1}(R))$. W.l.o.g. assume $Q = \mathbb{Z}_p^m$ for some $m$ and $q_0 = (0, \dots, 0)$. We prove the claim by constructing a formula $\psi_\mathfrak{A}(\overline{x}) \in \mathrm{FO}$ stating that there is a successful run of $\mathfrak{A}$ on some word $w \in \mathrm{fold}(\mathrm{val}_p^{-1}(\overline{x})))$. By assumption, if $\mathfrak{A}$ accepts one such word it accepts all words regardless of the number of leading zeros. The run of $\mathfrak{A}$ is encoded by a tuple $(q_0, \dots, q_{m-1}) \in \mathbb{N}^m$ of numbers such that the digits of $q_0, \dots, q_{m-1}$ at some position equal $k_0, \dots, k_{m-1}$ iff the automaton is in state $(k_0, \dots, k_{m-1})$ when scanning the input symbol at that position. Additionally, we have to find a position $s$ to the right of all positions carrying non-zero digits that we can take as length of the input. $\psi_\mathfrak{A}(\overline{x})$ has the form

$$\begin{aligned}\psi_\mathfrak{A}(x_0, \dots, x_{n-1}) := \exists q_0 \cdots \exists q_{m-1} \exists s[&\mathrm{ADM}(\overline{x}, \overline{q}, s) \land \mathrm{START}(\overline{x}, \overline{q}, s) \land \\ &\mathrm{RUN}(\overline{x}, \overline{q}, s) \land \mathrm{ACC}(\overline{x}, \overline{q}, s)],\end{aligned}$$

where the admissibility condition $\mathrm{ADM}(\overline{x}, \overline{q}, s)$ states that $s$ is some position greater than each $x_i$, $\mathrm{START}(\overline{x}, \overline{q}, s)$ says that the first state is $\overline{0}$, $\mathrm{ACC}(\overline{x}, \overline{q}, s)$ that the last one is final, and $\mathrm{RUN}(\overline{x}, \overline{q}, s)$ ensures that all transitions are correct.

We use the abbreviation $\mathrm{Sym}_{\overline{a}}(\overline{x}, z) := \bigwedge_i \mathrm{dig}_{a_i}(x_i, z)$ stating that the digits of $\overline{x}$ at position $z$ are $\overline{a}$. $\mathrm{ADM}(\overline{x}, \overline{q}, s)$ must express that $s$ is a power of $p$ which is greater than any of the $x_i$.

$$\mathrm{ADM}(\overline{x}, \overline{q}, s) := P_p s \wedge \bigwedge_{i<n} x_i < s$$

$\mathrm{START}(\overline{x}, \overline{q}, s)$ and $\mathrm{ACC}(\overline{q}, \overline{x}, s)$ simply say that the first symbol of $\overline{q}$ is $\overline{0}$ and that the last symbol of $\overline{q}$ is in $F$, respectively.

$$\mathrm{START}(\overline{x}, \overline{q}, s) := \mathrm{Sym}_{0\dots 0}(\overline{q}, 1)$$
$$\mathrm{ACC}(\overline{x}, \overline{q}, s) := \bigvee_{\overline{k} \in F} \mathrm{Sym}_{\overline{k}}(\overline{q}, s)$$

Finally, $\mathrm{RUN}(\overline{x}, \overline{q}, s)$ states that at every position a valid transition is used.

$$\mathrm{RUN}(\overline{x}, \overline{q}, s) := \forall z \Big( P_p z \wedge z < s \rightarrow \bigvee_{\tau \in \Delta} \mathrm{Trans}_\tau(\overline{x}, \overline{q}, s) \Big)$$

where $\mathrm{Trans}_\tau(\overline{x}, \overline{q}, z)$ describes a single transition $\tau$.

$$\mathrm{Trans}_{(\overline{k}, \overline{a}, \overline{k}')}(\overline{x}, \overline{q}, z) := \mathrm{Sym}_{\overline{k}}(\overline{q}, z) \wedge \mathrm{Sym}_{\overline{a}}(\overline{x}, z) \wedge \mathrm{Sym}_{\overline{k}'}(\overline{q}, p \cdot z)$$

□

Using this theorem twice we can transform every formula $\varphi$ into an automaton $\mathfrak{A}$ and back to $\psi_{\mathfrak{A}}$. Hence, in $\mathfrak{N}_p$ every formula $\varphi$ is equivalent to a formula of the form $\psi_{\mathfrak{A}}$ for some automaton $\mathfrak{A}$. We call $\psi_{\mathfrak{A}}$ the *automaton normal form* of $\varphi$.

**Corollary 4.4.** *In* $\mathfrak{N}_p$ *every* $\mathrm{FO}(\exists^\omega)$*-formula is equivalent to some* $\Delta_2$*-formula.*

*Proof.* Let $\psi_{\mathfrak{A}}$ be the automaton normal form of the given formula. We have to count its quantifier nesting. 0 and 1 can be defined as

$$\mathrm{DEF}(0,1) := 0 + 0 = 0$$
$$\wedge \forall x \forall y (x + y = 1 \rightarrow (x = 0 \wedge y = 1) \vee (x = 1 \wedge y = 0))$$

which is in $\Pi_1$. Furthermore

$$x < y \in \Sigma_1, \quad \mathrm{ADM} \in \Sigma_1, \quad \mathrm{ACC} \in \Sigma_1,$$
$$\mathrm{dig}_k(x, y) \in \Sigma_1, \quad \mathrm{START} \in \Sigma_1, \quad \mathrm{RUN} \in \Pi_2.$$

Therefore, if $\psi_{\mathfrak{A}}$ is written as

$$\exists \overline{q} \exists s \exists 0 \exists 1 [\mathrm{DEF} \wedge \mathrm{ADM} \wedge \mathrm{START} \wedge \mathrm{RUN} \wedge \mathrm{ACC}]$$

we see that $\psi_{\mathfrak{A}} \in \Sigma_3$. In order to obtain the stronger claim we have to rewrite RUN to some $\Pi_1$-formula. This can be done by expressing that all invalid transitions do not occur instead of listing all valid transitions.

$$\mathrm{RUN}'(\overline{x}, \overline{q}, s) := \forall z \Big( P_p z \wedge z < s \rightarrow \bigwedge_{\tau \notin \Delta} \neg \mathrm{Trans}_\tau(\overline{x}, \overline{q}, s) \Big)$$

$\psi_{\mathfrak{A}}$, as constructed above, is in $\Sigma_2$. Since we can take $\mathfrak{A}$ to be deterministic, an equivalent definition is

$$\forall \overline{q} \forall s \forall 0 \forall 1 [\mathrm{DEF} \wedge \mathrm{ADM} \wedge \mathrm{START} \wedge \mathrm{RUN}' \rightarrow \mathrm{ACC}]$$

which is in $\Pi_2$. □

The last corollary can be strengthened to give an explicit bound on the number of quantifiers which depends only on the number of free variables appearing in the formula. For $w \in \{\exists, \forall\}^*$ let $[w]$ denote the class of all first-order formulae which are equivalent to some FO-formula with the quantifier prefix $w$.

**Corollary 4.5.** *In* $\mathfrak{N}_p$ *every* $\mathrm{FO}(\exists^\omega)$*-formula* $\varphi(x_0, \dots, x_{n-1})$ *is in* $[\exists^{10}\forall^{3n+10}]$ *and in* $[\forall^7\exists^{3n+10}]$.

*Proof.* We use the same idea as in the previous corollary but have to encode the run in only one variable $q$. Let $\mathfrak{A} = \big(Q, \mathbb{Z}_p^n, \Delta, \{0\}, \{m-1\}\big)$ be a non-deterministic automaton belonging to $\varphi$ with states $Q = \{0, \dots, m-1\}$. We store only every $(m+1)^{\text{th}}$ state in $q$. Thus we can use $m+1$ digits of $q$ for each state $k \in Q$. We encode $k$ as sequence $1^{k+1}0^{m-k}$. Note that there is always at least one 1 and one 0. First, we define a generalisation of $\mathrm{dig}_k(x, y)$ to sequences of digits.

$$\mathrm{digs}_{k_0 \dots k_{r-1}}(x, y) := \exists s \exists t \Big( x = s + \Big(\sum_{i<r} p^i k_i\Big) \cdot y + t \\ \wedge\ t < y\ \wedge\ p^r \cdot y \mid_p s \Big) \in [\exists^3]$$

Furthermore, we have $\mathrm{DEF} \in [\forall^2]$ and, using that

$$x < y \equiv \exists z(x + z = y \wedge x \neq y) \equiv \forall z(y + z \neq x) \in [\exists] \cap [\forall],$$

we obtain

$$\mathrm{START}(\overline{x}, q, s) := \mathrm{digs}_{10^m}(q, 1) \in [\exists^3],$$
$$\mathrm{ACC}(\overline{x}, q, s) := \mathrm{digs}_{1^m 0}(q, s) \in [\exists^3].$$

ADM has to ensure that $q$ is of the right form.

$$\mathrm{ADM}(\overline{x}, q, s) := \\ P_p s \wedge \bigwedge_{i<n} x_i < s \\ \wedge \neg\exists z \bigvee \big\{ \mathrm{dig}_w(q, z) \bigm| |w| = m+1 \text{ and } w \text{ is not a factor of } \\ 1^{i+1}0^{m-i}1^{j+1}0^{m-j} \text{ for all } i, j < m \big\} \in [\forall^4].$$

In order to define RUN we need a formula describing the effect of a sequence of $m+1$ transitions

$$\mathrm{Trans}_{(k, \overline{a}_0 \dots \overline{a}_m, k')}(\overline{x}, q, z) := \\ \mathrm{digs}_{1^{k+1}0^{m-k}}(q, z) \wedge \mathrm{digs}_{1^{k'+1}0^{m-k'}}(q, p^{m+1} \cdot z) \\ \wedge \bigwedge_{i<n} \mathrm{digs}_{(a_0)_i \dots (a_m)_i}(x_i, z) \in [\exists^{3n+6}],$$

and a formula defining those positions where the encoding of a state starts

$$\mathrm{POS}(q, z) := P_p z \wedge \bigvee_{w \in \{0,1\}^{m-1}} \mathrm{digs}_{1w0}(q, z) \in [\exists^3].$$

Let $\Delta^{m+1}$ denote the set of all tuples $(k, \overline{a}_0 \dots \overline{a}_m, k') \in Q \times \mathbb{Z}_p^{m+1} \times Q$ describing sequences of $m + 1$ transitions permitted by $\Delta$. Setting

$$\mathrm{RUN}(\overline{x}, q, s) := \forall z \Big( \mathrm{POS}(q, z) \to \bigwedge_{\tau \notin \Delta^{m+1}} \neg \mathrm{Trans}_\tau(\overline{x}, q, z) \Big) \in [\forall^{3n+10}]$$

we obtain

$$\exists q \exists s \exists 0 \exists 1 [\mathrm{DEF} \wedge \mathrm{ADM} \wedge \mathrm{START} \wedge \mathrm{RUN} \wedge \mathrm{ACC}] \in [\exists^4 \exists^3 \exists^3 \forall^{3n+10}],$$
$$\forall q \forall s \forall 0 \forall 1 [\mathrm{DEF} \wedge \mathrm{ADM} \wedge \mathrm{START} \wedge \mathrm{RUN} \to \mathrm{ACC}] \in [\forall^4 \forall^3 \exists^{3n+10}].$$

□

## 4.2 $\omega$-Languages

In this and the following sections we repeat the program of the last one for, respectively, $\omega$-, tree, and $\omega$-tree languages. In the case of $\omega$-languages the structures corresponding to $\mathfrak{N}_p$ and $\mathfrak{W}(\Sigma)$ are

$$\mathfrak{R}_p := (\mathbb{R}, +, \leq, |_p, 1) \qquad \text{and} \qquad \mathfrak{W}^\omega(\Sigma) := (\Sigma^{\leq\omega}, (\sigma_a)_{a\in\Sigma}, \preceq, \mathrm{el}),$$

where $+$, $\leq$, and 1 have their usual meaning, $p \in \mathbb{N} \setminus \{0, 1\}$, and

$$\begin{aligned}
x \,|_p\, y \quad &:\text{iff } \exists n, k \in \mathbb{Z} : x = p^n \text{ and } y = kx, \\
\sigma_a(x) \quad &:= \begin{cases} xa & \text{if } x \in \Sigma^*, \\ x & \text{if } x \in \Sigma^\omega, \end{cases} \\
x \preceq y \quad &:\text{iff } \exists z : xz = y, \\
\mathrm{el}(x, y) \quad &:\text{iff } |x| = |y| .
\end{aligned}$$

Again, the first step is to prove their equivalence. In order to simplify one direction we additionally introduce the structure $\mathfrak{R}_p^+ := (\mathbb{R}^{\geq 0}, +, |_p, 1)$.

What makes matters slightly more complicated in the case of reals is the fact that some real numbers have two encodings. For instance, in base 10 the numbers $0.999\dots$ and $1.000\dots$ are the same. The first case is called the *low encoding*, the second the *high encoding*.

**Proposition 4.6.** $\mathfrak{R}_p \rightleftharpoons_{\mathrm{FO}} \mathfrak{R}_p^+ \rightleftharpoons_{\mathrm{FO}} \mathfrak{W}^\omega(\mathbb{Z}_p)$.

*Proof.* ($\mathfrak{R}_p \leq_{\mathrm{FO}} \mathfrak{R}_p^+$) The interpretation represents non-negative numbers $x \in \mathbb{R}$ by the pair $(0, x)$ and non-positive numbers $x$ by $(1, -x)$.

$$\begin{aligned}
\delta(\overline{x}) \quad &:= x_0 = 0 \vee x_0 = 1, \\
\varepsilon(\overline{x}, \overline{y}) \quad &:= (x_0 = y_0 \wedge x_1 = y_1) \vee (x_1 = 0 \wedge y_1 = 0), \\
\varphi_1(\overline{x}) \quad &:= x_0 = 0 \wedge x_1 = 1, \\
\varphi_{|_p}(\overline{x}, \overline{y}) &:= x_0 = 0 \wedge x_1 \,|_p\, y_1, \\
\varphi_\leq(\overline{x}, \overline{y}) &:= (x_0 = 1 \wedge y_0 = 0) \vee (x_1 = 0 \wedge y_1 = 0) \\
&\quad \vee (x_0 = 0 \wedge y_0 = 0 \wedge \exists z (x_1 + z = y_1)) \\
&\quad \vee (x_0 = 1 \wedge y_0 = 1 \wedge \exists z (y_1 + z = x_1)).
\end{aligned}$$

To define addition we have to handle each combination of signs separately.

$$\begin{aligned}\varphi_+(\overline{x},\overline{y},\overline{z}) := {} & (x_0 = 0 \wedge y_0 = 0 \wedge z_0 = 0 \wedge x_1 + y_1 = z_1) \\ & \vee (x_0 = 1 \wedge y_0 = 1 \wedge z_0 = 1 \wedge x_1 + y_1 = z_1) \\ & \vee (x_0 = 0 \wedge y_0 = 1 \wedge z_0 = 0 \wedge x_1 = y_1 + z_1) \\ & \vee (x_0 = 1 \wedge y_0 = 0 \wedge z_0 = 1 \wedge x_1 = y_1 + z_1) \\ & \vee (x_0 = 1 \wedge y_0 = 0 \wedge z_0 = 0 \wedge y_1 = x_1 + z_1) \\ & \vee (x_0 = 0 \wedge y_0 = 1 \wedge z_0 = 1 \wedge y_1 = x_1 + z_1)\end{aligned}$$

($\mathfrak{R}_p^+ \leq_{\mathrm{FO}} \mathfrak{W}^\omega(\mathbb{Z}_p)$) We represent a number $\sum_i m_i p^i$ (in high encoding) by the pair $(m_0 \dots m_r, m_{-1}m_{-2}\dots)$ and define, using the same abbreviations as in the previous section,

$$\begin{aligned}&\mathrm{Inf}(x) := \forall y(x \preceq y \rightarrow x = y), && \varepsilon \ : \forall x(\varepsilon \preceq x), \\ &\mathrm{Fin}(x) := \neg\mathrm{Inf}(x), && 0^\omega : \mathrm{Inf}(0^\omega) \wedge \forall x\, \mathrm{dig}_0(0^\omega, x).\end{aligned}$$

The universe of the interpretation consists of all pairs whose fractional part does not end with $(p-1)^\omega$.

$$\delta(\overline{x}) := \mathrm{Fin}(x_0) \wedge \mathrm{Inf}(x_1) \wedge \neg\exists y \forall z(|z| > |y| \rightarrow \mathrm{dig}_{p-1}(x_1, z))$$

Two pairs are equal if their fractional parts are identical and their integer parts differ only by the number of initial zeros.

$$\begin{aligned}\varepsilon(\overline{x},\overline{y}) &:= x_1 = y_1 \wedge \forall z \bigwedge_{k<p} (\mathrm{dig}_k(x_0, z) \leftrightarrow \mathrm{dig}_k(y_0, z)) \\ \varphi_1(\overline{x}) &:= \varepsilon(\overline{x}, (\sigma_1\varepsilon, 0^\omega))\end{aligned}$$

For $x \mid_p y$ we have to check whether $x$ is an integer or less than 1, and handle both cases separately.

$$\begin{aligned}\varphi_{|_p}(\overline{x},\overline{y}) &:= [x_1 = y_1 = 0^\omega \wedge \psi^1_{|_p}(x_0, y_0)] \vee [x_0 \preceq 0^\omega \wedge \psi^2_{|_p}(x_1, y_1)] \\ \psi^1_{|_p}(x,y) &:= \exists z\big(\mathrm{dig}_1(x,z) \wedge \forall z'(|z'| \neq |z| \rightarrow \mathrm{dig}_0(x,z')) \wedge \\ &\qquad\qquad \forall z'(|z'| < |z| \rightarrow \mathrm{dig}_0(y,z'))\big) \\ \psi^2_{|_p}(x,y) &:= \exists z\big(\mathrm{dig}_1(x,z) \wedge \forall z'(|z'| \neq |z| \rightarrow \mathrm{dig}_0(x,z')) \wedge \\ &\qquad\qquad \forall z'(|z'| > |z| \rightarrow \mathrm{dig}_0(y,z'))\big)\end{aligned}$$

Unsurprisingly, addition is the most complicated part. Again there has to be a number $\overline{u}$ encoding the carry.

$$\begin{aligned}\varphi_+(\overline{x},\overline{y},\overline{z}) := \exists\overline{u}[\delta(\overline{u}) &\wedge \forall v(\mathrm{dig}_0(u_0, v) \vee \mathrm{dig}_1(u_0, v)) \\ &\wedge \forall v(\mathrm{dig}_0(u_1, v) \vee \mathrm{dig}_1(u_1, v)) \\ &\wedge \psi^1_+(\overline{x},\overline{y},\overline{z},\overline{u}) \wedge \psi^2_+(\overline{x},\overline{y},\overline{z},\overline{u})]\end{aligned}$$

$\psi^1_+$ and $\psi^2_+$ handle, respectively, the integer and fractional part of the addition and check whether each digit is correct using the set $A$ defined above.

$$\psi^1_+(\overline{x},\overline{y},\overline{z},\overline{u}) := \forall v \bigvee_{(a,b,c,d,d')\in A} \big(\mathrm{dig}_a(x_0,v) \land \mathrm{dig}_b(y_0,v) \land \mathrm{dig}_c(z_0,v) \land \mathrm{dig}_d(u_0,v) \land \mathrm{dig}_{d'}(u_0,\sigma_0 v)\big)$$

$$\psi^2_+(\overline{x},\overline{y},\overline{z},\overline{u}) := \forall v \bigvee_{(a,b,c,d,d')\in A} \big(\mathrm{dig}_a(x_1,v) \land \mathrm{dig}_b(y_1,v) \land \mathrm{dig}_c(z_1,v) \land \mathrm{dig}_d(u_1,v) \land [\exists s(|v| = |s| + 1 \land \mathrm{dig}_{d'}(u_1,s)) \lor (v = \varepsilon \land \mathrm{dig}_{d'}(u_0,\varepsilon))]\big)$$

($\mathfrak{W}^\omega(\mathbb{Z}_p) \leq_{\mathrm{FO}} \mathfrak{R}_p$) Finite words $m_1 \dots m_r \in \mathbb{Z}_p^*$ are encoded by the number

$$p^{-r+1} + \sum_{i=1}^{r} m_i p^{-i} + 2 \in [2,3].$$

We cannot just map infinite words $m_1 m_2 \dots \in \mathbb{Z}_p^\omega$ to $\sum_i m_i p^{-i} \in [0,1]$ because, e.g., the words $0(p-1)^\omega$ and $10^\omega$ would be mapped to the same number. Therefore we choose the encoding as

$$\pm \sum_i m_i p^{-i} \in [-1,1]$$

such that numbers in $[0,1]$ encode the word corresponding to their high encoding and numbers in $[-1,0]$ encode words corresponding to the low encoding of their absolute value. This results in most words having two encodings. Set

$$\begin{aligned}
\mathrm{LastDigit}(x,y) &:= y \mid_p x \land p \cdot y \nmid_p x,\\
\mathrm{Inf}(x) &:= -1 \leq x \leq 1,\\
\mathrm{Fin}(x) &:= 2 \leq x \leq 3 \ \land\ \exists y(\mathrm{LastDigit}(x,y) \land p \cdot y \mid_p x - y),\\
\mathrm{Ambig}(x) &:= \mathrm{Inf}(x) \land \neg\exists y\, \mathrm{LastDigit}(x,y).
\end{aligned}$$

We obtain the interpretation

$$\begin{aligned}
\delta(x) &:= \mathrm{Inf}(x) \lor \mathrm{Fin}(x),\\
\varepsilon(x,y) &:= x = y \lor [\mathrm{Ambig}(x) \land \mathrm{Ambig}(y) \land x = -y],\\
\varphi_{\sigma_i}(x,y) &:= [\mathrm{Inf}(x) \land \varepsilon(x,y)] \lor\\
&\quad\ [\mathrm{Fin}(x) \land \mathrm{Fin}(y) \land \exists z(\mathrm{LastDigit}(x,z) \land \mathrm{LastDigit}(y,z/p)\\
&\qquad\qquad \land y = x - z + i \cdot z + z/p)],\\
\varphi_{\mathrm{el}}(x,y) &:= [\mathrm{Inf}(x) \land \mathrm{Inf}(y)] \lor\\
&\quad\ [\mathrm{Fin}(x) \land \mathrm{Fin}(y) \land \exists z(\mathrm{LastDigit}(x,z) \land \mathrm{LastDigit}(y,z))],\\
\varphi_{\preceq}(x,y) &:= \varepsilon(x,y)\\
&\quad \lor \big[\mathrm{Fin}(x) \land \exists z\big(\mathrm{LastDigit}(x,z) \land [(\mathrm{Inf}(y) \land \psi^1_{\preceq}(x,y,z)) \lor\\
&\qquad\qquad (\mathrm{Fin}(y) \land \psi^2_{\preceq}(x,y,z))]\big)\big],\\
\psi^1_{\preceq}(x,y,z) &:= [(0 \leq y \lor \mathrm{Ambig}(y)) \ \land\ 0 \leq |y| - (x-z) < p \cdot z] \lor\\
&\quad\ [y < 0 \land \neg\mathrm{Ambig}(y) \ \land\ 0 < |y| - (x-z) \leq p \cdot z],\\
\psi^2_{\preceq}(x,y,z) &:= \exists z'(\mathrm{LastDigit}(y,z') \ \land\ 0 \leq (y-z') - (x-z) < p \cdot z).
\end{aligned}$$

□

This time we use $\mathfrak{W}^\omega(\Sigma)$, mainly because the construction of an $\omega$-automatic presentation of $\mathfrak{N}_p$ is quite involved. (See [BRW98] for a similar result.)

**Theorem 4.7.** *$R \subseteq (\Sigma^\omega)^n$ is* FO*-definable in $\mathfrak{W}^\omega(\Sigma)$ if and only if* fold$(R)$ *is $\omega$-regular.*

*Proof.* W.l.o.g. assume $\Sigma = \mathbb{Z}_p$ for some $p > 1$.

(⇒) Set

$$\mathbb{Z}_{p\Box} := \mathbb{Z}_p \cup \{\Box\}, \qquad \mathrm{id} := \{[\tfrac{i}{i}] \mid i \in \mathbb{Z}_p\}, \qquad \mathrm{id}_\Box := \{[\tfrac{i}{i}] \mid i \in \mathbb{Z}_{p\Box}\}.$$

The desired presentation of $\mathfrak{W}^\omega(\Sigma)$ is

$$\mathfrak{d} := \big(\mathrm{id}, \mathbb{Z}_{p\Box}, L_\delta, L_\varepsilon, (L_{\sigma_i})_{i<p}, L_\preceq, L_{\mathrm{el}}\big)$$

where

$$L_\delta := \mathbb{Z}_p^*\Box^\omega \cup \mathbb{Z}_p^\omega, \qquad L_\preceq := L_\delta^{\otimes 2} \cap \big[\mathrm{id}^\omega \cup \mathrm{id}^*\{[\tfrac{\Box}{i}] \mid i \in \mathbb{Z}_{p\Box}\}^\omega\big],$$
$$L_\varepsilon := L_\delta^{\otimes 2} \cap \mathrm{id}_\Box^\omega, \qquad L_{\sigma_i} := L_\delta^{\otimes 2} \cap \big[\mathrm{id}^\omega \cup \mathrm{id}^*\,[\tfrac{\Box}{i}]\,[\tfrac{\Box}{\Box}]^\omega\big],$$
$$L_{\mathrm{el}} := (\mathbb{Z}_p^2)^\omega \cup (\mathbb{Z}_p^2)^*\,[\tfrac{\Box}{\Box}]^\omega.$$

(⇐) The proof is analogous to the one above. Let $\mathfrak{A} = (Q, \mathbb{Z}_p^n, \Delta, \overline{0}, \mathscr{F})$ with $Q = \mathbb{Z}_p^m$ be a Muller-automaton which recognises fold$(R)$. We construct a formula $\psi_\mathfrak{A}$ defining $R$.

$$\psi_\mathfrak{A}(x_0, \ldots, x_{n-1}) := \exists q_0 \cdots \exists q_{m-1}[\mathrm{ADM}(\overline{q}, \overline{x}) \wedge \mathrm{START}(\overline{q}, \overline{x}) \wedge \mathrm{RUN}(\overline{q}, \overline{x}) \wedge \mathrm{ACC}(\overline{q}, \overline{x})]$$

with

$$\begin{aligned}
\mathrm{Inf}(x) &:= \forall y(x \preceq y \to x = y),\\
\mathrm{Sym}_{\overline{a}}(\overline{x}, z) &:= \bigwedge_i \mathrm{dig}_{a_i}(x_i, z),\\
\mathrm{ADM}(\overline{q}, \overline{x}) &:= \bigwedge_{i<m} \mathrm{Inf}(q_i) \wedge \bigwedge_{i<n} \mathrm{Inf}(x_i),\\
\mathrm{START}(\overline{q}, \overline{x}) &:= \mathrm{Sym}_{\overline{0}}(\overline{q}, \varepsilon),\\
\mathrm{RUN}(\overline{q}, \overline{x}) &:= \\
&\forall z \bigvee_{(\overline{k}, \overline{a}, \overline{k}') \in \Delta} (\mathrm{Sym}_{\overline{k}}(\overline{q}, z) \wedge \mathrm{Sym}_{\overline{a}}(\overline{x}, z) \wedge \mathrm{Sym}_{\overline{k}'}(\overline{q}, \sigma_0 z)),\\
\mathrm{ACC}(\overline{q}, \overline{x}) &:= \bigvee_{F \in \mathscr{F}} \Big( \bigwedge_{\overline{k} \in F} \forall z \exists z'(|z'| > |z| \wedge \mathrm{Sym}_{\overline{k}}(\overline{q}, z'))\\
&\qquad \wedge \bigwedge_{\overline{k} \notin F} \neg\forall z \exists z'(|z'| > |z| \wedge \mathrm{Sym}_{\overline{k}}(\overline{q}, z'))\Big).
\end{aligned}$$

□

## 4.3 Tree Languages

Let $R$ be a ring and $M$ a monoid. The semiring $R\langle\langle M \rangle\rangle$ of formal power series over $M$ consists of all maps $r : M \to R$. We write $(r, m)$ for the value of $m$

under $r$. Addition, product, and Hadamard product are defined as

$$\begin{aligned}(r_1 + r_2, m) &:= (r_1, m) + (r_2, m),\\ (r_1 \cdot r_2, m) &:= \sum_{m_1 \cdot m_2 = m} (r_1, m_1) \cdot (r_2, m_2),\\ (r_1 \odot r_2, m) &:= (r_1, m) \cdot (r_2, m).\end{aligned}$$

Note that the product is undefined if the sum diverges. We denote by $R\langle M\rangle$ the semiring of formal polynomials over $M$, i.e., power series $r$ with $(r, m) = 0$ for all but a finite number of $m$.

In this section we consider the structures

$$\mathfrak{P}_p := \big(\mathbb{Z}_p\langle\{X, Y\}^*\rangle, +, \odot, \cdot X, \cdot Y\big) \quad \text{and} \quad \mathfrak{T}_p := (T, +, \cdot, s_0, s_1)$$

where, for $p \in \mathbb{N} \setminus \{0, 1\}$, $\mathfrak{P}_p$ is the semiring of formal polynomials in two non-commuting variables with addition, Hadamard product and right-multiplication by the variables, and

$$\begin{aligned} T &:= \big\{\, t \in T^{\omega}_{\mathbb{Z}_p} \bigm| t^{-1}(i) \text{ is finite for all } i \neq 0 \,\big\},\\ (t_1 + t_2)(x) &:= t_1(x) + t_2(x) && \text{for all } x \in \{0,1\}^*,\\ (t_1 \cdot t_2)(x) &:= t_1(x) \cdot t_2(x) && \text{for all } x \in \{0,1\}^*,\\ (s_i t)(x) &:= t(xi) && \text{for } i \in \{0,1\}.\end{aligned}$$

**Proposition 4.8.** $\mathfrak{P}_p =_{\mathrm{FO}} \mathfrak{T}_p$.

*Proof.* Note that each tree $t : \mathrm{dom}(t) \to \mathbb{Z}_p$ can be regarded as a formal polynomial in $\mathbb{Z}_p\langle\{0, 1\}^*\rangle$. Hence, both structures are nearly isomorphic but for the definition of $s_i$ and $\cdot X$, where the arguments are reversed. $s_0 t = r$ iff $r \cdot X = t$. □

We encode each $t \in T_\Sigma$ as tree in $\mathfrak{T}_p$ by marking its frontier with 1's. Formally

$$\mathrm{code}(t) := \begin{cases} t(x) & \text{if } x \in \mathrm{dom}(t),\\ 1 & \text{if } x \notin \mathrm{dom}(t),\ x = yi,\ i \in \{0,1\} \text{ and } y \in \mathrm{dom}(t),\\ 0 & \text{otherwise.}\end{cases}$$

**Theorem 4.9.** *Let $R \subseteq (T_\Sigma)^n$. $\mathrm{code}(R)$ is FO-definable in $\mathfrak{T}_{|\Sigma|}$ if and only if $\mathrm{fold}(R)$ is recognisable.*

*Proof.* (⇒) A presentation of $\mathfrak{T}_p$ is given by

$$\mathfrak{d} := \big(\mathrm{id}, \mathbb{Z}_p, T_{\mathbb{Z}_p}, L(\mathfrak{A}_\varepsilon), L(\mathfrak{A}_+), L(\mathfrak{A}_\cdot), L(\mathfrak{A}_{s_0}), L(\mathfrak{A}_{s_1})\big)$$

where

$$\begin{aligned}\mathfrak{A}_\varepsilon &:= \big(\{q_0\}, \mathbb{Z}_p^2, \Delta_\varepsilon, \{q_0\}\big),\\ \mathfrak{A}_\oplus &:= \big(\{q_0\}, \mathbb{Z}_p^3, \Delta_\oplus, \{q_0\}\big),\\ \mathfrak{A}_{s_i} &:= \big(\{q_0, \dots, q_{p-1}\}, \mathbb{Z}_p^2, \Delta_i, \{q_0, \dots, q_{p-1}\}\big)\end{aligned}$$

for $\oplus \in \{\cdot, +\}$ and $i \in \{0,1\}$ with

$$\begin{aligned}
\Delta_\varepsilon &:= \big\{(q_0,(a,a),q,q') \mid a \in \mathbb{Z}_p, q,q' \in \{q_0,\Box\}\big\},\\
\Delta_\oplus &:= \big\{(q_0,(a_1,a_2,a_1 \oplus a_2),q,q') \mid a_1,a_2 \in \mathbb{Z}_p, q,q' \in \{q_0,\Box\}\big\}\\
&\quad\cup \big\{(q_0,(a,\Box,a\oplus 0),q,q'),(q_0,(\Box,a,0\oplus a),q,q') \mid\\
&\qquad\qquad a \in \mathbb{Z}_p, q,q' \in \{q_0,\Box\}\big\},\\
\Delta_0 &:= \big\{(q_a,(a,b),q_b,q) \mid a,b \in \mathbb{Z}_p, q \in \{q_0,\dots,q_{p-1},\Box\}\big\}\\
&\quad\cup \big\{(q_a,(a,\Box),\Box,q) \mid a \in \mathbb{Z}_p, q \in \{q_0,\dots,q_{p-1},\Box\}\big\},\\
\Delta_1 &:= \big\{(q_a,(a,b),q,q_b) \mid a,b \in \mathbb{Z}_p, q \in \{q_0,\dots,q_{p-1},\Box\}\big\}\\
&\quad\cup \big\{(q_a,(a,\Box),q,\Box) \mid a \in \mathbb{Z}_p, q \in \{q_0,\dots,q_{p-1},\Box\}\big\}.
\end{aligned}$$

($\Leftarrow$) First we define some auxiliary formulae. The 0-tree is defined by $0+0=0$. The $m$-fold product of some tree $t$ is

$$0t := 0, \qquad mt := t + \dots + t.$$

In order to access the nodes of a tree we use trees containing a single node labelled by 1.

$$\begin{aligned}
\text{SingleNonZero}(t) &:= \forall s \bigvee_{i<p} t \cdot s = it,\\
\text{SingleOne}(t) &:= \text{SingleNonZero}(t) \wedge \exists^{=p} s(t \cdot s = s).
\end{aligned}$$

The root is

$$\text{Root}(t) := \text{SingleOne}(t) \wedge s_0 t = 0 \wedge s_1 t = 0,$$

and the successors of some node are defined by

$$\begin{aligned}
\text{Succ}_0(s,t) &:= \text{SingleOne}(s) \wedge \text{SingleOne}(t) \wedge s_0 t = s,\\
\text{Succ}_1(s,t) &:= \text{SingleOne}(s) \wedge \text{SingleOne}(t) \wedge s_1 t = s,\\
\text{Succ}(s,t_0,t_1) &:= \text{Succ}_0(s,t_0) \wedge \text{Succ}_1(s,t_1).
\end{aligned}$$

Additionally, we need a formula characterising those trees all of whose nodes are either labelled with 1 and posses at least one child also labelled with 1, or are labelled with 0

$$\begin{aligned}
\text{Inf}(t) := {}& \forall r[\text{SingleOne}(r) \to (t \cdot r = r \vee t \cdot r = 0)]\\
&\wedge \forall r \forall s_0 \forall s_1 [\text{Succ}(r,s_0,s_1) \to (t \cdot r = 0 \ \vee\ t \cdot s_0 \neq 0 \ \vee\ t \cdot s_1 \neq 0)],
\end{aligned}$$

and a formula defining those positions of a tree whose successors all are labelled with 0

$$\begin{aligned}
\text{Box}(s,t) := {}& \big[s \cdot t = 0 \wedge \exists r(\text{Inf}(r) \ \wedge\ r \cdot t = t \ \wedge\ r \cdot s = 0)\big]\\
&\vee \big[s \cdot t = t \wedge \forall v(\,\text{Succ}_0(t,v) \vee \text{Succ}_1(t,v) \to\\
&\qquad \exists r(\text{Inf}(r) \wedge r \cdot v = v \wedge r \cdot s = 0))\big].
\end{aligned}$$

We construct a formula stating that the tree automaton $\mathfrak{A} = (Q, \mathbb{Z}_p^n, \Delta, F)$ with $Q = \mathbb{Z}_p^m$ accepts some tuple $(t_0, \dots, t_{n-1}) \in T_{\mathbb{Z}_p}^n$.

$$\psi_{\mathfrak{A}}(t_0, \dots, t_{n-1}) := \exists q_0 \cdots \exists q_{m-1}[\mathrm{RUN}(\overline{q}, \overline{t}) \wedge \mathrm{ACC}(\overline{q}, \overline{t})]$$

where

$$\begin{aligned}
\mathrm{Sym}_{\overline{a}}(\overline{x}, r) &:= \bigwedge_{i:\ a_i \neq \square} (x_i \cdot r = a_i r \wedge \neg\mathrm{Box}(x_i, r)) \wedge \bigwedge_{i:\ a_i = \square} \mathrm{Box}(t_i, r), \\
\mathrm{RUN}(\overline{q}, \overline{t}) &:= \\
&\forall r \forall s_0 \forall s_1 \Big(\mathrm{Succ}(r, s_0, s_1) \rightarrow \\
&\bigvee_{(\overline{k}, \overline{a}, \overline{k}_0, \overline{k}_1) \in \Delta} \big(\mathrm{Sym}_{\overline{k}}(\overline{q}, r) \wedge \mathrm{Sym}_{\overline{a}}(\overline{t}, r) \wedge \mathrm{Sym}_{\overline{k}_0}(\overline{q}, s_0) \wedge \mathrm{Sym}_{\overline{k}_1}(\overline{q}, s_1)\big)\Big), \\
\mathrm{ACC}(\overline{q}, \overline{t}) &:= \exists r \Big(\mathrm{Root}(r) \wedge \bigvee_{\overline{k} \in F} \mathrm{Sym}_{\overline{k}}(\overline{q}, r)\Big).
\end{aligned}$$

□

## 4.4 ω-Tree Languages

This last section holds no surprises. A bored reader may skip it without missing anything. The structures are

$$\mathfrak{P}_p^\omega := \big(\mathbb{Z}_p\langle\langle \{X, Y\}^* \rangle\rangle, +, \odot, \cdot X, \cdot Y\big) \quad \text{and} \quad \mathfrak{T}_p^\omega := (T_{\mathbb{Z}_p}^\omega, +, \cdot, s_0, s_1)$$

where, $p \in \mathbb{N} \setminus \{0, 1\}$, $\mathfrak{P}_p$ is the semiring of formal power series in two non-commuting variables with addition, Hadamard product and right-multiplication by the variables, and

$$\begin{aligned}
(t_1 + t_2)(x) &:= t_1(x) + t_2(x) && \text{for all } x \in \{0,1\}^*, \\
(t_1 \cdot t_2)(x) &:= t_1(x) \cdot t_2(x) && \text{for all } x \in \{0,1\}^*, \\
(s_i t)(x) &:= t(xi) && \text{for } i \in \{0,1\}.
\end{aligned}$$

**Proposition 4.10.** $\mathfrak{P}_p^\omega =_{\mathrm{FO}} \mathfrak{T}_p^\omega$.

*Proof.* same as above. □

**Theorem 4.11.** *$R \subseteq (T_\Sigma^\omega)^n$ is* FO*-definable in $\mathfrak{T}_{|\Sigma|}^\omega$ if and only if* fold$(R)$ *is recognisable.*

*Proof.* (⇒) The desired presentation of $\mathfrak{T}_p^\omega$ is

$$\mathfrak{d} := \big(\mathrm{id}, \mathbb{Z}_p, T_{\mathbb{Z}_p}^\omega, L(\mathfrak{A}_\varepsilon), L(\mathfrak{A}_+), L(\mathfrak{A}_\cdot), L(\mathfrak{A}_{s_0}), L(\mathfrak{A}_{s_1})\big)$$

where

$$\begin{aligned}
\mathfrak{A}_\varepsilon &:= \big(\{q_0\}, \mathbb{Z}_p^2, \Delta_\varepsilon, \{q_0\}, \{q_0\}\big), \\
\mathfrak{A}_\oplus &:= \big(\{q_0\}, \mathbb{Z}_p^3, \Delta_\oplus, \{q_0\}, \{q_0\}\big), \\
\mathfrak{A}_{s_i} &:= \big(\{q_0, \dots, q_{p-1}\}, \mathbb{Z}_p^2, \Delta_i, \{q_0, \dots, q_{p-1}\}, \mathscr{P}(\{q_0, \dots, q_{p-1}\})\big)
\end{aligned}$$

for $\oplus \in \{\cdot, +\}$ and $i \in \{0,1\}$ with

$$\begin{aligned}
\Delta_\varepsilon &:= \big\{(q_0, (a,a), q_0, q_0) \bigm| a \in \mathbb{Z}_p\big\},\\
\Delta_\oplus &:= \big\{(q_0, (a_1, a_2, a_1 \oplus a_2), q_0, q_0) \bigm| a_1, a_2 \in \mathbb{Z}_p\big\},\\
\Delta_0 &:= \big\{(q_a, (a,b), q_b, q) \bigm| a, b \in \mathbb{Z}_p, q \in \{q_0, \dots, q_{p-1}\}\big\},\\
\Delta_1 &:= \big\{(q_a, (a,b), q, q_b) \bigm| a, b \in \mathbb{Z}_p, q \in \{q_0, \dots, q_{p-1}\}\big\}.
\end{aligned}$$

($\Leftarrow$) Using the same auxiliary formulae as in the case of finite trees we construct a formula stating that the $\omega$-tree automaton $\mathfrak{A} = (Q, \mathbb{Z}_p^n, \Delta, Q_0, \mathscr{F})$ with $Q = \mathbb{Z}_p^m$ accepts some tuple $(t_0, \dots, t_{n-1}) \in (T^\omega_{\mathbb{Z}_p})^n$.

$$\psi_{\mathfrak{A}}(t_0, \dots, t_{n-1}) := \exists q_0 \cdots \exists q_{m-1}[\mathrm{START}(\overline{q}, \overline{t}) \wedge \mathrm{RUN}(\overline{q}, \overline{t}) \wedge \mathrm{ACC}(\overline{q}, \overline{t})]$$

where

$$\begin{aligned}
\mathrm{Sym}_{\overline{a}}(\overline{x}, r) \quad &:= \bigwedge_i x_i \cdot r = a_i r,\\
\mathrm{START}(\overline{q}, \overline{t}) &:= \exists r\Big(\mathrm{Root}(r) \wedge \bigvee_{\overline{k} \in Q_0} \mathrm{Sym}_{\overline{k}}(\overline{q}, r)\Big),
\end{aligned}$$

$$\begin{aligned}
&\mathrm{RUN}(\overline{q}, \overline{t}) :=\\
&\quad \forall r \forall s_0 \forall s_1 \Big(\mathrm{Succ}(r, s_0, s_1) \to\\
&\quad \bigvee_{(\overline{k}, \overline{a}, \overline{k}_0, \overline{k}_1) \in \Delta} \big(\mathrm{Sym}_{\overline{k}}(\overline{q}, r) \wedge \mathrm{Sym}_{\overline{a}}(\overline{t}, r) \wedge \mathrm{Sym}_{\overline{k}_0}(\overline{q}, s_0) \wedge \mathrm{Sym}_{\overline{k}_1}(\overline{q}, s_1)\big)\Big),\\
&\mathrm{ACC}(\overline{q}, \overline{t}) :=\\
&\quad \bigvee_{F \in \mathscr{F}} \Big( \bigwedge_{\overline{k} \in F} \forall r[\mathrm{Inf}(r) \to \exists s(\mathrm{SingleOne}(s) \wedge r \cdot s = s \wedge \mathrm{Sym}_{\overline{k}}(\overline{q}, s)]\\
&\qquad \wedge \bigwedge_{\overline{k} \notin F} \neg \forall r[\mathrm{Inf}(r) \to \exists s(\mathrm{SingleOne}(s) \wedge r \cdot s = s \wedge \mathrm{Sym}_{\overline{k}}(\overline{q}, s)]\Big).
\end{aligned}$$

□

# Chapter 5

# Classes of Automatic Structures

We are now ready to investigate the four classes of automatic structures. After developing tools to obtain negative results and looking at the closure of [$\omega$-][T]AutStr under certain products we will determine the relationship between them.

## 5.1 Growth Rates and Length Sequences

So far, our only tool to prove that some structure is not automatic was to show that its theory is undecidable. In this section we develop another method which unfortunately is only applicable in case of AutStr. The arguments used are slight generalisations of a result of Khoussainov and Nerode [KN95, Lemma 4.5].

When trying to show that a structure has no automatic presentation one suffers from the lack of knowledge about how elements are encoded. If such information were available one could use standard techniques from formal language theory to prove non-regularity. So far, the best we can do is to give bounds on the length of the encoding of some element.

**Proposition 5.1** (implicit in [KN95, Lemma 4.5]). *Let $\mathfrak{A} \in$ AutStr, $\mathfrak{d}$ an injective presentation of $\mathfrak{A}$, and let $f : A^n \to A$ be a function of $\mathfrak{A}$. Then there is a constant $m$ such that for all $\overline{a} \in A^n$*

$$\lambda^{\mathfrak{d}}(f(\overline{a})) \leq m + \max\{\lambda^{\mathfrak{d}}(a_0), \dots, \lambda^{\mathfrak{d}}(a_{n-1})\}.$$

*Proof.* As $\mathfrak{d}$ is injective there is a single word $w$ encoding the value of $f(\overline{a})$. Let $m$ be the number of states in the automaton recognising the graph of $f$. Suppose that $w$ is more than $m$ symbols longer than the encoding of each argument. Then the automaton recognises a word of the form

$$((\Sigma \cup \{\Box\})^n \times \Sigma)^*(\{\Box\}^n \times \Sigma)^{m+1}.$$

As there has to be a repetition of states in the suffix of this word the automaton recognises infinitely many words with the same prefix. But this prefix completely contains the arguments of the function so the image of $\overline{a}$ has infinitely many representations. Contradiction. $\Box$

**Corollary 5.2.** *Let* $\mathfrak{A} \in \mathrm{AutStr}$*,* $\mathfrak{d}$ *an injective presentation of* $\mathfrak{A}$*, and let* $R \subseteq A^{n+k}$ *be a relation of* $\mathfrak{A}$ *such that for all* $\overline{a} \in A^n$ *the number of* $\overline{b} \in A^k$ *with* $(\overline{a}, \overline{b}) \in R$ *is finite. Then there is a constant* $m$ *such that for all* $(\overline{a}, \overline{b}) \in R$

$$\max\{\lambda^{\mathfrak{d}}(b_0), \dots, \lambda^{\mathfrak{d}}(b_{k-1})\} \leq m + \max\{\lambda^{\mathfrak{d}}(a_0), \dots, \lambda^{\mathfrak{d}}(a_{n-1})\}.$$

*Proof.* Define the function $f : A^n \to A$ by

$$\begin{aligned} f(\overline{a}) = c \ :\text{iff} \ \ &\exists \overline{b}(R\overline{a}\overline{b} \wedge \text{“}c \text{ appears in } \overline{b}\text{”}) \\ &\wedge \forall \overline{b}\Big(R\overline{a}\overline{b} \to \bigwedge_{i<k} \lambda^{\mathfrak{d}}(b_i) \leq \lambda^{\mathfrak{d}}(c)\Big). \end{aligned}$$

By assumption on $R$ $f$ is well-defined, and it should be clear that there is some automaton recognising the graph of $f$. Therefore the result follows from the preceding proposition. □

In the case of Presburger Arithmetic Proposition 5.1 seems to indicate that we do not have much choice with regard to the encoding.

**Lemma 5.3.** *For any automatic presentation* $\mathfrak{d}$ *of Presburger Arithmetic we have* $\lambda^{\mathfrak{d}}(n) \in \Theta(\log n)$*.*

*Proof.* The lower bound immediately follows from the fact that there are only $|\Sigma|^n$ strings of length $n$ over $\Sigma$. To prove the upper bound we show by induction on $n$ that

$$\lambda^{\mathfrak{d}}(n) \leq m\lceil \log_2 n \rceil + \lambda^{\mathfrak{d}}(1)$$

where $m$ is the constant from the previous lemma.

($n = 1$) $\lambda^{\mathfrak{d}}(1) \leq m\lceil \log_2 1 \rceil + \lambda^{\mathfrak{d}}(1)$.

($n > 1$) Set $k = \lceil \log_2 n \rceil$. Then $n = 2^{k-1} + (n - 2^{k-1})$ and we obtain from the previous lemma and the induction hypothesis

$$\begin{aligned} \lambda^{\mathfrak{d}}(n) &= \lambda^{\mathfrak{d}}(2^{k-1} + (n - 2^{k-1})) \\ &\leq m + \max\{\lambda^{\mathfrak{d}}(2^{k-1}), \lambda^{\mathfrak{d}}(n - 2^{k-1})\} \\ &\leq m + m(k-1) + \lambda^{\mathfrak{d}}(1) \\ &= m\lceil \log_2 n \rceil + \lambda^{\mathfrak{d}}(1). \end{aligned}$$

□

Corollary 5.2 can be paraphrased such that it yields lower bounds.

**Corollary 5.4.** *Let* $\mathfrak{A} \in \mathrm{AutStr}$*,* $\mathfrak{d}$ *an injective presentation of* $\mathfrak{A}$*, and let* $f : A^n \to A$ *be a function of* $\mathfrak{A}$ *such that for all* $b \in A$ *the set* $f^{-1}(b)$ *is finite. Then there is a constant* $m$ *such that for all* $\overline{a} \in A^n$

$$\lambda^{\mathfrak{d}}(f(\overline{a})) \geq \max\{\lambda^{\mathfrak{d}}(a_0), \dots, \lambda^{\mathfrak{d}}(a_{n-1})\} - m.$$

*Proof.* The relation $R := \{\, (b, \overline{a}) \mid f(\overline{a}) = b \,\}$ satisfies the conditions of the corollary above. □

The above results deal with a single application of a function or relation. In the remaining part of this section we will study the effect of applying functions iteratively, that is, we will consider some definable subset of the universe and calculate upper bounds on the length of the encodings of elements in the substructure generated by it. First we need bounds for the (encodings of) elements of some definable subsets.

**Lemma 5.5.** *Let $\mathfrak{A}$ be a structure in* AutStr *with presentation $\mathfrak{d}$, and let $B$ be an* $\mathrm{FO}(\exists^\omega)$*-definable subset of $A$. Then $\lambda^{\mathfrak{d}}(B)$ is a finite union of arithmetical progressions.*

*Proof.* Denote by $L$ the regular language representing $B$ and let $h : \Sigma^* \to \{1\}^*$ be the projection with $h(a) := 1$ for all $a \in \Sigma$. Then $h(L)$ is regular, too, and

$$\{\, |x| \mid x \in L \,\} = \{\, |x| \mid x \in h(L) \,\}.$$

As $h(L)$ is a regular language over an unary alphabet the claim follows (see, e.g., [Eil74, Proposition V.1.1]). □

Before proceeding we apply this lemma to our favourite example, Presburger Arithmetic.

**Lemma 5.6.** *Let $(\mathbb{N}, +, P) \in$ AutStr for some unary predicate $P$, and let $k_1 < k_2 < \cdots$ be an enumeration of $P$. There exists a constant $c$ such that $k_i \leq 2^{ci}$.*

*Proof.* Fix a presentation $\mathfrak{d}$ of $(\mathbb{N}, +, P)$. Obviously, the set $P$ is definable in this structure. By the preceding lemma, there is a constant $m$ such that $\lambda^{\mathfrak{d}}(k_i) \leq mi$ for all $i$, and because of $\lambda^{\mathfrak{d}}(k_i) \in \Theta(\log k_i)$ there is some $c$ such that

$$\tfrac{1}{c} \log_2 k_i \leq \lambda^{\mathfrak{d}}(k_i) \leq mi \implies k_i \leq 2^{cmi}.$$

□

The example $(\mathbb{N}, +, P_p) \in$ AutStr shows that this result is optimal, where $P_p$ is the set of all powers of $p$.

In the process of generating a substructure we have to count the number of applications of functions. This is made precise by

**Definition 5.7.** Let $\mathfrak{A} \in$ AutStr with presentation $\mathfrak{d}$, let $f_0, \dots, f_r$ be finitely many operations of arity $r_0, \dots, r_r$, respectively, and let $E = \{e_1, e_2, \dots\}$ be some subset of $A$ with $\lambda^{\mathfrak{d}}(e_1) \leq \lambda^{\mathfrak{d}}(e_2) \leq \cdots$. Then $G_n(E)$, the $n^{\text{th}}$ *generation* of $E$, is defined as

$$G_1(E) := \{e_1\},$$
$$G_n(E) := G_{n-1}(E) \cup \{e_n\} \cup \{\, f_i(\bar{a}) \mid \bar{a} \in G_{n-1}^{r_i}(E),\ i \leq r \,\}.$$

Putting everything together we obtain the following important result. The case of finitely generated substructures already appeared in [KN95].

**Proposition 5.8.** *Let $\mathfrak{A} \in$ AutStr with injective presentation $\mathfrak{d}$, let $f_0, \dots, f_r$ be finitely many operations of $\mathfrak{A}$, and let $E$ be some definable subset of $A$. Then there is a constant $m$ such that*

$$\lambda^{\mathfrak{d}}(a) \leq mn \text{ for all } a \in G_n(E).$$

*In particular, $|G_n(E)| \leq |\Sigma|^{mn+1}$ where $\Sigma$ is the alphabet of $\mathfrak{d}$.*

*Proof.* According to Proposition 5.1 and Lemma 5.5 there are constants $m'$ and $m_0, \dots, m_r$ with

$$\begin{aligned} \lambda^{\mathfrak{d}}(e_n) &\leq m'n, \\ \lambda^{\mathfrak{d}}(f_i(a_0, \dots, a_{r_i-1})) &\leq m_i + \max\{\lambda^{\mathfrak{d}}(a_0), \dots, \lambda^{\mathfrak{d}}(a_{r_i-1})\} \end{aligned}$$

for $i \leq r$. Set $m := \max\{m', m_0, \dots, m_r\}$. We prove the claim by induction on $n$.

($n = 1$) $G_1(E) = \{e_1\}$ and $\lambda^{\mathfrak{d}}(e_1) \leq m' \leq m$.

($n > 1$) Let $a \in G_n(E)$. There are three possible cases. If $a \in G_{n-1}(E)$ then the induction hypothesis yields

$$\lambda^{\mathfrak{d}}(a) \leq m(n-1) < mn.$$

If $a = e_n$ then

$$\lambda^{\mathfrak{d}}(a) \leq m'n \leq mn.$$

If $a = f_i(a_0, \dots, a_{r_i-1})$ for some $\overline{a} \in G_{n-1}^{r_i}(E)$ and $i \leq r$ then

$$\begin{aligned} \lambda^{\mathfrak{d}}(a) &= \lambda^{\mathfrak{d}}(f_i(a_0, \dots, a_{r_i-1})) \\ &\leq m_i + \max\{\lambda^{\mathfrak{d}}(a_0), \dots, \lambda^{\mathfrak{d}}(a_{r_i-1})\} \\ &\leq m_i + m(n-1) \qquad \text{(by ind. hyp.)} \\ &\leq m + m(n-1) = mn. \end{aligned}$$

□

*Remark.* Clearly, the claim remains valid if we replace some of the generating functions by relations which satisfy the conditions of Corollary 5.2.

We give two applications. Obviously, in free structures you can construct many different elements by few applications of functions. Therefore it should not be surprising that the free monoid is not automatic.

*Example.* Let $\mathfrak{M}$ be a trace monoid with at least two non-commuting generators $a$ and $b$. Then $\mathfrak{M} \notin$ AutStr. In particular, $(\Sigma^*, \cdot, \varepsilon) \notin$ AutStr for any non-unary alphabet $\Sigma$.

*Proof.* We show by induction on $n$ that

$$\{a, b\}^{\leq 2^n} \subseteq G_{n+1}(a, b).$$

For $n = 1$ we have $\{a, b\} \subseteq \{a, aa, b\} = G_2(a, b)$, and for $n > 1$

$$\begin{aligned} G_{n+1}(a, b) &= \{\, uv \mid u, v \in G_n(a, b) \,\} \\ &\supseteq \{\, uv \mid u, v \in \{a, b\}^{\leq 2^{n-1}} \,\} \\ &= \{a, b\}^{\leq 2^n}. \end{aligned}$$

Therefore, $|G_n(a, b)| \geq 2^{2^n}$ and the claim follows. □

*Example.* Let $\mathfrak{A}$ be any structure in which a pairing function $f$ can be defined. Then $\mathfrak{A} \notin$ AutStr.

*Proof.* Let $a$, $b$ be distinct elements of $\mathfrak{A}$. All words $w \in \{a,b\}^*$ of length $|w| = 2^n$ can be coded in $\mathfrak{A}$ using applications of $f$ nested $n$ levels deep. For instance, the word $abaa$ of length $2^2$ can be represented as $f(f(a,b), f(a,a))$. Let $c(w)$ be the code of $w$. Consider the generations of $\{a,b\}$. We have

$$\{ c(w) \mid w \in \{a,b\}^{2^n} \} \subseteq G_{n+1}(a,b).$$

which implies that $|G_{n+1}(a,b)| \geq 2^{2^n}$ as the coding is injective. □

The above proposition can be generalised to the case of an infinite number of definable generating functions.

**Definition 5.9.** Let $L \subseteq \Sigma^*$. By $\iota(L)$ we denote the index of the Nerode-congruence of $L$. Analogously, if $\mathfrak{d}$ is an automatic presentation and $\varphi \in \mathrm{FO}$ we define $\iota(\varphi) := \iota(\eta^{\mathfrak{d}}(\varphi))$.

**Lemma 5.10.** *Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ be an automatic presentation.*

$$\iota(\varphi \vee \psi) \leq \iota(\varphi)\iota(\psi) \qquad \iota(\neg\varphi) \leq \iota(L_\delta)\iota(\varphi)$$
$$\iota(\varphi \wedge \psi) \leq \iota(\varphi)\iota(\psi) \qquad \iota(\exists \overline{y} \varphi(\overline{x}, \overline{y})) \leq 2^{\iota(\varphi)}$$

*Proof.* To prove the first three inequations we show that $\iota(L_1 \oplus L_2) \leq \iota(L_1)\iota(L_2)$ for regular languages $L_1$, $L_2$, and $\oplus \in \{\cup, \cap, \setminus\}$. Let $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be the minimal deterministic automata recognising $L_1$ and $L_2$. Then, after choosing the right set of final states, the product automaton $\mathfrak{A}_1 \times \mathfrak{A}_2$ recognises $L_1 \oplus L_2$.

To construct an automaton for $\eta^{\mathfrak{d}}(\exists \overline{y} \varphi)$ we take the minimal deterministic automaton for $\varphi$, remove the components corresponding to $\overline{y}$ from the labels of every transition, and mark as final states all states from which, in the original automaton, a final state can be reached by using only transitions whose labels contain □ in the components corresponding to $\overline{y}$. Since in general this yields a nondeterministic automaton we have to apply the subset construction which may cause an exponential blowup of the state-space. □

*Example.* The question whether $(\mathbb{Q}, +)$ is automatic is open. If we assume that $(\mathbb{Q}, +)$ has an automatic presentation $\mathfrak{d}$, then there is a constant $m$ such that for all $n, q_0, \dots, q_l, k_0, \dots, k_l \in \mathbb{N}$

$$\lambda^{\mathfrak{d}}\Big(\frac{n}{q_0^{k_0} \cdots q_l^{k_l}}\Big) \leq \lambda^{\mathfrak{d}}(1) + m\lceil \log_2 n \rceil + \sum_{i=0}^{l} k_i 2^{m^{\log_2^2 q_i}}.$$

*Proof.* Set $m := \iota(x + y = z)$. As in the case of Presburger Arithmetic for $n \in \mathbb{N}$ we obtain the bound

$$\lambda^{\mathfrak{d}}(n) \leq m\lceil \log_2 n \rceil + \lambda^{\mathfrak{d}}(1).$$

It remains to show that $\iota(y = x/q) \leq 2^{m^{\log_2^2 q}}$ for fixed $q \in \mathbb{N}$. Let $\{i_0, \dots, i_r\}$ be the set of digits of the binary encoding of $q$ which are 1. Then, $y = x/q$ or, equivalently, $x = q \cdot y$ can be defined as

$$\exists x_1 \cdots \exists x_{r-1} \exists y_1 \cdots \exists y_{\lfloor \log_2 q \rfloor} \Big( y_1 = y + y \wedge \bigwedge_j y_{j+1} = y_j + y_j$$
$$\wedge \bigwedge_{j>1} x_j = x_{j-1} + y_{i_j} \wedge x_1 = y_{i_0} + y_{i_1} \wedge x = x_{r-1} + y_{i_r} \Big).$$

($y_i$ contains $2^i y$, $x_i$ is used to calculate the sum of those $y_i$ needed.) We obtain the following bound

$$\iota(x = q \cdot y) \leq 2^{m^{\lfloor \log_2 q \rfloor \cdot r}} \leq 2^{m^{\log_2^2 q}}$$

proving our claim. □

## 5.2 Applications and Examples

In this section the tools developed in the previous one are used to investigate whether some structures do or do not have automatic presentations. We start with some simple applications to linear orders, equivalence and permutation structures.

**Lemma 5.11.** *Let* $\mathfrak{A} = (A, <, R_0, \ldots, R_r) \in \text{AutStr}$ *be a structure with a discrete linear order* $<$ *and an injective presentation* $\mathfrak{d}$. *Denote by* $s$ *the successor function of* $<$. *Then there is some constant* $m$ *such that for all* $a \in A$ *and* $n \in \mathbb{Z}$

$$\lambda^{\mathfrak{d}}(s^n a) \leq \lambda^{\mathfrak{d}}(a) + |n|\, m.$$

*Proof.* Immediately from Proposition 5.1 as $s$ and $s^{-1}$ are definable. □

**Lemma 5.12.** *Let* $\mathfrak{A} = (A, <_0, <_1, R_0, \ldots, R_r) \in \text{AutStr}$ *be a structure with two discrete linear orders. Denote the successor functions of* $<_0$ *and* $<_1$ *by* $s_0$ *and* $s_1$, *respectively. There is some constant* $m$ *such that for all* $a \in A$ *and* $n \in \mathbb{Z}$

$$\left|\lambda^{\mathfrak{d}}(s_0^n a) - \lambda^{\mathfrak{d}}(s_1^n a)\right| \leq \lambda^{\mathfrak{d}}(a) + |n|\, m.$$

*Proof.* Take $m$ as maximum of the constants from the previous lemma for $<_0$ and $<_1$. □

**Lemma 5.13.** *Let* $\mathfrak{A} = (A, <, R_0, \ldots, R_r) \in \text{AutStr}$ *be a structure with a well-ordering* $<$ *and an injective presentation* $\mathfrak{d}$. *Then there exists a constant* $m$ *such that for every* $a \in A$

$$\lambda^{\mathfrak{d}}(b) \leq \lambda^{\mathfrak{d}}(a) + m \text{ for all } b \leq a.$$

*Proof.* Since for every $a \in A$ the set $\{\, b \in A \mid b \leq a \,\}$ is finite we can apply Corollary 5.2. □

**Lemma 5.14.** *If* $f : \mathbb{N} \to \mathbb{N}$ *is definable in* $\mathfrak{N}_p$ *and* $\sim\, \subseteq A \times A$ *is an equivalence relation with* $f(n)$ *classes of size* $n$ *for all* $n \in \mathbb{N}$ *and with* $r \leq \omega$ *classes of cardinality* $\aleph_0$ *then the structure* $\mathfrak{A} := (A, \sim)$ *has an automatic presentation.*

*Proof.* We show that $\mathfrak{A} \leq_{\text{FO}} \mathfrak{N}_p$. The $k^{\text{th}}$ element of the $m^{\text{th}}$ class of size $n$ is encoded by the tuple $(n, m, k)$ and the $k^{\text{th}}$ element of the $m^{\text{th}}$ infinite class is encoded by $(0, m, k)$. The interpretation is defined as

$$\begin{aligned}
\delta(\overline{x}) &:= (x_0 > 0 \wedge x_1 < f x_0 \wedge x_2 < x_0) \vee (x_0 = 0 \wedge x_1 < r), \\
&\qquad\qquad (\text{if } r = \omega \text{ then } x_1 < r \equiv \text{true}) \\
\varepsilon(\overline{x}, \overline{y}) &:= x_0 = y_0 \wedge x_1 = y_1 \wedge x_2 = y_2, \\
\varphi_\sim(\overline{x}, \overline{y}) &:= x_0 = y_0 \wedge x_1 = y_1.
\end{aligned}$$

□

**Lemma 5.15.** *Let $\mathfrak{A} = (A, \sim) \in \text{AutStr}$ where $\sim$ is an equivalence relation and let $\mathfrak{d}$ be an injective presentation of $\mathfrak{A}$. Then there is a constant $m$ such that for all finite equivalence classes $[a]_\sim$*

$$\left|\lambda^{\mathfrak{d}}(a) - \lambda^{\mathfrak{d}}(a')\right| \leq m \qquad \text{for all } a' \in [a]_\sim.$$

*Proof.* Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_\sim)$ and let $m$ be the index of the Nerode-congruence of $L_\sim$. If there are $x, y \in \Sigma^*$ such that

$$x \otimes y \in L_\sim \qquad \text{and} \qquad |y| \geq |x| + m$$

then, according to the Pumping Lemma, there are infinitely many $y' \in \Sigma^*$ with $x \otimes y \in L_\sim$. Contradiction. □

**Lemma 5.16.** *Let $\mathfrak{A} = (A, \sim) \in \text{AutStr}$ where $\sim$ is an equivalence relation. Let $n_0 < n_1 < \cdots$ be an enumeration of the cardinalities of the finite $\sim$-classes. Then $n_i \in 2^{\mathcal{O}(i)}$.*

*Proof.* Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_\sim)$ be an injective presentation of $\mathfrak{A}$. Consider the set $F$ defined by

$$\varphi(x) := \neg\exists^\omega y(x \sim y).$$

According to Lemma 5.5 there is a subset $\{a_1, a_2, \dots\} \subseteq F$ such that, for some constant $m'$, $\lambda^{\mathfrak{d}}(a_i) = m'i$. Let $m$ be the constant from the preceding lemma and set $k := \lfloor m/m' \rfloor + 1$. Then $a_i \not\sim a_{i+k}$ and

$$\begin{aligned} &\lambda^{\mathfrak{d}}([a_{ik}]_\sim) \leq m'i \cdot (\lfloor m/m' \rfloor + 1) + m \leq (m + m')(i + 1) \\ \Longrightarrow\ &\left|[a_{ik}]_\sim\right| \leq 2^{\mathcal{O}(i)}. \end{aligned}$$

As $\big(|[a_{ik}]_\sim|\big)_i$ is a subsequence of $(n_i)_i$ the claim follows. □

**Lemma 5.17.** *Let $f : \mathbb{N} \to \mathbb{N}$ be definable in $\mathfrak{N}_p$. Then $(A, \pi) \in \text{AutStr}$ where $A$ is countable and $\pi$ is a permutation of $A$ with $f(n)$ orbits of size $n$ and an arbitrary number of infinite orbits.*

*Proof.* For simplicity, we construct an interpretation of $(A, \pi)$ in $(\mathbb{Z}, +, |_p)$. Clearly, $f$ is also definable in this structure. Let $r \leq \omega$ be the number of infinite orbits. We encode the $k^{\text{th}}$ element of the $m^{\text{th}}$ orbit of size $n$ as $(k, m, n)$ and the elements of the $m^{\text{th}}$ infinite orbit as $(k, m, 0)$ for $k \in \mathbb{Z}$. Thus, we define

$$\begin{aligned} \delta(\overline{x}) &:= (x_2 > 0 \,\wedge\, 0 \leq x_0 < x_2 \,\wedge\, 0 \leq x_1 < fx_2) \vee (x_2 = 0 \wedge x_1 < r), \\ \varepsilon(\overline{x}, \overline{y}) &:= x_0 = y_0 \wedge x_1 = y_1 \wedge x_2 = y_2, \\ \varphi_\pi(\overline{x}, \overline{y}) &:= x_1 = y_1 \wedge x_2 = y_2 \wedge (x_2 > 0 \wedge x_0 + 1 < x_2 \wedge y_0 = x_0 + 1 \\ &\qquad\qquad \vee x_2 > 0 \wedge x_0 + 1 = x_2 \wedge y_0 = 0 \\ &\qquad\qquad \vee x_2 = 0 \wedge y_0 = x_0 + 1). \end{aligned}$$

□

**Reducts of Arithmetic.** We start by deriving limits on the possible presentations of Presburger Arithmetic. Recall from Lemma 5.3 that $\lambda^{\mathfrak{d}}(n) \in \Theta(\log n)$.

**Proposition 5.18.** *Let* $(\mathbb{N}, +, f) \in \mathrm{AutStr}$ *for* $f : \mathbb{N} \to \mathbb{N}$.

(i) $f(n) \in n^{\mathcal{O}(1)}$ *and* $n^{1+\varepsilon} \notin \mathcal{O}(f(n))$ *for all* $\varepsilon > 0$.

(ii) *If* $f(n) \in \mathcal{O}(n^{1-\varepsilon})$ *for some* $\varepsilon > 0$ *then* $f(n)$ *is bounded.*

(iii) *Let* $k_1 < k_2 < \cdots$ *be an enumeration of* $f^{-1}(n)$ *for some* $n$. *There exists a constant* $c$ *such that* $k_i \leq 2^{ci}$.

*Proof.* (i) Fix a presentation $\mathfrak{d}$ of $(\mathbb{N}, +, f)$. By Proposition 5.1 there is a constant $m$ such that for all $n \in \mathbb{N}$

$$\lambda^{\mathfrak{d}}(f(n)) \leq \lambda^{\mathfrak{d}}(n) + m.$$

Applying $f$ several times, we obtain

$$\lambda^{\mathfrak{d}}(f^k(n)) \leq \lambda^{\mathfrak{d}}(n) + km.$$

Because of $\lambda^{\mathfrak{d}}(n) \in \Theta(\log n)$ there are constants $c_0$ and $c_1$ such that for large enough $n$

$$\begin{aligned} &\tfrac{1}{c_0} \log_2 f^k(n) \leq \lambda^{\mathfrak{d}}(f^k(n)) \leq \lambda^{\mathfrak{d}}(n) + km \leq c_1 \log_2 n + km \\ \Longrightarrow\ & f^k(n) \leq 2^{c_0 km} n^{c_0 c_1}. \end{aligned}$$

Thus $f(n) \in n^{\mathcal{O}(1)}$.

Suppose that $n^r \in \mathcal{O}(f(n))$ for some $r > 1$, i.e., $f(n) > cn^r$ for some $c$ and all sufficiently large $n$. Thus,

$$f^k(n) > cc^r \cdots c^{r^{k-1}} n^{r^k} = c^{(r^k-1)/(r-1)} n^{r^k}.$$

Choosing $r^k > c_0 c_1$ we get a contradiction to $f^k(n) \leq 2^{c_0 km} n^{c_0 c_1}$ for large $n$.

(ii) Let $f(n)$ be unbounded. Then

$$g(n) := \min\{\, k \mid f(k) \geq n \,\}$$

is well-defined and monotone. Since $g$ is FO-definable in $(\mathbb{N}, +, f)$ the structure $(\mathbb{N}, +, g)$ has an automatic presentation as well, and $n^{1+\varepsilon} \notin \mathcal{O}(g(n))$ for all $\varepsilon > 0$ by (i).

Suppose $f(n) \in \mathcal{O}(n^r)$ for some $r < 1$. Then $f(n) < cn^r$ for some $c$ and all sufficiently large $n$. Thus,

$$f(n) < cn^r \implies n = g(f(n)) < g(cn^r) \implies c^{-1} n^{1/r} \leq g(n)$$

in contradiction to $n^{1/r} \notin \mathcal{O}(g(n))$.

(iii) Since the set $f^{-1}(n)$ is definable the claim immediately follows from Lemma 5.6. □

So far, the only reduct of Arithmetic we looked at was the additive one. Now we turn to Skolem Arithmetic $(\mathbb{N}, \cdot)$ and the divisibility poset $(\mathbb{N}, |)$.

**Proposition 5.19.** $(\mathbb{N}, |) \notin \mathrm{AutStr}$.

*Proof.* Suppose $(\mathbb{N}, |) \in \text{AutStr}$. We define the set of primes

$$Px \ :\text{iff} \ x \neq 1 \wedge \forall y(y \mid x \rightarrow y = 1 \vee y = x),$$

the set of powers of some prime

$$Qx \ :\text{iff} \ \exists y(Py \wedge \forall z(z \mid x \wedge z \neq 1 \rightarrow y \mid z)),$$

and a relation containing all pairs $(n, pn)$ where $p$ is a prime divisor of $n$

$$Sxy \ :\text{iff} \ x \mid y \wedge \exists^{=1} z(Qz \wedge \neg Pz \wedge z \mid y \wedge \neg z \mid x).$$

The least common multiple of two numbers is

$$\text{lcm}(x, y) = z \ :\text{iff} \ x \mid z \ \wedge \ y \mid z \ \wedge \ \neg\exists u(u \neq z \ \wedge \ x \mid u \ \wedge \ y \mid u \ \wedge \ u \mid z).$$

For every $n \in \mathbb{N}$ there are only finitely many $m$ with $(n, m) \in S$. Therefore $S$ satisfies the conditions of Corollary 5.2. Consider the set generated by $P$ via $S$ and lcm, and let $\gamma(n) := |G_n(P)|$ be the cardinality of $G_n(P)$. If $(\mathbb{N}, |)$ is in AutStr then $(\mathbb{N}, |, P, Q, S) \in \text{AutStr}$ and $\gamma(n) \in 2^{\mathcal{O}(n)}$ by Proposition 5.8. Let $P = \{p_1, p_2, \dots\}$. For $n = 1$ we have $G_1(P) = \{p_1\}$. Generally, $G_n(P)$ consists of

(1) numbers of the form $p_1^{k_1}$,
(2) numbers of the form $p_2^{k_2} \cdots p_n^{k_n}$, and
(3) numbers of a mixed form.

In $n$ steps we can create

(1) $p_1, \dots, p_1^n$ (via $S$),
(2) $\gamma(n-1)$ numbers with $k_1 = 0$, and
(3) $\gamma(n-2) - 1$ numbers of a mixed form for every $0 < k_1 < n$ (via lcm).

All in all we obtain

$$\begin{aligned}
\gamma(n) &\geq n + \gamma(n-1) + (n-1)(\gamma(n-2) - 1) \\
&= \gamma(n-1) + (n-1)\gamma(n-2) + 1 \\
&\geq n\gamma(n-2) && (\text{as } \gamma(n-1) > \gamma(n-2)) \\
&\geq n(n-2)\cdots 3\gamma(1) && (\text{w.l.o.g. assume that } n \text{ is odd}) \\
&= n(n-2)\cdots 3 \\
&\geq ((n+1)/2)! \\
&\in 2^{\Omega(n \log n)}.
\end{aligned}$$

Contradiction. □

The importance of the following corollary lies in the fact that it is possible to construct a tree-automatic presentation of Skolem Arithmetic (cf. Section 5.3) which implies that $\text{AutStr} \neq \text{TAutStr}$.

**Corollary 5.20.** $(\mathbb{N}, \cdot) \notin \text{AutStr}$.

*Proof.* $(\mathbb{N}, |) \leq_{\text{FO}} (\mathbb{N}, \cdot)$. □

*Example.* If we replace divisibility by the predicate $\perp$ defined by

$$x \perp y \;:\text{iff}\; x \text{ and } y \text{ have no common divisors}$$

the resulting structure $(\mathbb{N}, \perp)$ is automatic.

*Proof.* We construct an interpretation $(\mathbb{N}, \perp) \leq_{\mathrm{FO}} \mathfrak{N}_p$. A number $n$ is encoded by the pair $(k, m)$ where the $i^{\text{th}}$ digit of $k$ is 1 iff the $i^{\text{th}}$ prime divides $n$ and the second component enumerates all numbers with the same set of prime divisors. Thus, $(k, m) \perp (k', m')$ holds iff there is no position at which both $k$ and $k'$ carry the digit 1. We obtain the interpretation

$$\begin{aligned}
\delta(\overline{x}) &:= \forall z(P_p z \rightarrow \mathrm{dig}_0(x_0, z) \vee \mathrm{dig}_1(x_0, z)),\\
\varepsilon(\overline{x}, \overline{y}) &:= x_0 = y_0 \wedge x_1 = y_1,\\
\varphi_\perp(\overline{x}, \overline{y}) &:= \neg\exists z(\mathrm{dig}_1(x_0, z) \wedge \mathrm{dig}_1(y_0, z)).
\end{aligned}$$

□

**Proposition 5.21.** $(\mathbb{N}, +, \perp) \notin \mathrm{AutStr}$.

*Proof.* The set of primes can be defined as

$$Px \;:\text{iff}\; x > 1 \wedge \forall y(y < x \rightarrow x \perp y).$$

We start by constructing a function mapping numbers $x$ to the least prime greater than $x$.

$$fx = y \;:\text{iff}\; y > x \wedge Py \wedge \neg\exists z(x < z < y \wedge Pz).$$

Let $g(x) := fx \cdot ffx$. Since $g(x) > x^2$, the claim follows if we can define $g$ in $(\mathbb{N}, +, \perp)$. We use the auxiliary relation $Mxy$ which holds iff $fx$ and $ffx$ are the only prime divisors of $y$. Thus, $g(x)$ returns the least such $y$.

$$\begin{aligned}
Mxy \;&:\text{iff}\; \neg(y \perp fx) \wedge \neg(y \perp ffx)\\
&\qquad \wedge \forall z[\neg(y \perp z) \rightarrow \neg(z \perp fx) \vee \neg(z \perp ffx)],\\
g(x) = y \;&:\text{iff}\; y > ffx \wedge Mxy \wedge \neg\exists z[ffx < z < y \wedge Mxz].
\end{aligned}$$

□

## 5.3 Composition of Structures

**Generalised Products.** We begin our investigation of the closure properties of automatic structures with Feferman-Vaught like products (see [Tho97a, Zei94, Hod93]). A generalised product—as it is defined below—is a generalisation of a direct product, a disjoint union, and an ordered sum. Hence, we will be able to prove closure under all of these operations with just one—unfortunately quite technical—theorem.

The relations of the new structure are defined in terms of the types of the components of its elements.

**Definition 5.22.** Let $\tau$ be a finite relational signature, $\mathfrak{A}$ a $\tau$-structure, and $\overline{a} \in A^n$. For $\overline{k} \in \mathbb{N}^*$ we define the *$\overline{k}$-type* $T^{\overline{k}}(\mathfrak{A}, \overline{a})$ of $(\mathfrak{A}, \overline{a})$ as

$$\begin{aligned}
T^\varepsilon(\mathfrak{A}, \overline{a}) &:= \{\, \varphi \in \mathrm{FO}^n[\tau] \mid \varphi \text{ is atomic, } (\mathfrak{A}, \overline{a}) \models \varphi \,\},\\
T^{\overline{k}m}(\mathfrak{A}, \overline{a}) &:= \{\, T^{\overline{k}}(\mathfrak{A}, \overline{a}\overline{b}) \mid \overline{b} \in A^m \,\}.
\end{aligned}$$

The set $\mathscr{T}^{\overline{k}}(n)$ of all $\overline{k}$-types with $n$ parameters is

$$\mathscr{T}^{\varepsilon}(n) := \{\, \varphi \in \mathrm{FO}^n[\tau] \mid \varphi \text{ is atomic} \,\},$$
$$\mathscr{T}^{\overline{k}m}(n) := \mathscr{P}(\mathscr{T}^{\overline{k}}(n+m)).$$

For each type there exists a so-called *Hintikka-formula* defining the tuples of this type (see [EF95] for the definition).

In order to understand the next definition let us first look at how a direct product and an ordered sum can be defined using types.

*Example.* (1) Let $\mathfrak{A} := \mathfrak{A}_0 \times \mathfrak{A}_1$ where $\mathfrak{A}_i = (A_i, R_i)$, for $i \in \{0,1\}$, and $R$ is a binary relation. The universe of $\mathfrak{A}$ is $A_0 \times A_1$. Some pair $(\overline{a}, \overline{b})$ belongs to $R$ iff $(a_0, b_0) \in R_0$ and $(a_1, b_1) \in R_1$. This is equivalent to the condition that the $\varepsilon$-types of $a_0 b_0$ and of $a_1 b_1$ both include the formula $Rx_0x_1$.

(2) Let $\mathfrak{A} := \mathfrak{A}_0 + \mathfrak{A}_1$ where $\mathfrak{A}_i = (A_i, <_i)$, for $i \in \{0,1\}$, and $<_0$, $<_1$ are partial orders. The universe of $\mathfrak{A}$ is $A_0 \,\dot\cup\, A_1 \cong A_0 \times \{\Diamond\} \cup \{\Diamond\} \times A_1$, and we have

$$\begin{aligned} \overline{a} < \overline{b} \text{ iff } & \overline{a} = (a_0, \Diamond),\ \overline{b} = (b_0, \Diamond) \text{ and } a_0 <_0 b_0, \\ \text{or } & \overline{a} = (\Diamond, a_1),\ \overline{b} = (\Diamond, b_1) \text{ and } a_1 <_1 b_1, \\ \text{or } & \overline{a} = (a_0, \Diamond),\ \overline{b} = (\Diamond, b_1). \end{aligned}$$

Again, the condition $a_i <_i b_i$ can be expressed using $\varepsilon$-types.

**Definition 5.23.** Let $\tau = \{R_0, \dots, R_r\}$ be a finite relational signature, $r_j$ the arity of $R_j$, and $\hat{r} := \max\{r_0, \dots, r_r\}$. Let $n \in \mathbb{N}$ and $(\mathfrak{A}_i)_{i \in I}$ be a sequence of $\tau$-structures, and let $\mathfrak{I}$ be an arbitrary relational $\sigma$-structure with universe $I$.

Fix for each $k \leq \hat{r}$ an enumeration $\{\tau_0^k, \dots, \tau_{t_k}^k\}$ of $\mathscr{T}^{\varepsilon}(n+k)$ and set

$$\sigma_k := \sigma \,\dot\cup\, \{D_0, \dots, D_{k-1}\} \,\dot\cup\, \{\, T_l^m \mid m \leq k, l \leq t_m \,\}.$$

The $\sigma_k$-expansion $\mathfrak{I}(\overline{b})$ of $\mathfrak{I}$ belonging to a sequence $\overline{b} \in \big(\prod_{i \in I}(A_i \,\dot\cup\, \{\Diamond\})\big)^k$ is given by

$$D_l^{\mathfrak{I}(\overline{b})} := \{\, i \in I \mid (b_l)_i \neq \Diamond \,\},$$
$$\begin{aligned} (T_l^m)^{\mathfrak{I}(\overline{b})} := \{\, i \in I \mid & \{\, j \mid (b_j)_i \neq \Diamond \,\} = \{j_0, \dots, j_{m-1}\} \text{ and} \\ & T^{\varepsilon}(\mathfrak{A}_i, (b_{j_0})_i \dots (b_{j_{m-1}})_i) = \tau_l^m \,\}. \end{aligned}$$

Then $\mathcal{C} := (\mathfrak{I}, D, \beta_0, \dots, \beta_r)$ with $D \subseteq \mathbb{B}^I$ and $\beta_j \in \mathrm{FO}[\sigma_{r_j}]$ defines the *generalised product* $\mathcal{C}(\mathfrak{A}_i)_{i \in I} := (A, R_0, \dots, R_r)$ of $(\mathfrak{A}_i)_{i \in I}$ where

$$A := \bigcup_{\overline{d} \in D} \prod_{i \in I} \chi_{d_i}(\{\Diamond\}, A_i), \qquad R_i := \{\, \overline{b} \in A^{r_i} \mid \mathfrak{I}(\overline{b}) \models \beta_i \,\},$$

and $\chi_b(a_0, a_1) := a_b$.

*Example.* (continued)

(1) For the direct product of $\mathfrak{A}_0 \times \mathfrak{A}_1$ we would set

$$\begin{aligned} &\mathfrak{I} := (I) \qquad \text{with } I = \{0,1\}, \\ &D := \{(1,1)\}, \\ &\beta := \bigvee_{l \in L} T_l^2 0 \wedge \bigvee_{l \in L} T_l^2 1, \end{aligned}$$

where $L$ is the set of $\varepsilon$-types containing the formula $Rx_0x_1$.

(2) In this case we would set

$$\begin{aligned}
\mathfrak{I} &:= (I) \qquad \text{with } I = \{0,1\},\\
D &:= \{(1,0),(0,1)\},\\
\beta &:= \Big(D_0 0 \wedge D_1 0 \wedge \bigvee_{l\in L} T_l^2 0\Big) \vee \Big(D_0 1 \wedge D_1 1 \wedge \bigvee_{l\in L} T_l^2 1\Big) \vee (D_0 0 \wedge D_1 1),
\end{aligned}$$

where $L$ is the set of $\varepsilon$-types containing the formula $x_0 < x_1$.

**Theorem 5.24.** *Let $\tau = \{R_0,\dots,R_r\}$ be a finite relational signature, and $\mathscr{K}$ a class of $\tau$-structures containing all finite $\tau$-structures and a structure $\mathfrak{C}$ which is complete for $\mathscr{K}$ with regard to many-dimensional* FO*-interpretations.*

*Let $\mathfrak{I}$ be a finite relational $\sigma$-structure, $(\mathfrak{A}_i)_{i\in I}$ a sequence of structures in $\mathscr{K}$, and $\mathcal{C} = (\mathfrak{I}, D, \overline{\beta})$ a generalised product. Then $\mathcal{C}(\mathfrak{A}_i)_{i\in I} \in \mathscr{K}$, and an interpretation $\mathcal{C}(\mathfrak{A}_i)_{i\in I} \leq_{\mathrm{FO}} \mathfrak{C}$ can be constructed effectively from the interpretations $\mathfrak{A}_i \leq_{\mathrm{FO}} \mathfrak{C}$ and $\mathfrak{I} \leq_{\mathrm{FO}} \mathfrak{C}$.*

*Proof.* W.l.o.g. let $I = \{0,\dots,|I|-1\}$ and assume that $\mathfrak{C}$ contains constants 0 and 1. We have to construct an interpretation of $\mathfrak{A} := \mathcal{C}(\mathfrak{A}_i)_{i\in I}$ in $\mathfrak{C}$. Let $r_j$ be the arity of $R_j$. Consider $n_i$-dimensional interpretations

$$\mathcal{I}^i := \big(h^i, \delta^i(\overline{x}^i), \varepsilon^i(\overline{x}^i,\overline{y}^i), \varphi_0^i(\overline{x}_0^i,\dots,\overline{x}_{r_0-1}^i),\dots,\varphi_r^i(\overline{x}_0^i,\dots,\overline{x}_{r_r-1}^i)\big)$$

of $\mathfrak{A}_i$ in $\mathfrak{C}$. We represent an element $a$ of $\mathfrak{A}$ by an $(|I| + n_0 + \cdots + n_{|I|-1})$-tuple

$$\overline{x} := \big(\overline{d}, \overline{x}^0,\dots,\overline{x}^{|I|-1}\big)$$

where $\overline{d} \in D$ determines which components are empty and $\overline{x}^i$ encodes the $i^{\text{th}}$ component of $a$. The desired interpretation is constructed as follows.

$$\mathcal{I} := \big(h, \delta(\overline{x}), \varepsilon(\overline{x},\overline{y}), \varphi_0(\overline{x}_0,\dots,\overline{x}_{r_0-1}),\dots,\varphi_r(\overline{x}_0,\dots,\overline{x}_{r_r-1})\big)$$

where

$$\begin{aligned}
h(\overline{d},\overline{x}^0,\dots,\overline{x}^{|I|-1}) &:= \big(\chi_{d_0}\big(\Diamond, h^0(\overline{x}^0)\big),\dots,\chi_{d_{|I|-1}}\big(\Diamond, h^{|I|-1}(\overline{x}^{|I|-1})\big)\big),\\
\delta(\overline{d},\overline{x}^0,\dots,\overline{x}^{|I|-1}) &:= \bigvee_{\overline{c}\in D}\Big(\overline{d} = \overline{c} \wedge \bigwedge_{i:\ c_i=1} \delta^i(\overline{x}^i)\Big),
\end{aligned}$$

and

$$\varepsilon(\overline{d},\overline{x}^0,\dots,\overline{x}^{|I|-1},\overline{e},\overline{y}^0,\dots,\overline{y}^{|I|-1}) := \overline{d} = \overline{e} \wedge \bigwedge_{i<|I|}\big(d_i = 1 \rightarrow \varepsilon^i(\overline{x}^i,\overline{y}^i)\big).$$

In order to define $\varphi_j$ we consider an interpretation

$$\mathcal{I}^I := \big(h^I, \delta^I(\overline{x}), \varepsilon^I(\overline{x},\overline{y}), \varphi_0^I(\overline{x}_0,\dots,\overline{x}_{s_0-1}),\dots,\varphi_s^I(\overline{x}_0,\dots,\overline{x}_{s_s-1})\big)$$

of $\mathfrak{I}$ in $\mathfrak{C}$. Since $\mathfrak{I}$ is finite such an interpretation exists. Let $\alpha_j := \beta_j^{\mathcal{I}^I}$ be the formula defining $R_j$. Note that $\beta_j$ contains additional relations $D_l$ and $T_l^m$ which are not in $\sigma$. Thus $\alpha_j$ is a sentence over the signature $\tau$ extended by the symbols $D_l$ and $T_l^m$ for appropriate $l$ and $m$. We have to be replace them in order to obtain a definition of $\varphi_j$. Let $\overline{x}_0,\dots,\overline{x}_{r_j-1}$ be the parameters of $\varphi_j$ where

$$\overline{x}_k = (\overline{d}_k, \overline{x}_k^0,\dots,\overline{x}_k^{|I|-1})$$

for $k < r_j$. $D_l$ can be defined by

$$D_l i := (d_l)_i = 1.$$

To define $T_l^m$ consider the Hintikka-formula $\vartheta_l^m(x_0, \dots, x_{r_j-1})$ defining the corresponding type and set

$$T_l^m i := (\vartheta_l^m)^{\mathcal{I}^i}(\overline{x}_0^i, \dots, \overline{x}_{r_j-1}^i).$$

Note that those definitions are only valid because $i$ ranges over a finite set. $\varphi_j$ can now be defined as $\alpha_j$ with $D_l$ and $T_l^m$ replaced by the above definitions.

Obviously, all steps in the construction above are effective. □

**Corollary 5.25.** [$\omega$-][T]AutStr *is effectively closed under finitary generalised products.*

As promised we immediately obtain closure under several types of compositions.

**Corollary 5.26.** *Let $\tau = \{R_0, \dots, R_r\}$ be a finite relational signature, $I$ a finite set, and $\mathfrak{A}$ and $\mathfrak{A}_i$, $i \in I$, $\tau$-structures with automatic presentation. Then there exist automatic presentations of*

(i) *the direct product $\prod_{i \in I} \mathfrak{A}_i$ of $(\mathfrak{A}_i)_{i \in I}$,*

(ii) *the disjoint union $\dot{\bigcup}_{i \in I} \mathfrak{A}_i$ of $(\mathfrak{A}_i)_{i \in I}$, and*

(iii) *the $\omega$-fold disjoint union $\omega \cdot \mathfrak{A}$ of $\mathfrak{A}$.*

*Proof.* (i) We have $\prod_{i \in I} \mathfrak{A}_i = \mathcal{C}(\mathfrak{A}_i)_{i \in I}$ for $\mathcal{C} := (\mathfrak{I}, D, \beta_0, \dots, \beta_r)$ with $\mathfrak{I} := (I)$, $D := \{(1, \dots, 1)\}$, and

$$\beta_j := \forall i \bigvee_{l:\ R_j \overline{x} \in \tau_l^{r_j}} T_l^{r_j} i.$$

(ii) We have $\dot{\bigcup}_{i \in I} \mathfrak{A}_i = \mathcal{C}(\mathfrak{A}_i)_{i \in I}$ for $\mathcal{C} := (\mathfrak{I}, D, \overline{\beta})$ with $\mathfrak{I} := (I)$ and

$$D := \{(1, 0, \dots, 0), \dots, (0, \dots, 0, 1)\},$$
$$\beta_j := \exists i \Big( \bigwedge_{l < r_j} D_l i \wedge \bigvee_{l:\ R_j \overline{x} \in \tau_l^{r_j}} T_l^{r_j} i \Big).$$

(iii) $\mathfrak{N} = (\omega)$ has an automatic presentation. We have $\omega \cdot \mathfrak{A} = \mathcal{C}(\mathfrak{A}, \mathfrak{N})$ for $\mathcal{C} := (\mathfrak{I}, D, \overline{\beta})$ with $\mathfrak{I} := (\{0, 1\}, 0, 1)$, $D := \{(1, 1)\}$, and

$$\beta_j := \bigvee_{l:\ R_j \overline{x} \in \tau_l^{r_j}} T_l^{r_j} 0 \wedge \bigwedge_{i_0, i_1 < r_j} \bigvee_{l:\ x_{i_0} = x_{i_1} \in \tau_l^{r_j}} T_l^{r_j} 1.$$

□

**Corollary 5.27.** *Let $\tau = \{<, R_0, \dots, R_r\}$ be a finite relational signature, $I$ a finite ordered set, and $\mathfrak{A}$ and $\mathfrak{A}_i$, $i \in I$, ordered $\tau$-structures with automatic presentation. Then there exist automatic presentations of*

(i) *the ordered sum $\sum_{i \in I} \mathfrak{A}_i$ of $(\mathfrak{A}_i)_{i \in I}$ and*

(ii) *the $\omega$-fold ordered sum $\sum_{i \in \omega} \mathfrak{A}$ of $\mathfrak{A}$.*

*Proof.* (i) We have $\sum_{i\in I}\mathfrak{A}_i = \mathcal{C}(\mathfrak{A}_i)_{i\in I}$ for $\mathcal{C} := (\mathfrak{I}, D, \beta_<, \overline{\beta})$ with $\mathfrak{I} := (I, <)$ and

$$D := \{(1,0,\dots,0),\dots,(0,\dots,0,1)\},$$
$$\beta_j := \exists i\Big(\bigwedge_{l<r_j} D_l i \wedge \bigvee_{l:\ R_j\overline{x}\in\tau_l^{r_j}} T_l^{r_j} i\Big),$$
$$\beta_< := \exists i\Big(D_0 i \wedge D_1 i \wedge \bigvee_{l:\ x_0<x_1\in\tau_l^2} T_l^2 i\Big) \vee \exists i_0 \exists i_1 (D_0 i_0 \wedge D_1 i_1 \wedge i_0 < i_1).$$

(ii) The structure $\mathfrak{N} = (\omega, <)$ has an automatic presentation. Construct $\mathcal{C} := (\mathfrak{I}, D, \beta_<, \overline{\beta})$ with $\mathfrak{I} := (\{0,1\}, 0, 1)$, $D := \{(1,1)\}$, and

$$\beta_j := \bigvee_{l:\ R_j\overline{x}\in\tau_l^{r_j}} T_l^{r_j} 0 \wedge \bigwedge_{i_0,i_1<r_j}\ \bigvee_{l:\ x_{i_0}=x_{i_1}\in\tau_l^{r_j}} T_l^{r_j} 1,$$
$$\beta_< := \Big(\bigvee_{l:\ x_0<x_1\in\tau_l^2} T_l^2 0 \wedge \bigvee_{l:\ x_0=x_1\in\tau_l^2} T_l^2 1\Big) \vee \bigvee_{l:\ x_0<x_1\in\tau_l^2} T_l^2 1.$$

Then $\sum_{i\in\omega}\mathfrak{A} = \mathcal{C}(\mathfrak{A}, \mathfrak{N})$. □

**Weak Direct Powers.** A case not covered in the preceding section are weak and $\omega$-fold direct powers. Clearly, for cardinality reasons [T]AutStr cannot be closed under $\omega$-fold direct powers, and even in the weak case we obtain a negative result.

**Theorem 5.28.** AutStr *is not closed under weak direct powers.*

*Proof.* Presburger Arithmetic $(\mathbb{N}, +)$ possesses an automatic presentation. But its weak direct power is isomorphic to Skolem Arithmetic which according to Corollary 5.20 is not in AutStr. □

It turns out that tree presentations on the other hand are closed under weak powers.

**Theorem 5.29.**

(i) TAutStr *is closed under weak direct powers.*

(ii) $\omega$-TAutStr *is closed under weak and $\omega$-fold direct powers.*

*Proof.* Let $\mathfrak{A} \in$ TAutStr with presentation $\mathfrak{d} = (\nu, \Sigma, T_\delta, T_\varepsilon, T_{R_0}, \dots, T_{R_r})$. In order to construct a tree-automatic presentation of the weak direct power $\mathfrak{A}^*$ of $\mathfrak{A}$ we encode a tuple $(t_0, \dots, t_n)$ of trees from $T_\delta$ by the tree $t$ with

$$\operatorname{dom}(t) := \{\varepsilon, 0, \dots, 0^n\} \cup \bigcup_{i\le n} 0^i 1 \operatorname{dom}(t_i),$$
$$t(0^i) := 0,$$
$$t(0^i 1x) := t_i(x).$$

Let $\mathfrak{B} = (Q, \Sigma, \Delta, F)$ be a tree-automaton recognising one of the languages $T_\delta$, $T_\varepsilon$, $T_{R_0}, \dots, T_{R_r}$. The tree-automaton $\mathfrak{B}^*$ for the corresponding language in the presentation of $\mathfrak{A}^*$ is $\mathfrak{B}^* := (Q \uplus \{q_0\}, \Sigma, \Delta', \{q_0\})$ with

$$\Delta' := \Delta \uplus \{\, (q_0, 0, q_0, q), (q_0, 0, \Box, q) \mid q \in F \,\}.$$

The proofs of the other claims are analogous. □

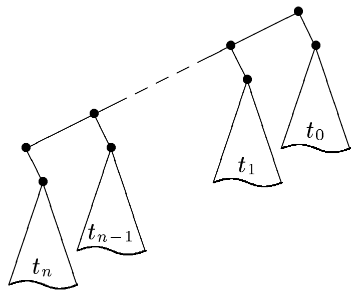



Figure 5.1: Encoding of $(t_0, \dots, t_n)$

*Example.* (1) $(\mathbb{N}, \cdot) \in$ TAutStr as $(\mathbb{N} \setminus \{0\}, \cdot) \cong (\mathbb{N}, +)^*$ via the isomorphism taking $(n_0, n_1, \dots)$ to the number $p_0^{n_0} p_1^{n_1} \cdots$ where $p_0, p_1, \dots$ is an enumeration of all primes.

(2) Similarly, $(\mathbb{Q}^{>0}, \cdot) \in$ TAutStr as $(\mathbb{Q}^{>0}, \cdot) \cong (\mathbb{Z}, +)^*$.

## 5.4 The Class Hierarchy

Finally, we are able to compare the various classes of automatic structures.

**Theorem 5.30.** [T]AutStr $\subset \omega$-[T]AutStr.

*Proof.* We construct an interpretation $\mathfrak{N}_p \leq_{\mathrm{FO}} \mathfrak{R}_p^+$.

$$\begin{aligned} \delta(x) &:= 1 \mid_p x, & \varphi_+(x, y, z) &:= x + y = z, \\ \varepsilon(x, y) &:= x = y, & \varphi_{\mid_p}(x, y) &:= x \mid_p y. \end{aligned}$$

Because $|\mathbb{R}| > |\mathbb{N}|$ there is no interpretation in the other direction, hence the inclusion is proper.

The case of tree-automatic presentations is analogous. □

**Theorem 5.31.**

(i) AutStr $\subset$ TAutStr

(ii) $\omega$-AutStr $\subseteq \omega$-TAutStr

*Proof.* (i) We show that $\mathfrak{N}_p \leq_{\mathrm{FO}} \mathfrak{T}_p$. We define formulae which state that the left branch of a tree $t$ is labelled 1, respectively, from the root or from some vertex $r$ to some other vertex, and every other vertex is labelled 0.

$$\begin{aligned} &\mathrm{LeftPath}(t) := \\ &\quad \forall r \forall s \forall s' (\mathrm{Succ}(r, s, s') \\ &\qquad \rightarrow (t \cdot r = r \vee t \cdot r = 0) \wedge (t \cdot r = r \rightarrow t \cdot s' = 0) \\ &\qquad \wedge (t \cdot r = 0 \rightarrow t \cdot s = 0 \wedge t \cdot s' = 0)) \\ &\mathrm{LeftPathSuffix}(t, r) := \\ &\quad \mathrm{SingleOne}(r) \\ &\quad \wedge \exists s_1 \exists s_2 (\mathrm{LeftPath}(s_1) \wedge \mathrm{LeftPath}(s_2) \\ &\qquad \wedge s_1 \cdot t = t \wedge s_2 \cdot t = 0 \wedge t + s_2 = s_1 \wedge s_1 \cdot r = r \wedge s_2 \cdot r = 0 \\ &\qquad \wedge \forall v (\mathrm{Succ}_0(v, r) \rightarrow s_2 \cdot v = v)) \end{aligned}$$

To check the digits at one position the following formula can be used. It states that the labels at the position $r$ in the trees $t_0$, $t_1$, $t_2$, and $t_3$ are labelled $x_0$, $x_1$, $x_2$, and $x_3$, respectively, and the position $r'$ in $t_3$ is labelled $x'_3$.

$$\mathrm{Add}_{a_0a_1a_2a_3a'_3}(t_0,t_1,t_2,t_3,r,r') := \bigwedge_{i<4} r \cdot t_i = a_i r \wedge r' \cdot t_3 = a'_3 r.$$

A number $n \in \mathbb{N}$ is encoded by a tree whose left branch is labelled with the digits of $n$.

$$\begin{aligned}
\delta(t) &:= \exists s(\mathrm{LeftPath}(s) \wedge s \cdot t = t),\\
\varepsilon(t,t') &:= t = t',\\
\varphi_{|_p}(t,t') &:= \exists s(\mathrm{LeftPathSuffix}(s,t) \wedge s \cdot t' = t'),\\
\varphi_+(t_0,t_1,t') &:= \exists s\Big(\delta(s) \wedge \forall r \forall r' \Big(\mathrm{Succ}_0(r,r') \rightarrow\\
&\qquad \bigvee_{(a,b,c,d,d')\in A} \mathrm{Add}_{abcdd'}(t_0,t_1,t',s,r,r')\Big)\Big),
\end{aligned}$$

where we used the set $A$ of correct digits defined in Section 4.1.

The inclusion is proper, as $(\mathbb{N},\cdot) \in \mathrm{TAutStr} \setminus \mathrm{AutStr}$.

(ii) analogous. □

We have seen that, simply for cardinality reasons, $\omega$-[T]AutStr \ [T]AutStr is non-empty. The question occurs whether cardinality is the only reason. A first step to answer this question is

**Theorem 5.32.** *Let* $\mathfrak{A} \in \omega\text{-AutStr}$ *be countable.* $\mathfrak{A} \in \mathrm{AutStr}$ *if and only if it has an injective* $\omega$*-automatic presentation.*

*Proof.* (⇒) Let $\mathfrak{d}$ be an injective automatic presentation of $\mathfrak{A}$. We obtain an injective $\omega$-automatic presentation by changing each encoding $x$ to $x\Box^\omega$ for some padding symbol $\Box$.

(⇐) Let $\mathfrak{d} = (\nu, \Sigma, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ be an injective $\omega$-automatic presentation of a countable structure $\mathfrak{A}$. Then $L_\delta$ is countable, too. As it is $\omega$-regular we have

$$L_\delta = \bigcup_{i\le n} U_i V_i^\omega$$

for regular languages $U_0,\dots,U_n$, $V_0,\dots,V_n \subseteq \Sigma^*$. In this expression $V_0,\dots,V_n$ can be chosen one-elementary as, otherwise, $|V_i^\omega| \ge |\{0,1\}^\omega| = 2^{\aleph_0} > |L_\delta|$. Thus,

$$L_\delta = \bigcup_{i\le n} U_i\{v_i\}^\omega.$$

We construct an automatic presentation $\mathfrak{d}' = (\nu', \Sigma', L'_\delta, L'_\varepsilon, L'_{R_0}, \dots, L'_{R_r})$ of $\mathfrak{A}$.

$$\Sigma' := \Sigma \uplus \{0,\dots,n\} \qquad \nu'(ix) := \nu(xv_i^\omega)$$

$$L'_\delta := \bigcup_{i\le n} iU_i \qquad L'_\varepsilon := (L'_\delta)^{\otimes 2} \cap \Big\{ \left[\begin{smallmatrix}a\\a\end{smallmatrix}\right] \;\Big|\; a \in \Sigma' \Big\}^*$$

Consider a Büchi-automaton $\mathfrak{B} = (Q, (\Sigma \cup \{\Box\})^{r_j}, \Delta, q_0, F)$ recognising $L_{R_j}$, and for $\bar{l} \in \{0, \dots, n\}^{r_j}$ and $i_0 < |v_{l_0}|, \dots, i_{r_j-1} < |v_{l_{r_j-1}}|$, denote by $R^{\bar{l}}_{\bar{\imath}}$ the set of states from which $\mathfrak{B}$ accepts the word

$$\big(s_{i_0}(v_{l_0}) \otimes \cdots \otimes s_{i_{r_j-1}}(v_{l_{r_j-1}})\big)^\omega$$

where $s_i$ is the cyclic shift by $i$ letters to the left

$$s_i(a_0 \cdots a_n) := a_i \cdots a_n a_0 \cdots a_{i-1}.$$

Let $k := \max\{|v_0|, \dots, |v_n|\}$ and $v_i = v_{i0} \cdots v_{i(|v_i|-1)}$. The following automaton recognises $L'_{R_j}$. Set $\mathfrak{B}' := \big(Q', (\Sigma' \cup \{\Box\})^{r_j}, \Delta', q'_0, F'\big)$ where

$$\begin{aligned} Q' &:= Q \times \{0, \dots, n\}^{r_j} \times \{0, \dots, k-1\}^{r_j} \cup \{q'_0\}, \\ \Delta' &:= \big\{\, \big(q'_0, \bar{l}, (q_0, \bar{l}, \bar{0})\big) \bigm| \bar{l} \in \{0, \dots, n\}^{r_j} \,\big\} \\ &\quad \cup \big\{\, \big((q, \bar{l}, \bar{\imath}), \bar{a}, (q', \bar{l}, \bar{\imath}')\big) \bigm| (q, \bar{b}, q') \in \Delta \text{ where, for } s < r_j, \\ &\qquad\qquad (b_s = a_s,\ i'_s = i_s = 0, \text{ and } a_s \in \Sigma) \text{ or} \\ &\qquad\qquad (b_s = v_{l_s i_s},\ i'_s = i_s + 1 \bmod |v_{l_s}|, \text{ and } a_s = \Box) \,\big\}, \\ F' &:= \big\{\, (q, \bar{l}, \bar{\imath}) \bigm| q \in R^{\bar{l}}_{\bar{\imath}} \,\big\}. \end{aligned}$$

Intuitively, $\mathfrak{B}'$ determines from the first letter which infinite part the words in each track of the input have, and when the end of a word is reached it simulates the work of $\mathfrak{B}$ on the word $v_i^\omega$ until the end of the whole input is reached. As there are only finitely many possible ways the infinite parts are shifted relative to each other $\mathfrak{B}'$ can determine whether the input is accepted by $\mathfrak{B}$. $\Box$

*Open Problem.* Does every countable $\mathfrak{A} \in \omega$-AutStr possess an injective presentation, or, equivalently, is every countable $\mathfrak{A} \in \omega$-AutStr already in AutStr?

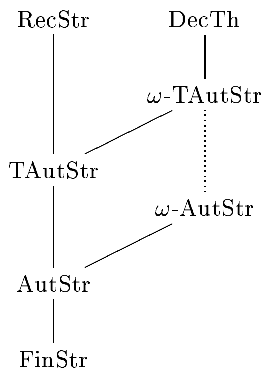
A first step in the investigation of this question is

**Lemma 5.33.** *Let $\mathfrak{A} \in \omega$-AutStr and let $a \in A$ be definable. Then in every $\omega$-automatic presentation $\mathfrak{d}$ of $\mathfrak{A}$ there is an ultimately periodic $\omega$-word encoding $a$.*

*Proof.* Let $\varphi(x)$ be the formula defining $a$. The claim immediately follows from the fact that every non-empty regular $\omega$-language contains an ultimately periodic word, and because $\eta^{\mathfrak{d}}(\varphi)$ is non-empty. $\Box$

*Open Problem.* Does every $\mathfrak{A} \in \omega$-AutStr in which every element is definable belong to AutStr?

We have obtained the following hierarchy of classes



where FinStr is the class of finite, RecStr the class of recursive, and DecTh the class of structures with decidable FO-theory, and where solid lines indicate proper inclusion. Examples for the proper inclusions FinStr $\subset$ AutStr, TAutStr $\subset$ RecStr, and $\omega$-TAutStr $\subset$ DecTh are Presburger Arithmetic $(\mathbb{N}, +)$, full Arithmetic $(\mathbb{N}, +, \cdot)$, and any set with cardinality greater than $2^{\aleph_0}$, respectively.

The following example, due to Eric Rosen, shows that cardinality is not the only reason for the proper inclusion of $\omega$-TAutStr in DecTh.

**Lemma 5.34.** DecTh $\setminus$ $\omega$-TAutStr *contains a countable structure.*

*Proof.* The structure $\mathfrak{D}$ is constructed via diagonalisation. Consider the class $\mathscr{K}$ of graphs consisting of finite disjoint cycles. Let $(\mathfrak{A}_i)_{i \in \mathbb{N}}$ be an enumeration of all $\omega$-tree automatic structures in $\mathscr{K}$. Define $\mathfrak{D}$ as follows: $\mathfrak{D}$ contains one cycle of length $n$ iff $\mathfrak{A}_n$ does contain no such cycle. Obviously, $\mathfrak{D} \notin \omega$-TAutStr. On the other hand $\mathfrak{D} \in$ DecTh because, for every $\varphi \in$ FO, whether $\varphi \in \mathrm{Th}(\mathfrak{D})$ depends only on the existence of cycles up to a certain length. This length can be effectively determined from the quantifier rank of $\varphi$. Because of the effective semantics of automatic structures the question whether a cycle of length $n$ exists can be answered by constructing $\mathfrak{A}_n$. $\square$

# Chapter 6

# Model Theory

We turn back to logic. After showing that the compactness theorem fails for the class of automatic structures we will take a closer look at the theory of $\mathfrak{N}_p$.

## 6.1 Compactness

Very often, if one restricts the class of models—say to finite or recursive models or to constraint databases—many important tools and results of classic model theory fail. The most prominent example is compactness. Unsurprisingly in automatic model theory it also does not hold.

**Theorem 6.1.** *The compactness theorem fails for the classes* [$\omega$-][T]AutStr.

*Proof.* (Adapted from the proof for the case of recursive structures in [HH96].) Let $A \subseteq \mathbb{N}$ be any non-recursive set. Define

$$\Phi := \{\varphi_<, \varphi_S\} \cup \{\, \varphi_k \mid k \in \mathbb{N}\}$$

where

$$\begin{aligned}\varphi_< := {} & \forall xyz(\neg x < x \wedge (x < y \wedge y < z \rightarrow x < z) \\ & \qquad\qquad \wedge (x < y \vee x = y \vee y < x)) \\ & \wedge \exists x \neg \exists y (y < x) \\ & \wedge \forall x \exists y (x < y \wedge \neg \exists z (x < z \wedge z < y)),\end{aligned}$$

("$<$ is a discrete linear order with least element.")

$$\varphi_S := \forall xy(Sxy \leftrightarrow (x < y \wedge \neg \exists z (x < z \wedge z < y))),$$

("$S$ is the successor relation with respect to $<$.")

$$\varphi_k := \exists x_0 \cdots x_k \Big( \neg \exists y (y < x_0) \wedge \bigwedge_{i<k} S x_i x_{i+1} \wedge \psi_k(\overline{x}) \Big),$$

$$\psi_k := \bigwedge_{i \in \{0,\dots,k\} \cap A} U x_i \wedge \bigwedge_{i \in \{0,\dots,k\} \setminus A} \neg U x_i .$$

("$U = A \cap \{0, \dots, k\}$")

Then every finite subset $\Phi' \subseteq \Phi$ has the automatic model $(\mathbb{N}, <, S, U)$ with the usual ordering and successor relation, and $U := A \cap \{0, \ldots, m\}$ where $m := \max\{\, k \mid \varphi_k \in \Phi' \,\}$.

Suppose $\Phi$ has an automatic model $\mathfrak{A}$. Then the following algorithm can decide $A$:

> Input: $n$
> $\varphi := \exists x_0 \cdots x_n \big(\neg\exists y(y < x_0) \wedge \bigwedge_{i<n} Sx_ix_{i+1} \wedge Ux_n\big)$
> **if** $\mathfrak{A} \models \varphi$ **then**
>     **return** true
> **else**
>     **return** false

□

**Corollary 6.2.** *There is no sound and complete proof system for the set of sentences valid in* $[\omega\text{-}]$[T]AutStr.

*Proof.* We show that the existence of such a system would imply the compactness theorem. Assume there is a proof system such that $\Phi \vdash \psi$ iff $\Phi \models \psi$. If $\Phi$ is unsatisfiable then there is a proof of $\Phi \vdash$ false. In this proof only a finite number of sentences of $\Phi$ would be used. Therefore there is a finite subset $\Phi' \subseteq \Phi$ with $\Phi' \vdash$ false. By completeness this would imply $\Phi' \models$ false. Thus there is a finite unsatisfiable subset of $\Phi$. □

## 6.2 Axiomatisation of Th($\mathfrak{N}_p$)

We present an axiom system for Th($\mathfrak{N}_p$). In order to simplify the task we first construct one for the structure $\mathfrak{S}_p := (\mathbb{N}, <, s_p, (D_k)_{k<p})$ where

$$D_k := \{\, (x, y) \mid y \text{ is a power of } p \text{ and the digit of } x \text{ at position } y \text{ is } k \,\},$$
$$s_p x := p \cdot x.$$

**Proposition 6.3.** $\mathfrak{N}_p \equiv_{\mathrm{FO}} \mathfrak{S}_p$.

The proof is straightforward. It follows that any axiom system for the theory of one structure yields an axiomatisation of the other one.

We have seen in Section 4.1 that in $\mathfrak{N}_p$ every formula can be transformed into automaton normal form. This can be used to derive an axiom system of Th($\mathfrak{N}_p$) or, equivalently, one of Th($\mathfrak{S}_p$).

**Definition 6.4** (Axiom system of Th($\mathfrak{S}_p$))**.** We introduce the following abbreviations. The set $P$ of *Positions* is defined as $Px := D_1xx$. The least element of $<$ is denoted by 0, the next one by 1. Let $\mathfrak{A} = (\mathbb{Z}_p^m, \mathbb{Z}_p^n, \overline{0}, \Delta, F)$ be a deterministic automaton. The corresponding formula (see Section 4.1) is defined as

$$\psi_{\mathfrak{A}}(\overline{x}) := \exists\overline{q}\exists s[\mathrm{ADM} \wedge \mathrm{START} \wedge \mathrm{RUN} \wedge \mathrm{ACC}]$$

where

$$\begin{aligned}
\mathrm{ADM}(\overline{x},\overline{q},s) &:= Ps \wedge \bigwedge_{i<n} x_i < s,\\
\mathrm{START}(\overline{x},\overline{q},s) &:= \mathrm{Sym}_{\overline{0}}(\overline{q},1),\\
\mathrm{RUN}(\overline{x},\overline{q},s) &:= \forall z\Big(z < s \wedge Pz \to \bigvee_{\tau\in\Delta} \mathrm{Trans}_\tau(\overline{x},\overline{q},z)\Big),\\
\mathrm{ACC}(\overline{x},\overline{q},s) &:= \bigvee_{\overline{k}\in F} \mathrm{Sym}_{\overline{k}}(\overline{q},s),\\
\mathrm{Trans}_{(\overline{k},\overline{a},\overline{k}')}(\overline{x},\overline{q},z) &:= \mathrm{Sym}_{\overline{k}}(\overline{q},z) \wedge \mathrm{Sym}_{\overline{a}}(\overline{x},z) \wedge \mathrm{Sym}_{\overline{k}'}(\overline{q},s_pz),\\
\mathrm{Sym}_{\overline{a}}(\overline{x},z) &:= \bigwedge_i D_{a_i}x_iz.
\end{aligned}$$

The axiom system consists of:

(P1) $<$ is a discrete linear order with first but without last element.

$$\begin{aligned}
&\forall x\, \neg x < x\\
&\forall x\forall y\forall z(x < y \wedge y < z \to x < z)\\
&\forall x\forall y(x < y \vee x = y \vee y < x)\\
&\forall x\exists y(x < y \wedge \neg\exists z(x < z \wedge z < y))\\
&\forall x[\exists y\, y < x \to \exists y(y < x \wedge \neg\exists z(y < z \wedge z < x))]\\
&\exists x\forall y\, x \le y
\end{aligned}$$

(P2) $s_p$ is monotone.

$$\begin{aligned}
&\forall x(x > 0 \to s_px > x)\\
&s_p0 = 0
\end{aligned}$$

(P3) The least element of $P$ is 1, $P$ is unbounded, and $s_p$ is the successor function on $<|_P$.

$$\begin{aligned}
&\neg P0 \wedge P1\\
&\forall x\exists y(x < y \wedge Py)\\
&\forall x(Px \to Ps_px \wedge \neg\exists z(Pz \wedge x < z < s_px))\\
&\forall x(Px \wedge x > 1 \to \exists y(Py \wedge x = s_py))
\end{aligned}$$

(P4) Each number has exactly one colour at every position and no colour at non-positions.

$$\begin{aligned}
&\forall x\forall y \bigwedge_{i\neq k}(\neg D_ixy \vee \neg D_kxy)\\
&\forall x\forall y\Big(Py \leftrightarrow \bigvee_{k<p} D_kxy\Big)
\end{aligned}$$

(P5) Numbers are uniquely identified by their colouring.

$$\forall x\forall y[x = y \leftrightarrow \forall z(Pz \to \mathrm{SameDigit}(x,z;y,z))]$$

where $\mathrm{SameDigit}(x_1,z_1;x_2,z_2) := \bigvee_{k<p}(D_kx_1z_1 \wedge D_kx_2z_2)$.

(P6) Every number eventually has colour zero.

$$\forall x \exists y (Py \wedge \forall z (Pz \wedge z \geq y \rightarrow D_0 xz))$$

(P7) Positions have the colouring $0 \cdots 010 \cdots$.

$$\forall x \forall y (Px \wedge Py \wedge x \neq y \rightarrow D_0 xy)$$

(P8) Definition of $<$ and $s_p$ in terms of colours.

$$\forall x \forall y \Big( x < y \leftrightarrow \exists z \Big( Pz \wedge \bigvee_{i<k} (D_i xz \wedge D_k yz) \\ \wedge \forall z' (z' > z \rightarrow \mathrm{SameDigit}(x, z'; y, z')) \Big) \Big)$$

$$\forall x \forall y \bigwedge_{k<p} (D_k xy \leftrightarrow D_k s_p x s_p y)$$

$$\forall x (\exists y (x = s_p y) \leftrightarrow D_0 x 1)$$

(P9) Every periodic colouring exists. For all numbers $n \in \mathbb{N} \setminus \{0\}$ and every word $w = a_0 \cdots a_{n-1} \in \mathbb{Z}_p^n$ of length $n$ we have the axiom

$$\forall x \forall s \forall t \exists y \Big( Ps \wedge Pt \wedge s_p^n s \leq t \\ \rightarrow \forall z (z < s \vee z > t \rightarrow \mathrm{SameDigit}(x, z; y, z)) \\ \wedge \forall z (s \leq z \wedge s_p^n z \leq t \rightarrow \mathrm{SameDigit}(y, z; y, s_p^n z)) \\ \wedge \bigwedge_{i<n} D_{a_i} y s_p^i s \\ \wedge \bigvee_{i<n} \exists z \Big( s_p^n z = t \wedge \bigwedge_{k=i}^{n} D_{a_k} y s_p^{k-i} z \wedge \bigwedge_{k=0}^{i-1} D_{a_k} y s_p^{n-(i-1)+k} z \Big) \\ \wedge \forall z \Big( s \leq z \wedge s_p^{2n} z \leq t \rightarrow \bigvee_{i<n} \bigwedge_{k<n} D_{a_k} y s_p^{i+k} z \Big) \Big).$$

(Intuitively, this axiom says that for every number $x$ and all positions $s$ and $t$ of $x$ there is some other number $y$ which differs from $x$ only at the positions between $s$ and $t$. The part of $y$ between $s$ and $t$ is periodic with period $n$, it starts with $w$, ends with some suffix of $w$, and every interval of length $n$ in between contains some cyclic permutation of $w$.)

(P10) Every deterministic automaton has a unique run on each input. For all $n, m \in \mathbb{N}$, $m > 0$ and all transition relations $\Delta \subseteq \mathbb{Z}_p^m \times \mathbb{Z}_p^n \times \mathbb{Z}_p^m$ of some finite *total deterministic* automaton $(\mathbb{Z}_p^m, \mathbb{Z}_p^n, \Delta, \overline{0}, F)$—i.e., for all $\overline{q} \in \mathbb{Z}_p^m$ and $\overline{a} \in \mathbb{Z}_p^n$ there is exactly one $\overline{q}' \in \mathbb{Z}_p^m$ with $(\overline{q}, \overline{a}, \overline{q}') \in \Delta$—we have the axiom

$$\forall \overline{x} \forall s \exists^{=1} \overline{q} [\mathrm{START}(\overline{x}, \overline{q}, s) \wedge \mathrm{RUN}(\overline{x}, \overline{q}, s) \wedge \mathrm{END}(\overline{q}, s)]$$

where

$$\mathrm{END}(\overline{q}, s) := \forall z (Pz \wedge z > s \rightarrow \mathrm{Sym}_{\overline{0}}(\overline{q}, z)).$$

Note that we allow automata without input, i.e., $n = 0$. Such automata are of the form $\mathfrak{A} = (\mathbb{Z}_p^m, \mathbb{Z}_p^0, \Delta, \overline{0}, F)$ where $\mathbb{Z}_p^0 = \{\Box\}$ ($\Box$ denotes the empty tuple), $\Delta \subseteq \mathbb{Z}_p^m \times \mathbb{Z}_p^m \cong \mathbb{Z}_p^m \times \{\Box\} \times \mathbb{Z}_p^m$ and $L(\mathfrak{A})$ is either $\{\Box\}$ or $\emptyset$ depending on whether there is some $\overline{q} \in F$ with $(\overline{0}, \overline{q}) \in \mathrm{TC}(\Delta)$.

(P11) The subset construction works. For all deterministic automata $\mathfrak{A}$ and $\mathfrak{B}$ such that $\mathfrak{B}$ recognises the set defined by $\exists y\, \psi_{\mathfrak{A}}(\overline{x}y)$ we have the axiom

$$\forall \overline{x}[\exists y \psi_{\mathfrak{A}}(\overline{x}y) \leftrightarrow \psi_{\mathfrak{B}}(\overline{x})].$$

**Theorem 6.5.** *The axiom system* (P1)–(P11) *is complete.*

*Proof.* We show that (P1)–(P11) imply that each formula is equivalent to its automaton normal form using the minimal automaton. Therefore, if $\varphi$ is a sentence it has an automaton normal form $\psi_{\mathfrak{A}}$ with $\mathfrak{A} = (\{0\}, \{\Box\}, \Delta, 0, F)$ where $\Delta = \{(0, \Box, 0)\}$ and $F$ is either $\{0\}$ or $\emptyset$. In the first case (P1)–(P11) $\models \varphi$, in the other case (P1)–(P11) $\models \neg\varphi$. Thus, (P1)–(P11) is complete.

By (P1)–(P3) the set of positions is some discrete linear order with first element 1 and without last element. By (P4) every number can be seen as colouring of $P$ which by (P6) eventually becomes 0; by (P5) the colouring is unique.

By (P7) and (P8), if $z$ is a position then $x < z$ iff $D_0 x z'$ for all positions $z' \geq z$. Let $\mathfrak{A}$ be a deterministic automaton. Consider

$$\psi_{\mathfrak{A}}(\overline{x}) := \exists \overline{q} \exists s[\mathrm{ADM} \wedge \mathrm{START} \wedge \mathrm{RUN} \wedge \mathrm{ACC}].$$

By (P3) there is some $s$ satisfying ADM and by (P10) there is a unique tuple $\overline{q}$ which, given $s$, satisfies START $\wedge$ RUN. Therefore $\psi_{\mathfrak{A}}$ holds if and only if the unique run of $\mathfrak{A}$ on $\overline{x}$ contains some final state somewhere after the last position of $\overline{x}$ carrying a non-zero digit.

Now we a ready to prove the equivalence of atomic formulae to their automata. We start with equality. Let $\mathfrak{A}_= := \big(\{0,1\}, \mathbb{Z}_p^2, \Delta_=, 0, \{0\}\big)$ with

$$\begin{aligned}\Delta_= := &\{\, (0,(a,a),0) \mid a \in \mathbb{Z}_p \,\} \cup \{\, (0,(a,b),1) \mid a \neq b \,\} \\ &\cup \{\, (1,(a,b),1) \mid a, b \in \mathbb{Z}_p \,\}.\end{aligned}$$

Because of (P9) the colourings $00\cdots$ and $0\cdots01\cdots10\cdots$ exist. Therefore, by (P10) and (P6) the unique run of $\mathfrak{A}$ on some $\overline{x}$ is of one of these forms. If $x_0 = x_1$ it can only be the former, and if $x_0 \neq x_1$ it can only be the latter. Thus $\mathfrak{A}$ accepts $\overline{x}$ if and only if $x_0 = x_1$.

The other relations are handled similarly. Define

$$\begin{aligned}\mathfrak{A}_< &:= \big(\{0,1\}, \mathbb{Z}_p^2, \Delta_<, 0, \{1\}\big), \\ \mathfrak{A}_{D_k} &:= \big(\{0,1,2\}, \mathbb{Z}_p^2, \Delta_{D_k}, 0, \{1\}\big), \\ \mathfrak{A}_{s_p} &:= \big(\{0,\dots,p\}, \mathbb{Z}_p^2, \Delta_{s_p}, 0, \{0\}\big)\end{aligned}$$

with

$$\begin{aligned}\Delta_< := &\{\, (q,(a,b),0) \mid a > b, q \in \{0,1\} \,\} \\ &\cup \{\, (q,(a,b),1) \mid a < b, q \in \{0,1\} \,\} \\ &\cup \{\, (q,(a,a),q) \mid a \in \mathbb{Z}_p, q \in \{0,1\} \,\},\end{aligned}$$

$$\begin{aligned}\Delta_{s_p} := &\{\, (a,(b,a),b) \mid a, b \in \mathbb{Z}_p \,\} \cup \{\, (p,(a,b),p) \mid a, b \in \mathbb{Z}_p \,\} \\ &\cup \{\, (c,(a,b),p) \mid b \neq c, a, b, c \in \mathbb{Z}_p \,\},\end{aligned}$$

$$\begin{aligned}\Delta_{D_k} := \{\, &(0,(a,0),0),(1,(a,0),1) \mid a \in \mathbb{Z}_p \,\}\\ &\cup \{(0,(k,1),1)\} \cup \{\,(0,(a,1),2) \mid a \neq k\,\}\\ &\cup \{\,(0,(a,b),2) \mid b > 1, a,b \in \mathbb{Z}_p\,\}\\ &\cup \{\,(1,(a,b),2) \mid b \neq 0, a,b \in \mathbb{Z}_p\,\}\\ &\cup \{\,(2,(a,b),2) \mid a,b \in \mathbb{Z}_p\,\}.\end{aligned}$$

We have $x_0 < x_1$ iff, by (P8), there is some position $z$ such that the digit of $x_0$ at $z$ is greater than the digit of $x_1$ at $z$ and the digits of $x_0$ and $x_1$ are the same at all greater positions. This is the case iff in the run of $\mathfrak{A}_<$ on $(x_0, x_1)$ the state at position $s_p z$ is 1 and remains 1 until all non-zero digits of $(x_0, x_1)$ are passed. Again, the last equivalence follows since by (P9) such a run exists and by (P10) it is unique. Therefore, $x_0 < x_1$ iff $\psi_{\mathfrak{A}_<}(x_0, x_1)$.

Similarly, $D_k x_0 x_1$ iff, by (P9) and (P10), the run of $\mathfrak{A}_{D_k}$ on $(x_0, x_1)$ has the form $0\cdots 01\cdots 1$. Therefore, $D_k x_0 x_1$ iff $\psi_{\mathfrak{A}_{D_k}}(x_0, x_1)$.

Finally, $s_p x_0 = x_1$ iff, by (P9) and (P19), the run of $\mathfrak{A}_{s_p}$ on $(x_0, x_1)$ has the form $(x_1, 0)$. Therefore, $s_p x_0 = x_1$ iff $\psi_{\mathfrak{A}_{s_p}}(x_0, x_1)$.

It remains to prove that the equivalence is preserved when applying boolean connectives and quantifiers. Let $\mathfrak{A}_i = (\mathbb{Z}_p^{m_i}, \mathbb{Z}_p^n, \Delta_i, \overline{0}, F_i)$, for $i = 0, 1$, be deterministic automata recognising some set of numbers. In particular the acceptance of $\mathfrak{A}_i$ does not depend on the number of leading zeros.

$\neg\psi_{\mathfrak{A}_0}(\overline{x})$ holds iff the unique run $\overline{q}$ of $\mathfrak{A}_0$ on $\overline{x}$ does not contain a final state after the last non-zero position of $\overline{x}$ iff, by assumption, $\overline{q}$ contains some non-final state at such a position iff $\overline{\mathfrak{A}}_0 := (\mathbb{Z}_p^{m_0}, \mathbb{Z}_p^n, \Delta_0, \overline{0}, \mathbb{Z}_p^{m_0} \setminus F_0)$ accepts $\overline{x}$ iff $\psi_{\overline{\mathfrak{A}}_0}(\overline{x})$ holds.

$\psi_{\mathfrak{A}_0}(\overline{x}) \vee \psi_{\mathfrak{A}_1}(\overline{x})$ holds iff the unique run $\overline{q}_0$ of $\mathfrak{A}_0$ on $\overline{x}$ or the run $\overline{q}_1$ of $\mathfrak{A}_1$ contains a final state after the last non-zero position of $\overline{x}$ iff the run $(\overline{q}_0, \overline{q}_1)$ of

$$\mathfrak{A} := (\mathbb{Z}_p^{m_0+m_1}, \mathbb{Z}_p^n, \Delta, \overline{0}, F_0 \times \mathbb{Z}_p^{m_1} \cup \mathbb{Z}_p^{m_0} \times F_1),$$

with $\Delta$ defined componentwise according to $\Delta_0$ and $\Delta_1$, contains a final state at such a position if $\mathfrak{A}$ accepts $\overline{x}$ iff $\psi_{\mathfrak{A}}(\overline{x})$ holds.

The case of the existential quantifier immediately follows from (P11).

It remains to prove that each automaton can be minimised. Let $\overline{q}$ be the run of some automaton $\mathfrak{A}$ on input $\overline{x}$. The run $\overline{q}'$ of the minimal automaton $\mathfrak{B}$ can be obtained from $\overline{q}$ by mapping each state to the corresponding state of $\mathfrak{B}$. (Note that minimising some automaton means merging equivalent states.) If $\overline{q}'$ exists it follows by (P10) that $\psi_{\mathfrak{B}} \equiv \psi_{\mathfrak{A}}$. Consider the automaton $\mathfrak{C}$ whose states are the states of $\mathfrak{B}$ and which on input $\overline{q}$, after reading one symbol of $\overline{q}$ enters the corresponding state of the minimal automaton. Hence, the run of $\mathfrak{C}$ on input $\overline{q}$ is $\overline{0}\overline{q}'$. As $\overline{0}\overline{q}' = s_p\overline{q}'$ (with obvious abbreviations) the existence of $\overline{q}'$ follows from (P8). □

## 6.3 Non-Standard Models

The axiom system of the previous section can be used to construct non-standard models of $\mathrm{Th}(\mathfrak{S}_p)$ and $\mathrm{Th}(\mathfrak{N}_p)$. Of course, we are mainly interested in non-standard models which are automatic, but so far the author has only been able to construct a recursive one.

**Definition 6.6.** $\tilde{\mathfrak{S}}_p := (S, <, s_p, (D_k)_k)$ is the structure of "intermediately periodic" $(\omega+\zeta)$-words where $\zeta = \omega^* + \omega$ is the order type of the integers and the universe $S$ consists of all words $w \in \mathbb{Z}_p^{\omega+\zeta}$ such that there are finite words $x$, $y$, $z \in \mathbb{Z}_p^*$ with $w = xy^\omega y^{\omega^*} z0^\omega$. The relations are defined the canonical way:

$$\begin{aligned} x < y \ &:\text{iff}\ \ x = uiv,\ y = u'kv \text{ for some } u, u', v, \text{ and } i < k \in \mathbb{Z}_p, \\ D_k xy \ &:\text{iff}\ \ x = ukv \text{ and } y = 0^{|u|}10\cdots \text{ for some } u,\ v, \\ s_p x \ &:= 0x. \end{aligned}$$

**Proposition 6.7.** *$\tilde{\mathfrak{S}}_p$ is a recursive non-standard model of* $\mathrm{Th}(\mathfrak{S}_p)$.

*Proof.* $\tilde{\mathfrak{S}}_p$ obviously satisfies (P1)–(P9). Consider two runs $\overline{q}_0$, $\overline{q}_1$ of some deterministic automaton $\mathfrak{A}$ on input $\overline{x}$. By definition there are decompositions

$$\overline{q}_0 = \overline{x}_0\overline{y}_0^\omega\overline{y}_0^{\omega^*}\overline{z}_0\overline{0}^\omega \qquad \text{and} \qquad \overline{q}_1 = \overline{x}_1\overline{y}_1^\omega\overline{y}_1^{\omega^*}\overline{z}_1\overline{0}^\omega.$$

Clearly, the initial parts of both runs must be identical $\overline{x}_0\overline{y}_0^\omega = \overline{x}_1\overline{y}_1^\omega$. Thus, $\overline{x}_0\overline{y}_0^\omega\overline{y}_0^{\omega^*} = \overline{x}_1\overline{y}_1^\omega\overline{y}_1^{\omega^*}$ and therefore $\overline{q}_0 = \overline{q}_1$ which yields (P10). Analogously, (P11) holds because, when reading the initial part of the input $\overline{xy}^\omega\overline{y}^{\omega^*}\overline{z}0^\omega$, the set of reachable states must eventually become periodic and this period is preserved faithfully when crossing the infinite gap.

Since the order types of $\mathfrak{S}_p$ and $\tilde{\mathfrak{S}}_p$ are different, they cannot be isomorphic and $\tilde{\mathfrak{S}}_p$ is really non-standard.

Each element $xy^\omega y^{\omega^*}z$ of $\tilde{\mathfrak{S}}_p$ can be stored as $(x, y, z_1, z_2) \in (\mathbb{Z}_p^*)^4$ where $z_1$ is the part of $z = z_1z_2$ which lies before position $\omega + \omega^*$. Obviously, using this encoding all relations can be checked effectively. Thus, $\tilde{\mathfrak{S}}_p$ is recursive. □

From $\tilde{\mathfrak{S}}_p$ one easily obtains a recursive non-standard model $\tilde{\mathfrak{N}}_p$ of $\mathrm{Th}(\mathfrak{N}_p)$ by applying the interpretation $\mathfrak{N}_p \leq_{\mathrm{FO}} \mathfrak{S}_p$.

*Open Problem.* Is there an automatic non-standard model of $\mathrm{Th}(\mathfrak{N}_p)$?

Since the order type of $\tilde{\mathfrak{N}}_p$ is $\omega + \zeta\eta$ this problem is related to the question whether $(\mathbb{Q}, +)$ is automatic.

**Lemma 6.8.** *If $\tilde{\mathfrak{S}}_p$ as constructed above is in* AutStr *then* $(\mathbb{Q}, +, |_p) \in$ AutStr *where $+$ and $|_p$ are defined the canonical way.*

*Proof.* We proceed in several steps. First applying the interpretation of $\mathfrak{N}_p$ in $\mathfrak{S}_p$ we obtain an automatic non-standard model $\tilde{\mathfrak{N}}_p$ of $\mathrm{Th}(\mathfrak{N}_p)$. Since the set $I$ of infinite powers of $p$ is $\mathrm{FO}(\exists^\omega)$-definable by

$$\varphi(x) := P_p x \wedge \exists^\omega y\, y \leq x,$$

the expansion $(\tilde{N}, +, |_p, I)$ of $\tilde{\mathfrak{N}}_p$ is automatic as well. Finally, we construct an interpretation of $(\mathbb{Q}, +, |_p, I)$ in this structure by identifying two elements of $\tilde{N}$ if their difference is finite.

$$\delta(x) := \text{true}, \qquad \varepsilon(x, y) := \forall z (Iz \rightarrow |x - y| < z).$$

$+$ and $|_p$ are defined the obvious way.

$$\varphi_+(x, y, z) := \varepsilon(x + y, z), \qquad \varphi_{|_p}(x, y) := Ix \wedge x \,|_p\, y.$$

□

*Open Problem.* Are $(\mathbb{Q}, +)$ or $(\mathbb{Q}, +, |_p)$ in $[\omega$-][T]AutStr?

Another partial answer to the first problem provides the following observation.

**Proposition 6.9.** *If there exists an automatic non-standard model* $\mathfrak{A}$ *of* $\mathfrak{N}_p$ *then* $\mathfrak{A}$ *is not a reduct of a non-standard model of Peano Arithmetic.*

*Proof.* It is a well known result of recursive model theory that in any non-standard model of Peano Arithmetic both addition and multiplication are not recursive (see e.g. [Sch98]). □

# Chapter 7

# Unary Presentations

The kind of automatic presentations we have used so far have two main disadvantages. While the FO-theories of automatic structures are decidable, their complexity can be non-elementary and more expressive logics like FO(DTC) are already undecidable. The other problem is of a methodical nature. It seems to be very difficult to show that some structure is not automatic and thus to give exact characterisations of the various classes of automatic structures.

In this chapter we will investigate a certain restricted type of presentations in the hope that stronger logics become decidable, the complexity of various operations decreases, or that at least more powerful theoretical techniques become available.

Our main method in the investigation of presentations was to calculate bounds on the length of encodings. In the special case of languages over a unary alphabet a word is completely determined by its length. Therefore, we take a closer look at this case.

The class of structures $\mathfrak{A} \in \mathrm{AutStr}[\tau]$ possessing a unary automatic presentations, i.e., a presentation over a unary alphabet, is denoted by $1\mathrm{AutStr}[\tau]$. Many of the basic properties proved in Chapter 3 for automatic structures—such as the effective semantics for $\mathrm{FO}(\exists^\omega)$—remain valid for 1AutStr. One notable exception is that 1AutStr is only closed under 1-dimensional $\mathrm{FO}(\exists^\omega)$-interpretations.

## 7.1 Complete Structure

Again, our aim will be to characterise 1AutStr via a complete structure. This structure is $\mathfrak{N}_1 := \big(\mathbb{N}, \leq, (n \mid x)_{n\in\mathbb{N}}\big)$, the natural numbers with ordering and divisibility predicates or, equivalently,

$$\mathfrak{N}_1' := \big(\mathbb{N}, s, \leq, 0, (x \equiv k \pmod n))_{k,n\in\mathbb{N}}\big)$$

where $s$ is the successor function, $\leq$ is the natural order, and $x \equiv k \pmod n$ denotes those numbers which are congruent $k$ modulo $n$.

**Definition 7.1.** Let $\overline{x}, \overline{y} \in \mathbb{N}^n$. Define

$$o(\overline{x}) := \{\, \pi \in S_n \mid x_{\pi 0} \leq \cdots \leq x_{\pi(n-1)} \,\},$$
$$\Delta(\overline{x}) := \big(x_{\pi 0}, x_{\pi 1} - x_{\pi 0}, \dots, x_{\pi(n-1)} - x_{\pi(n-2)}\big) \quad \text{for some/all } \pi \in o(\overline{x}).$$

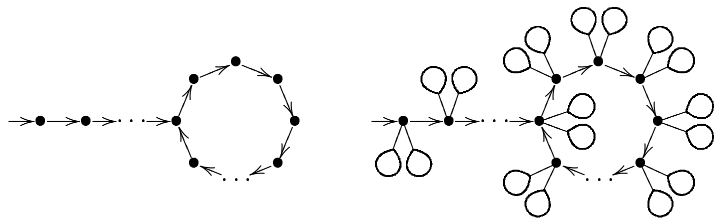


Figure 7.1: Automata over $1^*$ and $1^* \otimes 1^*$

$\sim_{l,p}$ is the equivalence relation defined by

$$\overline{x} \sim_{l,p} \overline{y} \;:\text{iff}\; o(\overline{x}) = o(\overline{y}) \text{ and } \Delta(\overline{x})_i \sim_{l,p} \Delta(\overline{y})_i \text{ for all } i < n,$$

where by abuse of notation $\sim_{l,p}$ denotes the equivalence relation

$$x \sim_{l,p} y \;:\text{iff}\; \text{either } x = y < l, \text{ or } x, y \geq l \text{ and } x \equiv y \pmod{p}.$$

Our main lemma to prove the completeness of $\mathfrak{N}_1$ is the following characterisation of regular languages. The general structure of automata over $1^* \otimes \cdots \otimes 1^*$ is depicted in Figure 7.1. The inner loop of the second automaton is labelled by $\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$, the outer loops by $\left[\begin{smallmatrix}\Box\\1\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix}1\\\Box\end{smallmatrix}\right]$, respectively.

**Lemma 7.2.** *$L \subseteq (1^*)^{\otimes n}$ is regular if and only if there are constants $l$, $p \in \mathbb{N}$ such that for all $\overline{x}$, $\overline{y} \in \mathbb{N}^n$ with $\overline{x} \sim_{l,p} \overline{y}$ it holds that*

$$1^{x_0} \otimes \cdots \otimes 1^{x_{n-1}} \in L \iff 1^{y_0} \otimes \cdots \otimes 1^{y_{n-1}} \in L.$$

*Proof.* ($\Rightarrow$) Induction on $n$.

($n = 1$) $L \subseteq \{1\}^*$ is regular iff it is a finite union of arithmetical progressions (see [Eil74, Proposition V.1.1]).

($n > 1$) Let $\mathfrak{A} = (Q, \{1\}, \delta, q_0, F)$ be a deterministic automaton recognising $L$. For each pair $q \in Q$, $R \subseteq Q$ denote by $\mathfrak{A}_{qR}$ the automaton $\mathfrak{A}_{qR} := (Q, \{1\}, \delta, q, R)$, and let $\mathfrak{A}^i_{qR}$ be the automaton obtained from $\mathfrak{A}_{qR}$ by erasing all transitions whose label has as $i^{\text{th}}$ component a $\Box$, and by removing the $i^{\text{th}}$ component of all other labels. Then, if $x_i = \max\{x_0, \ldots, x_{n-1}\}$ we have

$$\begin{aligned} &1^{x_0} \otimes \cdots \otimes 1^{x_{n-1}} \in L(\mathfrak{A}) \\ \text{iff } &1^{x_0} \otimes \cdots \otimes 1^{x_{i-1}} \otimes 1^{x_{i+1}} \otimes \cdots \otimes 1^{x_{n-1}} \in L(\mathfrak{A}^i_{q_0\{q\}}) \end{aligned}$$

for some $q \in Q$ such that

$$\varepsilon^{\otimes i-1} \otimes 1^{\Delta(\overline{x})_{n-1}} \otimes \varepsilon^{\otimes n-i} \in L(\mathfrak{A}_{qF}).$$

Let $l^i_q$, $p^i_q \in \mathbb{N}$ be the constants for $L(\mathfrak{A}^i_{q_0\{q\}})$ provided by the induction hypothesis and let $\tilde{l}^i_q$, $\tilde{p}^i_q \in \mathbb{N}$ be the corresponding constants for the language

$$L(\mathfrak{A}^i_{qF}) \cap (\Box^{i-1} \times \{1\} \times \Box^{n-i})^*$$

(as language over the unary alphabet $\{(\Box, \ldots, \Box, 1, \Box, \ldots, \Box)\}$). Define

$$l := \max\{\, l^i_q, \tilde{l}^i_q \mid i < n,\, q \in Q \,\}, \qquad p := \prod_{i,q} p^i_q \tilde{p}^i_q.$$

Then we obtain for all $\overline{x}, \overline{y} \in \mathbb{N}^n$ with $\overline{x} \sim_{l,p} \overline{y}$ that

$$\begin{aligned}
&1^{x_0} \otimes \cdots \otimes 1^{x_{n-1}} \in L(\mathfrak{A}) \\
\text{iff } &1^{x_0} \otimes \cdots \otimes 1^{x_{i-1}} \otimes 1^{x_{i+1}} \otimes \cdots \otimes 1^{x_{n-1}} \in L(\mathfrak{A}^i_{q_0\{q\}}) \\
&\text{for } i = o(\overline{x})(n-1) \text{ and some } q \in Q \text{ such that} \\
&\quad \varepsilon^{\otimes i-1} \otimes 1^{\Delta(\overline{x})_{n-1}} \otimes \varepsilon^{\otimes n-i} \in L(\mathfrak{A}_{qF}) \\
\text{iff } &1^{y_0} \otimes \cdots \otimes 1^{y_{i-1}} \otimes 1^{y_{i+1}} \otimes \cdots \otimes 1^{y_{n-1}} \in L(\mathfrak{A}^i_{q_0\{q\}}) \\
&\text{for } i = o(\overline{x})(n-1) = o(\overline{y})(n-1) \text{ and some } q \in Q \text{ such that} \\
&\quad \varepsilon^{\otimes i-1} \otimes 1^{\Delta(\overline{y})_{n-1}} \otimes \varepsilon^{\otimes n-i} \in L(\mathfrak{A}_{qF}) \\
\text{iff } &1^{y_0} \otimes \cdots \otimes 1^{y_{n-1}} \in L(\mathfrak{A}).
\end{aligned}$$

($\Leftarrow$) For each $\sim_{l,p}$-class one can easily construct an automaton recognising this class. As regular languages are closed under union the claim follows. □

For lack of a better name, we call the numbers $l$ and $p$ of the preceding lemma the *loop constants* of $L$.

**Definition 7.3.** The *loop constants* of a unary presentation $\mathfrak{d}$ consists of a pair $(l, p)$ such that $l$ and $p$ are loop constants of every language of $\mathfrak{d}$. W.l.o.g. we always assume that $l < p$.

For $R \subseteq \mathbb{N}^n$ define $\operatorname{code}(R) := \{\, 1^{x_0} \otimes \cdots \otimes 1^{x_{n-1}} \mid (x_0, \dots, x_{n-1}) \in R \,\}$.

**Theorem 7.4.** *$R \subseteq \mathbb{N}^n$ is* FO*-definable in $\mathfrak{N}_1$ if and only if* $\operatorname{code}(R)$ *is regular.*

*Proof.* ($\Rightarrow$) $\mathfrak{N}_1$ has a unary automatic presentation

$$\mathfrak{d} := \big(\nu, \{1\}, L_\delta, L_\varepsilon, L_\leq, (L_n)_n\big)$$

with

$$\nu(1^x) := x, \qquad L_\delta := 1^*, \qquad L_n := (1^n)^*,$$
$$L_\varepsilon := \begin{bmatrix}1\\1\end{bmatrix}^*, \qquad L_\leq := \begin{bmatrix}1\\1\end{bmatrix}^* \begin{bmatrix}\square\\1\end{bmatrix}^*.$$

($\Leftarrow$) If $\operatorname{code}(R)$ is regular then it is a union of some $\sim_{l,p}$-classes where $(l, p)$ are the loop constants of $\operatorname{code}(R)$. One such class can be defined (in $\mathfrak{N}'_1$) by the formula

$$\varphi(\overline{x}) = \bigwedge_{i<n-1} x_{\pi i} \leq x_{\pi(i+1)} \wedge \psi_0(x_{\pi 0}) \wedge \bigwedge_{i<n-1} \psi_{i+1}(x_{\pi(i+1)} - x_{\pi i})$$

where each $\psi_i(x - y)$ is either of the form

$$\psi(x-y) := x - y = m \quad \big(\equiv x = s^m y\big)$$

or

$$\psi(x-y) := x - y \geq l \wedge x - y \equiv k \pmod p$$
$$\Big(\equiv x \geq s^l y \wedge \bigvee_{i<p} (x \equiv i + k \pmod p \wedge y \equiv i \pmod p)\Big).$$

Hence, $\operatorname{code}(R)$ can be defined by a disjunction with one such formula for each $\sim_{l,p}$-class contained in $R$. □

As a corollary we obtain the desired characterisation of 1AutStr in terms of a complete structure.

**Corollary 7.5.** $\mathfrak{A} \in$ 1AutStr iff $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{N}_1$ *via a* 1-*dimensional interpretation.*

We will see below that 1AutStr is not closed under products and hence under many-dimensional interpretations. A more robust class is obtained if we take the closure of $\mathfrak{N}_1$ under many-dimensional FO-interpretations. This corresponds to presentations where all languages are subsets of $(1^*)^{\otimes k}$, for some $k$, instead of $1^*$. In the following we only consider 1AutStr, which is simple enough to permit precise characterisations of the structures it contains.

## 7.2 Structures with Unary Presentation

The following example shows that unary presentations are much weaker than those with a binary alphabet.

*Example.* Presburger Arithmetic $(\mathbb{N}, +)$ has no unary automatic presentation.

*Proof.* Suppose $(\mathbb{N}, +)$ has a unary presentation $\mathfrak{d}$. Define

$$N_n := \{\, m \in \mathbb{N} \mid \lambda^{\mathfrak{d}}(m) \leq n \,\} \setminus \{0\}$$

and let $m_n := \max N_n$. Then $|N_n + m_n| = |N_n|$, and since $\lambda^{\mathfrak{d}}(x) > n$ for all $x \notin N_n$ there is some $x_n \in N_n$ with

$$\lambda^{\mathfrak{d}}(x_n + m_n) \geq \lambda^{\mathfrak{d}}(m_n) + |N_n| = \max\{\lambda^{\mathfrak{d}}(m_n), \lambda^{\mathfrak{d}}(x_n)\} + |N_n| \,.$$

As $|N_n|$ is unbounded for $n \to \infty$ we get a contradiction to Proposition 5.8. □

**Proposition 7.6.**

(i) 1AutStr *is not closed under products.*

(ii) 1AutStr *is closed under finite disjoint unions and finite ordered sums.*

*Proof.* (i) Consider $(\mathbb{N}, s)$, the $s$-reduct of $\mathfrak{N}_1'$. We claim that $(\mathbb{N}^2, s) := (\mathbb{N}, s) \times (\mathbb{N}, s)$ has no unary presentation. Let

$$M := \{\, (n,0),\ (0,n) \in \mathbb{N}^2 \mid n \in \mathbb{N} \},$$

which is definable by $\varphi(x) := \neg\exists y(x = sy)$. Consider the sequence $(G_n(M))_n$ of generations of $M$. As $\langle x\rangle_s \cap \langle y\rangle_s = \emptyset$ for all different $x$, $y \in M$ the size $\gamma(n)$ of $G_n(M)$ is equal to

$$\gamma(n) = \gamma(n-1) + n - 1 + 1 = \gamma(n-1) + n = \sum_{i=1}^{n} i = n(n-1)/2.$$

But, according to Proposition 5.8, $|G_n(M)| \leq mn$ for some $m$ because in the unary case there can be only one word of each length.

(ii) Let, for $i \in \{0, 1\}$, $\mathfrak{A}_i \in$ 1AutStr with presentation

$$\mathfrak{d}_i = (\nu_i, \{1\}, L^i_\delta, L^i_\varepsilon, L^i_{R_0}, \ldots, L^i_{R_r}).$$

Define the homomorphism $h : \{1, \Box\}^* \to \{1, \Box\}^*$ by $h(1) := 11$, $h(\Box) := \Box\Box$. We identify $h$ with its extension to $1^{\otimes k}$ (defined componentwise). Then $\mathfrak{A}_0 \uplus \mathfrak{A}_1$ has the presentation $\mathfrak{d} := (\nu, \{1\}, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$ where

$$\begin{aligned}
\nu(1^k) &:= \begin{cases} \nu_0(1^{k/2}) & \text{if } k \text{ is even,} \\ \nu_1(1^{(k-1)/2}) & \text{if } k \text{ is odd,} \end{cases} \\
L_\delta &:= h(L^0_\delta) \cup 1h(L^1_\delta), \\
L_\varepsilon &:= h(L^0_\varepsilon) \cup \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right] h(L^1_\varepsilon), \\
L_{R_j} &:= h(L^0_{R_j}) \cup \begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} h(L^1_{R_j}).
\end{aligned}$$

That is, elements of $\mathfrak{A}_0$ are mapped to even numbers, those of $\mathfrak{A}_1$ to odd ones. In case of the ordered sum we additionally define

$$\begin{aligned}
L_\leq := {} & h(L^0_\leq) \cup \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right] h(L^1_\leq) \\
& \cup \left(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\right)^* \left(\left[\begin{smallmatrix}\Box\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}\Box\\1\end{smallmatrix}\right]\right)^* \left[\begin{smallmatrix}\Box\\1\end{smallmatrix}\right] \\
& \cup \left(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\right)^* \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]\left[\begin{smallmatrix}1\\\Box\end{smallmatrix}\right] \left(\left[\begin{smallmatrix}1\\\Box\end{smallmatrix}\right]\left[\begin{smallmatrix}1\\\Box\end{smallmatrix}\right]\right)^*.
\end{aligned}$$

□

**Corollary 7.7.** 1AutStr *is not closed under many-dimensional* FO*-interpretations.*

In the remainder of this section we try to give precise characterisations of those structures having a unary presentation. The main work is done in the following technical lemmas. Let $f : A \times A^n \to A$. Define

$$f^0(a, \bar{b}) := a, \qquad f^{i+1}(a, \bar{b}) := f(f^i(a, \bar{b}), \bar{b}).$$

The set $f^*(a, \bar{b}) := \{ f^n(a, \bar{b}) \mid n \in \mathbb{N} \}$ is called the *$f$-chain of $a$* (with parameters $\bar{b}$).

**Lemma 7.8.** *Let $(A, f) \in$ 1AutStr for some $f : A \to A$. There are only finitely many disjoint infinite $f$-chains.*

*Proof.* Let $\mathfrak{d}$ be a unary presentation of $(A, f)$ and let $m$ be some constant such that $\lambda^\mathfrak{d}(f(a)) \leq \lambda^\mathfrak{d}(a) + m$. Suppose there are infinitely many infinite $f$-chains $f^*(a_0), f^*(a_1), \dots$. Let $k := \max\{ \lambda^\mathfrak{d}(a_i) \mid i \leq m \}$. For each $i \leq m$ let $b_i \in f^*(a_i)$ the element with minimal length $\lambda^\mathfrak{d}(b_i) \geq k$. W.l.o.g. assume that $\lambda^\mathfrak{d}(b_0) < \dots < \lambda^\mathfrak{d}(b_m)$. By minimality, $\lambda^\mathfrak{d}(b_m) < k + m$. Thus

$$k \leq \lambda^\mathfrak{d}(b_0) < \dots < \lambda^\mathfrak{d}(b_m) < k + m.$$

Contradiction. □

**Lemma 7.9.** *Let $\mathfrak{d}$ be a unary presentation of $(A, f)$ where $f : A \times A^n \to A$. The sequence*

$$\left(\lambda^\mathfrak{d}(f^{i+1}a\bar{b}) - \lambda^\mathfrak{d}(f^i a\bar{b})\right)_{i \in \mathbb{N}}$$

*is eventually periodic for all $a$ and $\bar{b}$ in $A$. Furthermore, the period can be chosen to be independent of $a$ and $\bar{b}$.*

*Proof.* If $f^n a\bar{b} = f^{n+k} a\bar{b}$ for some $n$ and $k$ the claim follows immediately. Otherwise, let $(l, p)$ be the loop constants of $\mathfrak{d}$. W.l.o.g. assume that $l > \lambda^{\mathfrak{d}}(b_j)$ for all parameters $b_j$. Choose $i_0$ large enough such that $\lambda^{\mathfrak{d}}(f^i a\bar{b}) > l$ for all $i \geq i_0$. We claim that

$$\lambda^{\mathfrak{d}}(a) \equiv \lambda^{\mathfrak{d}}(a') \pmod{p} \quad \text{implies} \quad \lambda^{\mathfrak{d}}(fa\bar{b}) - \lambda^{\mathfrak{d}}(a) = \lambda^{\mathfrak{d}}(fa'\bar{b}) - \lambda^{\mathfrak{d}}(a')$$

for all $a$, $a' \in A$ such that $\lambda^{\mathfrak{d}}(a)$, $\lambda^{\mathfrak{d}}(a')$, $\lambda^{\mathfrak{d}}(fa\bar{b})$, and $\lambda^{\mathfrak{d}}(fa'\bar{b})$ are greater than or equal to $l$. The result follows since the sequence $(\lambda^{\mathfrak{d}}(f^i a\bar{b}) \bmod p)_i$ for $i_0 \leq i \leq i_0 + p$ must contain at least one number twice. Hence by the claim, the part in between is repeated infinitely. Furthermore, we can choose $p!$ as period which is independent of $a$ and $\bar{b}$.

To prove the claim suppose by symmetry, $\lambda^{\mathfrak{d}}(a) \leq \lambda^{\mathfrak{d}}(a')$. Since $f$ is a function, either $\lambda^{\mathfrak{d}}(a) - l < \lambda^{\mathfrak{d}}(fa\bar{b}) < \lambda^{\mathfrak{d}}(a) + l$ or $\lambda^{\mathfrak{d}}(fa\bar{b}) < l$. If $\lambda^{\mathfrak{d}}(a) > l$ and $\lambda^{\mathfrak{d}}(fa\bar{b}) > l$ then

$$(\lambda^{\mathfrak{d}}(a), \lambda^{\mathfrak{d}}(fa\bar{b})) \sim_{l,p} (\lambda^{\mathfrak{d}}(a) + ip, \lambda^{\mathfrak{d}}(fa\bar{b}) + ip)$$

for all $i > 0$. Thus, if $\lambda^{\mathfrak{d}}(a') \equiv \lambda^{\mathfrak{d}}(a) \pmod{p}$ then $\lambda^{\mathfrak{d}}(a') = \lambda^{\mathfrak{d}}(a) + ip$ for some $i$. Therefore, $\lambda^{\mathfrak{d}}(fa'\bar{b}) = \lambda^{\mathfrak{d}}(fa\bar{b}) + ip$, and

$$\lambda^{\mathfrak{d}}(fa'\bar{b}) - \lambda^{\mathfrak{d}}(a') = \lambda^{\mathfrak{d}}(fa\bar{b}) + ip - \lambda^{\mathfrak{d}}(a) - ip = \lambda^{\mathfrak{d}}(fa\bar{b}) - \lambda^{\mathfrak{d}}(a).$$

□

**Lemma 7.10.** *Let $\mathfrak{A} \in$ 1AutStr, $f$ a unary function of $\mathfrak{A}$, and $a$ some element of $\mathfrak{A}$. Every presentation $\mathfrak{d}$ of $\mathfrak{A}$ can effectively be extended to one of $(\mathfrak{A}, R)$ where $R := \{ (a, b) \mid b \in f^*(a) \}$.*

*Proof.* Let $\mathcal{I}$ be an interpretation of $\mathfrak{A}$ in $\mathfrak{N}_1$, and let $\mathfrak{d}$ be the corresponding presentation. For notational simplicity we identify elements of $\mathfrak{A}$ with their encodings in $\mathbb{N}$. We have to construct a formula $\varphi(x, y)$ for $R$.

In a first step we define a formula $\psi_a(y)$ describing $f^*a$ for fixed $a$. If $f^*a$ is finite $\psi_a(y)$ simply consists of an enumeration of its elements. Otherwise, by Lemma 7.9, there is a constant $q$ such that

$$f^{i+1}a - f^i a = f^{q+i+1}a - f^{q+i}a$$

for all $i$ greater than some $i_0$. (Recall that we identify $a$ with $\lambda^{\mathfrak{d}}(a)$.) Thus, $f^{q+i}a = f^i a + \Delta$ for some $\Delta$ which is positive by infinity of $f^*a$. Hence, we can set

$$\psi_a(y) := \bigvee_{i \leq i_0} y = f^i a \vee \bigvee_{i_0 < i \leq i_0 + q} \big(y \geq f^i a \wedge y \equiv f^i a \pmod{\Delta}\big).$$

In the second step we construct $\varphi$. Let $(l, p)$ be the loop constants of $\mathfrak{d}$. Choose the threshold $m := l(p + 2)$. The $f$-chains of all elements less than $m$ are defined by

$$\chi(x, y) := \bigvee_{k < m} \big(x = k \wedge \psi_k(y)\big).$$

For each $k < p$, the $f$-chains of all elements $a \equiv m + k \pmod{p}$ greater than $m$ are handled by a single formula $\vartheta_k(x, y)$. Consider the $f$-chain of $m + k$. By the preceding lemma, there is some number $i_k$ such that the sequence

$$(f^{i+1}(m + k) - f^i(m + k))_i$$

is periodic for $i \geq i_k$. Denote the period by $p_k$ and let $\Delta_k$ be the constant such that $f^{p_k+i}(m+k) = f^i(m+k) + \Delta_k$ for $i \geq i_k$. Note that

(i) either $a - l < fa < a + l$ or $fa < l$;

(ii) if $a > l$ and $fa > l$ then for all $b \geq a$ with $b \equiv a \pmod p$ we have $(a, fa) \sim_{l,p} (b, b + fa - a)$. Thus $fb = fa + b - a$.

Suppose that $f^i(m+k) > l$ for all $i \leq j$. Then, by (ii),

$$f^i(m + k + pn) = f^i(m + k) + pn$$

for $i \leq j$ and all $n \geq 0$. By choice of $m$ and (i), we either have $f^i(m+k) > l$ for $i \leq p$, or there is some $j < p$ such that $f^i(m+k) > 2l$ for $i < j$ and $f^j(m+k) < l$.

First consider the second case. We have

$$f^i(m + k + pn) = \begin{cases} f^i(m+k) + pn & \text{for } i < j, \\ f^i(m+k) & \text{for } i \geq j, \end{cases}$$

where the case $i \geq j$ follows because of

$$\big(f^{j-1}(m+k), f^j(m+k)\big) \sim_{l,p} \big(f^{j-1}(m+k) + pn, f^j(m+k)\big).$$

Thus we can define

$$\vartheta_k(x, y) := \bigvee_{i<j} \big(y - x = f^i(m+k) - (m+k)\big) \vee \psi_{f^j(m+k)}(y).$$

Note that $f^i(m+k) - (m+k)$ is a constant.

Now assume $f^i(m+k) > l$ for every $i \leq p$. Then

$$f^i(m + k + pn) = f^i(m + k) + pn$$

for all $i \leq p$. As the sequence $(f^i(m+k))_{i \leq p}$ must contain two elements which are congruent modulo $p$, the first period appears before position $p$, i.e., $i_k + p_k \leq p$. To define $\vartheta_k(x, y)$ we consider the following cases.

If $\Delta_k > 0$ then $f^i(m+k) > l$ for all $i$. Thus we define

$$\begin{aligned} \vartheta_k(x, y) := & \bigvee_{i<i_k} y - x = f^i(m+k) - (m+k) \\ & \vee \bigvee_{i_k \leq i < i_k + q} \big(y - x \geq f^i(m+k) - (m+k) \wedge \\ & \qquad\qquad y - x \equiv f^i(m+k) - (m+k) \pmod{\Delta_k}\big). \end{aligned}$$

If $\Delta_k = 0$ then

$$\vartheta_k(x, y) := \bigvee_{i<i_k+q} y - x = f^i(m+k) - (m+k).$$

The most complicated case is $\Delta_k < 0$. We split the definition into two parts by choosing some intermediate element $c \in f^*(m+k)$ with $l < c < m$. The initial part of the chain up to $c$ is defined by

$$\begin{aligned} \vartheta_k^1(x, y) := & \bigvee_{i<i_k} y - x = f^i(m+k) - (m+k) \\ & \vee \bigvee_{i_k \leq i < i_k + p_k} \big(y \geq l \wedge y - x \leq f^i(m+k) - (m+k) \wedge \\ & \qquad\qquad y - x \equiv f^i(m+k) - (m+k) \pmod{\Delta_k}\big), \end{aligned}$$

and the final part by

$$\vartheta_k^2(x,y) := \exists z\big(l < z < m \wedge \vartheta_k^1(x,z) \wedge \chi(z,y)\big).$$

Thus $\vartheta_k(x,y) := \vartheta_k^1(x,y) \vee \vartheta_k^2(x,y)$.

Altogether we obtain

$$\varphi(x,y) := \chi(x,y) \vee \bigvee_{k<p} \big(x \equiv m+k \pmod p \wedge x \geq m \wedge \vartheta_k(x,y)\big).$$

It should be clear that all constants needed in the above construction can be obtained effectively. □

**Unary functions.** Analogously to Proposition 5.18 we obtain

**Proposition 7.11.** *Let* $(\mathbb{N}, s, f) \in 1\text{AutStr}$ *where* $s$ *is the successor function and* $f : \mathbb{N} \to \mathbb{N}$.

(i) *There is a constant* $c$ *such that* $f(n) \leq n + c$ *for all* $n \in \mathbb{N}$.

(ii) *If* $\liminf_{n\to\infty} f(n) = \infty$ *then there are constants* $c_0$ *and* $c_1$ *such that*

$$n - c_0 \leq f(n) \leq n + c_1$$

*for all but finitely many* $n$.

*Proof.* (i) By the Lemma 7.9 applied to $s$, there is a constant $q$ such that

$$\lambda^{\mathfrak{d}}(s^{i+1}0) - \lambda^{\mathfrak{d}}(s^i 0) = \lambda^{\mathfrak{d}}(s^{q+i+1}0) - \lambda^{\mathfrak{d}}(s^{q+i}0)$$

for large enough $i$. Thus $\lambda^{\mathfrak{d}}(s^{q+i}0) = \lambda^{\mathfrak{d}}(s^i 0) + \Delta$ for some $\Delta$. If $f(n) - n$ is unbounded then for all $m$ there is some $n$ with

$$f(n) > n + mq \implies \lambda^{\mathfrak{d}}(f(n)) > \lambda^{\mathfrak{d}}(n) + m\Delta + r$$

where $r := \min\{\, \lambda^{\mathfrak{d}}(s^{n+i}0) - \lambda^{\mathfrak{d}}(s^n 0) \mid i < q \,\}$. But $\lambda^{\mathfrak{d}}(f(n)) - \lambda^{\mathfrak{d}}(n)$ is bounded. Contradiction.

(ii) Define

$$g(n) := \min\{\, f(k) \mid k \geq n \,\}, \qquad h(n) := \max\{\, k \mid g(k) \leq n \,\}.$$

Both functions are monotone. Suppose $n - f(n) \leq n - g(n)$ is unbounded, i.e., for all $c$ there are $n$ with

$$n - c \geq g(n) \implies h(n-c) \geq h(g(n)) \geq n.$$

Thus, for all $c$ there are $n$ with $h(n) \geq n + c$ in contradiction to (i). □

For structures with a permutation a precise characterisation is possible.

**Theorem 7.12** (Khoussainov, Rubin [KR99]). *Let* $f : A \to A$ *be a bijective function.* $(A, f) \in 1\text{AutStr}$ *if and only if*

(i) *the cardinality of the finite orbits of* $f$ *is bounded and*

(ii) *there are only finitely many infinite orbits of* $f$.

*Proof.* ($\Leftarrow$) Since 1AutStr is closed under finite unions and it contains every finite structure, we only need to prove the claim for structures with one infinite orbit and structures with infinitely many finite orbits of the same size. For the first case we construct an interpretation $\mathcal{I} = (h, \delta, \varepsilon, \varphi_s)$ of $(\mathbb{Z}, s)$ in $\mathfrak{N}_1$ where

$$h(n) := \begin{cases} 2n & \text{if } n \geq 0, \\ -2n - 1 & \text{if } n < 0, \end{cases}$$

$\delta$ and $\varepsilon$ are trivial, and

$$\varphi_s(x, y) := (2 \mid x \wedge y = x + 2) \vee (2 \nmid x \wedge y + 2 = x) \vee (x = 1 \wedge y = 0).$$

For the other case consider the structure $(\mathbb{N}, f)$ where

$$f(x) := \begin{cases} x + 1 & \text{if } n \nmid (x + 1), \\ x + 1 - n & \text{otherwise}, \end{cases}$$

which has infinitely many orbits of size $n$. $f$ can be defined in $\mathfrak{N}_1$ by

$$f(x) = y \ :\text{iff } (y = x + 1 \wedge n \nmid y) \vee (y + n - 1 = x \wedge n \mid y).$$

($\Rightarrow$) (i) By Lemma 7.9 there is a constant $q$ such that

$$\lambda^{\mathfrak{d}}(f^{i+1}a) - \lambda^{\mathfrak{d}}(f^i a) = \lambda^{\mathfrak{d}}(f^{q+i+1}a) - \lambda^{\mathfrak{d}}(f^{q+i}a)$$

for all $a \in A$ and large enough $i$. Let $\Delta$ be a finite orbit. For $a \in \Delta$ this implies $f^{q+i}a = f^i a$ as $\Delta$ would be infinite otherwise. Thus, $|\Delta| \leq q$.

(ii) Let $\Delta$ be an infinite orbit and choose some $a \in \Delta$. $f^*(a) \subseteq \Delta$ is infinite. Thus, each infinite orbit contains an infinite $f$-chain, of which, by Lemma 7.8, there are only finitely many. □

As an immediate corollary we obtain a characterisation of structures with an equivalence relation.

**Theorem 7.13** (Khoussainov, Rubin [KR99])**.** *Let $\sim \subseteq A \times A$ be an equivalence relation. $(A, \sim) \in$ 1AutStr if and only if*

(i) *the cardinality of the finite $\sim$-classes is bounded and*

(ii) *there are only finitely many infinite $\sim$-classes.*

*Proof.* ($\Leftarrow$) Again, it is sufficient to prove the claim for structures with one infinite class and structures with infinitely many classes of the same size. Clearly, $(A, A \times A) \in$ 1AutStr, and for each $n > 1$, the relation

$$x \sim y \ :\text{iff } \exists z (n \mid z \ \wedge \ z \leq x < z + n \ \wedge \ z \leq y < z + n)$$

has infinitely many classes of size $n$.

($\Rightarrow$) By Lemma 3.6 there is a well-ordering $\leq$ such that $(A, \sim, \leq) \in$ 1AutStr. Define $f : A \to A$ by

$$f(x) := \begin{cases} \min\{\, y \mid y \sim x \wedge y > x \,\} & \text{if such a } y \text{ exists}, \\ \min\{\, y \mid y \sim x \,\} & \text{otherwise}. \end{cases}$$

Clearly, $f$ is definable in $(A, \sim, \leq)$. Thus, $(A, f) \in$ 1AutStr. Since the orbits of $f$ are exactly the $\sim$-classes, the claim follows from the preceding theorem. □

**Orderings.** Next we turn to linear orderings. Again, Khoussainov and Rubin obtained a precise characterisation.

**Proposition 7.14.** *Let $(A, \leq) \in$ 1AutStr be a linear order. Every set $B \subseteq A$ such that there are infinitely many elements of $A$ between any two elements of $B$ is finite.*

*Proof.* Let $\mathfrak{d}$ be a unary presentation of $(A, \leq)$ with loop constants $(l, p)$. We claim that $|B| < p(p+2) + l$. Otherwise, there are elements $a_0 < \cdots < a_{p+1}$ of $B$ with $\lambda^{\mathfrak{d}}(a_i) \equiv \lambda^{\mathfrak{d}}(a_j) \pmod p$ and $\lambda^{\mathfrak{d}}(a_i) > l$ for all $i$, $j$. Denote by $J_i$ the set of numbers $k$ such that the interval between $a_i$ and $a_{i+1}$ contains infinitely many elements $a$ with $\lambda^{\mathfrak{d}}(a) \equiv k \pmod p$. There have to be two sets $J_i$, $J_k$ with $J_i \cap J_k \neq \emptyset$. Choose elements $a_i < b < a_{i+1}$ and $a_j < c < a_{j+1}$ with

$$\begin{aligned} &\lambda^{\mathfrak{d}}(b) \equiv \lambda^{\mathfrak{d}}(c) \equiv m \pmod p && \text{for some } m \in J_i \cap J_k, \\ &\lambda^{\mathfrak{d}}(b), \lambda^{\mathfrak{d}}(c) > \lambda^{\mathfrak{d}}(a_{i+1}) + l, \lambda^{\mathfrak{d}}(a_j) + l. \end{aligned}$$

Then $\big(\lambda^{\mathfrak{d}}(b), \lambda^{\mathfrak{d}}(a_{i+1})\big) \sim_{l,p} \big(\lambda^{\mathfrak{d}}(c), \lambda^{\mathfrak{d}}(a_j)\big)$ but $b \leq a_{i+1}$ and $c > a_j$. Contradiction. □

**Theorem 7.15** (Khoussainov, Rubin [KR99])**.** *Let $\leq$ be a linear order. $(A, \leq)$ has a unary presentation if and only if it is a finite sum of linear orders of type $1$, $\omega$, or $\omega^*$.*

*Proof.* ($\Leftarrow$) immediately follows from the closure of 1AutStr under finite ordered sums. ($\Rightarrow$) Each structure satisfying the condition of the previous proposition can be written as such a sum. □

**Corollary 7.16** (Khoussainov, Rubin [KR99])**.** *Let $\alpha$ be an ordinal. $(\alpha, \leq)$ has a unary presentation if and only if $\alpha < \omega^2$.*

**Graphs.** A graph is in 1AutStr iff it has a certain "ladder structure."

**Theorem 7.17.** *Let $\mathfrak{G} = (V, E)$ be a graph. $\mathfrak{G} \in$ 1AutStr if and only if there are finite graphs $\mathfrak{H}$, $\mathfrak{H}'$ and a partition $(A, B_0, B_1, \dots)$ of $V$ such that the following conditions hold.*

(i) *$\mathfrak{G}|_A \cong \mathfrak{H}$ and $\mathfrak{G}|_{B_i} \cong \mathfrak{H}'$ for all $i$.*

(ii) *The edges between $A$ and $B_i$ do not depend on $i$ for $i \geq 1$, and the edges between $B_i$ and $B_k$ do not depend on $i$ and $k$ for $|i - k| > 1$. Formally, let $A = \{a_0, \dots, a_{r-1}\}$, $B_i = \{b_0^i, \dots, b_{s-1}^i\}$.*

$$\begin{aligned} &(a_k, b_l^i) \in E &&\text{iff } (a_k, b_l^j) \in E && \text{for all } i, j \geq 1, \\ &(b_k^i, b_l^j) \in E &&\text{iff } (b_k^{i'}, b_l^{j'}) \in E && \text{for all } i - j, i' - j' > 1 \text{ or} \\ &&&&& \quad i - j, i' - j' < -1, \\ &(b_k^i, b_l^{i+1}) \in E &&\text{iff } (b_k^j, b_l^{j+1}) \in E && \text{for all } i, j. \end{aligned}$$

*Proof.* ($\Rightarrow$) Fix a presentation $\mathfrak{d}$ of $\mathfrak{G}$ with loop constants $(l, p)$. Set

$$\begin{aligned} A &:= \{\, v \in G \mid \lambda^{\mathfrak{d}}(v) < l \,\}, \\ B_i &:= \{\, v \in G \mid pi + l \leq \lambda^{\mathfrak{d}}(v) < p(i+1) + l \,\}. \end{aligned}$$

Each condition can easily be verified. For example, to prove the first item of condition (ii) let $a \in A$, $b^i \in B_i$, and $b^j \in B_j$. Then

$$(\lambda^{\mathfrak{d}}(a), \lambda^{\mathfrak{d}}(b^i)) \sim_{l,p} (\lambda^{\mathfrak{d}}(a), \lambda^{\mathfrak{d}}(b^j))$$

and thus $(a, b^i) \in E$ iff $(a, b^j) \in E$.

($\Leftarrow$) We construct an interpretation of $\mathfrak{G}$ in $\mathfrak{N}_1$. Let $r := |A|$, $s := |B_i|$. The elements of $A$ are encoded as numbers less than $r$, and those of $B_i$ as $si + r, \dots, s(i+1) + r - 1$. We can define formulae expressing that $x$ is the $k^{\text{th}}$ element of $B_i$ for some $i$, and that $x \in B_i$ and $y \in B_{i+1}$ for some $i$ by

$$\begin{aligned}\psi_k(x) &:= x - r \equiv k \pmod{s},\\ \chi(x,y) &:= \exists z(\psi_0(z) \wedge z - s \leq x < z \leq y < z + s).\end{aligned}$$

The desired formula $\varphi_E(x,y)$ for $E$ can be constructed as disjunction over the cases $x, y \in A$; $x \in A$, $y \in B_0$; $x \in A$, $y \in B_i$ for $i > 0$; $x \in B_i$, $y \in B_k$ for $|i - k| > 1$, and so on. Each case can be handled using $\psi_k(x)$ and $\chi(x,y)$. □

**Corollary 7.18.** *The Random Graph $\mathfrak{R}$ has no unary presentation.*

*Proof.* Suppose there is a partition $(A, B_0, B_1, \dots)$ of $\mathfrak{R}$ satisfying the conditions of the preceding theorem. Set $X := A \cup B_0 \cup \dots \cup B_4$. By the extension axioms, there is some node $v \notin X$ which is connected to all elements of $X$ except those of $B_3$. Since $v \in B_i$ for some $i \geq 5$ we have $(b^1_k, v) \in E$ iff $(b^3_k, v) \in E$, by the second condition of part (ii) above. Contradiction. □

**Groups.** As far as groups are concerned unary presentations only suffice to describe finite structures.

**Theorem 7.19.** *Let $\mathfrak{G} = (G, \cdot)$ be a group. $\mathfrak{G} \in$ 1AutStr if and only if $\mathfrak{G}$ is finite.*

*Proof.* ($\Leftarrow$) is immediate. ($\Rightarrow$) Suppose $\mathfrak{G}$ is infinite. Fix an injective presentation $\mathfrak{d}$ with loop constants $(l, p)$. Choose elements $a$ and $b$ such that $2p < \lambda^{\mathfrak{d}}(a) < \lambda^{\mathfrak{d}}(b) - p$.

If $\lambda^{\mathfrak{d}}(a \cdot b) < \lambda^{\mathfrak{d}}(b) - l$ then choose $c$ such that $\lambda^{\mathfrak{d}}(c) = \lambda^{\mathfrak{d}}(b) + p$. Because of

$$(\lambda^{\mathfrak{d}}(a), \lambda^{\mathfrak{d}}(b), \lambda^{\mathfrak{d}}(a \cdot b)) \sim_{l,p} (\lambda^{\mathfrak{d}}(a), \lambda^{\mathfrak{d}}(c), \lambda^{\mathfrak{d}}(a \cdot b))$$

we have $a \cdot b = a \cdot c$. But this implies $b = c$. Contradiction.

If $\lambda^{\mathfrak{d}}(a \cdot b) \geq \lambda^{\mathfrak{d}}(b) - l$ then choose $c$ such that $\lambda^{\mathfrak{d}}(c) = \lambda^{\mathfrak{d}}(a) - p$. Because of

$$(\lambda^{\mathfrak{d}}(a), \lambda^{\mathfrak{d}}(b), \lambda^{\mathfrak{d}}(a \cdot b)) \sim_{l,p} (\lambda^{\mathfrak{d}}(c), \lambda^{\mathfrak{d}}(b), \lambda^{\mathfrak{d}}(a \cdot b))$$

we again obtain a contradiction. □

**Corollary 7.20.** *Let $\mathfrak{A}$ be a ring or field. $\mathfrak{A} \in$ 1AutStr if and only if $\mathfrak{A}$ is finite.*

Let $\mathfrak{G} = (G, \cdot)$ be a group, $S$ a set of semigroup generators of $\mathfrak{G}$, and set $f_a(x) := x \cdot a$. If we do not require full multiplication to be presented but use groups in the form $(G, (f_a)_{a \in S})$ instead, there are also infinite groups in 1AutStr.

**Lemma 7.21.** *Let* $\mathfrak{G} = (G, \cdot)$ *be a group and* $S$, $S' \subseteq G$ *be sets of semigroup generators of* $\mathfrak{G}$. $(G, (f_a)_{a \in S}) =_{\mathrm{FO}} (G, (f_a)_{a \in S'})$.

*Proof.* Each $g \in S$ can be written as $g = g'_0 \cdots g'_n$ for some $g'_0, \ldots, g'_n \in S'$. Thus, $f_g$ can be defined by $f_g x := f_{g'_n} \cdots f_{g'_0} x$. □

**Proposition 7.22.** *Let* $\mathfrak{G} = (G, (f_a)_{a \in S})$ *be an abelian group.* $\mathfrak{G} \in 1\mathrm{AutStr}$ *if and only if* $\mathfrak{G}$ *is either finite or* $\mathfrak{G} \cong \mathbb{Z} \oplus \mathbb{Z}_{p_0} \oplus \cdots \oplus \mathbb{Z}_{p_n}$ *for some* $p_0, \ldots, p_n$.

*Proof.* ($\Rightarrow$) Suppose $\mathfrak{G} \cong \mathbb{Z} \oplus \mathbb{Z} \oplus \mathfrak{H}$ and let $a$ and $b$ be generators of the subgroups $\mathbb{Z}$. By the preceding lemma we can assume that $a$, $b \in S$. Since $f_a^*(b^n) \cap f_a^*(b^m) = \emptyset$ for all $n \neq m$ there are infinitely many disjoint $f_a$-chains in contradiction to Lemma 7.8.

($\Leftarrow$) Let $\mathfrak{G} = \mathbb{Z} \oplus \mathfrak{H}$ for finite $\mathfrak{H}$. W.l.o.g. assume that $S = \{a\} \cup T$ where $a$ generates $\mathbb{Z}$ and $T$ generates $\mathfrak{H}$. Let $n := |H|$. We identify $H$ with the set $\{0, \ldots, n-1\}$, and construct an interpretation of $\mathfrak{G}$ in $\mathfrak{N}_1$ by encoding elements $(k, l) \in \mathbb{Z} \times H$ by

$$h(k,l) := \begin{cases} 2kn + l & \text{if } k \geq 0, \\ (-2k-1)n + l & \text{if } k < 0. \end{cases}$$

The generating functions can be defined by

$$\begin{aligned} f_a(x) = y \ &:\text{iff} \ \left[y = x + 2n \wedge \exists z(z < n \ \wedge \ x - z \equiv 0 \pmod{2n})\right] \\ &\qquad \vee \left[y + 2n = x \wedge \exists z(z < n \ \wedge \ x - z \equiv n \pmod{2n})\right] \\ &\qquad \vee \left[n \leq x < 2n \ \wedge \ y = x - n\right], \\ f_b(x) = y \ &:\text{iff} \ \exists z\Big(z \equiv 0 \pmod{n} \wedge \bigvee_{h \in H} (x = z + h \ \wedge \ y = z + g_b(h))\Big). \end{aligned}$$

where $b \in T$ and $g_b : H \to H$ is the right-multiplication by $b$ in $\mathfrak{H}$. □

**Proposition 7.23.** *Let* $\mathfrak{G} = (G, (f_a)_{a \in S}) \in 1\mathrm{AutStr}$ *be a group. If* $a$ *and* $b$ *are elements of infinite order then there are some constants* $k$, $l \in \mathbb{Z} \setminus \{0\}$ *such that* $a^k = b^l$.

*Proof.* W.l.o.g. assume $a \in S$. Consider the $f_a$-chains of $b^i$ for $i \geq 0$. Each chain is infinite since

$$b^i a^n = b^i a^m \implies a^n = a^m \implies n = m.$$

By Lemma 7.8, only finitely many chains can be disjoint. Hence, there are $i$, $j \in \mathbb{N}$ such that $f_a^*(b^i) \cap f_a^*(b^j) \neq \emptyset$, i.e.,

$$b^i a^n = b^j a^m \implies b^{i-j} = a^{m-n}$$

for some $n$, $m \in \mathbb{N}$. □

Equivalently, the above proposition can be stated as, if $\mathfrak{G}$ is in 1AutStr and $a$ is of infinite order then $|\mathfrak{G} : \langle a \rangle|$, the index of $\langle a \rangle$, is finite.

## 7.3 Complexity

After having defined unary presentations and having shown that they are much weaker than general automatic presentations the question arises whether we have gained anything by this restriction. A first positive effect is a drastic decrease in complexity.

We will show that every quantifier $\exists x\varphi$ can be replaced by a bounded version $(\exists x \leq m)\varphi$ for some $m$.

**Definition 7.24.** For $\overline{a}$, $\overline{b} \in \mathbb{N}^k$ and $n, \delta \in \mathbb{N}$ we define

$$\overline{a} \sim_{n,\delta} \overline{b} \ :\text{iff} \ \ d(a_i, a_j) =_{\delta 3^n} d(b_i, b_j) \text{ and } a_i \equiv b_i \pmod{\delta} \text{ for all } i, j < k$$

where

$$d(a_0, a_1) := a_1 - a_0 \qquad \text{and} \qquad a =_l b \ :\text{iff} \ \ a = b \text{ or } a, b > l.$$

The following lemma ensures that if $\exists x\varphi$ is satisfied then there is some element $b$ which is not too large such that $\varphi(b)$ holds.

**Lemma 7.25.** *Let $\overline{a}$, $\overline{b} \in \mathbb{N}^k$ with $a_0 = b_0 = 0$ and $\overline{a} \sim_{n+1,\delta} \overline{b}$, and let $m \in \mathbb{N}$ be such that $b_0, \dots, b_{k-1} \leq m$. For every $a' \in \mathbb{N}$ there is some $b' \in \mathbb{N}$ with $b' \leq m + \delta(3^n + 1)$ such that $\overline{a}a' \sim_{n,\delta} \overline{b}b'$.*

*Proof.* W.l.o.g. assume $a_0 \leq \cdots \leq a_{k-1}$, and let $a_i \leq a' \leq a_{i+1}$ for some $i$. The case $a_{k-1} < a'$ is proved analogously.

If $d(a_i, a') \leq \delta 3^n$ then choose $b' := b_i + d(a_i, a')$. It follows that

$$\begin{aligned} &d(a', a_{i+1}) =_{\delta 3^n} d(b', b_{i+1}) \\ \text{and } &a' \equiv a_i + d(a_i, a') \equiv b_i + d(a_i, a') \equiv b' \pmod{\delta}. \end{aligned}$$

Thus, $\overline{a}a' \sim_{n,\delta} \overline{b}b'$.

If $d(a_i, a') > \delta 3^n$ but $d(a', a_{i+1}) \leq \delta 3^n$ then choose $b' := b_{i+1} - d(a_i, a')$. Again, we have $\overline{a}a' \sim_{n,\delta} \overline{b}b'$.

Finally, if both distances are more than $\delta 3^n$ then choose some $b'$ such that $b_i + \delta 3^n < b' < b_{i+1} - \delta 3^n$ and $b' \equiv a' \pmod{\delta}$. This is possible because $d(b_i, b_{i+1}) > \delta 3^{n+1}$ and

$$\begin{aligned} \{\, a \bmod \delta \mid a_i < a < a_{i+1} \,\} &= \{0, \dots, \delta - 1\} \\ &= \{\, b \bmod \delta \mid b_i < b < b_{i+1} \,\}. \end{aligned}$$

Furthermore $b'$ can be chosen such that $d(b_i, b') \leq \delta 3^n + \delta$. Therefore $b' < m + \delta(3^n + 1)$. ☐

**Proposition 7.26.** *Let $\varphi = Q_0 x_0 \cdots Q_{n-1} x_{n-1} \psi(\overline{x}, \overline{y})$ for quantifiers $Q_0, \dots, Q_{n-1} \in \{\exists, \forall\}$ and let $n_0, \dots, n_r$ be the constants appearing in divisibility predicates $n \,|\, x$. Denote the least common multiple of $n_0, \dots, n_r$ by $\delta$, and for $\overline{a} \in \mathbb{N}^k$ let $m := \max\{a_0, \dots, a_k\}$. Then the model-checking problem $\mathfrak{N}_1 \models \varphi(\overline{a})$ is in*

$$\text{DSPACE}\big[\mathcal{O}(n + \log|\varphi| + \log\delta + \log m)\big].$$

*Proof.* Obviously, $\overline{b} \sim_{0,\delta} \overline{b}'$ implies $\mathfrak{N}_1 \models \psi(\overline{b}, \overline{a})$ iff $\mathfrak{N}_1 \models \psi(\overline{b}', \overline{a})$. By the preceding lemma there are bounds $m_0, \dots, m_{n-1}$ such that we can find $b'_i < m_i$, $i < n$ with $\overline{b} \sim_{0,\delta} \overline{b}'$. We have

$$\begin{aligned} m_0 &:= m + \delta(3^{n-1} + 1), \\ m_{i+1} &:= m_i + \delta(3^{n-i} + 1), \qquad \text{for } i < n - 1. \end{aligned}$$

Which yields

$$\begin{aligned} m_i &= m + \sum_{j \le i} \delta(3^{n-j-1} + 1) \\ &= m + \delta(i+1) + \delta 3^{n-i-1} \sum_{j \le i} 3^j \\ &= m + \delta(i+1) + \delta 3^{n-i-1} \frac{3^{i+1} - 1}{3 - 1} \\ &\le m + \delta(i+1) + \tfrac{1}{2}\delta 3^n. \end{aligned}$$

Therefore, a Turing machine can evaluate $\varphi(\overline{a})$ by cycling through all values of $b_i$ for $i < n$ on its tape, and checking whether $\psi(\overline{a}, \overline{b})$ holds, which can be done in Logspace. The space used to store $\overline{b}$ is

$$\begin{aligned} \log \sum_{i<n} m_i &= \log\bigl(nm + \tfrac{1}{2}\delta n(n+1) + \tfrac{1}{2}\delta n 3^n\bigr) \\ &\le \log\bigl(nm + \delta 2^{\mathcal{O}(n)}\bigr) \\ &\le \mathcal{O}(n + \log \delta + \log m). \end{aligned}$$

□

Hence, using the same conventions as in Section 3.4 we obtain the following bound on the complexity of the model-checking problem for 1AutStr.

**Corollary 7.27.** *Let $\tau$ be a relational signature. Given the presentation $\mathfrak{d}$ of a structure $\mathfrak{A} \in$ 1AutStr$[\tau]$, a tuple $\overline{a}$ in $\mathfrak{A}$, and a formula $\varphi(\overline{x}) \in$ FO$[\tau]$, the model-checking problem for $(\mathfrak{A}, \overline{a}, \varphi)$ is in* Dspace$\bigl[\mathcal{O}(|\varphi|^2 |\mathfrak{d}|^6 + \log \lambda^{\mathfrak{d}}(\overline{a}))\bigr]$.

*Proof.* Construct an interpretation $\mathcal{I}$ of $\mathfrak{A}$ in $\mathfrak{N}_1$ via the translation of automata to formulae given above. A closer look reveals that the length of each formula $\psi$ defining one $\sim_{l,p}$-class is in $\mathcal{O}(l + p^2)$. There are at most $|\mathfrak{d}|$ such classes (one for each final state). Since $l, p \le |\mathfrak{d}|$ we obtain $|\psi| \in \mathcal{O}(|\mathfrak{d}|^3)$. The translation of $\mathfrak{d}$ to $\mathcal{I}$ can be performed in Dtime$[\mathcal{O}(|\mathfrak{d}|^3)]$.

Further, note that the interpretation maps each $a \in A$ to the number $\lambda^{\mathfrak{d}}(a)$. By the preceding proposition we can decide $\mathfrak{N}_1 \models \varphi^{\mathcal{I}}(\overline{a}^{\mathcal{I}})$ in

$$\text{Dspace}\bigl[\mathcal{O}(n + \log|\varphi^{\mathcal{I}}| + \log \delta + \log \lambda^{\mathfrak{d}}(\overline{a}))\bigr].$$

Since $|\varphi^{\mathcal{I}}| \in \mathcal{O}(|\varphi|\,|\mathfrak{d}|^3)$ we have $n \in \mathcal{O}(|\varphi|\,|\mathfrak{d}|^3)$, $n_0, \dots, n_r \in 2^{\mathcal{O}(|\varphi||\mathfrak{d}|^3)}$, and hence

$$\delta \le n_0 \cdots n_r \in (2^{\mathcal{O}(|\varphi||\mathfrak{d}|^3)})^{\mathcal{O}(|\varphi||\mathfrak{d}|^3)} = 2^{\mathcal{O}(|\varphi|^2|\mathfrak{d}|^6)}.$$

□

## 7.4 Decidability

We start our investigation as to what logics are decidable by showing that $\mathfrak{N}_1'$ allows quantifier elimination. To simplify the task an intermediate structure is introduced.

**Lemma 7.28.** *The structure* $\big(\mathbb{Z}, s, \leq, (x \equiv k \pmod n)_{k,n}\big)$ *admits quantifier elimination.*

*Proof.* It is well known that $(\mathbb{Z}, s, \leq)$ admits quantifier elimination. In [KK71] it is shown that each formula $\exists x\varphi \in \mathrm{FO}[s, \leq]$ with quantifier-free $\varphi$ can be transformed into a disjunction of formulae of the form

$$\exists x\Big(\bigwedge_i x < t_i \wedge \bigwedge_i u_i < x \wedge \bigwedge_i x = v_i\Big).$$

Analogously, formulae $\exists x\varphi \in \mathrm{FO}[s, \leq, (x \equiv k \pmod n))_{k,n}]$ can be brought into the form

$$\exists x\Big(\bigwedge_i x < t_i \wedge \bigwedge_i u_i < x \wedge \bigwedge_i x = v_i \wedge \bigwedge_i (x \equiv k_i \pmod{n_i})\Big)$$

by using the following additional rules:

$$\begin{aligned} s^m x \equiv k \pmod n \quad &\equiv \quad x \equiv k - m \pmod n, \\ \neg(x \equiv k \pmod n) \quad &\equiv \quad \bigvee_{i \neq k} x \equiv i \pmod n. \end{aligned}$$

Furthermore, we can ensure that all moduli $n_i$ are equal by replacing them by their least common multiple. Thus, we obtain

$$\exists x\Big(\bigwedge_i x < t_i \wedge \bigwedge_i u_i < x \wedge \bigwedge_i x = v_i \wedge \bigwedge_i (x \equiv k_i \pmod n)\Big).$$

If there are more than one atom of the form $x \equiv k_i \pmod n$ with different $k_i$ then the formula is false. If there is no such atom then we can eliminate the quantifier as in the case of $(\mathbb{Z}, s, \leq)$. Hence, we only need to consider the case

$$\exists x\Big(\bigwedge_i x < t_i \wedge \bigwedge_i u_i < x \wedge \bigwedge_i x = v_i \wedge x \equiv k \pmod n\Big).$$

If there is at least one atom of the form $x = v$ then we can replace $x$ by $v$ everywhere. Otherwise, let the free variables be among $\{y_0, \dots, y_s\}$. Then the formula is equivalent to

$$\exists x \bigvee_{k_0,\dots,k_s < n} \Big(\bigwedge_{i \leq s} y_i \equiv k_i \pmod n \wedge \varphi_{k_0 \dots k_s}\Big)$$

where $\varphi_{\overline{k}}$ is obtained from $\varphi$ be removing the modulo-atom and modifying all other atoms according to the following rules:

$$\begin{aligned} x < s^l y_i \quad &\longrightarrow \quad x < s^{l - \Delta_i} y_i, \\ s^l y_i < x \quad &\longrightarrow \quad s^{l + \Delta_i} y_i < x, \end{aligned}$$

where $\Delta_i := k_i - k \pmod n$. In the resulting formula the quantifier can be eliminated as in the case of $(\mathbb{Z}, s, \leq)$. ☐

**Corollary 7.29.** $\mathfrak{N}_1'$ *admits quantifier elimination.*

*Proof.* It follows from the preceding lemma that

$$\mathfrak{Z}_1' := \big(\mathbb{Z}, s, \leq, 0, (x \equiv k \pmod{n})_{k,n}\big)$$

admits elimination of quantifiers (just replace 0 by some new variable, eliminate all quantifiers, and replace the new variable by 0, see e.g. [KK71]). $\mathfrak{N}_1'$ is the substructure of $\mathfrak{Z}_1'$ defined by $\delta(x) := 0 \leq x$. Let $\varphi(\overline{x}) \in \mathrm{FO}$. By $\varphi^\delta$ we denote the relativisation of $\varphi$ to the set defined by $\delta$. There is some quantifier-free $\psi(\overline{x}) \in \mathrm{FO}$ such that

$$\begin{aligned}
&\mathfrak{Z}_1' \models \varphi^\delta(\overline{x}) \leftrightarrow \psi(\overline{x}) \\
\text{iff } &\mathfrak{Z}_1' \models \varphi^\delta(\overline{a}) \iff \mathfrak{Z}_1' \models \psi(\overline{a}) \text{ for all } \overline{a} \text{ in } \mathbb{Z} \\
\Longrightarrow\ &\mathfrak{Z}_1' \models \varphi^\delta(\overline{a}) \iff \mathfrak{Z}_1' \models \psi(\overline{a}) \text{ for all } \overline{a} \text{ in } \mathbb{N}.
\end{aligned}$$

As $\psi$ is quantifier-free this implies

$$\begin{aligned}
&\mathfrak{Z}_1' \models \varphi^\delta(\overline{a}) \iff \mathfrak{N}_1' \models \psi(\overline{a}) \text{ for all } \overline{a} \text{ in } \mathbb{N} \\
\text{iff } &\mathfrak{N}_1' \models \varphi(\overline{a}) \iff \mathfrak{N}_1' \models \psi(\overline{a}) \text{ for all } \overline{a} \text{ in } \mathbb{N} \\
\text{iff } &\mathfrak{N}_1' \models \varphi(\overline{x}) \leftrightarrow \psi(\overline{a}).
\end{aligned}$$

□

We have quantifier elimination not only for FO but also for FO(R), the extension of FO by Ramsey-quantifiers. The formula $\mathsf{R}x_0 \ldots x_{n-1}\varphi$ holds iff there is some infinite set $X$ such that $\varphi(\overline{a})$ is true for all distinct $a_0, \ldots, a_{n-1}$ in $X$.

**Lemma 7.30.** $\mathfrak{N}_1'$ *admits quantifier elimination for* FO(R).

*Proof.* We have to show that for every formula $\varphi(\overline{y}) = \mathsf{R}x_0 \ldots x_{n-1}\psi(\overline{y}, \overline{x})$ with $\psi \in \mathrm{FO}$ there is an equivalent formula $\varphi'(\overline{y}) \in \mathrm{FO}$. If we can prove that

$$\begin{aligned}
\mathfrak{N}_1' \models \varphi(\overline{a}) \text{ iff } &\text{there are } k, p \in \mathbb{N} \text{ with } k > a_j + p \text{ for all } j \text{ such that} \\
&\mathfrak{N}_1' \models \psi(\overline{a}, k + i_0 p, \ldots, k + i_{n-1} p) \\
&\text{for all different } i_0, \ldots, i_{n-1} \in \mathbb{N},
\end{aligned}$$

then it follows that

$$\begin{aligned}
\varphi(\overline{y}) \equiv \exists z \Big( &\bigwedge_i y_i + p < z \\
&\wedge \forall \overline{x} \Big( \bigwedge_i (x_i \geq z \wedge x_i \equiv z \pmod{p}) \wedge \bigwedge_{i \neq j} x_i \neq x_j \rightarrow \psi(\overline{y}, \overline{x}) \Big)\Big).
\end{aligned}$$

Thus it remains to prove the above claim. ($\Leftarrow$) is trivial. ($\Rightarrow$) Let $(l, p)$ be the loop constants of some unary presentation $\mathfrak{d}$ of $\mathfrak{N}_1'$. Let $X \subseteq \mathbb{N}$ be a maximal infinite set satisfying $\mathfrak{N}_1' \models \psi(\overline{a}, b_0, \ldots, b_{n-1})$ for all distinct $b_0, \ldots, b_{n-1} \in X$ and with $b \geq a_j + p$ for all $j$ and $b \in X$. By the Pigeonhole Principle there is some constant $c < p$ such that

$$Y := \{\, b \in X \mid b \equiv c \pmod{p} \,\}$$

is infinite. Note that, if $b_0 < \cdots < b_{n-1} \in Y$ and $b_{i+1} \geq b_i + 2p$, then

$$\mathfrak{N}_1' \models \psi(\bar{a}, b_0, \ldots, b_i, b_{i+1} \pm p, \ldots, b_{n-1} \pm p).$$

Let $b_0 < \cdots < b_{n-1}$ be the least elements of $Y$. Applying the above observation several times we obtain

$$\mathfrak{N}_1' \models \psi(\bar{a}, b_0 + i_0 p, \ldots, b_{n-1} + i_{n-1} p)$$

for all $i_0, \ldots, i_{n-1} \in \mathbb{N}$ such that $b_j + i_j p + p \leq b_{j+1} + i_{j+1} p$ for all $j < n$. Thus, when setting $k := b_{n-1}$, it follows that

$$\mathfrak{N}_1' \models \psi(\bar{a}, k + i_0 p, \ldots, k + i_{n-1} p)$$

for all distinct $i_0, \ldots, i_{n-1} \in \mathbb{N}$. □

Unfortunately, despite the weakness of unary presentations we have not gained much as far as decidability of stronger logics is concerned.

**Proposition 7.31.** *There are structures with undecidable* FO(DTC)*-theory in* 1AutStr.

*Proof.* Immediately from Lemma 2.7 as $(\mathbb{N}, s) \in$ 1AutStr. □

There is only a very special case in which we obtain decidability. Denote by FO(closed DTC) the restriction of FO(DTC) to those formulae such that in every subformula of the form $[\mathrm{DTC}_{x,y}\ \psi(x,y)](u,v)$ the only free variables of $\psi$ are $x$ and $y$.

**Theorem 7.32.** 1AutStr *is effectively closed under* FO(closed $\mathrm{DTC}^1$)*-interpretations.*

*Proof.* Define

$$f(x) := \begin{cases} y & \text{if } y \text{ is the unique element such that } \psi(x,y), \\ x & \text{otherwise.} \end{cases}$$

Then $[\mathrm{DTC}_{x,y}\ \psi(x,y)](u,v)$ holds iff $v \in f^*(u)$. Therefore, the claim follows from Lemma 7.10. □

# Chapter 8

# Other Types of Presentations

The restriction to unary alphabets turned out to yield an interesting subclass of automatic structures where model-checking has an acceptable complexity and which permits many precise characterisations. In this chapter we look at different kinds of restrictions hoping to obtain other interesting subclasses.

While the class studied in the first section has many pleasant theoretical properties it seems doubtful whether weak presentations are strong enough to be of practical value. The classes defined in the second section even lack important theoretical properties—like closure under first-order interpretations—and are only included for the sake of completeness.

## 8.1 Weak Presentations

The choice we made concerning the encoding of tuples is not the only one possible. In this section we investigate an alternative encoding where a tuple $(x_0, \dots, x_{n-1})$ of words is encoded by the word $x_0\Box \cdots \Box x_{n-1}$. As it turns out this model is considerably weaker than those we have used so far.

**Definition 8.1.** Let $x_0, \dots, x_{n-1} \in \Sigma^*$. The *weak convolution* of $\overline{x}$ is defined as

$$x_0 \otimes_{\mathrm{w}} \cdots \otimes_{\mathrm{w}} x_{n-1} := x_0\Box \cdots \Box x_{n-1}.$$

The notion of *weak presentation* (called "strong presentation" in [KN95]) is defined analogously to automatic presentations where

(i) convolution is replaced by weak convolution everywhere and

(ii) the language $L_\varepsilon$ defining equality is left out, i.e., the presentation is always injective.

The class of $\tau$-structures with weak presentation is denoted by $\mathrm{WAutStr}[\tau]$.

The reason for the restriction to injective presentations is that identity cannot be weakly presented. Therefore only finite structures would be presentable without it (see Corollary 8.6 and Theorem 8.7 below).

As in the case of automatic structures one can effectively evaluate FO-formulae on weakly presentable structures with the notable exception of equality. In the following we denote by $\mathfrak{L}_{\neq}$ the logic $\mathfrak{L}$ without equality. If we want to emphasise that equality is allowed the notation $\mathfrak{L}_{=}$ is used.

**Lemma 8.2.** *There is a recursive function $\eta$ assigning to every weak presentation $\mathfrak{d}$ of some $\mathfrak{A} \in \mathrm{WAutStr}[\tau]$ and every formula $\varphi \in \mathrm{FO}_{\neq}[\tau]$ a weak presentation of $(\mathfrak{A}, \varphi^{\mathfrak{A}})$.*

*Proof.* Analogous to the proof for automatic structures with obvious modifications for the case of quantifiers. □

In order to give a characterisation in terms of a complete structure we have to choose a different logic.

**Definition 8.3.** FFO is the restriction of $\mathrm{FO}_{\neq}$ to boolean combinations of monadic $\mathrm{FO}_{\neq}$-formulae, i.e., formulae with only one free variable.

Obviously, every such formula can be written in the form

$$\varphi(x_0, \dots, x_{n-1}) = \bigvee_{i<m} \bigwedge_{j<n} \psi^i_j(x_j)$$

with $\psi^i_j \in \mathrm{FO}$.

**Theorem 8.4.** *Let $R \subseteq N^n$. Let $\mathrm{code}_p(x)$ be the $p$-adic encoding of $x \in \mathbb{N}$ and define*

$$\mathrm{code}_p(R) := \{\, \mathrm{code}_p(x_0) \otimes_{\mathrm{w}} \cdots \otimes_{\mathrm{w}} \mathrm{code}_p(x_{n-1}) \mid (x_0, \dots, x_{n-1}) \in R \,\}.$$

*$\mathrm{code}_p(R)$ is regular if and only if $R$ is FFO-definable in $\mathfrak{N}_p$.*

*Proof.* ($\Rightarrow$) Let $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$ be a deterministic automaton recognising $\mathrm{code}_p(R)$. For each pair $(q, q') \in Q$ we can construct formulae $\psi_{qq'}(x)$ saying that if $\mathfrak{A}$ starts in state $q$ it reaches state $q'$ after reading $\mathrm{code}_p(x)$. Then $R$ can be defined by

$$\varphi(x_0, \dots, x_{n-1}) := \bigvee \Big\{ \psi_{q_0 q_1}(x_0) \wedge \bigwedge_{i<n-1} \psi_{\delta(q_i, \Box), q_{i+1}}(x_i) \;\Big|\; q_0, \dots, q_{n-1} \in Q, q_n \in F \Big\}.$$

($\Leftarrow$) Let $\varphi(x_0, \dots, x_{n-1}) = \bigvee_i \bigwedge_j \psi^i_j(x_j)$. The languages $L^i_j$ defined by $\psi^i_j$ are regular. Therefore, the language $L = \bigcup_i L^i_0 \Box \cdots \Box L^i_{n-1}$ is regular as well. □

**Corollary 8.5.** *A structure $\mathfrak{A} = (A, R_0, \dots, R_r)$ has a weak automatic presentation if and only if $\mathfrak{A} \leq_{\mathrm{FFO}} \mathfrak{N}_p$ for some/all $p \in \mathbb{N} \setminus \{0, 1\}$.*

**Corollary 8.6** (Khoussainov, Nerode [KN95])**.** *If $\mathfrak{A} \in \mathrm{WAutStr}$ then for every relation $R$ of $\mathfrak{A}$ with arity $r$ there are $X_{ik} \subseteq A$ such that*

$$R = \bigcup_{i<m} X_{i0} \times \cdots \times X_{i(r-1)}.$$

*Proof.* Take as $X_{ik}$ the sets defined by the formulae $\psi^i_k$ in the definition of $R$. □

**Theorem 8.7.** *Let $\mathfrak{A}$ be a relational $\tau$-structure.* $\mathfrak{A} \in$ WAutStr *if and only if there is some congruence $\sim$ such that $\mathfrak{A}/\sim$ is finite.*

*Proof.* ($\Rightarrow$) Let $\bigvee_i \bigwedge_k \psi^j_{ik}$ be the formula defining $R_j$. Set

$$x \sim y \;:\text{iff}\; \psi^j_{ik}(x) \iff \psi^j_{ik}(y) \text{ for all } j, i, k.$$

Then $\sim$ is the required congruence of finite index.

($\Leftarrow$) Let $\mathfrak{A} = (A, R_0, \dots, R_r)$ and let $\sim$ be a congruence of finite index. We construct a weak automatic presentation of $\mathfrak{A}$ as follows. Let $[a_0], \dots, [a_n]$ be an enumeration of $A/\sim$, and $n_i := \big|[a_i]\big|$. Denote the $k^{\text{th}}$ member of $[a_i]$ by $a_{ik}$. We encode $a_{ik}$ as $1^i\#1^k$. The presentation is $\mathfrak{d} := (\nu, \{1, \#\}, L_\delta, L_{R_0}, \dots, L_{R_r})$ where

$$L_\delta := \bigcup_{i \leq n} 1^i\#1^{<n_i}, \qquad \nu(1^i\#1^k) := a_{ik},$$

$$L_{R_j} := \bigcup \Big\{ 1^{i_0}\#1^*\square \cdots \square 1^{i_{r_j - 1}}\#1^* \;\Big|\; ([a_{i_0}], \dots, [a_{i_{r_j-1}}]) \in R_j/\sim \Big\}.$$

□

In case of structures with functions $f$ the condition above, applied to the graph of $f$, means that the image of $f$ is finite.

Theorem 8.7 shows that weakly presentable structures are just finite structures blown up. Therefore we can reduce most problems to the finite case which usually is decidable. We call a logic $\mathfrak{L}$ *invariant under congruences* if for all structures $\mathfrak{A}$, congruences $\sim$, and formulae $\varphi(\overline{x}) \in \mathfrak{L}$ it holds that

$$\mathfrak{A} \models \varphi(\overline{a}) \text{ iff } \mathfrak{A}/\sim \models \varphi([\overline{a}]_\sim).$$

**Theorem 8.8.** *Let $\mathfrak{L}$ be a logic invariant under congruences.* WAutStr *is closed under $\mathfrak{L}$-interpretations.*

*Proof.* As WAutStr is closed under reducts it is sufficient to show that given $\mathfrak{A} \in$ WAutStr and $\varphi \in \mathfrak{L}$ we can construct an FFO-interpretation of $(\mathfrak{A}, \varphi^{\mathfrak{A}})$ in $\mathfrak{N}_p$. According to Theorem 8.7 there is a congruence $\sim$ of finite index. Let $\mathcal{I} = (h, \delta, \varepsilon, \varphi_{R_0}, \dots, \varphi_{R_r})$ be an FFO-interpretation $\mathfrak{A} \leq_{\text{FFO}} \mathfrak{N}_p$ and let $\vartheta_{[a]_\sim}(x)$ be the formula defining the $\sim$-class of $a$ in $\mathfrak{N}_p$. By assumption on $\mathfrak{L}$ the formula

$$\psi(\overline{x}) := \bigvee \Big\{ \bigwedge_{i<n} \vartheta_{[a_i]_\sim}(x_i) \;\Big|\; ([a_0]_\sim, \dots, [a_{n-1}]_\sim) \in \varphi^{\mathfrak{A}/\sim} \Big\}$$

defines $\varphi^{\mathfrak{A}}$. Thus $(\mathcal{I}, \psi)$ is an FFO-interpretation of $(\mathfrak{A}, \varphi^{\mathfrak{A}})$ in $\mathfrak{N}_p$. □

Some logic satisfying the condition above is $\text{FO}_{\neq}(\text{PFP})$. Logics not covered are e.g., $\text{FO}_=$ or $\text{FO}_{\neq}(\#)$. What about SO?

**Proposition 8.9.** WAutStr *is not closed under* $\text{SO}_{\neq}$*-interpretations.*

*Proof.* Equality is definable in $\text{SO}_{\neq}$.

$$\begin{aligned} u = v \;:\text{iff}\; & \exists{\neq}[\neg u \neq v \land \forall x \neg x \neq x \\ & \land \forall R(\forall x \neg Rxx \to \forall x \forall y(Rxy \to x \neq y))]. \end{aligned}$$

□

The above proof uses a two-dimensional relation variable. This leaves the case of monadic second-order logic unanswered. In fact, WAutStr is closed under $\mathrm{MSO}_{\neq}$-interpretations despite it not being invariant under congruences.

**Proposition 8.10.** WAutStr *is closed under* $\mathrm{MSO}_{\neq}$*-interpretations.*

*Proof.* Let $\mathfrak{A} = (A, R_0, \dots, R_r)$ be a relational structure and $\sim$ a congruence of $\mathfrak{A}$. We denote by $\mathfrak{A}_m^\sim = (A_m^\sim, (R_0)_m^\sim, \dots, (R_r)_m^\sim)$ the substructure of $\mathfrak{A}$ which contains exactly $m$ elements of each $\sim$-classes of size at least $m$, and all elements of smaller $\sim$-classes. Note that, since $\sim$ is a congruence, $\mathfrak{A}_m^\sim$ is uniquely determined up to isomorphisms. Let $P \subseteq A$ be a unary relation. The refinement $\sim_P$ of $\sim$ induced by $P$ is defined as

$$a \sim_P b \;:\text{iff}\;\; a \sim b \text{ and } a \in P \iff b \in P.$$

Let $\psi(\overline{x}) = Q_0 P_0 \cdots Q_{n-1} P_{n-1} \varphi(\overline{x}, \overline{P}) \in \mathrm{MSO}_{\neq}$ with $Q_0, \dots, Q_{n-1} \in \{\exists, \forall\}$ and $\varphi \in \mathrm{FO}_{\neq}$. We prove by induction on $n$ that

$$\mathfrak{A} \models \psi(\overline{a}) \;\text{ iff }\; \mathfrak{A}_{2^n}^\sim \models \psi(\overline{a}') \text{ for some } \overline{a}' \sim \overline{a}.$$

The case $n = 0$ is immediate as $\mathrm{FO}_{\neq}$ is invariant under congruences. For the induction step we prove:

*Claim.* There is a surjective mapping $'$ associating to every $P \subseteq A$ a relation $P' \subseteq A_{2^n}^\sim$ such that

$$(\mathfrak{A}, P)_{2^{n-1}}^{\sim_P} \cong (\mathfrak{A}_{2^n}^\sim, P')_{2^{n-1}}^{\sim_P}.$$

Then it follows that

$$\begin{array}{lll}
& \mathfrak{A} \models \exists/\forall P \psi(\overline{a}) & \\
\text{iff for some/all } P \subseteq A: & (\mathfrak{A}, P) \models \psi(\overline{a}) & \\
\text{iff for some/all } P \subseteq A: & (\mathfrak{A}, P)_{2^{n-1}}^\sim \models \psi(\overline{a}') & \text{(ind. hyp.)} \\
\text{iff for some/all } P \subseteq A: & (\mathfrak{A}_{2^n}^\sim, P')_{2^{n-1}}^\sim \models \psi(\overline{a}') & \text{(Claim)} \\
\text{iff for some/all } P \subseteq A_{2^n}^\sim: & (\mathfrak{A}_{2^n}^\sim, P)_{2^{n-1}}^\sim \models \psi(\overline{a}') & \text{(surjectivity)} \\
\text{iff for some/all } P \subseteq A_{2^n}^\sim: & (\mathfrak{A}_{2^n}^\sim, P) \models \psi(\overline{a}'') & \text{(ind. hyp.)} \\
\text{iff } \mathfrak{A}_{2^n}^\sim \models \exists/\forall P \psi(\overline{a}''). & &
\end{array}$$

It remains to prove the claim. Consider each $\sim$-class $[a]$ in turn. What we have to do is to decide how many elements of $[a]$ are to be included in $P'$.

If $|[a]| \leq 2^n$ then $[a] \subseteq A_{2^n}^\sim$ and we can put all $b \in [a] \cap P$ into $P'$. Otherwise let $n_1 := \big|[a] \cap P\big|$ and $n_2 := \big|[a] \setminus P\big|$, and set $n_1' := \min\{n_1, 2^{n-1}\}$, $n_2' := \min\{n_2, 2^{n-1}\}$. Then we can add $n_1'$ elements form $[a]$ to $P'$ and there are still at least $n_2'$ elements left which are not in $P'$. Therefore in both cases we have

(i) either $\big|[a] \cap P\big| = \big|[a] \cap P'\big|$ or $\big|[a] \cap P\big|, \big|[a] \cap P'\big| \geq 2^{n-1}$, and

(ii) either $\big|[a] \setminus P\big| = \big|[a] \setminus P'\big|$ or $\big|[a] \setminus P\big|, \big|[a] \setminus P'\big| \geq 2^{n-1}$.

Hence,

$$(\mathfrak{A}, P)_{2^{n-1}}^{\sim_P} \cong (\mathfrak{A}_{2^n}^\sim, P')_{2^{n-1}}^{\sim_P}.$$

It remains to show that $'$ is surjective. Let $\tilde{P} \subseteq A_{2^n}^\sim$. Construct a relation $P \subseteq A$ by including $\big|[a] \cap \tilde{P}\big|$ elements of each $\sim$-class $[a]$ into $P$. Then $P' = \tilde{P}$. □

**Theorem 8.11.** WAutStr $\subset$ 1AutStr

*Proof.* Let $\mathfrak{A} = (A, R_0, \dots, R_r) \in$ WAutStr and let $\sim$ be the congruence defined in Theorem 8.7. Fix an enumeration $[a_0], \dots, [a_{n-1}]$ of $A/\sim$ and denote the $k^{\text{th}}$ member of $[a_i]$ by $a_{ik}$. Set $n_i := \big|[a_i]\big|$. We construct a unary presentation

$$\mathfrak{d} := (\nu, \{1\}, L_\delta, L_\varepsilon, L_{R_0}, \dots, L_{R_r})$$

of $\mathfrak{A}$ by encoding $a_{ik}$ by the string of length $kn + i$.

$$\begin{aligned}
\nu(1^l) &:= a_{ik} \qquad \text{where } k := \lfloor l/n \rfloor,\ i := l \pmod n, \\
L_\delta &:= \bigcup_{i<n} 1^i (1^n)^{<n_i}, \\
L_\varepsilon &:= \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]^*, \\
L_{R_j} &:= \bigcup \big\{\, 1^{i_0}(1^n)^* \otimes \dots \otimes 1^{i_{r_j-1}}(1^n)^* \bigm| ([a_{i_0}], \dots, [a_{i_{r_j-1}}]) \in R_j/\sim \big\}.
\end{aligned}$$

□

## 8.2 Star-free and Locally Threshold Testable Presentations

When looking at restrictions of regular languages one naturally thinks of star-free and locally threshold testable languages. As far as automatic presentations are concerned those classes of languages are unsuitable as the following remark shows.

**Lemma 8.12** (see e.g. [Tho97b, page 412])**.** *The classes of star-free and locally threshold testable languages are not closed under projections.*

Therefore we only have closure under quantifier-free interpretations.

**Lemma 8.13.** *Let $\mathfrak{A}$ be a structure with a star-free or locally threshold testable presentation. Then, for every quantifier-free formula $\varphi$, $(\mathfrak{A}, \varphi^{\mathfrak{A}})$ has a presentation of the same type.*

*Proof.* By definition, the class of star-free languages forms a boolean algebra. By the logical characterisation of locally threshold testable languages the same is true in case of the second class. Therefore, by the same proof as for AutStr we obtain the desired result. □

The structures in question are

$$\mathfrak{S}_p^{\leq} := (\mathbb{N}, \leq, (D_i)_{i \in \mathbb{Z}_p}) \qquad \text{and} \qquad \mathfrak{S}_p^{s} := (\mathbb{N}, s_p, (D_i)_{i \in \mathbb{Z}_p})$$

where

$$D_i xy \ :\text{iff}\ \operatorname{dig}_i(x, y) \qquad \text{and} \qquad s_p := \{\, (x, px) \mid x \in \mathbb{N} \}.$$

Again, for the characterisation via interpretations we need to define the right logic. We consider only structures with universe $\mathbb{N}$ and define $\mathrm{W}_p\mathrm{FO}$ to be the restriction of FO to quantification over powers of $p$.

As in Proposition 4.2 we encode words $w \in \mathbb{Z}_p^*$ be the number $\operatorname{val}_p(w1)$.

**Theorem 8.14.** *Let $R \subseteq \mathbb{N}^n$.*

(i) *$R$ is $\mathrm{W}_p\mathrm{FO}$-definable in $\mathfrak{S}_p^{\leq}$ if and only if $\mathrm{fold}(\mathrm{val}_p^{-1}(R))$ is star-free.*

(ii) *$R$ is $\mathrm{W}_p\mathrm{FO}$-definable in $\mathfrak{S}_p^{s}$ if and only if $\mathrm{fold}(\mathrm{val}_p^{-1}(R))$ is locally threshold testable.*

*Proof.* We prove only (i). The other case is analogous.

($\Rightarrow$) Let $\varphi(\overline{y}) \in \mathrm{FO}[\leq, (Q_i^k)_{k,i}]$ define $L$ where $Q_i^k$ is the set of positions at which the symbol $i$ appears in the $k^{\text{th}}$ component of the word. We construct a formula $\varphi^*(\overline{x}, \overline{y}) \in \mathrm{W}_p\mathrm{FO}$ such that

$$\begin{aligned} &w_0 \otimes \cdots \otimes w_{n-1} \models \varphi(r_0, \dots, r_{m-1}) \\ \text{iff } &\mathfrak{B}_p \models \varphi^*(\mathrm{val}_p(w_0), \dots, \mathrm{val}_p(w_{n-1}), p^{r_0}, \dots, p^{r_{m-1}}). \end{aligned}$$

First we define a formula specifying those positions which lie in the domain of the $k^{\text{th}}$ word, which is the case if there is a greater position carrying the digit 1.

$$\mathrm{dom}_k(y) := \exists_p z(y < z \wedge D_1 x_k z), \qquad \mathrm{dom}(y) := \bigvee_{k<n} \mathrm{dom}_k(y).$$

The translation is

$$\begin{aligned} (Q_i^k y)^* &:= \mathrm{dom}_k(y) \wedge D_i x_k y \qquad \text{for } i \neq \Box, \\ (Q_\Box^k y)^* &:= \neg\mathrm{dom}_k(y), \\ (y_i = y_j)^* &:= y_i = y_j, \\ (y_i \leq y_j)^* &:= y_i \leq y_j, \\ (\neg\varphi)^* &:= \neg\varphi^*, \\ (\varphi \vee \psi)^* &:= \varphi^* \vee \psi^*, \\ (\exists y \varphi(\overline{y}))^* &:= \exists_p y(\mathrm{dom}(y) \wedge \varphi^*(\overline{x}, \overline{y}y)). \end{aligned}$$

($\Leftarrow$) Let $\varphi(\overline{x}, \overline{y}) \in \mathrm{W}_p\mathrm{FO}$ where the variables $\overline{y}$ are guaranteed to range over powers of $p$. As variables in $\mathfrak{S}_p^{\leq}$ are unbounded whereas the positions in word models are bounded by the length of the longest word, we need to store additional information about those variables whose values are too large. Therefore we define for any tuple $(r_0, \dots, r_{m-1}) \in \mathbb{N}$

$$\mathrm{type}_n(\overline{r}) := (t_{ik})_{i,k<m}$$

where

$$t_{ik} := \begin{cases} r_i - r_k & \text{if } |r_i - r_k| < 2^n, \\ \infty & \text{if } r_i - r_k \geq 2^n, \\ -\infty & \text{if } r_i - r_k \leq -2^n. \end{cases}$$

We write $t \models y_i \leq y_k$ for some type $t$ iff $t_{ik} \leq 0$ and similarly for other formulae. Now, we can construct a formula $\varphi_t^*(\overline{y}) \in \mathrm{FO}$ such that

$$\begin{aligned} &\mathfrak{B}_p \models \varphi(\mathrm{code}_p(w_0), \dots, \mathrm{code}_p(w_{n-1}), p^{r_0}, \dots, p^{r_{m-1}}) \\ \text{iff } &w_0 \otimes \cdots \otimes w_{n-1} \models \varphi_t^*(r_{i_0}, \dots, r_{i_k}) \end{aligned}$$

where

$$l := \max\{|w_0|, \dots, |w_{n-1}|\},$$
$$t := \mathrm{type}_{\mathrm{qr}(\varphi)}(l-1, \max\{l-1, r_0\}, \dots, \max\{l-1, r_{m-1}\}),$$
$$\{r_{i_0}, \dots, r_{i_k}\} := \{\, r \in \{r_0, \dots, r_{m-1}\} \mid r < l \,\}.$$

First, we simplify $\varphi$ to $\varphi'$ by applying the following rules.

$$\begin{aligned}
(x_i = x_k)' &:= \forall_p z \bigwedge_j (D_j x_i z \leftrightarrow D_j x_k z),\\
(x_i = y)' &:= D_1 x_i y \wedge \forall_p z (z \neq y \rightarrow D_0 x_i z),\\
(y = x_i)' &:= (x_i = y)',\\
(x_i \leq y)' &:= x_i = y \vee \forall_p z (z \geq y \rightarrow D_0 x_i z),\\
(y \leq x_i)' &:= y = x_i \vee \neg (x_i \leq y)',\\
(x_i \leq x_k)' &:= x_i = x_k \vee \exists_p z \Big[ \bigvee_{j<j'} (D_j x_i z \wedge D_{j'} x_k z)\\
&\qquad \wedge \forall z' \Big( z' > z \rightarrow \bigwedge_j (D_j x_i z' \leftrightarrow D_j X_k z') \Big) \Big],\\
(D_i y_j y_k)' &:= \begin{cases} y_j \neq y_k & \text{if } i = 0,\\ y_j = y_k & \text{if } i = 1,\\ \text{false} & \text{otherwise}, \end{cases}\\
(D_i x_k x_j)' &:= \exists_p z (x_j = z \wedge D_i x_k z),\\
(D_i y x_k)' &:= \exists_p z (x_k = z \wedge D_i y z).
\end{aligned}$$

Thus, only the following cases remain. For the boolean connectives we define

$$\begin{aligned}
(\neg \varphi)^*_t &:= \neg \varphi^*_t,\\
(\varphi \vee \psi)^*_t &:= \varphi^*_t \vee \psi^*_t,\\
(\exists_p z \varphi(\overline{x}, \overline{y}))^*_t &:= \exists z \varphi^*_{t[z=l-1]}(\overline{y} z)\\
&\quad \vee \bigvee \{\, \varphi^*_{t[z=r]}(\overline{y}) \mid t \models r \leq y + 2^{\mathrm{qr}(\varphi)} \text{ for some } y \,\},
\end{aligned}$$

where we denoted by $t[z = r]$ the extension of $t$ by an additional variable with value $r$, and for the atomic formulae

$$(D_i x_k y)^*_t := \begin{cases} Q^k_i y & \text{if } t \models y < l,\\ \text{true} & \text{if } i = 1 \text{ and } t \models y = l,\\ \text{true} & \text{if } i = 0 \text{ and } t \models y > l,\\ \text{false} & \text{otherwise}, \end{cases}$$

$$(y_i = y_j)^*_t := \begin{cases} y_i = y_j & \text{if } t \models y_i < l \wedge y_j < l,\\ \text{true} & \text{if } t \models y_i = y_j \geq l,\\ \text{false} & \text{otherwise}, \end{cases}$$

$$(y_i \leq y_j)^*_t := \begin{cases} y_i \leq y_j & \text{if } t \models y_i < l \wedge y_j < l,\\ \text{true} & \text{if } t \models y_i \leq y_j \wedge l \leq y_j,\\ \text{false} & \text{otherwise}. \end{cases}$$

□

**Corollary 8.15.** (i) *A structure* $\mathfrak{A} = (A, R_0, \ldots, R_r)$ *has a star-free presentation if and only if* $\mathfrak{A} \leq_{\mathrm{W}_p\mathrm{FO}} \mathfrak{S}_p^{\leq}$ *for some* $p \in \mathbb{N} \setminus \{0, 1\}$.

(ii) *A structure* $\mathfrak{A} = (A, R_0, \ldots, R_r)$ *has a locally threshold testable presentation if and only if* $\mathfrak{A} \leq_{\mathrm{W}_p\mathrm{FO}} \mathfrak{S}_p^{s}$ *for some* $p \in \mathbb{N} \setminus \{0, 1\}$.

# Chapter 9

# Conclusion

We studied various classes of structures which can be presented in some way or other by automata. The resulting hierarchy is depicted in Figure 9.1. A common characteristic of those classes is that they allow effective—even automatic—evaluation of first-order queries. In the case of AutStr several complexity results were obtained. They are summarised in Table 9.1.

One of the most fundamental results was that in each case we were able to give an equivalent characterisation in terms of interpretations. Each class investigated turned out to be the closure of some complete structure under interpretations. This view can be applied to various other fields. For instance, the class of recursive structures can be defined as the closure of Arithmetic under $\Delta_1$-interpretations.

Another example are constraint databases. A constraint database consists of a fixed structure, called context structure, extended by relations that can be defined by quantifier-free formulae in this structure. Extensions of this kind can be regarded as interpretations of a particularly simple form. Hence the class of constraint databases using a fixed context structure is the closure of this structure under a restricted type of interpretations.

A natural generalisation of both automatic structures and constraint databases therefore consists of classes defined as the closure of some given structure under interpretations of some kind. Form a practical point of view it would be of particular interest to find classes where either the complexity of evaluating a query is acceptable or REACHABILITY becomes decidable.

Another area of possible further research would be to develop methods for proving non-membership in one of the automatic classes. To the knowledge of

| | | Structure-Complexity | Expression-Complexity |
|---|---|---|---|
| Model-Checking | $\Sigma_0$ | LOGSPACE-complete | ALOGTIME-complete |
| | $\Sigma_0$ + fun | NLOGSPACE | PTIME-complete |
| | $\Sigma_1$ | NPTIME-complete | PSPACE-complete |
| Query-Evaluation | $\Sigma_0$ | LOGSPACE | PSPACE |
| | $\Sigma_1$ | PSPACE | EXPSPACE |

Table 9.1: Complexity results for AutStr

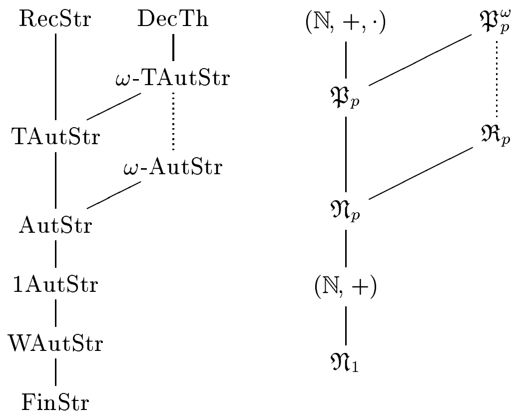



Figure 9.1: Hierarchy of automatic classes and complete structures

the author up to now only two such methods are available: showing that the FO-theory is undecidable and proving a more than exponential lower bound on the cardinality of generations. In particular there is no tool to separate $\omega$-TAutStr from $\omega$-AutStr.

Finally, many questions in model theory remain unresolved. Besides compactness there are several other results in classical model theory which fail for most restricted classes, e.g., Craig's Interpolation Theorem, Beth's Definability Theorem, Lyndon's Lemma, and other preservation properties. Up to now it is unknown whether these results do or do not hold in the case of automatic structures. A first step to answer those questions could be to show that there are no automatic non-standard models of Th($\mathfrak{N}_p$). In that case it would be possible to axiomatise a well-ordering, and if, furthermore, it were possible to reduce this axiom system to a finite one, one would have a tool which perhaps could be used to answer the above questions.

# Bibliography


[BHMV94] V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire, *Logic and p-recognizable sets of integers*, Bull. Belg. Math. Soc. **1** (1994), 191–238.

[BRW98] B. Boigelot, S. Rassart, and P. Wolper, *On the Expressiveness of Real and Integer Arithmetic Automata*, LNCS **1443** (1998), 152–163.

[Bus87] S. R. Buss, *The boolean formula value problem is in* ALOGTIME, Proc. 19th ACM Symp. on the Theory of Computing, 1987, pp. 123–131.

[ECH+92] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P Thurston, *Word Processing in Groups*, Jones and Bartlett Publishers, Boston, 1992.

[EF95] H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer, New York, 1995.

[EFT94] H.-D. Ebbinghaus, J. Flum, and W. Thomas, *Mathematical Logic*, 2nd ed., Springer, New York, 1994.

[Eil74] S. Eilenberg, *Automata, Languages, and Machines*, vol. A, Academic Press, New York, 1974.

[GS97] F. Gécseg and M. Steinby, *Tree Languages*, Handbook of Formal Languages (G. Rozenberg and A. Salomaa, eds.), vol. 3, Springer, New York, 1997, pp. 1–68.

[Grä90] E. Grädel, *Simple Interpretations among Complicated Theories*, Information Processing Letters **35** (1990), 235–238.

[HH96] T. Hirst and D. Harel, *More about Recursive Structures: Descriptive Complexity and Zero-One Laws*, Proc. 11th IEEE Symp. on Logic in Comp. Sci., 1996, pp. 334–348.

[HRS76] H. B. Hunt III, D. J. Rosenkrantz, and T. G. Szymanski, *On the Equivalence, Containment, and Covering Problems for the Regular and Context-Free Languages*, Journal of Computer and System Sciences **12** (1976), 222–268.

[HU79] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Mass., 1979.

[Hod93] W. Hodges, *Model theory*, Cambridge University Press, 1993.

[Imm98] N. Immerman, *Descriptive Complexity*, Springer, New York, 1998.

[KK71] G. Kreisel and J. L. Krivine, *Elements of Mathematical Logic*, North-Holland, Amsterdam, 1971.

[KN95] B. Khoussainov and A. Nerode, *Automatic Presentations of Structures*, LNCS **960** (1995), 367–392.

[KR99] B. Khoussainov and S. Rubin, *Finite Automata and Isomorphism Types*, unpublished, 1999.

[MS73] A. R. Meyer and L. J. Stockmeyer, *Word problems requiring exponential time*, Proc. 5th ACM Symp. on the Theory of Computing, 1973, pp. 1–9.

[Pap94] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, Mass., 1994.

[RS97] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages*, vol. 1–3, Springer, New York, 1997.

[Sch98] J. H. Schmerl, *Recursive Models and the Divisibility Poset*, Notre Dame Journal of Formal Logic **39** (1998), no. 1, 140–148.

[Tho90] W. Thomas, *Automata on infinite objects*, Handbook of Theoretical Computer Science (J. van Leeuwen, ed.), vol. B, Elsevier, Amsterdam, 1990, pp. 133–191.

[Tho97a] W. Thomas, *Ehrenfeucht Games, the Composition Method, and the Monadic Theory of Ordinal Words*, Structures in Logic and Computer Science (J. Mycielski, ed.), DIMACS Series in Discrete Math. and Theor. Comp. Sci., no. 29, Am. Math. Soc., 1997, pp. 25–40.

[Tho97b] W. Thomas, *Languages, Automata, and Logic*, Handbook of Formal Languages (G. Rozenberg and A. Salomaa, eds.), vol. 3, Springer, New York, 1997, pp. 389–455.

[Zei94] R. S. Zeitman, *The Composition Method*, Ph. D. Thesis, Wayne State Univ., Michigan, 1994.

# Index


$\sim_{l,p}$, **69**
1AutStr, **69**

Alphabet, 10, 69
Arithmetic, 7, 15–17
  Peano —, 68
  Presburger —, 10, 17, 44, 45, 50, 72
  Skolem —, 51, 57, 58
Automaton, **3**, 31, 62
  — normal form, **32**, 62
  deterministic —, **3**, 19
  nondeterministic —, 19
  $\omega$- —, **3**, 37
  $\omega$-tree —, **5**, 41
  tree —, **5**, 40
AutStr, **9**, 29
Axioms of Th($\mathfrak{N}_p$), 62

Borel hierarchy, **4**, 11

$f$-Chain, **73**, 74
Closure, 16, 72
  — of regular languages, 3
  — under generalised products, 54, 55
  — under interpretations, 15, 29, 69, 73, 89
Complete structure, **29**, 54, 72, 88, 94, 95
Complexity, 18–27, 81, 82
  expression —, **18**
  structure —, **18**
Congruence, 89, 90
  Nerode- —, **3**, 14, 47
Convolution, **4**, 5, 9
  weak —, **87**

Decidability, 7, 15, 17, 60, 85, 89
Definitional equivalent, **7**

$\eta^{\mathfrak{d}}$, **13**, 88
Equivalence relation, 48, 49, 77

FFO, **88**

FO, **6**, 13, 15
FO(#), **6**, 17
FO(closed DTC), 85
FO(DTC), **6**, 17, 85
FO($\exists^\omega$), **6**, 14
fold, **4**
FO(PFP), 89
FO(R), **84**

Generation, **45**, 45–47, 51, 72
Graph, 78
  Random —, 79
Group, 79, 80

$\iota$, **47**
Index, 14, 47, 80, 89
Injective, 11, 58
Interpretation, **6**, 15, 29, 54, 69, 72, 88, 89, 94, 95

$\lambda^{\mathfrak{d}}$, **10**
Length, 10, 43–45, 47, 48, 73
Loop constants, 71, **71**, 74, 78, 84

Model-checking, 18, **18**, 22–24, 81, 82
Monoid
  free —, 46
  trace —, 46
MSO, **6**, 90
Mutually interpretable, **7**

$\mathfrak{N}_1$, **69**, 84
$\mathfrak{N}_p$, **29**, 62–68, 88

Ordered sum, 53, 55, 72
Ordering, 11, 48, 78
  alphabetic —, **4**
  lexicographic —, **4**
Ordinal, 78

$\mathfrak{P}_p$, **38**
$\mathfrak{P}_p^\omega$, **40**
Pairing function, 46
Permutation, 49, 76
Power, 56

Power series, **37**
Presentation
  automatic —, **9**
  locally threshold testable —, **91**
  star-free —, **91**
  unary —, **69**
  weak —, **87**
Product, 72
  direct —, 53, 55
  generalised —, **53**
Pumping Lemma, **4**, 11, 15, 26, 49

Quantifier
  — elimination, 32, 33, 83
  — -free, 22, 23, 26, 91
  Ramsey- —, 84
Query-evaluation, 12, **18**, 26, 27

$\mathfrak{R}_p$, **34**
$\mathfrak{R}_p^+$, **34**
REACHABILITY, 17, 23
Reduct, **7**, 16

$\mathfrak{S}_p$, **62**
Signature, **6**
SO, **6**, 89
Substructure, 16

$\mathfrak{T}_p$, **38**
$\mathfrak{T}_p^\omega$, **40**
Tree, **5**, 37
Type, **52**, 92

unfold, **4**
Union, 55, 72

$\mathfrak{W}(\Sigma)$, **29**
$\mathfrak{W}^\omega(\Sigma)$, **34**
WAutStr, **87**
$\mathrm{W}_p$FO, **91**