Definability on ω-Automatic Structures

Faried Abu Zaid
Matrikelnummer 251727

Diplomarbeit
vorgelegt der Fakultät
für Mathematik, Informatik und Naturwissenschaften
der Rheinisch-Westfälischen Technischen Hochschule Aachen

im Februar 2011 (letzte Version: 10.02.2011)

angefertigt am
Lehr- und Forschungsgebiet
Mathematische Grundlagen der Informatik
Prof. Dr. Erich Grädel

Hiermit versichere ich, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Aachen, den 10. Februar 2011

(Faried Abu Zaid)

Contents

List of Figures

Chapter 1

Introduction

Many of the problems that we face in computer science can be modeled as the question whether a property φ holds in a structure $\mathfrak{A}$. As an example one may think of the numerous graph problems such as *reachability* or *3-colorability*. The aim of computer science is to solve these problems in a generic but efficient way. In the most generic case this would mean that we provide an algorithm that gets $\mathfrak{A}$ and φ as inputs and decides if φ holds in $\mathfrak{A}$.

To provide such an algorithm we need to encode $\mathfrak{A}$ and φ in a way that is accessible to the algorithm. The coding of φ seems to be best done by restricting to properties that can be formulated in some formal logic. The coding of $\mathfrak{A}$ can also be done very naturally if we restrict ourselves to finite structures.

This setting was very successfully issued by finite model theory which has influenced many other fields of computer science. For example it has been shown that there is a strong correspondence between the logic $\mathfrak{L}$ that is chosen and the complexity of the problem that we obtain when we fix some property φ that is expressible in $\mathfrak{L}$. Most notably here are in particular Fagin's theorem that existential second order logic captures NP on the class of all finite structures [6], and the Immerman-Vardi theorem which states that least fixed point logic captures PTIME at least on ordered finite structures [10, 20].

In the last years there have been some efforts to include infinite structures into this consideration. Here the question of the structure encoding arises again. It is obvious that not every structure can be encoded in a finite way and therefore we need to restrict ourselves to suitable classes of structures. Another aspect that must be taken into account is that there is a trade-off between the complexity of the structures that can be represented and the complexity of the queries that can be solved algorithmically.

For example recursive structures have been considered, that is structures over a recursive domain whose relations are also recursive. These structures can be encoded by Turing machines that recognize the relations, but the resulting class of structures is far too complex for our purposes. For instance there is no uniform way to decide, given a recursive relation R, if $\exists x Rx$ holds.

A more suitable class of structures, namely the automatic structures, was proposed by Khoussainov and Nerode in [12]. Here the structure is encoded by finite automata that recognize the domain and the relations of the structure. Automata theory is a very well studied field of computer science and especially the relation to logic is extensively investigated. Thus, automatic structures give us a robust framework to start with.

Automatic structures can be considered over finite words or trees and over infinite words or trees which yields [ω][tree]-automatic structures. A lot of the fundamental work on these classes of structures was done by Blumensath in [2].

One of the major open tasks today is to explore the boundaries of these classes. Therefore we need tools and techniques that allow us to show that a structure is not automatic. For finite word automatic structures some techniques have been developed but for the other classes not too much is known.

This work focuses on ω-automatic structures. Our main goal is to develop techniques that enable us to prove the non-automaticity of a structure.

In Chapter 2 we repeat the basic notations of formal logic.

Chapter 3 is devoted to languages over infinite words. We explain the general concepts of ω-languages with an special emphasis on ω-regular languages. We discuss several possibilities to describe ω-regular languages and take a deeper look at the recognition of ω-automatic languages by ω-semigroups. In the last section we define ω-automatic presentations as a special case of general ωLang presentations.

In Chapter 4 we review some of the basic and current results on automatic structures. In the last few years some important open questions on ω-automatic structures were answered.

For example, Bárány, Kaiser and Rubin show in [11] that first order theory of ω-automatic structures remains decidable even if one adds cardinality and modulo counting quantifiers. As a consequence they derive that a countable structure is ω-automatic iff it is automatic. Another interesting result is due to Hjorth, Khoussainov, Montalbán and Nies. They answer in [9] the injectivity problem for ω-automatic structures by showing that there are ω-automatic that do not have an injective presentation.

Additionally we transfer the notion of growth rates to presentations over infinite words. We do so by restricting to words that are periodic from some point onwards and measure the length of the non-periodic prefix as well as the length of the period. We show that the growth rates we can observe in this setting are much larger than what can be found in the finite word case.

Chapter 5 discusses pairing functions. We discuss why growth rate related arguments, as they were used for the finite word case, will fail. As a replacement we consider ω-automatic functions $f : L^k \to L$ and words that are equal from some fixed position m onwards (denoted by $v \sim_e^m w$). By investigating the pictures of the $\sim_e^m$-classes of L we show that the minimal picture size $|f(X^k)|$ for finite sets $X \subseteq L$ is bounded above by a function that grows linear in the size of X. We use this result to show that there are no injective presentations of pairing functions.

Chapter 6 examines substructures of automatic structures. Given an ω-automatic presentation $\mathcal{L}$ of a structure $\mathfrak{A}$, and a substructure $\mathfrak{B}$ of $\mathfrak{A}$, we investigate the complexity of the presentations $\mathcal{L}'$ of $\mathfrak{B}$ that are contained in $\mathcal{L}$. Following the work of Bárány, Kaiser and Rubin in [11] and the work of Kuske in [14] we show that every ω-automatic presentation of an uncountable linear order contains an injective ω-automatic presentation of the structure $(\{0,1\}^\omega, \prec)$, where $\prec$ is the lexicographic order on $\{0,1\}^\omega$. This directly leads to the proof that an ordinal is ω-automatic if and only if it is smaller than ω^ω. As another result we use the existence of an injective subpresentation of an uncountable substructure to show that the results of Chapter 5 can be expanded to (not necessarily injective) ω-automatic presentations of uncountable ordered structures.

In Chapter 7 we take a look at the field of reals $(\mathbb{R}, +, \cdot)$. We explore some basic approaches to show that the real field is not automatic and apply the concepts that have been developed in the previous chapters.

We show that all reducts of $(\mathbb{R}, +, \cdot)$ are automatic and examine the coding length of several definable elements in ω-automatic presentations of $(\mathbb{R}, +, \cdot)$. We show that there are natural numbers n such that their coding length is logarithmic in n but in general the best bound we can give is $n^{\mathcal{O}((\log\log n)^2)}$.

Additionally we show that for every ω-automatic presentation of $(\mathbb{R}, +, \cdot)$ the number of natural numbers that are encoded by words of one $\sim_e$ class is bounded by a constant and investigate the encodings of the rational numbers.

Acknowledgements First I want to thank Prof. Erich Grädel for the possibility to work on this topic, his guiding and his patience. I also want to thank Dr. Łukasz Kaiser for the insightful discussions and suggestions which helped me a lot during this thesis. A special thanks goes to Sebastian Siebertz, Simon Lessenich and Wied Pakusa for the great time that we could spend while I was writing this thesis. This apllies also to the staff of the MGI and I7 who created a wonderful working environment.

At last I am most grateful to my wife Monika for her careful corrections and, which is unmeasurably more valuable, for her love.

Chapter 2

Logic and Structures

We assume that the reader is familiar with the basic concepts of formal logic. Therefore we will only shortly sum up the notions that we will need in this work. For a detailed introduction we refer for example to [5, 19].

A signature τ is a set of relation and function symbols where every symbol has an associated arity. We say τ is relational iff τ contains no function symbols. As a convention we will denote relation symbols with capital letters and function symbols with small letters. For $\tau = \{R_1, \dots, R_n, f_1, \dots, f_m\}$, where r_i is the arity of R_i and s_j is the arity of f_j, a τ-structure $\mathfrak{A}$ is a tuple $(A, R_1^{\mathfrak{A}}, \dots, R_n^{\mathfrak{A}}, f_1^{\mathfrak{A}}, \dots, f_m^{\mathfrak{A}})$ that consists of a nonempty set A, relations $R_i^{\mathfrak{A}} \subseteq A^{r_i}$ and functions $f_j^{\mathfrak{A}} : A^{s_j} \to A$. We will not always make a notational difference between a function or relation symbol and the interpretation of this symbol in a structure $\mathfrak{A}$. A substructure of $\mathfrak{A}$ is a structure $\mathfrak{B} = (B, R_0^{\mathfrak{B}}, \dots, R_n^{\mathfrak{B}}, f_0^{\mathfrak{B}}, \dots, f_m^{\mathfrak{B}})$ with

- $B \subseteq A$,
- $R_i^{\mathfrak{B}} = R_i^{\mathfrak{A}} \cap B^{r_i}$,
- $f_i^{\mathfrak{B}} = f_i^{\mathfrak{A}} \restriction B$ and
- $f_i^{\mathfrak{B}}(B) \subseteq B$.

A congruence relation $\sim$ on a structure $\mathfrak{A} = (A, f_1, \dots, f_n, R_1, \dots, R_m)$ is an equivalence relation on A that fulfills the following properties:

- If $\overline{x} \sim \overline{y}$ then also $f_i(\overline{x}) \sim f_i(\overline{y})$.

- If $\overline{x} \sim \overline{y}$ and $\overline{x} \in R_j$ holds then also $\overline{y} \in R_j$.

For $a \in A$ we write $[a]_\sim$ for the set $\{x \in A : x \sim a\}$ and $A/_\sim$ for $\{[a] : a \in A\}$. The factor structure $\mathfrak{A}/_\sim$ is the structure $(A/_\sim, f_1^{\mathfrak{A}/\sim}, \dots, f_n^{\mathfrak{A}/\sim}, R_1^{\mathfrak{A}/\sim}, \dots, R_m^{\mathfrak{A}/\sim})$ where $f_i^{\mathfrak{A}/\sim}([x_1]_\sim, \dots, [x_k]_\sim) = [f_i^{\mathfrak{A}}(x_1, \dots, x_k)]_\sim$ and $R_i^{\mathfrak{A}/\sim} = \{([x_1]_\sim, \dots, [x_k]_\sim) : (x_1, \dots, x_k) \in R_i^{\mathfrak{A}}\}$

The first order logic $\mathrm{FO}(\tau)$ is the set of formulas that can be constructed from atomic formulas using the boolean operations $\wedge, \vee, \neg$ and the quantifiers $\exists$ and $\forall$. The atomic formulas are built up from terms using the relation symbols in τ and the equality symbol $=$. This means, if $t_1, \dots, t_{r_i}$ are terms then $R_i(t_1, \dots, t_{r_i})$ and $t_i = t_j$ are atomic formulas. At last terms can be constructed inductively using variable symbols $x_0, x_1, x_2, \dots$ and the function symbols in τ. First every variable symbol x is a term and if $t_1, \dots, t_{s_i}$ are terms then $f_i(t_1, \dots, t_{s_i})$ is also a term.

We can evaluate a formula $\varphi(\overline{x}) \in \mathrm{FO}(\tau)$ on a τ-structure $\mathfrak{A}$ if we assign elements of $\mathfrak{A}$ to the free variables $\overline{x}$ of φ. This assignment is given by a function $\theta : \overline{x} \to A$. We extend the definition of θ to terms by the rule $\theta(f(t_1, \dots, t_r)) = f^{\mathfrak{A}}(\theta(t_1), \dots, \theta(t_r))$. We write $\theta[x \leftarrow a]$ for the assignment $\theta' : \overline{x} \cup \{x\}$ with

$$\theta'(y) = \begin{cases} a, & \text{if } y = x \\ \theta(y), & \text{else} \end{cases}$$

The evaluation of $\varphi(\overline{x})$ is then defined inductively as described below.

$$\mathfrak{A}, \theta \models t_1 = t_2 \text{ iff } \theta(t_1) = \theta(t_2)$$
$$\mathfrak{A}, \theta \models R_i(t_1, \dots, t_{r_i}) \text{ iff } (\theta(t_1), \dots, \theta(t_{r_1})) \in R_i^{\mathfrak{A}}$$
$$\mathfrak{A}, \theta \models \varphi \wedge \psi \text{ iff } \mathfrak{A}, \theta \models \varphi \text{ and } \mathfrak{A}, \theta \models \psi$$
$$\mathfrak{A}, \theta \models \varphi \vee \psi \text{ iff } \mathfrak{A}, \theta \models \varphi \text{ or } \mathfrak{A}, \theta \models \psi$$
$$\mathfrak{A}, \theta \models \neg\varphi \text{ iff it does not hold that } \mathfrak{A}, \theta \models \varphi$$
$$\mathfrak{A}, \theta \models \exists x \varphi(x, \overline{y}) \text{ iff } \mathfrak{A}, \theta[x \leftarrow a] \models \varphi(x, \overline{y}) \text{ for some } a \in A$$
$$\mathfrak{A}, \theta \models \forall x \varphi(x, \overline{y}) \text{ iff } \mathfrak{A}, \theta[x \leftarrow a] \models \varphi(x, \overline{y}) \text{ for every } a \in A$$

We also want to mention here the extension of first order logic by the quantifiers $\exists^{(k \mod m)}, \exists^{\leq \aleph_0}, \exists^{> \aleph_0}$. The semantic is

- $\mathfrak{A}, \theta \models \exists^{(k \mod m)} x\varphi$ iff $\varphi^{\mathfrak{A}}$ is finite and $|\varphi^{\mathfrak{A}}| = k \mod m$.
- $\mathfrak{A}, \theta \models \exists^{\leq \aleph_0} x\varphi$ iff there are at most countably many $a \in A$ such that $\mathfrak{A}, \theta[x \leftarrow a] \models \varphi$.
- $\mathfrak{A}, \theta \models \exists^{> \aleph_0} x\varphi$ iff there are uncountably many $a \in A$ such that $\mathfrak{A}, \theta[x \leftarrow a] \models \varphi$.

We denote this extension by FOC.

In this work we will only consider relational signatures. This is not a huge restriction since every structure can be made relational by replacing every function f by its graph G_f. Here G_f is the relation $\{(\overline{x}, y) : f(\overline{x}) = y\}$.

We also want to mention here the notion of an interpretation which is closely related to the notion of ωLang presentation that we will focus on. The idea is to describe a structure $\mathfrak{A}$ within a structure $\mathfrak{B}$ using formulas of a specified logic $\mathfrak{L}$. The elements of $\mathfrak{A}$ are coded by tuples of elements of $\mathfrak{B}$. The set off all tuples that code elements and the relations is then described by formulas.

Definition 2.0.1. Let $\mathfrak{L}$ be a logic, $\mathfrak{A} = (A, R_1, \ldots, R_n)$ where the R_i are relations of arity r_i and $\mathfrak{B} = (B, R'_1, \ldots, R'_m)$ some structure. A (k-dimensional) $\mathfrak{L}$ interpretation of $\mathfrak{A}$ in $\mathfrak{B}$ is a tuple $(\varphi_\delta(\overline{x}), \varphi_\approx(\overline{x}, \overline{y}), \varphi_{R_1}(\overline{x}_1, \ldots, \overline{x}_{r_1}), \ldots, \varphi_{R_n}(\overline{x}_1, \ldots, \overline{x}_{r_n}))$ of $\mathfrak{L}$ formulas over the signature of $\mathfrak{B}$ together with a surjective naming function $f : \varphi_\delta^{\mathfrak{B}} \to A$ such that:

- For every $\overline{b} \in (\varphi_\delta^{\mathfrak{B}})^{r_i}$ it holds that $\mathfrak{B} \models \varphi_{R_i}(\overline{b})$ if and only if $\mathfrak{A} \models R_i f(\overline{b})$
- For every $b, b' \in \varphi_\delta^{\mathfrak{B}}$ it holds that $b \approx b'$ if and only if $f(b) = f(b')$.

Chapter 3

Languages and Presentations

3.1 Languages Over Infinite Words

Formal languages are one of the main concepts of theoretical computer science. We will shortly introduce some of the basic notions concerning languages over infinite words. For a broader introduction see for example [18, 19, 16].

Generally a language is a set of words and words are concatenations of letters of a specified alphabet. As a convention we will use the symbols Σ and Γ for the alphabets and u, v, w for both, finite and infinite words. If we want to make a clearer distinction between finite and infinite words we will use Greek letters $\alpha, \beta, \ldots$ for infinite words.

A finite word of length n over the alphabet Σ can be seen as a function $w : \{0, \ldots, n-1\} \to \Sigma$. We write Σ^* for the set of all finite words of any length over Σ. This definition of a finite word can naturally be modified to a notion of an infinite word. An infinite word is a function $\alpha : \mathbb{N} \to \Sigma$. We write Σ^ω for the set of all infinite words over Σ.

There are some classes of ω-languages that are of special importance for us. We address these classes in the next sections.

3.2 Regular Languages

Regular languages form a very well studied class of languages that are applicable in many fields of computer science. One reason for their popularity is that there are several ways to encode this languages by finite objects such that they can be handled algorithmically. For instance the basic set theoretic operations can be carried out effectively and important decision problems like the membership or emptiness problem are decidable on this presentations. One of the most important ways of presenting regular languages is by finite automata, therefore we call regular languages also automatic.

In the classical setting regular languages range over finite words and their equivalents for infinite words are called an ω-regular or ω-automatic languages. But if it is obvious from the context that we refer to infinite words we will sometimes call these languages *regular* or *automatic*.

Finite word regular languages can be seen as a special case of ω-regular languages. More precisely one can easily show that for any finite word language L over the alphabet Σ it holds that L is regular iff $L\Box^\omega = \{w\Box^\omega : w \in L\}$ is ω-regular. Here $\Box$ is a new "dummy" letter that is not contained in Σ.

3.2.1 Regular Expressions

We define the ω-regular languages by the so called ω-regular expressions. These expressions are built up from regular expressions. The regular languages are exactly those languages that can be represented by a regular expression. We define the regular expressions inductively as follows:

- Every $a \in \Sigma$ is a regular expression.
- If α is a regular expression then α^* is also a regular expression.
- If α and β are regular expressions then $\alpha\beta$ and $(\alpha+\beta)$ are also regular expressions.

The ω-regular expressions are then exactly the expressions of the form $\alpha_1\beta_1^\omega + \ldots + \alpha_n\beta_n^\omega$ where the α_i, β_i are regular expressions.

With every (ω-)regular expression α we associate a language $L(\alpha) \subseteq \Sigma^+$ (Σ^ω). First, for the regular expressions we define:

- $L(a) := \{a\}$ for every $a \in \Sigma$.
- $L(\alpha^*) := \{w_1w_2\ldots w_n : n \in \mathbb{N}, w_i \in L(\alpha)\}$
- $L(\alpha + \beta) := L(\alpha) \cup L(\beta)$
- $L(\alpha\beta) := \{vw : v \in L(\alpha), w \in L(\beta)\}$

Then for the ω-regular expressions we define

$$L(\alpha_1\beta_1^\omega + \ldots + \alpha_n\beta_n^\omega) := \bigcup_{1\leq i\leq n} \{w_0w_1w_2w_3\ldots : w_0 \in L(\alpha), w_i \in L(\beta) - \{\epsilon\} \text{ for } i > 0\}.$$

For notational convenience we will often directly identify the regular expression α with its associated language $L(\alpha)$.

3.2.2 Automata

Besides regular expressions, automata are the most usual way to describe regular languages. A finite word automaton is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ with Q the finite set of states, input alphabet Σ, initial state q_0, transition relation $\Delta \subseteq Q \times \Sigma \times Q$ and the accepting states $F \subseteq Q$.

Similar, a finite ω-word automaton is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{F})$ with Q the finite set of states, input alphabet Σ, initial state q_0, transition relation $\Delta \subseteq Q \times \Sigma \times Q$ and the accepting runs $F \subseteq Q^\omega$. We say that a run ρ of $\mathcal{A}$ is consistent with a word w if $\rho[0] = q_0$ and $(\rho[i], w[i], \rho[i+1]) \in \Delta$ for all $i \in \mathbb{N}$. We say w is accepted by $\mathcal{A}$ if there is an accepting run that is consistent with w. With $L(\mathcal{A})$ we denote the language recognized by $\mathcal{A}$.

To use this definition in practice, we need to represent $\mathcal{F}$ in a finite way. By means of how $\mathcal{F}$ is represented we distinguish between several kinds of ω-word automata. We mention here commonly used acceptance conditions. For a run ρ we write $\mathit{inf}(\rho)$ for the set all $q \in Q$ such that q occurs infinitely often in ρ.

- The Büchi condition is described as a set $F \subseteq Q$ which stands for the set of accepting runs $\mathcal{F} = \{\rho \in Q^\omega : \mathit{inf}(\rho) \cap F \neq \emptyset\}$.
- The Muller condition is described by a set $F \subseteq \mathcal{P}(Q)$ which stands for the set of accepting runs $\mathcal{F} = \{\rho \in Q^\omega : \mathit{inf}(\rho) \in F\}$.
- The parity condition is given by a coloring function $\Omega : Q \to \{0, \ldots, n\}$ and a run is accepting iff the minimal color that occurs infinitely often is even. The explicit set of accepting runs is therefore $\mathcal{F} = \{\rho \in Q^\omega : \min(\Omega(\mathit{inf}(\rho))) \text{ is even}\}$.

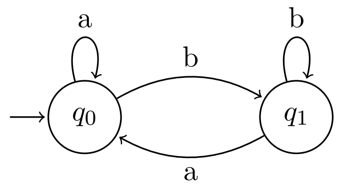

Figure 3.1: Büchi-automaton recognizing $(b^*a)^\omega$

The acceptance conditions presented here are all equivalent in the sense that they all capture the same class of languages. The only difference is whether or not nondeterminism is needed.

Theorem 3.2.1. *Let $L \subseteq \Sigma^\omega$ the following statements are equivalent:*

- *L is regular.*
- *L is recognized by some nondeterministic Büchi automaton.*
- *L is recognized by some deterministic Muller automaton.*
- *L is recognized by some deterministic parity automaton.*

So for parity and Muller conditions nondeterministic automata have the same expressive power as deterministic ones but for Büchi conditions the deterministic automata are strictly weaker than the nondeterministic ones.

Example 3.2.1. The language $(b^*a)^\omega$ can be recognized by the automaton shown in Figure 3.1 with the following acceptance conditions:

- $F = \{q_0\}$ (Büchi condition)
- $F = \{\{q_0\}, \{q_0, q_1\}\}$ (Muller condition)
- $\Omega(q_i) = i$ (Parity condition)

The state q_0 indicates that the last seen letter was an a and q_1 indicates that the last seen letter was a b. So obviously the automaton accepts iff infinitely often an a is seen for every given acceptance condition.

3.2.3 ω-Semigroups

Another interesting way to characterize regular languages is via so called ω-semigroups. We want to give a short overview on the concepts needed to define the recognition of a language by an ω-semigroup. Concerning this we will stick to the presentation given by Perrin and Pin in [16].

A ω-semigroup is a two-sorted structure $S = (S_f, S_\omega, \cdot, *, \pi)$ with the following properties:

- $(S_f, \cdot)$ is a semigroup.
- $* : S_f \times S_\omega \to S_\omega$ is the mixed product satisfying for $x, y \in S_f$, $z \in S_\omega$

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

- $\pi : S_f^\omega \to S_\omega$ is the infinite product that satisfies

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

- Additionally we demand some kind of associativity rule namely that for every strictly increasing sequence of positive integers $(k_i)_{i\in\mathbb{N}}$ it holds that

$$\pi(x_1, x_2, x_3, \ldots) = \pi((x_1 x_2 \ldots x_{k_1}), (x_{k_1+1} x_{k_1+2} \ldots x_{k_2}), \ldots).$$

Because of the last two items we can write the product $x_0 * \pi(x_1, x_2, x_3, \ldots)$ without ambiguity as $x_0 x_1 x_2 x_3 \ldots$. We will also sometimes write $(x_i)_{i\in\mathbb{N}}$ for the product $\pi(x_1, x_2, x_3, \ldots)$.

Example 3.2.2. For any set X the free ω-semigroup generated by X is the structure $X^\infty = (X^+, X^\omega, \cdot, *, \pi)$ where the operators are the usual concatenation products.

The first problem we encounter with these ω-semigroups is that even in the case that S_f and S_ω are finite sets, S is not a finite object because the domain of π is still uncountable.

Wilke showed in [21] that this problem can be bypassed since π is completely determined by the products of the form $x^\omega = xxx\ldots$. He uses so called Wilke algebras. Those are structures $(S_f, S_\omega, \cdot, *, {}^\omega)$ where $(S_f, \cdot)$ is a semigroup and the products $* : S_f \times S_\omega \to S_\omega$ and ${}^\omega : S_f \to S_\omega$ satisfy

$$x * (yx)^\omega = (xy)^\omega$$

and

$$(x^n)^\omega = x^\omega.$$

He shows that every finite Wilke algebra can be extended in a unique way to a finite ω-semigroup. Therefore we can represent each finite ω-semigroup by the corresponding Wilke algebra.

This property follows from an application of Ramsey's theorem to ω-semigroups. Since this technique will be very important for us in a slightly different context, we will shortly explain the main results being used. First, we will recall Ramsey's theorem for the case of undirected graphs.

Theorem 3.2.2 (Ramsey)**.** *Let (V, E) be the countably infinite, complete and undirected Graph and $c : E \to C$ a symmetric coloring function into some finite set of colors C. Then there is an infinite subset V' of V such that $c(u, v) = c(u', v')$ for all $(u, v), (u', v') \in E \cap (V' \times V')$.*

We can apply this to ω-semigroups by associating every product $\rho = x_0x_1x_2\ldots$ with a coloring c_ρ.

Lemma 3.2.1. Let $(S_f, S_\omega, \cdot, *, \pi)$ be a finite ω-semigroup. Then for every product $\rho = x_0x_1x_2\ldots$ there is a $s \in S_f$ and a strictly increasing sequence $(h_i)_{i\in\mathbb{N}}$ with $x_{h_i}x_{(h_i+1)}\ldots x_{(h_{i+1}-1)} = s$ for every $i \in \mathbb{N}$.

Proof. Let $G = (V, E)$ be the complete, countably infinite, undirected Graph. We define a coloring $c_\rho : E \to S_f$. To do so we may assume that $V = \mathbb{N}$. We define:

$$c(i, j) := x_i \cdot \ldots \cdot x_{j-1}$$

By Ramsey's theorem there is a monochromatic subset $H = \{h_1 < h_2 < h_3 \ldots\}$ of $\mathbb{N}$. This means $c(h_i, h_j) = c(h_{i'}, h_{j'})$ for every $i \neq j, i' \neq j'$. We now set $s = c(h_1, h_2)$.

Then by the definition of the coloring s and the sequence $(h_i)_{i\in\mathbb{N}}$ have the wanted properties. □

We will see later in this section that this property can be very useful when we examine a language that is recognized by a certain ω-semigroup.

First we need to define how a language can be recognized by an ω-semigroup. For this we need the notion of an ω-semigroup morphism. Given two ω-semigroups S and T a ω-semigroup morphism denoted by $g : S \to T$ is a pair (g_f, g_ω) where:

- g_f is a semigroup morphism from $(S_f, \cdot)$ to $(T_f, \cdot)$.
- The function $g_\omega : S_\omega \to T_\omega$ preserves the mixed and the infinite product. This means that for every sequence $(x_i)_{i\in\mathbb{N}}$ it holds that

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

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

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

If a certain function is clearly implied by the context, we will simplify the notation and omit the subscript.

We can now use morphisms from Σ^∞ to finite ω-semigroups to define the recognition of languages.

Definition 3.2.3. Let $L \subseteq \Sigma^\omega$ be some language and $g : \Sigma^\infty \to S$ a morphism into some finite ω-semigroup S. We say L is recognized by S via g if and only if $g_\omega^{-1}(g_\omega(L)) = L$.

This means that every $X \subseteq S_\omega$ defines a language namely $g^{-1}(X)$. Observe that every morphism from $g : \Sigma^\infty \to S$ is completely determined by the values $g_f(a), a \in \Sigma$. Therefore we can represent every such morphism in a finite way.

Now that we know how to recognize a language we state that in fact the languages that can be recognized be finite ω-semigroups are exactly the regular languages.

Theorem 3.2.4. *From a finite ω-semigroup S given as its corresponding Wilke algebra and a morphism $g : \Sigma^\infty \to S$ that recognize the language L one can effectively compute a Büchi-automaton that recognizes L and vice versa.*

For a proof of this theorem we refer to [16, section 9 and 10].

Example 3.2.3. The Language $(b^*a)^\omega$ is recognized by the ω-semigroup $(\{0,1\},\{0,1\},\cdot,*,\pi)$ with

- $x \cdot y = \min\{x, y\}$,
- $x * y = y$ and
- $x_1x_2x_3\ldots = \begin{cases} 0, & \text{if } x_i = 0 \text{ for infinitely many } i \\ 1, & \text{else} \end{cases}$

via the morphism g induced by $g(a) = 0$ and $g(b) = 1$. This holds since a word w is in $g_\omega^{-1}(0)$ iff w contains infinitely many a iff $w \in (b^*a)^\omega$.

As the comparison with example 3.2.1 may suggest, the representation of a language by an ω-semigroup often does not give the same intuitive idea on how the words of this language can be constructed. But the strength of such presentations becomes obvious when we look at common techniques used to prove properties of regular languages. This particularly applies to proofs that involve the decomposition and rearranging of some given words.

Usually one wants to decompose one or more words in a way such that we can repeat or permute certain parts to create new words whose acceptance behavior is determined by the acceptance behavior of the original words. The key to this capability is again the application of Ramsey's theorem. One can use it to directly obtain suitable decompositions of words.

Definition 3.2.5. Let $g : \Sigma^\infty \to S = (S_f, S_\omega)$ be an ω-semigroup morphism, $w \in \Sigma^\omega$, $H = \{h_0 < h_1 < h_2 < \ldots\} \subseteq \mathbb{N}$ and $s \in S_f$.

We say that H is a g, s-homogeneous factorization of w iff for all $i < j \in \mathbb{N}$ it holds that $g(w[h_i, h_j)) = s$.

Example 3.2.4. Consider again the morphism from Example 3.2.3. For the word $w = abab^2ab^3a\ldots$ the set

$$H = \left\{ \frac{n(n+1)}{2} : n \in \mathbb{N} \right\}$$

is a $g, 0$-homogeneous factorization since $g(w[h_i, h_{i+1})) = g(b^ia) = 0$.

Lemma 3.2.2. For $1 \leq i \leq n$ let $h_i : \Sigma_i^\infty \to S_i$ be some morphism into some finite ω-semigroup S_i and w_i some word over the corresponding alphabet. Then there is a $G = \{g_1 < g_2 < g_3 \ldots\} \subseteq \mathbb{N}$ such that for every i it holds that G is an g_i,e_i-homogeneous factorization of w_i for some idempotent semigroup element $e_i \in S_i$.

Proof. This lemma is more or less a direct consequence of Ramsey's theorem: We color every $\{k, l\} \subset \mathbb{N}$, $k < l$ with the tuple of semigroup elements

$$[h_i(w_i[k, l))]_{1 \leq i \leq n}.$$

With Ramsey's theorem we obtain a $G = \{g_1 < g_2 < g_3 \ldots\}$ such that all $\{g_i, g_j\}$, $i \neq j$ have the same color. Having set $e_i := h_i(w_i[g_1, g_2))$, G obviously is a h_i,e_i-homogeneous factorization of w_i and e_i is idempotent since

$$e_i e_i = h_i(w_i[g_1, g_2))h_i(w_i[g_2, g_3)) = h_i(w_i[g_1, g_3)) = e_i.$$

□

As we see we could now take these take these factorizations and permute or repeat every $w_i[g_l, g_k)$ to create new words that have the same acceptance behavior as the original word. Sometimes this property allows more convenient proofs since using automata usually involves the cutting and merging of their runs.

3.2.4 Regular Relations

In the following we want to interpret relations over infinite words also as languages. We do this in a straight forward way by using the convolution. This means we create from a given tuple of words another word over the Cartesian product of the original alphabets.

Definition 3.2.6. Let $\Sigma_1, \ldots, \Sigma_n$ be some alphabets and $(v_1, \ldots, v_n) \in \Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$. We define the convolution $\langle v_1, \ldots, v_n \rangle \in (\Sigma_1 \times \ldots \times \Sigma_n)^\omega$ as follows:

$$\langle v_1, \ldots, v_n \rangle[i] := (v_1[i], \ldots, v_n[i]),\ i \in \mathbb{N}$$

Further we define for $K \in \Sigma^\omega, L \in \Gamma^\omega$

$$K \otimes L := \{\langle u, v \rangle : u \in K, v \in L\}.$$

and use $L^{\otimes r}$ as an abbreviation for $\underbrace{L \otimes L \otimes \ldots \otimes L}_{r \text{ times}}$. Finally for $R \subseteq (\Sigma^\omega)^n$ we define

$$\otimes R := \{\langle x_1, \ldots, x_n \rangle : (x_1, \ldots, x_n) \in R\}$$

We call a relation $R \subseteq (\Sigma^w)^k$ regular iff its convolution $\otimes R$ is regular. There are a few regular relations that will play a special role in this work.

The ultimately equal relation $\sim_e$ is defined as

$$\sim_e := \{(u, v) \in (\Sigma^\omega)^2 : u[i, \omega) = v[i, \omega) \text{ for some } i \in \mathbb{N}\}.$$

So two words are ultimately equal iff they are equal starting from some position i onwards. Note that $\sim_e$ is an equivalence relation on Σ^ω. In some applications we want to ensure that the words are equal from some fixed position m onwards. Therefore we define a refinement of $\sim_e$ for every $m \in \mathbb{N}$. The relation $\sim_e^m$ is defined as

$$\sim_e^m := \{(u, v) \in (\Sigma^\omega)^2 : u[m, \omega) = v[m, \omega)\}.$$

Both types of relations are regular since they can be described by the regular expressions

$$\{\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) : a, b \in \Sigma\}^* \{\left(\begin{smallmatrix} a \\ a \end{smallmatrix}\right) : a \in \Sigma\}^\omega$$

and

$$\{\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) : a, b \in \Sigma\}^m \{\left(\begin{smallmatrix} a \\ a \end{smallmatrix}\right) : a \in \Sigma\}^\omega.$$

Another important relation on Σ^ω is the lexicographic order $\prec$. To define this order we need to fix an ordering $<$ on Σ. Then $u \prec v$ holds whenever on the first position i where u and v differ we have $u[i] < v[i]$. Also this relation is regular since it can be written as

$$\{\left(\begin{smallmatrix} a \\ a \end{smallmatrix}\right) : a \in \Sigma\}^* \{\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) : a < b\} \{\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) : a, b \in \Sigma\}^\omega .$$

3.3 Borel Languages

Another interesting class of languages consists of those languages that are found in the Borel hierarchy built upon a special topology defined on Γ^ω. This topology is defined by the open sets that are exactly the sets of the form $T\Gamma^\omega$ where T can be any subset of

Γ^*. We quickly verify that this is indeed a topology: the set Γ^ω can be written as $\{\epsilon\}\Gamma^\omega$ and $\emptyset$ can be written as $\emptyset\Gamma^\omega$. Further the open sets are closed under arbitrary unions and they are closed under finite intersections since

$$\begin{aligned} T\Gamma^\omega \cap T'\Gamma^\omega = (&\{v \in T : \exists w \in T'(v <_{\text{pref}} w)\} \\ &\cup \{v \in T' : \exists w \in T(w <_{\text{pref}} v)\})\Gamma^\omega. \end{aligned}$$

We now define $\mathbf{\Sigma}^0_1$ to be the open sets and $\mathbf{\Pi}^0_1$ to be the closed sets. From this starting point we create the Borel hierarchy by iteratively building all countable intersections of $\mathbf{\Sigma}^0_\alpha$ sets and all countable unions of $\mathbf{\Pi}^0_\alpha$ sets. Formally speaking we define for every ordinal $\alpha > 1$:

$$\begin{aligned} \mathbf{\Sigma}^0_\alpha &:= \{\cup_{i\in\mathbb{N}} A_i : A_i \in \mathbf{\Pi}^0_{\alpha_i} \text{ for some } \alpha_i < \alpha\} \\ \mathbf{\Pi}^0_\alpha &:= \{\cap_{i\in\mathbb{N}} A_i : A_i \in \mathbf{\Sigma}^0_{\alpha_i} \text{ for some } \alpha_i < \alpha\} \end{aligned}$$

A language L is called Borel iff $L \in \mathbf{\Sigma}^0_\alpha$ for some ordinal α. There is some kind of correspondence between the lower levels of the Borel hierarchy and the regular languages. It can be shown that every regular language is in the boolean closure of $\mathbf{\Pi}^0_2$.

3.4 Presentations Over Infinite Words

In this section we want to discuss how a structure $\mathfrak{A}$ can be represented by ω-languages. The main idea is to give a name (or possibly several names) to each element of $\mathfrak{A}$. The structure $\mathfrak{A}$ is then represented by the structure $\mathcal{L}$ whose universe is the set of all names given to elements of $\mathfrak{A}$. The relations of $\mathcal{L}$ are a special relation that indicates whether two names represent the same element and the relations that are induced by the relations of $\mathfrak{A}$. In our case these names will be infinite words over some finite alphabet Σ. We will now give a formal definition of what we mean by an ωLang presentation.

Definition 3.4.1. Let $\tau = \{R_1, \ldots, R_n\}$ be some finite relational signature, r_i the arity of R_i and $\mathfrak{A}$ a τ-structure. An ωLang presentation (over the alphabet Σ) of $\mathfrak{A}$ is a $\tau \uplus \{\approx\}$-structure $\mathcal{L} = (L, \approx, R_1, \ldots, R_n)$ together with a surjective labeling function $\pi : L \to A$ such that

- $L \subseteq \Sigma^\omega$,
- $R_i \subseteq (L)^{r_i}$, $1 \leq i \leq n$,
- $\approx = \{(x, y) : f(x) = f(y)\}$ and $\approx$ is a congruence on $R_1, \ldots, R_n$ and
- π is an isomorphism between $\mathcal{L}/_\approx$ and $\mathfrak{A}$.

We say $(\mathcal{L}, \pi)$ is an injective presentation iff π is injective. In this case we will omit $\approx$ in the signature of $\mathcal{L}$.

Depending on the relevance of the concrete labeling function it will go by the board from time to time.

We want to describe the complexity of these presentations from a formal language theoretic point of view. For that reason we also interpret the relations occurring in the presentation as languages. We do this by taking the convolution of the relation.

Definition 3.4.2. Let $\mathcal{L} = (L, \approx, R_1, \ldots, R_n)$ be an (injective) ωLang presentation of some structure $\mathfrak{A}$ and C a class of ω-languages. We say that $\mathcal{L}$ is a $(i)C$-presentation of $\mathfrak{A}$ if the languages $L, \otimes \approx, \otimes R_1, \ldots, \otimes R_n$ are in C. As a convention we will write $L_\approx$ for $\otimes \approx$ and L_{R_i} for $\otimes R_i$

We have to admit that that the name *presentation* may be a bit misleading for the general case of ωLang presentations since not every such presentation can be encoded in a finite way. For algorithmic purposes we are mainly interested in classes of ω-languages C for that all $L \in C$ can be encoded by finite objects. Further we want at least the basic set theoretic operations $\cup, \cap, \neg$ and the projection into an other alphabet to be effectively computable on such encodings. A well known class of languages that fulfills these properties is the class of all automatic languages which we will refer to as ωAut from now on. We call the class of all structures that have an ωAut presentation ωAutStr.

Example 3.4.1. Consider the structure $\mathfrak{A} = (\mathcal{P}(\mathbb{N}), \cap, \cup, \neg)$. We can represent every set $X \in \mathcal{P}(\mathbb{N})$ by the word over the alphabet $\{0, 1\}$ that is labeled with 1 on position n exactly if $n \in X$. The resulting (injective) automatic presentation is then $\mathcal{L} = (L, R_\cap^\mathcal{L}, R_\cup^\mathcal{L}, R_\neg^\mathcal{L}), \pi$ where

- $L = \{0,1\}^\omega$,
- $L_{\cap^\mathcal{L}} = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}^\omega$,
- $L_{\cup^\mathcal{L}} = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}^\omega$,
- $L_{\neg^\mathcal{L}} = \{ \binom{0}{1}, \binom{1}{0} \}^\omega$ and
- $\pi(v) = \{n : v[n] = 1\}$.

Since all relations are regular $\mathcal{L}$ is an automatic presentation of $\mathfrak{A}$.

Example 3.4.2. Let $\mathcal{L}, \pi$ be the presentation of $\mathfrak{A}$ from the previous example. Consider the structure $\mathfrak{A}/\sim_e$ where

$$X \sim_e Y :\Leftrightarrow X - Y \cup Y - X \text{ is finite.}$$

This structure can be presented by $\mathcal{L}_{/\sim_e} := (L, \sim_e, \cap, \cup, \neg)$ with labeling function $\pi'(x) = [\pi(x)]_{\sim_E}$. The induced relations are ω-automatic since

- $\otimes\cap = \{0,1\}^* \otimes \cap^\mathcal{L}$,
- $\otimes\cup = \{0,1\}^* \otimes \cup^\mathcal{L}$ and
- $\otimes\neg = \{0,1\}^* \otimes \neg^\mathcal{L}$.

Observe that this definition of $\sim_e$ on $\mathcal{P}(N)$ coincides with the definition of $\sim_e$ on $\{0,1\}^\omega$ if we identify every $w \in \{0,1\}^\omega$ with the set $\{n : w[n] = 1\}$.

This work focuses on ω-automatic structures. We will see in the next chapter why this class deserves special attention.

Chapter 4

Properties of Automatic Structures

Although ω-automatic presentations are by far not as well studied as automatic presentations, there are a lot of important results that can be more or less directly transferred from finite words to ω-words. There are very important features that are shared by both types of presentations. However there are also some nice properties that hold for automatic structures but for ω-automatic structures. In this chapter we want to present some of the the main results on ω-automatic structures and discuss how other techniques used to investigate finite word automatic presentations could be carried over to infinite words. For a broader overview we suggest [1].

4.1 Definability, Model Checking And Query Evaluation

Having fixed the notion of an ω-automatic presentation, we are now ready to examine its usability concerning the agenda described in the introduction.

The main problems we want to address here are the query evaluation and the model checking problem for automatic structures. The query evaluation problem asks given an automatic presentation $\mathcal{L}$ of a structure $\mathfrak{A}$ and a formula $\varphi(\overline{x})$ for a (e.g. automatic) presentation of $\varphi^{\mathfrak{A}}$. The model checking problem asks, given $\mathcal{L}$ and a sentence φ, whether $\mathfrak{A} \models \varphi$ holds or not.

Here we get to one of the big strengths of automatic structures. For first order logic and some of its extensions, automatic structures behave very well.

Theorem 4.1.1. *There is an effective procedure that, given an automatic presentation*

$\mathcal{L}$ of $\mathfrak{A}$ and a FOC formula $\varphi(\overline{x})$, computes an automaton that recognizes

$$\otimes\pi^{-1}(\varphi^{\mathfrak{A}}) = \{\langle v_1, \ldots, v_k\rangle : \mathfrak{A} \models \varphi(\pi(v_1), \ldots, \pi(v_k))\}.$$

Proof Sketch. Given an ω-automatic presentation $\mathcal{L} = (L, \approx, R_1, \ldots, R_n)$ and a FOC-formula $\varphi(x_1, \ldots, x_k)$ we first construct a FO-formula $\widehat{\varphi}(x_1, \ldots, x_k)$ over the vocabulary $\{L, \approx, R_1, \ldots, R_n, \sim_e\}$ such that for $\mathfrak{E}_{\mathcal{L}} = (\Sigma^\omega, L, \approx, R_1, \ldots, R_n, \sim_e)$ it holds that

$$\mathfrak{E}_{\mathcal{L}} \models \widehat{\varphi}(\overline{a}) \Leftrightarrow \overline{a} \in \pi^{-1}(\varphi^{\mathfrak{A}})$$

for every $\overline{a} \in (\Sigma^\omega)^k$. We first give the construction of $\widehat{\varphi}$ for $\varphi \in$ FO:

$$\begin{aligned}
\widehat{R_i x_{j_1} \ldots x_{j_{r_i}}} &:= R_i x_{j_1} \ldots x_{j_{r_i}} \\
\widehat{x_i = x_j} &:= x_i \approx x_j \\
\widehat{\neg\psi} &:= \neg\widehat{\psi} \\
\widehat{\psi_0 \wedge \psi_1} &:= \widehat{\psi_0} \wedge \widehat{\psi_1} \\
\widehat{\psi_0 \vee \psi_1} &:= \widehat{\psi_0} \vee \widehat{\psi_1} \\
\widehat{\exists x\psi} &:= \exists x(Lx \wedge \widehat{\psi}) \\
\widehat{\forall x\psi} &:= \forall x(Lx \rightarrow \widehat{\psi})
\end{aligned}$$

For the counting quantifiers we need the following lemma which we will state here without proof.

Lemma 4.1.1 ([11])**.** Let $\mathcal{L} = (L, \approx, R_1, \ldots, R_n)$ be an automatic presentation of a structure $\mathfrak{A}$. There is a constant c computable from $\mathcal{L}$ such that for every first order formula φ and every tuple $\overline{v} \in L$ the following statements are equivalent.

1. $\varphi(x, \overline{v})$ is satisfiable and $\varphi(x, \overline{v})^{\mathcal{L}}/_{\approx}$ contains countably many elements.
2. The are words $w_1, \ldots, w_c$ such that each satisfies $\varphi(x, \overline{v})$ and for every word w that satisfies $\varphi(x, \overline{v})$ there is a w' with $w \approx w'$ and $w' \sim_e w_i$ for some $i \in \{1, \ldots, c\}$.

Using this lemma we can replace the quantifiers $\exists^{\leq\aleph_0}$ and $\exists^{>\aleph_0}$ by a finite number of first order quantifiers. We replace formulas of the form $\exists^{\leq\aleph_0} x(\varphi(x))$ by

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

and formulas of the form $\exists^{>\aleph_0} x(\varphi(x))$ by

$$\exists x_1, \ldots, x_{c+1} \forall z(\bigwedge_{1 \leq i \leq c+1} \varphi(x_i) \wedge \bigwedge_{1 \leq i < j \leq c+1} x_i \approx z \rightarrow z \not\sim_e x_j)$$

The construction for the quantifier $\exists^{(k \mod m)}$ was first shown for injective presentations by Kuske and Lohrey in [15] and Bárány, Kaiser and Rubin show, using Lemma 4.1.1, that this construction can be adopted to the general case of ω-automatic presentations.

Using inductive definition of $\widehat{\varphi}$ and the fact the the base relations $L, \approx, \sim_e, R_1, \ldots R_n$ are automatic, it is not hard to see that that $\widehat{\varphi}^{\mathcal{E}_\mathcal{L}}$ is also automatic. The symbols $\wedge, \vee, \neg$ correspond to union, intersection and complementation. The quantifier $\exists$ corresponds to a projection into another alphabet. For example if $\mathcal{A} = (Q, \Sigma^{k+1}, q_0, \Delta, F)$ is a Büchi-automaton than recognizes $\varphi^{\mathcal{E}_\mathcal{L}}$ for a formula $\varphi(x_0, x_1, \ldots, x_n)$ then the automaton $\mathcal{A}' = (Q, \Sigma^k, q_0, \Delta', F)$ with

$$\Delta' = \{(q, (a_1, \ldots, a_n), q') : \exists a_0(q, (a_0, a_1, \ldots, a_n), q') \in \Delta\}$$

recognizes $(\exists x_0 \varphi)^{\mathcal{E}_\mathcal{L}}$.

The only technical complication that needs to be taken into consideration is that it is possible that the variables change their "position" in the corresponding languages. But this is not big obstacle since these operations correspond to projections of the input alphabet which preserve regularity. □

Corollary 4.1.2 (Bárány, Kaiser, Rubin)**.** *The following statements hold true for every automatic structure* $\mathfrak{A}$*.*

1. *Every expansion of* $\mathfrak{A}$ *by FOC definable relations is automatic.*
2. *The FOC theory of* $\mathfrak{A}$ *is decidable.*

The last item immediately yields a method to show that a structure is not automatic: If a structure has an undecidable FOC theory then it is not automatic.

Lemma 4.1.1 has another interesting consequence.

Theorem 4.1.3 (Bárány, Kaiser, Rubin)**.** *Let* $\mathfrak{A}$ *be a countable structure. Then the following statements are equivalent:*

1. $\mathfrak{A} \in AutStr$

2. $\mathfrak{A} \in i\omega AutStr$

3. $\mathfrak{A} \in \omega AutStr$

Proof. If $\mathfrak{A}$ is countable and $\mathcal{L}$ is an ω-automatic presentation of $\mathcal{A}$ then by Lemma 4.1.1 there are $\sim_e$-classes $X_1, \ldots, X_n$ such that every $\approx$-class has a representant in one of these X_i. Using this one can construct an ω-regular injective set of representatives from these X_i and obtain an injective ω-automatic presentation. An automatic presentation can be easily derived from an injective ω-automatic presentation of a countable structure. □

Given an automatic presentation $(\mathcal{L}, \pi)$ of a structure $\mathfrak{A}$, one might ask by what kind of relations R the structure $\mathfrak{A}$ can be expanded such that the presentation of $(\mathfrak{A}, R)$ induced by $(\mathcal{L}, \pi)$ is still automatic. This question motivates the notion of definability that we want to introduce now.

Definition 4.1.4. Let $(\mathcal{L}, \pi)$ be an automatic presentation of a structure $\mathfrak{A}$. We say that a relation $R \subseteq A^k$ is definable in $(\mathcal{L}, \pi)$ iff the relation $\{(w_1, \ldots, w_k) : (\pi(w_1), \ldots, \pi(w_k)) \in R\}$ is automatic.

Additionally we say $a \in A$ is definable in $(\mathcal{L}, \pi)$ iff $\{a\}$ is definable in $(\mathcal{L}, \pi)$.

As seen before every relation that is FOC definable $\mathfrak{A}$ is definable in every ω-automatic presentation $(\mathcal{L}, \pi)$ of $\mathfrak{A}$.

Another question that is closely related to the first one asks by what kind of relations R an automatic structure $\mathfrak{A}$ can be expanded such that $(\mathfrak{A}, R)$ is still automatic. One should observe that these questions ask for different things.

Lemma 4.1.2. There is an automatic structure $\mathfrak{A}$, an automatic presentation $(\mathcal{L}, \pi)$ of $\mathfrak{A}$ and a relation R over the universe of $\mathfrak{A}$ such that R is not definable in $(\mathcal{L}, \pi)$ but $(\mathfrak{A}, R)$ is still automatic.

Proof. Consider the structure $\mathfrak{A} = (\mathbb{N}, s)$ where s is the usual successor function. This structure can be represented by the (in some sense) unary presentation $(\mathcal{L}, \pi)$ where $L = 1^*0^\omega$, $\otimes R_s^{\mathcal{L}} = \binom{1}{1}^* \binom{0}{1} \binom{0}{0}^\omega$ and $\pi(1^n 0^\omega) = n$.

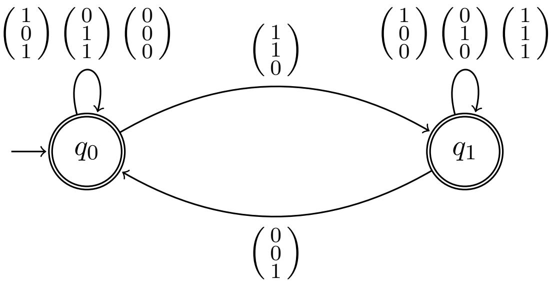

Figure 4.1: The automaton $\mathcal{A}_s$

We claim that the addition $+$ is not definable in $(\mathcal{L}, \pi)$. Suppose $+$ is definable in $(\mathcal{L}, \pi)$. Then there is a Büchi automaton $\mathcal{A} = (Q, \{0,1\}, q_0, \Delta, F)$ that recognizes

$$\begin{aligned} \otimes\pi^{-1}(+) &= \{\langle u, v, w\rangle : \exists n, m \in \mathbb{N}(\pi(u) = n \wedge \pi(v) = m \wedge \pi(w) = n + m)\} \\ &= \{\langle 1^n 0^\omega, 1^m 0^\omega, 1^{n+m} 0^\omega\rangle : n, m \in \mathbb{N}\}. \end{aligned}$$

Now consider the word $v := \langle 1^{|Q|+1}0^\omega, 1^{|Q|+1}0^\omega, 1^{2|Q|+2}0^\omega\rangle \in \otimes\pi^{-1}(+)$. Let ρ be an accepting run of $\mathcal{A}$ on v. We take a look at $\rho[|Q|+1, 2|Q|+2)$ and see that there must be some $|Q|+1 \leq i < j < 2|Q|+2$ with $\rho[i] = \rho[j]$. But then $\rho[0,j)\rho[i,j)\rho[j,\omega)$ is an accepting run of $\mathcal{A}$ on

$$v[0,j)v[i,j)v[j,\omega) = \langle 1^{|Q|+1}0^\omega, 1^{|Q|+1}0^\omega, 1^{2|Q|+2+(j-i)}0^\omega\rangle \notin \otimes\pi^{-1}(+)$$

which contradicts the assumption that $L(\mathcal{A}) = \otimes\pi^{-1}(+)$.

On the other hand $(\mathbb{N}, s, +)$ is automatic since it can be represented in binary by $\mathcal{L}' = (L', R'_s, R'_+), \pi'$ where $L' = \{0,1\}^*0^\omega$ and $\pi'(v) = \sum_{i=0}^{\infty} 2^i v[i]$. Then the induced relations s' and $+'$ are automatic since $\otimes R'_s = L(\mathcal{A}_s) \cap (L')^{\otimes 2}$ and $\otimes R'_+ = L(\mathcal{A}_+) \cap (L')^{\otimes 3}$ for the automata $\mathcal{A}_s, \mathcal{A}_+$ depicted in Figure 4.2 and 4.1.

□

A satisfying answer for this question would give an as strong as possible logic $\mathfrak{L}$ such that ωAutStr is closed under expansion with $\mathfrak{L}$-definable relations.

A nice direct consequence is that then ωAutStr is also closed under $\mathfrak{L}$-interpretations.

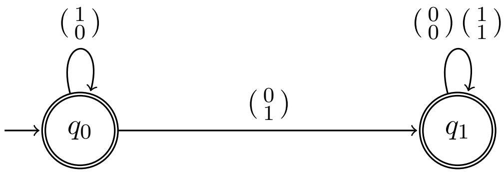

Figure 4.2: The automaton $\mathcal{A}_+$

Lemma 4.1.3. Let $\mathfrak{L}$ be some logic. If ωAutStr is closed under expansion by $\mathfrak{L}$-definable relations then ωAutStr is closed under $\mathfrak{L}$-interpretations.

Proof. Let $\mathcal{I} = (\varphi_\delta, \varphi_\approx, \varphi_{R_1}, \ldots, \varphi_{R_n})$, f be an $\mathfrak{L}$-interpretation of a structure $\mathfrak{B}$ in an automatic structure $\mathfrak{A}$. Since $\mathfrak{A}$ is automatic and ωAutStr is closed under expansion with $\mathfrak{L}$-definable relations there is an automatic presentation

$$\mathcal{L} = (L, \approx, R_1^{\mathcal{L}}, \ldots, R_n^{\mathcal{L}}, \varphi_\delta^{\mathcal{L}}, \varphi_\approx^{\mathcal{L}}, \varphi_{R_1}^{\mathcal{L}}, \ldots, \varphi_{R_n}^{\mathcal{L}})$$

with labeling function g of

$$(\mathfrak{A}, \varphi_\delta^{\mathfrak{A}}, \varphi_\approx^{\mathfrak{A}}, \varphi_{R_1}^{\mathfrak{A}}, \ldots, \varphi_{R_n}^{\mathfrak{A}}).$$

But then

$$\mathcal{L}' := (\varphi_\delta^{\mathcal{L}}, \varphi_\approx^{\mathcal{L}}, \varphi_{R_1}^{\mathcal{L}}, \ldots, \varphi_{R_n}^{\mathcal{L}})$$

together with the labeling function $f \circ g \restriction \varphi_\delta^{\mathcal{L}}$ is an automatic presentation of $\mathfrak{B}$. □

Corollary 4.1.5. *ωAutStr is closed under FOC interpretations.*

There are also bounds for the expressiveness of logics that preserve automaticity. For example it is easy to show that expansion by relations definable in FO expanded even by the simplest form of transitive closure does not preserve automaticity.

Theorem 4.1.6 (Blumensath [2])**.** *ωAutStr is not closed under the expansion with FO(DTC) definable relations.*

Proof. The Presburger arithmetic $(\mathbb{N}, +)$ is automatic and the multiplication is FO(DTC) definable in $(\mathbb{N}, +)$ by the formula

$$\varphi_\cdot(x, y, z) = [DTC_{x,y,z,x',y',z'}\ x = x' \wedge y = y' + 1 \wedge z + x = z'](x, y, 0, x, 0, z).$$

If ωAutStr would be closed under the expansion with FO(DTC) definable relations, then $(\mathbb{N}, +, \cdot)$ would be automatic as well. But the FO theory of $(\mathbb{N}, +, \cdot)$ is undecidable which stands in contradiction to Corollary 4.1.2 item 2. □

There is also a connection between an expansion by a certain relation R preserving automaticity and the definablility of R in a concrete automatic presentation. If $\mathfrak{A}$ is an automatic structure and R a relation over the universe of $\mathfrak{A}$ then $(\mathfrak{A}, R)$ is automatic if and only if R is definable in some automatic presentation of $\mathfrak{A}$.

4.2 Injective Presentations

It can easily be seen that every structure $\mathfrak{A}$ $\in$AutStr has an injective automatic presentation. This is because of the fact that there is an automatic well ordering on Σ^*. This well ordering is the so called length lexicographic order $<_{ll}$. It is defined as follows: $v <_{ll} w$ if and only if $|v| < |w|$ or v and w have the same length and $v \prec w$. Using this order one can, given an automatic presentation $\mathcal{L} = (L, \approx, R_1, \ldots, R_n)$ of $\mathfrak{A}$, pick an automatic set of representatives of the $\approx$-classes. It can be obtained by the first order formula $\varphi(x) = \neg\exists y(y <_{ll} x \wedge y \approx x)$. Since $<_{ll}$ is automatic, the language $L' := \varphi^{(\mathcal{L}, <_{ll})}$ is also automatic. The presentation $\mathcal{L}' = (L', R_1 \cap (L')^{r_1}, \ldots, R_n \cap (L')^{r_n})$ where r_i is the arity of R_i is then an injective automatic presentation of $\mathfrak{A}$.

It has been an open question for quite some time whether the same holds true for ω-automatic structures. First Kuske and Lohrey observed in [15] that it is not possible to create injective automatic presentations by selecting an ω-regular set of representatives of the $\approx$-classes from a given ω-automatic presentation. As they pointed out there is an automatic equivalence relation, namely the $\sim_e$ relation, that has no automatic set of representatives. This gave a first hint that the situation may be different then in the finite word case.

Recently Hjorth, Khoussainov, Montalbán and Nies gave a negative answer to this question in [9]. Actually they were able to show that there are automatic structures that don't even have an injective Borel presentation.

The proof uses results from descriptive set theory. We want to state these results here without proofs. As standard reference we refer to [14].

First we need some definitions.

Definition 4.2.1. A function $f : \Sigma_1^\omega \to \Sigma_2^\omega$ is called Borel iff the set $\{\langle x, f(x)\rangle : x \in \Sigma_1^\omega\}$ is Borel.

We also need the relation $\subseteq^*$ on $\mathcal{P}(\mathbb{N})/_{\sim_e}$.

Definition 4.2.2. For all $X, Y \in \mathcal{P}(\mathbb{N})/_{\sim_e}$ we write $X \subseteq^* Y$ iff $x - y$ is finite for every $x \in X$ and $y \in Y$.

The main key to the theorem is then the following lemma.

Lemma 4.2.1 ([8])**.** There is no Borel function $f : \{0,1\}^\omega \to \Sigma^\omega$ such that for all $x, y \in \{0,1\}^\omega$ it holds that $x \sim_e y$ iff $f(x) = f(y)$.

We will also need this interesting lemma which states that isomorphisms between Borel presentations of $(\mathcal{P}(\mathbb{N}), \subseteq)$ cannot be too complex in the sense that they are guaranteed to be Borel.

Lemma 4.2.2 ([9])**.** Let $\mathcal{L} = (L, \approx^{\mathcal{L}}, \subseteq^{\mathcal{L}})$ and $\mathcal{L}' = (L', \approx^{\mathcal{L}'}, \subseteq^{\mathcal{L}'})$ be Borel presentations of $(\mathcal{P}(\mathbb{N}), \subseteq)$. Then every isomorphism $f : \mathcal{L}/_{\approx^{\mathcal{L}}} \to \mathcal{L}'/_{\approx^{\mathcal{L}'}}$ is Borel.

Theorem 4.2.3 (Hjorth, Khoussainov, Montalbán and Nies)**.** *There is an automatic structure $\mathfrak{A}$ that has no injective Borel presentation.*

Proof. Let $\mathfrak{A}$ be the structure $((\mathcal{P}(\mathbb{N}) \cup \mathcal{P}(\mathbb{N})/_{\sim_e}), \leq, U, G)$ where $x \leq y$ holds iff $x, y \in \mathcal{P}(\mathbb{N})$ and $x \subseteq y$ or $x, y \in \mathcal{P}(\mathbb{N})/_{\sim_e}$ and $x \subseteq^* y$, $U = \mathcal{P}(\mathbb{N})$ and G is the graph of the function $g : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})/_{\sim_e}$ with $g(x) = [x]_{\sim_e}$. We give an automatic presentation $\mathcal{L} = (L, \approx, \leq, U, G), \pi$ of $\mathfrak{A}$. First we set $L := \{0,1\}^\omega \cup \{a,b\}^\omega$ and

$$\pi(v) := \begin{cases} \{n \in \mathbb{N} : v[n+1] = 1\} & \text{if } v \in \{0,1\}^\omega \\ [\{n \in \mathbb{N} : v[n+1] = b\}]_{\sim_e} & \text{if } v \in \{a,b\}^\omega. \end{cases}$$

The elements of $\mathcal{P}(\mathbb{N})$ are coded by the words $\{0,1\}^\omega$ and the elements of $\mathcal{P}(\mathbb{N})/_{\sim_e}$ are coded by the words $\{a,b\}^\omega$. It follows that

$$\begin{aligned}
\approx^{\mathcal{L}} &= \{(v,v) : v \in \{0,1\}^\omega\} \cup \{(v,w) \in \{a,b\}^\omega : v \sim_e w\}, \\
\leq^{\mathcal{L}} &= \{(v,w) \in (\{0,1\}^\omega)^2 : \forall i \in \mathbb{N}(v[i] = 1 \to w[i] = 1)\} \\
&\quad \cup \{(v,w) \in (\{a,b\}^\omega)^2 : \exists i \in \mathbb{N} \forall i' > i(v[i'] = b \to w[i'] = b)\}, \\
U^{\mathcal{L}} &= \{0,1\}^\omega \text{ and} \\
G^{\mathcal{L}} &= \{(v,w) \in \{0,1\}^\omega \times \{a,b\}^\omega : \exists i \forall j > i(v[j] = 1 \leftrightarrow w[j] = b)\}.
\end{aligned}$$

Observe that all these relations are regular.

Now assume that $\mathcal{L}' = (L', \leq', U', G')$ is an injective Borel-presentation of $\mathfrak{A}$. Let $f : \mathcal{L} \to \mathcal{L}'$ be an isomorphism between these presentations and f' the restriction of f to $\{0,1\}^\omega$. Then f' is an isomorphism between two (injective) presentations of $(\mathcal{P}(\mathbb{N}), \subseteq)$ and therefore f' is Borel by Lemma 4.2.2. Then the map $h := g' \circ f'$ is also Borel since Borel functions are closed under concatenation. But for $x, y \in \{0,1\}^\omega$ it holds that

$$\begin{aligned}
x \sim_e y &\Leftrightarrow g^{\mathfrak{A}}(\pi(x)) = g^{\mathfrak{A}}(\pi(y)) \\
&\Leftrightarrow f(\pi^{-1}(g^{\mathfrak{A}}(\pi(x)))) = f(\pi^{-1}(g^{\mathfrak{A}}(\pi(y)))) \\
&\Leftrightarrow g'(f'(x)) = g'(f'(y))
\end{aligned}$$

which contradicts Lemma 4.2.1. □

Corollary 4.2.4. *iωAutStr is a proper subclass of ωAutStr.*

From a technical point of view this result is rather unpleasant for us, because working with an equivalence relation to determine whether two words encode the same element significantly increases the complexity of proofs. From a more theoretical point of view this result is interesting since it shows that we indeed gain expressiveness when we allow to "blow up" the presentation with more than one name for every element. Moreover it motivates the study of injective and non-injective presentations by their own rights.

4.3 Growth Rates

The most successful concept for showing that a structure is not finite word automatic is that of growth rates. Therefore it is natural to ask if this concept can also be used in the infinite word case. But if we want to transfer the notion of growth rates to automatic presentations over infinite words, we first need to find a suitable replacement for the length of a word. One possibility is to restrict to ultimately periodic words w and replace the length by a tuple $\lambda(w) = (n, m)$ where n is the length of the non periodic prefix of w and m is the length of the period. This means n is the smallest number such that there are $u \in \Sigma^n$ and $v \in \Sigma^+$ with $w = uv^\omega$ and m is the smallest number such that there are $u \in \Sigma^n$ and $v \in \Sigma^m$ with $w = uv^\omega$. We denote the first component with $\lambda_p(w)$ and the second with $\lambda_c(w)$. The sum of both values will be denoted by $\lambda_{p+c}(w)$.

Definition 4.3.1. Let $\mathcal{A} = (Q, \Sigma_1 \times \Sigma_2, q_0, \Delta, F)$ be a Büchi-automaton and $w \in \Sigma_1^\omega$ ultimately periodic. The automaton $w\mathcal{A} = (Q', \Sigma_2, q_0', \Delta', F')$ is defined as

$$
\begin{aligned}
Q' &:= Q \times [\lambda_p(w) - 1] \times \{p\} \cup Q \times [\lambda_c(w) - 1] \times \{c\}. \\
q_0' &:= (q_0, 0, p). \\
\Delta' &:= \ \{((q,i,p), a, (q', i+1, p)) : i < \lambda_p(w) - 1 \text{ and } (q,a,q') \in \Delta\} \\
&\quad \cup \{((q, \lambda_p(w) - 1, p), a, (q', 0, c)) : (q,a,q') \in \Delta\} \\
&\quad \cup \{((q,i,c), a, (q', i+1 \mod \lambda_c(w), c)) : (q,a,q') \in \Delta\}. \\
F' &:= F \times [\lambda_c(w) - 1] \times \{c\}.
\end{aligned}
$$

A run of the automaton $w\mathcal{A}$ on a word $v \in \Sigma_2^\omega$ simulates a run of $\mathcal{A}$ on $\langle w, v\rangle$ and therefore it is easy to see that $w\mathcal{A}$ recognizes $\{v \in \Sigma_2^\omega : \langle w, v\rangle \in \mathcal{L}(\mathcal{A})\}$. With this observation we get the following corollary.

Corollary 4.3.2. *If $R \subseteq \Sigma_1^\omega \times \Sigma_2^\omega$ is an automatic relation and $w \in \Sigma_1^\omega$ is ultimately periodic then wR is also automatic.*

We can make a simple observation about definablility of elements in a presentation $(\mathcal{L}, \pi)$.

Lemma 4.3.1. Let $(\mathcal{L}, \pi)$ be an automatic presentation of $\mathfrak{A} = (A, R_1, \ldots, R_n)$. An element $a \in A$ is definable in $(\mathcal{L}, \pi)$ iff $\pi^{-1}(a)$ contains an ultimately periodic word.

Proof. The implication from left to right is trivial since every automatic set contains an ultimately periodic word. In the other direction we have that $\approx$ is automatic and there is an ultimately periodic word in $w \in \pi^{-1}(a)$. Thus by Corollary 4.3.2 the set $\{v : v \approx w\} = \{v : \pi(v) = \pi(w)\} = \pi^{-1}(a)$ is automatic. $\square$

Another observation that will help us to give a tight estimation of the growth rate is that every infinite run in $w\mathcal{A}$ starts with a sequence of exactly $\lambda_p(w)$ states from $Q \times [\lambda_p(w) - 1] \times \{p\}$ followed by an infinite sequence of states from $Q \times [\lambda_c(w) - 1] \times \{c\}$.

We can now investigate the relation between the values $\lambda(w_1), \dots, \lambda(w_k)$ and those λ-values that can be found in $w_1, \dots, w_k R$ for automatic relations R. We will see that the growth rates we face in this situation are significantly larger than what we find in the finite word counterpart.

Lemma 4.3.2. Let $R \subseteq (\Sigma^\omega)^{k+l}$ be an automatic relation. Then there is a constant c such that for all ultimately periodic words $w_1, \dots, w_k \in \Sigma^\omega$ with $\lambda(w_i) = (n_i, m_i)$ and $w_1, \dots, w_k R \neq \emptyset$ there is a tuple $(v_1, \dots, v_l)$ of ultimately periodic words in $w_1, \dots, w_k R$ with

$$\lambda_c(v_i) \leq \underset{1 \leq j \leq k}{lcm}(m_j) \cdot c$$

and

$$\lambda_p(v_i) \leq \max_{1 \leq j \leq k}(n_j) + \underset{1 \leq j \leq k}{lcm}(m_j) \cdot c - \lambda_c(v_i).$$

Proof. Let $\mathcal{A} = (Q, \Sigma^k \times \Sigma^l, q_0, \Delta, F)$ be a Büchi-automaton that recognizes R. Then $w_1, \dots, w_k R$ is recognized by the automaton $\langle w_1, \dots, w_k \rangle \mathcal{A}$. Note that every ultimately periodic successful run ρ on $\langle w_1, \dots, w_k \rangle \mathcal{A}$ induces a word $\langle v_1, \dots, v_l \rangle \in \mathcal{L}(\langle w_1, \dots, w_k \rangle \mathcal{A})$ (or several such words) with $\lambda(\langle v_1, \dots, v_l \rangle) \leq \lambda(\rho)$. Therefore we only need to show that a run ρ with the proposed constrains on $\lambda(\rho)$ exists. Every such run corresponds to a cycle in the graph of $\langle w_1, \dots, w_k \rangle \mathcal{A}$ that contains a final state together with a path from the initial state to this cycle.

Now we need to remember the definition of the state space of $\langle w_1, \dots, w_k \rangle \mathcal{A}$, the fact that $w_1, \dots, w_k R \neq \emptyset$ and the observations about the structure of runs of $\langle w_1, \dots, w_k \rangle \mathcal{A}$. We get that there must be a reachable cycle that contains a final state of length

$$r \leq \lambda_c(\langle w_1, \dots, w_k \rangle) \cdot |Q| = \underset{1 \leq i \leq k}{lcm}(m_i) \cdot |Q|$$

and a path to this cycle of length at most

$$\lambda_p(\langle w_1, \ldots, w_k\rangle) + \lambda_c(\langle w_1, \ldots, w_k\rangle) \cdot |Q| - r =$$
$$\max_{1\leq i\leq k}(n_i) + \underset{1\leq i\leq k}{lcm}(m_i) \cdot c - \lambda_c(\rho).$$

So the word $\langle v_1, \ldots, v_l\rangle$ induced by ρ satisfies also the proposed constraints and thus every v_i does. □

We can reformulate this lemma in terms of λ_{p+c}.

Corollary 4.3.3. *Let $R \subseteq (\Sigma^\omega)^{k+l}$ be an automatic relation. Then there is a constant c such that for all ultimately periodic words $w_1, \ldots, w_k \in \Sigma^\omega$ with $\lambda(w_i) = (n_i, m_i)$ and $w_1, \ldots, w_k R \neq \emptyset$ there is a tuple $(v_1, \ldots, v_l)$ of ultimately periodic words in $w_1, \ldots, w_k R$ with*

$$\lambda_{p+c}(v_i) \leq \max_{1\leq j\leq k}(n_j) + \underset{1\leq j\leq k}{lcm}(m_j) \cdot c = \mathcal{O}(\prod_{1\leq j\leq k} \lambda_{p+c}(w_j)).$$

The following lemma shows that this is the best possible estimation.

Lemma 4.3.3. There is an automatic function $f : L^k \to L$ such that for every n there are ultimately periodic words $w_1, \ldots, w_k \in L$ with $\lambda_{p+c}(w_i) \geq n$ and

$$\lambda_{p+c}(f(\overline{w})) = \prod_{1\leq i\leq k} \lambda_{p+c}(w_i).$$

Proof. Consider the function $f : (\{0,1\}^\omega)^k \to \{0,1\}^\omega$ defined by

$$f(w_1, \ldots, w_k)[i] := \begin{cases} 1 & , \text{ if } w_1[i] = \ldots = w_k[i] = 1 \\ 0 & , \text{ else} \end{cases}.$$

The function is obviously automatic. Now for $n \in \mathbb{N}$ we choose the words $w_1, \ldots, w_k$ as $w_i := (0^{p_i-1}1)^\omega$ where p_i is the i-th prime number that is greater than n. Then $f(\overline{w}) = (0^{(\prod_{1\leq i\leq k} p_i)-1}1)^\omega$ and with that

$$\lambda_{p+c}(f(\overline{w})) = \prod_{1\leq i\leq k} p_i = \prod_{1\leq i\leq k} \lambda_{p+c}(f(w_i)).$$

□

We also want to talk about the coding length of elements of the original structure $\mathfrak{A}$.

Definition 4.3.4. Let $(\mathcal{L}, \pi)$ be an automatic presentation of a structure $\mathfrak{A}$. For elements $\overline{a} = (a_1, \ldots, a_n)$ of $\mathfrak{A}$ that are definable in $(\mathcal{L}, \pi)$ we set

$$\lambda^{\pi}_{p+c}(\overline{a}) := \min\{\lambda_{p+c}(\langle v_1, \ldots, v_n\rangle) : v_i \in L \text{ is ultimately periodic and } \pi(v_i) = a_i\}.$$

We could also define λ^{π}_p and λ^{π}_c in the same way, but these definitions require some care, because $\lambda^{\pi}_p(a) = n$ and $\lambda^{\pi}_c(a) = m$ does not imply that there is a $v \in \pi^{-1}(a)$ with $\lambda_p(v) = n$ and $\lambda_c(v) = m$. Therefore we will only work with λ^{π}_{p+c}.

We quickly translate Corollary 4.3.3 to λ^{π}_{p+c}.

Corollary 4.3.5. *Let $(\mathcal{L}, \pi)$ be an automatic presentation of a structure $\mathfrak{A}$ and $R \subseteq A^{k+l}$ a relation of $\mathfrak{A}$. For every tuple $\overline{a} \in A^k$ of elements definable in $(\mathcal{L}, \pi)$ with $\overline{a}R \neq \emptyset$ there is a tuple $\overline{b} \in \overline{a}R$ with*

$$\lambda^{\pi}_{p+c}(\overline{b_i}) = \mathcal{O}\left(\prod_{1 \leq i \leq k} \lambda_{p+c}(a_i)\right).$$

We will apply these results in the following chapters. As it might already guess, we will see that in most cases these growth rates are too large to translate techniques used in proofs for *automatic* structures to ω-automatic structures.

Chapter 5

Automatic Pairing Functions

A pairing function on a set A is an injection from $A \times A$ to A. Pairing functions play an important role in many fields of computer science such as recursion theory and cryptography. Therefore it would be nice to have some automatic pairing functions. Unfortunately the situation does not look very promising.

5.1 Finite Word Automatic Pairing Functions

In the case of finite word automatic structures, pairing functions have been studied by Blumensath in [2]. Using the fact that in the finite word case every automatic structure has an injective presentation and that the output of an automatic function can only be larger than the input by a constant additive term he was able to show that no pairing function is automatic. We want to prove this theorem here since it is a good example for the techniques that can be used to show that a structure is not finite word automatic.

First we take a look at growth rates.

Lemma 5.1.1. Let $f : L^k \to L \subseteq \Sigma^*$ be a regular function. Then there is a constant c such that for every $w_1, \ldots, w_k \in L$ it holds that $|f(\overline{w})| \leq \max\{|w_1|, \ldots, |w_k|\} + c$.

Proof. Let $\mathcal{A} = (Q, \Sigma^{k+1}, q_0, \Delta, F)$ be an automaton that recognizes the graph of f. We claim that the proposition holds for $c = |Q|$. Suppose there are $w_1, \ldots, w_k \in L$ with $\max\{|w_1|, \ldots, |w_k|\} = m$ and $|f(\overline{w})| > m + c$. Let ρ be a successful run of $\mathcal{A}$ on $v = \langle \overline{w}, f(\overline{w})\rangle$. Since $|f(\overline{w})| - m > c = |Q|$ there must be some $m \leq i < j < |f(\overline{w})|$ with $\rho[i] = \rho[j]$. Then

$$\rho[0, j)\rho[i, j)\rho[j, |f(\overline{w})|)$$

is an accepting run of $\mathcal{A}$ on

$$v[0,j)v[i,j)v[j,|f(\overline{w})|) = \langle w_1,\dots,w_k, f(\overline{w})[0,j)f(\overline{w})[i,j)f(\overline{w})[j,|f(\overline{w})|)\rangle.$$

But since

$$f(\overline{w}) \neq f(\overline{w})[0,j)f(\overline{w})[i,j)f(\overline{w})[j,|f(\overline{w})|)$$

this contradicts the assumption that $\mathcal{A}$ recognizes the graph of f. □

We can use this lemma to obtain a bound for the size of the so called generations. We want to introduce these generations in a simple form.

Definition 5.1.1. For a function $f : A^k \to A$ and a set $E \subseteq A$ the generations of E with respect to f are defined by

$$G_0^f(E) = E,$$
$$G_{n+1}^f(E) = G_n(E) \cup \{f(\overline{x}) : \overline{x} \in G_n(E)^k\}.$$

We can see that the length of words found in the n-th generations of a finite set are bounded linearly in n.

Lemma 5.1.2. Let $f : L^k \to L \subseteq \Sigma^*$ be a regular function and $E \subseteq L$ a finite set with $\max\{|w| : w \in E\} = m$. Then there is a constant c such that for every n it holds that $\max\{|w| : w \in G_n^f(E)\} \leq m + nc$.

Proof. Let $\mathcal{A} = (Q, \Sigma^{k+1}, q_0, \Sigma, F)$ be a f.w. automaton that recognizes the graph of f. We set $c := |Q|$ and proof $\max\{|w| : w \in G_n^f(E)\} \leq m + nc$ by induction.

($n = 0$). $\max\{|w| : w \in G_0^f(E)\} = \max\{|w| : w \in E\} = m$ holds by assumption.

($n > 0$). Let $w_1,\dots,w_k$ be some elements of $G_{n-1}^f(E)$. By assumption it holds that $\max\{|w_1|,\dots,|w_k|\} \leq m + c(n-1)$ and therefore we get with Lemma 5.1.1 that $|f(w_1,\dots,w_k)| \leq m + c(n-1) + c = m + cn$. □

With this lemma we directly obtain a bound for the size of $G_n^f(E)$.

Corollary 5.1.2. *Let $f : L^k \to L \subseteq \Sigma^*$ be a regular function and $E \subseteq L$ a finite set with $\max\{|w| : w \in E\} = m$. Then for every n it holds that $|G_n^f(E)| \leq |\Sigma|^{m+cn+1}$.*

Theorem 5.1.3 (Blumensath)**.** *There is no structure $\mathfrak{A}$ $\in$AutStr in which a pairing function f is FOC definable.*

Proof. Suppose there is an automatic structure such that a pairing function f is definable in $\mathfrak{A}$. Then $(\mathfrak{A}, f)$ is also automatic and has an injective automatic presentation (L, f'). In this presentation f' is a regular function (or more precisely the graph of a regular function). From f' we define for every $n \in \mathbb{N}$ a function $g_n : L^{2^n} \to L$ as follows:

$$\begin{aligned} g_0(x) &= x \\ g_{n+1}(x_1, \ldots, x_{2^{n+1}}) &= f'(g_n(x_1, \ldots, x_{2^n}), g_n(x_{2^n+1}, \ldots, x_{2^{n+1}})) \end{aligned}$$

This means g_n is a nested application of f' of depth n. Observe that since f' is injective, all the g_n must also be injective. Now let a, b be distinct elements of L then we have $|g_n(\{a,b\}^{2^n})| = 2^{2^n}$. But $g_n(\{a,b\}^{2^n}) \subseteq G_n(\{a,b\})$ and $|G_n(\{a,b\})| \leq |\Sigma|^{\max\{|a|,|b|\}+cn+1}$ for some constant c. We obtain a contradiction by observing that there is a n' with $2^{2^{n'}} > |\Sigma|^{\max\{|a|,|b|\}+cn+1}$. ∎

Due to the equivalence of automatic structures and ω-automatic structures over countable domains we derive directly that there are no countable ω-automatic structures in which a pairing function can be defined.

5.2 Injective Presentations of Pairing Functions

At first glance it seems tempting to transfer the chain of arguments seen above to the infinite word case. The arguments deal with the length words and how this length is related to the length of their pictures under an automatic function. But with the concepts related to growth rates that we have developed this will fail since the growth rates can be so much larger than in the finite word case.

Lemma 5.2.1. Let $f : L^k \to L$ be an automatic function and $E \subseteq L$ be a finite set of ultimately periodic words with $\max\{\lambda_{p+c}(e) : e \in E\} = m$. Then there is a constant c such that $\max\{\lambda_{p+c}(e) : e \in G_n^f(E)\} \leq p(m, c, k, n)$. Here $p(m, c, k, n)$ is the function defined by:

$$p(m, c, k, 0) = m$$
$$p(m, c, k, n + 1) = cp(m, c, k, n)^k$$

Proof. We set c to be the constant from corollary 4.3.3 and proof the claim by induction over n.

($n = 0$). Here it holds that

$$\max\{\lambda_{p+c}(e) : e \in G_0^f(E)\} = \max\{\lambda_{p+c}(e) : e \in E\} = m = p(m, c, k, 0).$$

($n > 0$). Let $w_1, \ldots, w_k$ be elements from $G_{n-1}^f(E)$. By the induction hypothesis it holds that

$$\lambda_{p+c}(w_i) \le p(m, c, k, n - 1).$$

Then by corollary 4.3.3

$$\begin{aligned}\lambda_{p+c}(f(\overline{w})) &\le \max_{1\le i\le k}(\lambda_p(w_i)) + c \operatorname*{lcm}_{1<i<n}(\lambda_c(w_i)) \\ &\le c \prod_{1\le i\le k} \lambda_{p+c}(w_i) \\ &\le c \prod_{1\le i\le k} p(m, c, k, n-1) \\ &= cp(m, c, k, n-1)^k \\ &= p(m, c, k, n).\end{aligned}$$

□

This estimation is useless for obtaining an upper bound for the size of $G_n^f(E)$ since it is already naturally bounded by the function $g_n^f(E)$ defined by

$$g_0^f(E) = |E|,$$
$$g_{n+1}^f(E) = g_n^f(E)^k + g_n^f(E).$$

We will therefore present another technical property that can at least be used for injective presentations. This property deals with words, that are all equal from some position m onwards.

First we introduce a notation to keep track of the $\sim_e^m$-classes that are adjacent to a given tuple in a relation.

Definition 5.2.1. Let $R \subseteq (\Sigma^\omega)^{k+l}$ be a relation. We define the m-signature of $\overline{x} \in (\Sigma^\omega)^k$ under R as

$$sig_R^m(\overline{x}) := \{X \in (\Sigma^\omega)^{\otimes l}/\sim_e^m : \otimes(\overline{x}R) \cap X \neq \emptyset\}.$$

The following lemma basically states that for every regular relation the number of different m-signatures in $sig_R^m(X^k)$ for $\sim_e^m$-classes X is bounded by a constant.

Lemma 5.2.2. Let $R \subseteq (\Sigma^\omega)^{k+l}$ be a regular relation recognized by $\mathcal{A} = (Q, q_0, \Sigma^{k+l}, \Delta, F)$. Then $|sig_R^m(X^k)| \leq 2^{|Q|}$ for every $X \in \Sigma^\omega/\sim_e^m$.

Proof. Let $X \in \Sigma^\omega/\sim_e^m$ be a $\sim_e^m$-class. For $\overline{x} \in X^k$ let $\kappa_m(\overline{x})$ be the set of all $q \in Q$ such that there is a $\overline{y} \in (\Sigma^\omega)^l$ and an accepting run $\rho \in Q^\omega$ on $\langle \overline{x}, \overline{y} \rangle$ with $\rho[m] = q$.

We claim that if $\kappa_m(\overline{x}) = \kappa_m(\overline{y})$ then also $sig_R^m(\overline{x}) = sig_R^m(\overline{y})$. It suffices to show that if $\kappa_m(\overline{x}) = \kappa_m(\overline{y})$ then $sig_R^m(\overline{x}) \subseteq sig_R^m(\overline{y})$.

Let $r_{\overline{x}}$ be some element of $\overline{x}R$ and ρ and accepting run of $\mathcal{A}$ on $\langle \overline{x}, r_{\overline{x}} \rangle$. Since $\kappa_m(\overline{x}) = \kappa_m(\overline{y})$ there is an $r_{\overline{y}}$ such that there is an accepting run ρ' of $\mathcal{A}$ on $\langle \overline{y}, r_{\overline{y}} \rangle$ with $\rho[m] = \rho'[m]$. Then obviously $\rho'[0,m)\rho[m,\omega)$ is an accepting run of $\mathcal{A}$ on

$$\begin{aligned} &\langle \overline{y}, r_{\overline{y}} \rangle [0,m) \langle \overline{x}, r_{\overline{x}} \rangle [m,\omega) = \\ &\langle \overline{y}, r_{\overline{y}} \rangle [0,m) \langle \overline{y}, r_{\overline{x}} \rangle [m,\omega) = \\ &\quad \langle \overline{y}, r_{\overline{y}}[0,m) r_{\overline{x}}[m,\omega) \rangle. \end{aligned}$$

Here the first equation holds since $\langle \overline{x} \rangle \sim_e^m \langle \overline{y} \rangle$. But this means there is a word that is $\sim_e^m$-equivalent to $r_{\overline{x}}$ in $\overline{y}R$. □

We want to present another version of this lemma. This time we use ω-semigroups instead of automata. We see that automata theoretic arguments can be often translated quite naturally into the language of ω-semigroups.

Lemma 5.2.3. Let $R \subseteq (\Sigma^\omega)^{k+l}$ be a regular relation recognized by $S = (S_f, S_\omega)$ via the morphism $h : (\Sigma^{k+l})^\infty \to S$. Then $|sig_R^m(X^k)| \leq 2^{|S_f|}$ for every $X \in \Sigma^\omega/\sim_e^m$.

Proof. Let $X \in \Sigma^\omega / \sim_e^m$ be some $\sim_e^m$-class. For $\overline{x} \in X^k$ let $\iota_m(\overline{x})$ be the set of all $s \in S_f$ such that there is a $\overline{y} \in (\Sigma^\omega)^l$ with $(\overline{x}, \overline{y}) \in R$ and $h(\langle \overline{x}, \overline{y} \rangle [0,m)) = s$. Analogous to the proof of Lemma 5.2.2 claim that if $\iota_m(\overline{x}) = \iota_m(\overline{y})$ then also $sig_R^m(\overline{x}) = sig_R^m(\overline{y})$. Again it suffices to show that if $\iota_m(\overline{x}) = \iota_m(\overline{y})$ then $sig_R^m(\overline{x}) \subseteq sig_R^m(\overline{y})$.

Let $r_{\overline{x}}$ be some element of $\overline{x}R$. Since $\iota_m(\overline{x}) = \iota_m(\overline{y})$ there is an $r_{\overline{y}} \in \overline{y}R$ with $h(\langle \overline{x}, r_{\overline{x}} \rangle [0,m)) = h(\langle \overline{y}, r_{\overline{y}} \rangle [0,m))$. But then also

$$\begin{aligned} h(\langle \overline{x}, r_{\overline{x}} \rangle) &= h(\langle \overline{x}, r_{\overline{x}} \rangle [0,m)) * h(\langle \overline{x}, r_{\overline{x}} \rangle [m,\omega)) \\ &= h(\langle \overline{y}, r_{\overline{y}} \rangle [0,m)) * h(\langle \overline{x}, r_{\overline{x}} \rangle [m,\omega)) \\ &= h(\langle \overline{y}, r_{\overline{y}} \rangle [0,m) \langle \overline{x}, r_{\overline{x}} \rangle [m,\omega)) \\ &= h(\langle \overline{y}, r_{\overline{y}}[0,m) r_{\overline{x}}[m,\omega) \rangle). \end{aligned}$$

Here the last equation holds since $\langle \overline{x} \rangle \sim_e^m \langle \overline{y} \rangle$. But this means there is a word that is $\sim_e^m$-equivalent to $r_{\overline{x}}$ in $\overline{y}R$. □

For automatic functions we can give an even lower bound for the number of different m-signatures relative to the size of the automaton recognizing the function.

Lemma 5.2.4. Let $L \subseteq \Sigma^\omega$ and $f : L^k \to L$ be a regular function recognized by a Büchi-automaton $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$. Then $|sig_R^m(X^k)| \leq |Q|$ for every $X \in L/ \sim_e^m$.

Proof. Let X be any class from $L/ \sim_e^m$ With the same arguments as in the proof of the previous lemma it is easy to show that if for $\overline{x}, \overline{y} \in X^k$ there are accepting runs ρ, ρ' of $\mathcal{A}$ on $\langle \overline{x}, f(\overline{x}) \rangle$ and $\langle \overline{y}, f(\overline{y}) \rangle$ respectively, with $\rho[m] = \rho'[m]$ then $f(\overline{x}) \sim_e^m f(\overline{y})$. It follows that at most $|Q|$ $\sim_e^m$-classes can be found in $f(X^k)$. □

Observe that the constants in the above lemmata do not depend on the number m for which the $\sim_e^m$-classes are considered. As the number of elements that can be contained in one $\sim_e^m$-class increases with m we see that these properties are indeed a limitation for the complexity of regular relations.

Lemma 5.2.5. Let L be an infinite ω-regular language. Then there is an infinite $\sim_e$-class $X \in L/ \sim_e$.

Proof. We consider two cases.

Suppose L is countably infinite. Then

$$L = \bigcup_{1 \leq i \leq n} T_i v_i^\omega$$

for some regular $T_i \subseteq \Sigma^*$ and $v_i \in \Sigma^+$. Since L is infinite there is an i such that $T_i v_i^\omega$ is also infinite. Observe that $|T_i v_i^\omega / \sim_e| \leq |v_i|$ and therefore there is an infinite $X \in T_i v_i^\omega / \sim_e$.

Now consider the case that L is uncountable. Then there is a finite ω-semigroup S and a morphism $g : \Sigma^\infty \to S$ such that L is recognized by S via g. Since L is uncountable $L/ \sim_e$ is infinite. Now for $c := |S_f|$ choose $w_0, \ldots, w_c \in L$ such that $w_i \not\sim_e w_j$ for all $i \neq j$. By Lemma 3.2.2 there is a $H = \{h_0 < h_1 < \ldots\} \subseteq \mathbb{N}$ such that for $0 \leq i \leq n$, H is an g, a_i-homogeneous factorization of w_i for some $a_i \in S_f$.

Because of the fact that $|\{w_0, \ldots, w_c\}| > |S_f|$, we see that there must be some $i \neq j$ such that $a_i = a_j$. Now choose l such that $v_0 := w_i[h_0, h_l) \neq w_j[h_0, h_l) =: v_1$ which is possible since $w_i \not\sim_e w_j$. We will now construct an infinite $L' \subseteq L$ with $v \sim_e v'$ for all $v, v' \in L'$. Consider the language

$$L' := w_i[0, h_0)\{v_0, v_1\}^* v_0^\omega.$$

Since $|v_0| = |v_1|$ all words in L are pairwise $\sim_e$-equivalent. We now only need to show $L' \subseteq L$. By construction $g(v_0) = a_i = a_j = g(v_1)$ and therefore it holds for every $\alpha = \alpha_0 \ldots \alpha_{n-1} \in \{0,1\}^*, v_\alpha = v_{\alpha_1} \ldots v_{\alpha_{n-1}}$ that

$$\begin{aligned} g(w_i[0, h_0) v_\alpha v_0^\omega) &= g(w_i[0, h_0)) a_i^n a_i^\omega \\ &= g(w_i[0, h_0)) a_i^\omega \\ &= g(w_i[0, h_0) w_i[h_0, \omega)) \\ &= g(w_i) \end{aligned}$$

and this means $w_i[0, h_0) v_\alpha v_0^\omega \in L$. □

Lemma 5.2.6. Let L be an infinite ω-regular language. Then for every n one can choose an m such that there is a $Y \in L/ \sim_e^m$ which contains at least n elements.

Proof. Since L is infinite we can choose by Lemma 5.2.5 for every n distinct, pairwise $\sim_e^m$-equivalent words $w_1, \ldots, w_n \in L$ and set

$$m := \max\{z : w_i[z] \neq w_j[z] \text{ for some } 1 \leq i, j \leq n\}.$$

□

We now have everything together to show that there are no injective automatic presentations of pairing functions. We do so by proving a more general fact about $i\omega$Aut functions that deals with the minimal number of different elements that can be constructed by applying this function to elements from a subset of size n.

Definition 5.2.2. For every function $f : A^k \to A$ over an infinite set A we define the minimal image size $MIS_f : \mathbb{N} \to \mathbb{N}$:

$$MIS_f(n) = \min\{|f(X^k)| : |X| = n\}$$

We can now use our observations to show that the minimal image size can only grow linearly in n. The main idea is that if MIS_f grows too fast, we can ensure that for every $\sim_e^m$-class X which contains a certain number of elements $f(X^k)$ hits a $\sim_e^m$-class Y that contains more elements than X itself. This fact is then brought easily to a contradiction.

Lemma 5.2.7. For every injective-automatic structure (A, f) over an infinite universe it holds that $MIS_f(n) = \mathcal{O}(n)$.

Proof. Suppose there is an injective automatic presentation $\mathcal{L} = (L, f)$ of an infinite structure $(A, f^{\mathfrak{A}})$ with $f^{\mathfrak{A}} : A^k \to A$ and $MIS_{f^{\mathfrak{A}}}$ grows superlinear. It follows that also MIS_f growth superlinear. Let $\mathcal{A} = (Q, \Sigma^{k+1}, q_0, \Delta, F)$ be a Büchi automaton that recognizes f.

Now choose n such that $MIS_f(n') > |Q| \cdot n'$ for all $n' \geq n$. This is possible since MIS_f grows superlinear. By Lemma 5.2.6 we can choose m such that $L/\sim_e^m$ contains a class of size at least n. Let $Y \in L/\sim_e^m$ be such that for all $Z \in L/\sim_e^m$ it holds that $|Y| \geq |Z|$.

By Lemma 5.2.4, $f(Y^k)$ contains words from at most $|Q|$ different $\sim_e^m$-classes. Since $|f(Y^k)| > |Q| \cdot |Y|$, there are $\overline{x}_1, \ldots, \overline{x}_{(|Q| \cdot |Y|+1)} \in Y^k$ with $f(\overline{x}_i) \neq f(\overline{x}_j)$ for $i \neq j$. From these $\overline{x}_i$ at least

$$\lceil \frac{|Q| \cdot |Y| + 1}{|Q|} \rceil > |Y|$$

are mapped into the same $\sim_e^m$-class. But this means that there is a $\sim_e^m$-class that contains more elements than Y which contradicts the choice of Y. □

Corollary 5.2.1. There is no structure $\mathfrak{A} \in i\omega\text{AutStr}$ such that a pairing function is FOC-definable in $\mathfrak{A}$. This holds since for any pairing function f we have $MIS_f(n) = n^2$.

From this we obtain immediately our first examples of structures that have decidable theories but are not injective-automatic.

Example 5.2.1. The Structure $\mathfrak{A} = (\mathbb{N}, c)$, where c is the Cantor pairing function $c(x, y) = \frac{(x+y)(x+y+1)}{2} + y$, has a decidable theory [3] (even if one adds the usual successor function s) but from Corollary 5.2.1 we know $\mathfrak{A}$ is not injective-automatic.

Example 5.2.2. The structure $\mathfrak{A} = (\mathbb{N}, +, 2^x)$ has a decidable theory [17]. It is not injective-automatic since the function $f(x, y) = 2^x + 2^{y+1}$ is a pairing function that is obviously FO-definable in $\mathfrak{A}$.

In similar ways a pairing function can be defined on any structure $(\mathbb{N}, +, f(x))$ where for all x it holds that $f(x + 1) \geq 2f(x)$ [17], and therefore none of these structures is injective-automatic.

We should mention here that these results are also be derivable by the facts that pairing functions are not automatic [2] and every countable structure is automatic iff it is finite word automatic [11]. Nevertheless Corollary 5.2.1 holds also for uncountable structures.

Chapter 6

Substructures of Automatic Structures

6.1 Presentations of Substructures

In mathematics it is always an interesting question to ask what kind of substructures can be found in a given structure $\mathfrak{A}$. When considering structures given by some ωLANG-presentation $\mathcal{L}$, not only the existence of a substructure becomes interesting, but also the complexity of its presentations inside $\mathcal{L}$.

Definition 6.1.1. Let C_1,C_2 be classes of ω-languages, $\mathfrak{A}$,$\mathfrak{B}$ τ-structures and $\mathcal{L} = (L_A, \approx, (R)_{R\in\tau})$ a C_1-presentation of $\mathfrak{A}$. We say $\mathcal{L}$ contains a C_2-presentation of $\mathfrak{B}$ iff there is a $L_B \subseteq L_A$ such that $(L_B, \approx \cap L_B^2, (R \cap L_B^r)_{R\in\tau})$ is a C_2-presentation of $\mathfrak{B}$.

Observe that if a C_1-presentation of a structure $\mathfrak{A}$ contains a C_2-presentation of a structure $\mathfrak{B}$, then $\mathfrak{A}$ contains a substructure that is isomorphic to $\mathfrak{B}$. On the other side, if $\mathfrak{A}$ contains a substructure that is isomorphic to $\mathfrak{B}$ then every ωLANG-presentation of $\mathfrak{A}$ contains a ωLANG-presentation of $\mathfrak{B}$.

In [14] Kuske has examined the complexity of homogeneous sets in automatic (k,l)-partitions. He uses the notation

$$(\kappa, C_1) \to (\lambda, C_2)^l_k$$

where κ, λ are cardinals and C_1,C_2 are classes of ω-languages to denote that every C_1-presentation of a (k,l)-partition of size κ contains a C_2-presentation of a homogeneous set. We want to transfer this notation into a more general context.

Definition 6.1.2. Let C_1,C_2 be classes of ω-languages and $\mathcal{K}_1$,$\mathcal{K}_2$ classes of τ-structures. We write

$$(\mathcal{K}_1, (i)C_1) \Rightarrow (\mathcal{K}_2, [i]C_2)$$

to denote that every (injective) C_1-presentation of a structure $\mathfrak{A} \in \mathcal{K}_1$ contains an [injective] C_2-presentation of a structure $\mathfrak{B} \in \mathcal{K}_2$.

In the case that $\mathcal{K}_1$ or $\mathcal{K}_2$ contains only one structure $\mathfrak{A}$ we will simply write $\mathfrak{A}$ instead of $\{\mathfrak{A}\}$ in the above expression. For example we write $(\mathcal{K}_1, C_1) \Rightarrow (\mathfrak{A}, C_2)$ instead of $(\mathcal{K}_1, C_1) \Rightarrow (\{\mathfrak{A}\}, C_2)$.

Example 6.1.1. As seen in section 4.2 there are automatic structures $\mathfrak{A}$ that do not have an injective presentation. For every such $\mathfrak{A}$ it obviously holds that

$$(\mathfrak{A}, \omega\mathrm{Aut}) \not\Rightarrow (\mathfrak{A}, i\omega\mathrm{Aut}).$$

6.2 Elementary Substructures

By the famous Löwenheim-Skolem Theorem we know that every infinite structure (over a countable signature) has a countably infinite elementary substructure. It should not come as a surprise that we want to augment this fact in terms of the concepts developed in the previous section.

One could hope that every ω-automatic presentation of a structure contains an ω-automatic sub-presentation of an elementary substructure. If such a sub-presentation could be effectively constructed we could, given an ω-automatic presentation, construct the presentation of the countable elementary substructure and then construct from this presentation a finite word automatic presentation.

If the resulting presentation is not dramatically larger than the original one, this would have the advantage that model checking and query evaluation problem for finite word automatic presentations are of lower complexity than for automatic presentations over infinite words. Sadly this is not the case: as mentioned by Kaiser, Rubin and Bárány in [11, section 4.1] there are theories with automatic models but without countable automatic models.

For a structure $\mathfrak{A}$ we write $[\mathfrak{A}]^{\aleph_0}_{\equiv_e}$ for the class of all countably infinite structures $\mathfrak{B}$ with $\mathfrak{A} \equiv_e \mathfrak{B}$.

Theorem 6.2.1 (Bárány, Kaiser and Rubin). *There are ω-automatic structures $\mathfrak{A}$ with*

$$(\mathfrak{A}, \omega Aut) \not\Rightarrow ([\mathfrak{A}]^{\aleph_0}_{\equiv_e}, \omega Aut)$$

Proof. Consider the structure $\mathfrak{A} = (\mathcal{P}(\mathbb{N}), \cap, \cup, \neg)/\sim_e$. As seen in Example 3.4.2 $\mathfrak{A}$ is automatic. Since $\mathfrak{A}$ is an atomless Boolean Algebra any elementary substructure must also be an atomless Boolean Algebra. But as it was shown by Khoussainov, Nies, Rubin and Stephan in [13] the countable atomless Boolean Algebra is not finite word automatic and therefore not ω-automatic. Thus no automatic presentation of $\mathfrak{A}$ can contain an automatic sub-presentation of an structure in $[\mathfrak{A}]^{\aleph_0}_{\equiv_e}$. □

The best known bound for the complexity of a sub-presentation of an elementary substructure known so far was established independently by Bárány, Kaiser and Rubin in [11] and Hjorth, Khoussainov, Montalbán and Nies in [9].

For a language L we write L_{up} for the set of all ultimately periodic words in L and for a class of languages C we write C_{up} for the class $\{L_{up} : L \in C\}$.

Theorem 6.2.2. *For every automatic structure $\mathfrak{A}$ it holds that*

$$(\mathfrak{A}, \omega Aut) \Rightarrow ([\mathfrak{A}]^{\aleph_0}_{\equiv_e}, \omega Aut_{up}).$$

Proof. Let $\mathcal{L} = (L, \approx, R_1, \ldots, R_n)$ be an ωAut-presentation of $\mathfrak{A}$. We claim that the sub-presentation $\mathcal{L}_{up}$ induced by L_{up} is an ωAut_{up}-presentation of an elementary substructure. By the Tarski-Vaught criterion for elementary substructures it suffices to show that for every first order formula $\varphi(x, \overline{y})$ and all tuples of ultimately periodic words $\overline{v}$ it holds that

$$\mathcal{L} \models \exists x(\varphi(x, \overline{v})) \Rightarrow \mathcal{L}_{up} \models \exists x(\varphi(x, \overline{v})).$$

Suppose $\mathcal{L} \models \exists x(\varphi(x, \overline{v}))$ holds. By Corollary 4.1.2 the relation $\varphi(x, \overline{y})^{\mathcal{L}}$ is regular. Since the words $\overline{v}$ are ultimately periodic the set $\{w : \mathcal{L} \models \varphi(w, \overline{v})\}$ is also regular and therefore contains an ultimately periodic word. But this means that $\mathcal{L}_{up} \models \exists x(\varphi(x, \overline{v}))$. □

6.3 Uncountable Linear Orders

In this section we want to discuss how we can use sub-presentations to prove the non-automaticity of a structure.

One possibility to show that a class of structures $\mathcal{K}_1$ does not contain any automatic structures would be to show that $(\mathcal{K}_1, \omega\text{Aut}) \Rightarrow (\mathcal{K}_2, C)$ for a class $\mathcal{K}_2$ that does not contain any structure with C-presentation. One could call this a reduction like approach.

A slightly different approach is to show that $(\mathcal{K}_1, \omega\text{Aut}) \Rightarrow (\mathcal{K}_2, C)$ holds and then conclude that no structure from $\mathcal{K}_1$, which does not contain any structure of $\mathcal{K}_2$ as substructure, can be automatic.

When we try to carry out these approaches we face some technical complications. The classification of structures is often done by properties like transitivity or symmetry, but the standard ways of encoding regular relations (in our considerations mainly automata and ω-semigroups) represent these properties only rather implicitly.

To demonstrate how to overcome these complications on the class of all uncountable linear orders. We show that every automatic presentation of an uncountable linear order contains an injective automatic presentation of $(\{0,1\}^\omega, \prec)$. To prove this fact we make extensively use the transitivity of the linear order. Unfortunately we will see that the technical details required to make this property accessible are quite involved. The main ideas originate from [11]. Kuske [14] was the first to utilize these ideas to prove the following theorems.

Theorem 6.3.1. *Let $\mathcal{K}_{lin}^{>\aleph_0}$ be the class of all uncountable linear orders and $\mathfrak{A}_{lex}$ the structure $(\{0,1\}^\omega, \prec)$ than it holds that*

$$(\mathcal{K}_{lin}^{>\aleph_0}, \omega Aut) \Rightarrow (\mathfrak{A}_{lex}, i\omega Aut).$$

Proof. Let $\mathfrak{A}$ be a structure from $\mathcal{K}_{lin}^{>\aleph_0}$ and $\mathcal{L} = (L_A, \approx, <)$ an automatic presentation of $\mathfrak{A}$. Since $\mathcal{L}$ is automatic there are ω-semigroup morphisms to finite ω-semigroups $S_\delta = (S_f^\delta, S_\omega^\delta)$, $\delta \in \{A, \approx, <\}$

$$()^A : \Sigma^\infty \to S_A,$$
$$()^\approx : (\Sigma^2)^\infty \to S_\approx \text{ and}$$
$$()^< : (\Sigma^2)^\infty \to S_<$$

that recognize the corresponding relations. For $\delta \in \{A, \approx, <\}$ we set $F_\delta := (L_\delta)^\delta$.

We define c to be the size of the largest ω-semigroup among $S_L, S_\approx, S_<$, $C := c^3$ and k as the least common multiple of the exponents of those semigroups.

Our goal is to show that there are $u, v_0, v_1 \in \Sigma^+$ with $v_0 \neq v_1$, $|v_0| = |v_1|$ such that $u\{v_0, v_1\}^\omega \subseteq L_A$ and for all $\alpha, \beta \in \{0,1\}^\omega$ it holds that $uv_\alpha < uv_\beta$ iff $\alpha \prec \beta$ (here v_α stands for $v_{\alpha[0]}v_{\alpha[1]}v_{\alpha[2]}\ldots$). This obviously means, that $\mathcal{L}$ restricted to $u\{v_0, v_1\}^\omega$ is an injective ω-automatic presentation of $\mathfrak{A}_{lex}$.

We decompose the construction into several steps. First we need to show that there are two words and a factorization H such that regarding H, both words behave in the same way under every mentioned morphism. Later on we will use the obtained words to cut out suitable candidates for u, v_0 and v_1.

Lemma 6.3.1. There are $x_0, x_1 \in L_A$ with $[x_i]_{\sim_e} \cap [x_{1-i}]_\approx = \emptyset$ and a factorization $H = \{h_1 < h_2 < h_3 < \ldots\}$ such that for some idempotent $e_A \in S_A$, H is a $()^A, e_A$-homogeneous factorization of x_0 and x_1. For $\delta \in \{\approx, <\}$ there are idempotent e_δ, e_δ^{01} and e_δ^{10} such that

- H is a $()^\delta, e_\delta$-homogeneous factorization of $\langle x_0, x_0\rangle$ and $\langle x_1, x_1\rangle$ and
- H is a $()^\delta, e_\delta^{01}$-homogeneous factorization of $\langle x_0, x_1\rangle$ and
- H is a $()^\delta, e_\delta^{10}$-homogeneous factorization of $\langle x_1, x_0\rangle$.

Proof. Since $\mathcal{L}$ is a presentation of an uncountable structure, there is an infinite set X such that for all $x \neq y \in X$ it holds that $[x]_{\sim_e} \cap [y]_\approx = \emptyset$. We choose distinct elements $y_0, y_1, \ldots, y_C$ from X and look at their pictures under the given morphisms. For that we apply Lemma 3.2.2 to the pairs

- $(()^A, y_i)$ for $0 \leq i \leq C$,
- $(()^\approx, \langle y_i, y_j\rangle)$ for $0 \leq i, j \leq C$ and
- $(()^<, \langle y_i, y_j\rangle)$ for $0 \leq i, j \leq C$.

Then we obtain $H = \{h_1 < h_2 < h_3 < \ldots\} \subseteq \mathbb{N}$ such that for $0 \leq i, j \leq C$ there are idempotent $e_A^i, e_\approx^{ij}, e_<^{ij}$ with H is

- a $()^A, e_A^i$-homogeneous factorization of y_i,
- a $()^\approx, e_\approx^{ij}$-homogeneous factorization of $\langle y_i, y_j\rangle$ and

- a $()^<, e^{ij}_<$-homogeneous factorization of $\langle y_i, y_j \rangle$.

Observe that by Lemma 3.2.2 all these semigroup elements are idempotent.

If we look at the tuples $(e^i_A, e^{ii}_\approx, e^{ii}_<) \in (S^A_f \times S^\approx_f \times S^<_f), 0 \le i \le C$ we find that since $|S^A_f \times S^\approx_f \times S^<_f| \le c^3 = C$ there are some $i \ne j$ with $(e^i_A, e^{ii}_\approx, e^{ii}_<) = (e^j_A, e^{jj}_\approx, e^{jj}_<)$. This means y_i, y_j and H fulfill the properties that we were looking for. □

We may also assume that H is coarse enough that $x_0[h_l, h_{l+1}) \ne x_1[h_l, h_{l+1})$ for all $l \in \mathbb{N}$. We need to modify x_0, x_1 a bit to ensure all the properties we will need in following.

Lemma 6.3.2. There are $y_0, y_1 \in L_A$ with $y_0 \not\approx y_1$ and a factorization $G = \{g_1 < g_2 < g_3 < \ldots\}$ with the following properties:

- $y_0[0, g_1) = y_1[0, g_1)$ and $y_0[g_i, g_{i+1}) \ne y_1[g_i, g_{i+1})$ for all $i \in \mathbb{N}$.
- for $\delta \in \{\approx, <\}$ there is a $\rightarrow_\delta \in S^\delta_f$ and idempotent $\Box_\delta, \uparrow_\delta, \downarrow_\delta \in S^\delta_f$ such that
  - $\langle y_0, y_0 \rangle [0, g_1)^\delta = \rightarrow_\delta$,
  - $\langle y_0, y_0 \rangle [g_i, g_{i+1})^\delta = \langle y_1, y_1 \rangle [g_i, g_{i+1})^\delta = \Box_\delta$,
  - $\langle y_0, y_1 \rangle [g_i, g_{i+1})^\delta = \uparrow_\delta$,
  - $\langle y_1, y_0 \rangle [g_i, g_{i+1})^\delta = \downarrow_\delta$ and
  - $\rightarrow_\delta, \uparrow_\delta$ and $\downarrow_\delta$ absorb $\Box_\delta$.

Proof. We first construct y_0 and y_1 and then show that these have the desired properties. We define y_0 as

$$y_0 := x_1[0, h_2) x_0[h_2, \omega)$$

and y_1 by

$$y_1[0, h_2) := x_1[0, h_2) \text{ and}$$
$$y_1[h_{2l}, h_{2l+2}) := x_1[h_{2l}, h_{2l+1}) x_0[h_{2l+1}, h_{2l+2}) \text{ for } l \ge 1.$$

Due to our assumptions about H we have $y_0 \not\sim_e y_1$. We set $G = \{h_{2kl+2} : l \in \mathbb{N}\}$ (remember k is the least common multiple of the exponents of the involved semigroups). Then $y_0[0, g_1) = y_1[0, g_1)$ and $y_0[g_i, g_{i+1}) \ne y_1[g_i, g_{i+1})$ as postulated.

It is easy to see that G is a $()^A, e_A$-homogeneous factorization of y_0 and y_1 and therefore it holds that $(y_i)^A = x_1[0, g_2)^A e_A^\omega = (x_1)^A$ what implies that $y_i \in L_A$.

Next we show $y_0 \not\approx y_1$:

$$\begin{aligned}
\langle y_0, y_1\rangle^{\approx} &= \langle y_0, y_1\rangle[0, g_0)^{\approx} \cdot \pi_{\approx}((\langle y_0, y_1\rangle[g_i, g_{i+1})^{\approx})_{i\geq 0}) \\
&= \underbrace{\langle x_1, x_1\rangle[0, h_1)^{\approx}}_{:=s}\langle x_1, x_1\rangle[h_1, h_2)^{\approx} \cdot \pi_{\approx}((\langle y_0, y_1\rangle[g_i, g_{i+1})^{\approx})_{i\geq 1}) \\
&= se_{\approx} \cdot \pi_{\approx}((\langle x_0, x_1\rangle[h_{2i}, h_{2i+1})^{\approx} \cdot \langle x_0, x_0\rangle[h_{2i+1}, h_{2i+2})^{\approx})_{i\geq 1}) \\
&= se_{\approx}(e_{\approx}^{01} e_{\approx})^{\omega} \\
&= se_{\approx}e_{\approx}(e_{\approx}^{01} e_{\approx})^{\omega} \\
&= se_{\approx}(e_{\approx}e_{\approx}^{01})^{\omega} \\
&= \langle x_1, x_1\rangle[0, h_2)^{\approx} \cdot \pi_{\approx}((\langle x_1, x_1\rangle[h_{2i}, h_{2i+1})^{\approx} \cdot \langle x_0, x_1\rangle[h_{2i+1}, h_{2i+2})^{\approx})_{i\geq 1}) \\
&= \langle y_1, x_1\rangle^{\approx}
\end{aligned}$$

So if $y_0 \approx y_1$ then also $y_1 \approx x_1$ and therefore by transitivity $y_0 \approx x_1$. But $y_0 \sim_e x_0$ which means that $[x_0]_{\sim_e} \cap [x_1]_{\approx} \neq \emptyset$, which is contradicting the initial choice of x_0 and x_1.

We will now compute the values $\rightarrow_\delta, \uparrow_\delta, \downarrow_\delta$ and $\Box_\delta$ for $\delta \in \{\approx, <\}$:

$$\begin{aligned}
\rightarrow_\delta &= \langle y_0[0, g_0), y_0[0, g_0)\rangle^\delta \\
&= \langle x_1[0, h_2), x_1[0, h_2)\rangle^\delta \\
&= \langle x_1[0, h_1), x_1[0, h_1)\rangle^\delta e_\delta \\
\Box_\delta &= \langle y_0[g_i, g_{i+1}), y_0[g_i, g_{i+1})\rangle^\delta \\
&= \langle x_0[h_{2ki+2}, h_{2ki+2k+2}), x_0[h_{2ki+2}, h_{2ki+2k+2})\rangle^\delta \\
&= (e_\delta)^{2k} \\
&= e_\delta.
\end{aligned}$$

By similar computations we get $\uparrow_\delta = (e_\delta^{01} e_\delta)^k$ and $\downarrow_\delta = (e_\delta^{10} e_\delta)^k$.

Since e_δ is idempotent and k is a multiple of the exponent of S_f^δ, $\Box_\delta, \uparrow_\delta$ and $\downarrow_\delta$ are idempotent and $\rightarrow_\delta, \uparrow_\delta$ and $\downarrow_\delta$ absorb $\Box_\delta$.

It remains to show that $\langle y_1[g_i, g_{i+1}), y_1[g_i, g_{i+1})\rangle^\delta = \Box_\delta$.

$$
\begin{aligned}
\langle y_1[g_i, g_{i+1}), y_1[g_i, g_{i+1})\rangle^\delta &= \langle y_0, y_0\rangle[h_{2ki+2}, h_{2k(i+1)+2})^\delta \\
&= \langle x_1, x_1\rangle[h_{2ki+2}, h_{2ki+3})^\delta \langle x_0, x_0\rangle[h_{2ki+2}, h_{2ki+4})^\delta \\
&\quad \dots \\
&\quad \langle x_1, x_1\rangle[h_{2k(i+1)}, h_{2k(i+1)+1})^\delta \langle x_0, x_0\rangle[h_{2k(i+1)+1}, h_{2k(i+1)+2})^\delta \\
&= (e_\delta)^{2k} \\
&= \Box_\delta
\end{aligned}
$$

□

Now that we have y_0 and y_1 we are ready to construct u, v_0 and v_1. We set

$$u := y_1[0, g_0), \qquad v_0 := y_0[g_0, g_1) \text{ and } \qquad v_1 := y_1[g_0, g_1).$$

From this definition we immediately get for $\delta \in \{\approx, <\}$:

$$
\begin{aligned}
\langle u, u\rangle^\delta &= \rightarrow_\delta, \\
\langle v_0, v_0\rangle^\delta &= \langle v_1, v_1\rangle^\delta = \Box_\delta, \\
\langle v_0, v_1\rangle^\delta &= \uparrow_\delta \text{ and} \\
\langle v_1, v_0\rangle^\delta &= \downarrow_\delta .
\end{aligned}
$$

In the following we will omit the subscripts and just write $\rightarrow, \uparrow, \downarrow$ and $\Box$ since it will be obvious from the context which one is meant.

We will now show that $u\{v_0, v_1\}^\omega$ has all the properties that were announced at the beginning of the proof.

Lemma 6.3.3. $u\{v_0, v_1\}^\omega \subseteq L_A$

Proof. Let α be any sequence from $\{0, 1\}^\omega$.

$$
\begin{aligned}
(uv_\alpha)^A &= y_1[0, g_0)^A (y_{\alpha[i]}[g_0, g_1)^A)_{i\in\mathbb{N}} \\
&= x_1[0, g_0)^A (e_A)^\omega \\
&= (x_1)^A \in F_A
\end{aligned}
$$

This means every uv_α is in L_A and therefore $u\{v_0, v_1\}^\omega \subseteq L_A$. □

Next we show that at least some words from $u\{v_0, v_1\}^\omega$ do encode distinct elements.

Lemma 6.3.4. $\rightarrow (\uparrow\downarrow)^\omega \notin F_\approx$.

Proof. In a first step we see that $\rightarrow\uparrow^\omega \notin F_\approx$ since $\langle y_0, y_1\rangle^\approx = \rightarrow\uparrow^\omega$ and $y_0 \not\approx y_1$. We will make use of the transitivity of $\approx$ to show that also $\rightarrow (\uparrow\downarrow)^\omega \notin F_\approx$. Suppose $\rightarrow (\uparrow\downarrow)^\omega \in F_\approx$, then consider the words $u(v_0v_1v_0)^\omega$,$u(v_1v_0v_1)^\omega$ and $u(v_1v_1v_0)^\omega$. We have

$$\langle u(v_0v_1v_0)^\omega, u(v_1v_0v_1)^\omega\rangle^\approx = \rightarrow (\uparrow\downarrow\uparrow)^\omega = \rightarrow (\uparrow\downarrow)^\omega \text{ and}$$
$$\langle u(v_1v_0v_1)^\omega, u(v_1v_1v_0)^\omega\rangle^\approx = \rightarrow (\Box\uparrow\downarrow)^\omega = \rightarrow (\uparrow\downarrow)^\omega.$$

Hence $u(v_0v_1v_0)^\omega \approx u(v_1v_0v_1)^\omega$ and $u(v_1v_0v_1)^\omega \approx u(v_1v_1v_0)^\omega$ and so by transitivity $u(v_0v_1v_0)^\omega \approx u(v_1v_1v_0)^\omega$, but

$$\langle u(v_0v_1v_0)^\omega, u(v_1v_1v_0)^\omega\rangle^\approx = \rightarrow (\uparrow\Box\Box)^\omega = \rightarrow\uparrow^\omega \notin F_\approx$$

which leads to the wanted contradiction. ∎

We conclude our proof by showing that $<$ is indeed the lexicographic order on $u\{v_0, v_1\}^\omega$.

Lemma 6.3.5. There is an ordering of $\{0, 1\}$ such that for the induced lexicographic order $\prec$ on $\{0, 1\}^\omega$ it holds that $uv_\alpha < uv_\beta$ iff $\alpha \prec \beta$ for every $\alpha \neq \beta \in \{0, 1\}^\omega$.

Proof. First observe that $u(v_0v_1)^\omega \not\approx u(v_1v_0)^\omega$ since $\langle u(v_0v_1)^\omega, u(v_1v_0)^\omega\rangle^\approx = \rightarrow (\uparrow\downarrow)^\omega \notin F_\approx$. With this fact we know that either $u(v_0v_1)^\omega < u(v_1v_0)^\omega$ or $u(v_1v_0)^\omega < u(v_0v_1)^\omega$ holds. We carry out the proof for the case that $u(v_0v_1)^\omega < u(v_1v_0)^\omega$ holds. In this case we take the usual order on $\{0, 1\}$ as basis for $\prec$. For the other case take the reversed order on $\{0, 1\}$ as basis and interchange the roles of v_0 and v_1 as well as the roles of $\uparrow$ and $\downarrow$.

Look at the pictures of $\langle uv_\alpha, uv_\beta\rangle$, $\alpha \neq \beta \in \{0, 1\}^\omega$ under $()^<$. If we take the product $\langle u, u\rangle^<\langle v_{\alpha[0]}, v_{\beta[0]}\rangle^<\langle v_{\alpha[1]}, v_{\beta[1]}\rangle^< \dots$ and use idempotence and absorption to eliminate occurrences of $\uparrow\uparrow$, $\downarrow\downarrow$ and $\Box$ (except for occurrences in $\uparrow^\omega, \downarrow^\omega$ and $\Box^\omega$ at the end) we get, to a product of one of the following forms.

1. $\rightarrow (\uparrow\downarrow)^\omega$

2. $\rightarrow (\uparrow\downarrow)^n \uparrow^\omega$, $n \geq 0$

3. $\rightarrow (\uparrow\downarrow)^n \Box^\omega$, $n > 0$

4. $\rightarrow (\uparrow\downarrow)^n \uparrow \cdot \{\Box^\omega, \downarrow^\omega\}$, $n \geq 0$

5. $\rightarrow (\downarrow\uparrow)^\omega$

6. $\rightarrow (\downarrow\uparrow)^n \downarrow^\omega$, $n \geq 0$

7. $\rightarrow (\downarrow\uparrow)^n \Box^\omega$, $n > 0$

8. $\rightarrow (\downarrow\uparrow)^n \downarrow \cdot \{\Box^\omega, \uparrow^\omega\}$, $n \geq 0$

The product we obtain has one of the first four forms iff on the first position i where α and β differ we have $\alpha[i] = 0$ and $\beta[i] = 1$ i.e. $\alpha \prec \beta$. It has one of the last four forms iff the "reversed" product obtained from $\langle uv_\beta, uv_\alpha \rangle^<$ has one of the first four forms. If we can show that every product of the form 1, 2, 3 and 4 is in $F_\approx$, we get that $u\{v_0, v_1\}^\omega$ is ordered as desired by $<$. We already know that this is the case for $\rightarrow (\uparrow\downarrow)^\omega$. Once again we will use a transitivity argument to show it for products of the forms 2, 3 and 4. More precisely we will show that for every product ρ of the forms 2, 3 and 4 there are words w_1, w_2 and w_3 such that $\langle w_1, w_2 \rangle^< = \langle w_2, w_3 \rangle^< =\rightarrow (\uparrow\downarrow)^\omega$ and $\langle w_1, w_3 \rangle^< = \rho$. Since $\rightarrow (\uparrow\downarrow)^\omega \in F_<$ it holds that $w_1 < w_2$ and $w_2 < w_3$ and so by transitivity $w_1 < w_3$, but this means $\rho \in F_<$.

- $\rightarrow (\uparrow\downarrow)^n \uparrow^\omega$, $n \geq 0$:

$$
\begin{aligned}
w_1 &:= u(v_0v_1)^n(v_0v_0v_1)^\omega \\
w_2 &:= u(v_1v_0)^n(v_0v_1v_0)^\omega \\
w_3 &:= u(v_1v_0)^n(v_1v_0v_1)^\omega
\end{aligned}
$$

$$
\begin{aligned}
\langle w_1, w_2 \rangle^< &=\rightarrow (\uparrow\downarrow)^n(\Box \uparrow\downarrow)^\omega =\rightarrow (\uparrow\downarrow)^\omega \\
\langle w_2, w_3 \rangle^< &=\rightarrow \Box^{2n}(\uparrow\downarrow\uparrow)^\omega =\rightarrow (\uparrow\downarrow)^\omega \\
\langle w_1, w_3 \rangle^< &=\rightarrow (\uparrow\downarrow)^n(\uparrow \Box\Box)^\omega =\rightarrow (\uparrow\downarrow)^n \uparrow^\omega
\end{aligned}
$$

- $\rightarrow (\uparrow\downarrow)^n \Box^\omega$, $n > 0$:

$$\begin{aligned} w_1 &:= u(v_0v_1)^n(v_0v_1)^\omega \\ w_2 &:= u(v_0v_1)^n(v_1v_0)^\omega \\ w_3 &:= u(v_1v_0)^n(v_0v_1)^\omega \end{aligned}$$

$$\begin{aligned} \langle w_1, w_2\rangle^< &=\rightarrow \Box^{2n}(\uparrow\downarrow)^\omega =\rightarrow (\uparrow\downarrow)^\omega \\ \langle w_2, w_3\rangle^< &=\rightarrow (\uparrow\downarrow)^n(\downarrow\uparrow)^\omega =\rightarrow (\uparrow\downarrow)^\omega \\ \langle w_1, w_3\rangle^< &=\rightarrow (\uparrow\downarrow)^n(\Box\Box)^\omega =\rightarrow (\uparrow\downarrow)^n\Box^\omega \end{aligned}$$

- $\rightarrow (\uparrow\downarrow)^n \uparrow \Box^\omega$, $n \geq 0$:

$$\begin{aligned} w_1 &:= u(v_0v_1)^n v_0(v_0v_1)^\omega \\ w_2 &:= u(v_0v_1)^n v_0(v_1v_0)^\omega \\ w_3 &:= u(v_1v_0)^n v_1(v_0v_1)^\omega \end{aligned}$$

$$\begin{aligned} \langle w_1, w_2\rangle^< &=\rightarrow \Box^{2n+1}(\uparrow\downarrow)^\omega =\rightarrow (\uparrow\downarrow)^\omega \\ \langle w_2, w_3\rangle^< &=\rightarrow (\uparrow\downarrow)^n \uparrow (\downarrow\uparrow)^\omega =\rightarrow (\uparrow\downarrow)^\omega \\ \langle w_1, w_3\rangle^< &=\rightarrow (\uparrow\downarrow)^n \uparrow (\Box\Box)^\omega =\rightarrow (\uparrow\downarrow)^n \uparrow \Box^\omega \end{aligned}$$

- $\rightarrow (\uparrow\downarrow)^n \uparrow\downarrow^\omega$, $n \geq 0$:

$$w_1 := u(v_0v_1)^n v_0(v_1v_0v_1)^\omega$$
$$w_2 := u(v_0v_1)^n v_0(v_1v_1v_0)^\omega$$
$$w_3 := u(v_1v_0)^n v_1(v_0v_0v_1)^\omega$$

$$\langle w_1, w_2\rangle^< =\rightarrow \Box^{2n+1}(\Box \uparrow\downarrow)^\omega =\rightarrow (\uparrow\downarrow)^\omega$$
$$\langle w_2, w_3\rangle^< =\rightarrow (\uparrow\downarrow)^n \uparrow (\downarrow\downarrow\uparrow)^\omega =\rightarrow (\uparrow\downarrow)^\omega$$
$$\langle w_1, w_3\rangle^< =\rightarrow (\uparrow\downarrow)^n \uparrow (\downarrow \Box\Box)^\omega =\rightarrow (\uparrow\downarrow)^n \uparrow\downarrow^\omega$$

□

Taking all together we get that $\mathcal{L}$ restricted to $u\{v_0, v_1\}^\omega$ is an $i\omega$Aut-presentation of $(\{0,1\}^\omega, \prec)$. □

We can now use this theorem to see that certain linear orders do not have an ω-automatic presentation.

Theorem 6.3.2 (Kuske). *An ordinal* $(\alpha, <)$ *is* ω*-automatic iff* $\alpha < \omega^\omega$

Proof. Suppose there is an uncountable automatic ordinal $(\alpha, <)$. Let $\mathcal{L} = (L, \approx, <)$ be an automatic presentation of $(\alpha, <)$. By Theorem 6.3.1 $\mathcal{L}$ contains a sub-presentation of $(\{0,1\}^\omega, \prec)$. This means α contains a substructure isomorphic to $(\{0,1\}^\omega, \prec)$. But this is impossible since $(\{0,1\}^\omega, \prec)$ contains infinite decending chains. Therefore there are no uncountable automatic ordinals.

But we already know that every countable structure is automatic exactly if it is finite word automatic and that an ordinal is finite word automatic if and only if it is smaller than ω^ω [4]. □

This is a quite interesting result since it shows that for ordinals we can not obtain new structures by switching from automatic structures to ω-automatic structures.

We can generalize this result to scattered linear orders. A linear order is scattered iff it does not contain a densely ordered subset. This follows directly from these simple observations.

Lemma 6.3.6. $(\mathfrak{A}_{lex}, i\omega\text{Aut}) \Rightarrow ((\mathbb{R}, <), i\omega\text{Aut})$.

Proof. Let $\mathcal{L} = (L, <), \pi$ be an injective automatic presentation of $\mathfrak{A}_{lex}$. The substructure $\mathfrak{B} = (\{0,1\}^\omega - \{0,1\}^* 1^\omega, \prec)$ is isomorphic to $((0,1), <)$ and therefore also isomorphic to $(\mathbb{R}, <)$. So all we need to show is that $\mathcal{L}$ contains an injective automatic presentation of $\mathfrak{B}$. This is indeed the case since the set $\pi^{-1}(\{0,1\}^* 1^\omega)$ is definable in $(\mathcal{L}, \pi)$ by the first order formula $\varphi(x) := \exists y(x < y \wedge \neg \exists z(x < z \wedge z < y))$. Therefore we get an injective automatic presentation of $\mathfrak{B}$ with $(L - \varphi^{\mathcal{L}}, < \cap (L - \varphi^{\mathcal{L}})^2)$. □

Corollary 6.3.3. $(\mathcal{K}_{lin}^{>\aleph_0}, i\omega Aut) \Rightarrow ((\mathbb{R}, <), i\omega Aut)$.

Corollary 6.3.4 (Kuske)**.** *No uncountable scattered linear order is automatic.*

Hjorth, Khoussainov, Montalbán and Nies mention in [9] that there is also a strong result of Harrington and Shelah [7] which states that no Borel presentable linear order has a subset of order type $\aleph_1$.

6.4 Pairing Functions Revisited

We consider pairing function again. As we will see, we can use the results from the previous section to overcome the difficulties we faced when trying to lift the arguments used in Chapter 5 from injective to general ω-automatic presentations. The main Problem was that given an ω-automatic presentation we can not assume that there is a $\sim_e$-class of its domain such that this class codes infinitely many elements. But we will see that we can recover this property for the class of all uncountable linear ordered structures.

Lemma 6.4.1. Every ω-automatic presentation $\mathcal{L} = (L, \approx, <, R_1, \ldots, R_n), \pi$ of an uncountable linear ordered structure $\mathfrak{A}$ contains a $\sim_e$-class $X \in L/\sim_e^m$ such that $\pi(X)$ is infinite.

Proof. The structure $\mathcal{L}_< := (L, \approx, <)$ is an ω-automatic presentation of an uncountable linear order and therefore by Theorem 6.3.1 it contains an $i\omega$Aut-subpresentation $\mathcal{L}'_< = (L', (\approx \cap (L')^2), (< \cap (L')^2))$ of $(\{0,1\}^\omega, \prec)$. But since $\mathcal{L}'_<$ is injective there is an infinite $X \in L'/\sim_e$ such that $x \not\approx y$ for all $x \neq y \in X$ and since $X \subseteq Y$ for some $Y \in L/\sim_e$ the claim follows. □

Now we can apply the arguments seen in Chapter 5 in a slightly modified version to uncountable linear ordered structures.

Lemma 6.4.2. Let $\mathfrak{A}$ be an uncountable linear ordered ω-automatic structure and $f : A^k \to A$ a function of $\mathfrak{A}$. Then it holds that $MIS_f(n) = \mathcal{O}(n)$.

Proof. Suppose $MIS_f(n) \notin \mathcal{O}(n)$. Let $\mathcal{L} = (L, \approx, <, R_f, \ldots), \pi$ be an ω-automatic presentation of $\mathfrak{A}$.

By Lemma 6.4.1 there is an $X \in L/\sim_e$ such that $\pi(X)$ is infinite and therefore for every n we can choose a m such that there is a $Y \in L/\sim_e^m$ with $|\pi(Y)| \geq n$. By Lemma 5.2.2 there is a constant c such that $|sig^m_{R_f}(Y^k)| \leq c$ for every $m \in \mathbb{N}$ and $Y \in L/\sim_e^m$. Now choose n such that $MIS_f(n') > cn'$ for all $n' \geq n$, m as described above and $Y \in L/\sim_e^m$ such that $|\pi(Y)|$ is maximal.

Since $|\pi(Y)| \geq n$ there are $\overline{y}_0, \ldots, \overline{y}_{c|\pi(Y)|} \in Y^k$ such that $f(\pi(\overline{y}_i)) \neq f(\pi(\overline{y}_j))$ for $i \neq j$. This means for $i \neq j$ it holds that if $a \in \overline{y}_i R_f$ and $b \in \overline{y}_j R_f$ then $\pi(a) \neq \pi(b)$. Since there are only c different m-signatures among $sig^m_{R_f}(\overline{y}_0), \ldots, sig^m_{R_f}(\overline{y}_{c|\pi(Y)|})$, there are indexes $i_0, \ldots, i_{|\pi(Y)|}$ with $sig^m_{R_f}(\overline{y}_{i_r}) = sig^m_{R_f}(\overline{y}_{i_l})$. Now choose some $\sim_e^m$-class $Z \in sig^m_{R_f}(\overline{y}_{i_0})$. Then it holds that for every $j \in \{0, \ldots, n\}$ there is a $z_j \in Z$ such that $z_j \in \overline{y}_{i_j} R_f$ and therefore $|\pi(Z)| > |\pi(Y)|$. But this contradicts the choice of Y. $\square$

Corollary 6.4.1. *There are no uncountable linear ordered structures in $\omega AutStr$ in which a pairing function can be defined.*

Chapter 7

The Field of Reals

Although by now it is not known whether field of reals $\mathfrak{R} = (\mathbb{R}, +, \cdot)$ is automatic or not, we can gather some properties that would be fulfilled by every automatic presentation of $\mathfrak{R}$. One could hope to find some properties that give rise to a proof that $\mathfrak{R}$ is not automatic or, on the other side, give hints on how an automatic presentation could be constructed.

7.1 Reducts

Besides the fact that $(\mathbb{N}, +, \cdot)$ can not be automatic, since its first order theory is undecidable, one can also show that it is not automatic by showing that already one of its reducts is not automatic. While it is easy to show that $(\mathbb{N}, +)$ is automatic, Blumensath showed in [2] that $(\mathbb{N}, \cdot)$ is not automatic.

Carrying this idea over to $\mathfrak{R}$ it may be worthwhile to check whether the reducts $(\mathbb{R}, +)$ and $(\mathbb{R}, \cdot)$ are automatic. But as we will see in this section the situation here is a bit different than in the case of $(\mathbb{N}, +, \cdot)$. It is also not hard to see that $(\mathbb{R}, +)$ is automatic but $(\mathbb{R}, \cdot)$ behaves very similar to $(\mathbb{R}, +)$ and therefore it is possible to obtain an automatic presentation of $(\mathbb{R}, \cdot)$ from an automatic presentation of $(\mathbb{R}, +)$.

Lemma 7.1.1. $(\mathbb{R}, +) \in i\omega\text{AutStr}$.

Proof. The idea is to encode every $a \in \mathbb{R}$ by the convolution of two words v_0 and v_1, where v_0 is the binary presentation of the integer part of a (least significant bit first) followed by an infinite number of zeros and v_1 is the binary representation of the fractional digits (most significant bit first). To distinguish between positive and negative reals we prepend one of the signs $+, -$ to this coding. We get to a presentation $\mathcal{L} = (L, \approx, R_+), \pi$ over the alphabet $\Sigma = \{+, -\} \cup (\{0,1\} \times \{0,1\})$. We set

$$L := \{+, -\}L'$$

where L' is defined by

$$L' := \{\left(\begin{smallmatrix}0\\0\end{smallmatrix}\right), \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right), \left(\begin{smallmatrix}1\\1\end{smallmatrix}\right)\}^*\{\left(\begin{smallmatrix}0\\0\end{smallmatrix}\right), \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)\}^\omega.$$

and

$$\pi(\sigma \left(\begin{smallmatrix}a_0\\b_0\end{smallmatrix}\right) \left(\begin{smallmatrix}a_1\\b_1\end{smallmatrix}\right) \left(\begin{smallmatrix}a_2\\b_2\end{smallmatrix}\right) \dots) = \sigma \left(\sum_{i=0}^{\infty} 2^i a_i + \sum_{i=0}^{\infty} 2^{-(i+1)} b_i \right)$$

for $\sigma \in \{+, -\}$. We now need to check that the induced relation R_+ is automatic. First check that the relation

$$R'_+ := \{(u, v, w) \in L^3 : \pi(+u) + \pi(+v) = \pi(+w)\}$$

which is basically R_+ restricted to the codings of positive reals (including 0) is automatic. We informally describe a nondeterministic automaton $\mathcal{A}'$ that recognized R'_+. On an input

$$\langle \left(\begin{smallmatrix}a_0\\b_0\end{smallmatrix}\right) \left(\begin{smallmatrix}a_1\\b_1\end{smallmatrix}\right) \dots, \begin{pmatrix}a'_0\\b'_0\end{pmatrix} \begin{pmatrix}a'_1\\b'_1\end{pmatrix} \dots, \begin{pmatrix}a''_0\\b''_0\end{pmatrix} \begin{pmatrix}a''_1\\b''_1\end{pmatrix} \dots \rangle$$

The automaton first guesses if he has to carry over a bit from the addition of the fractional parts to the addition of the integer parts. Taking this carry into account $\mathcal{A}'$ then checks the addition of the integer part as it was seen in Lemma 4.1.2. Simultaneously $\mathcal{A}'$ checks the addition of the factional part. Here the automaton always remembers a variable $c \in \{0, 1\}$ that indicates that the automaton awaits an carry to appear or not (c is initialized with the first guess described above). The automaton then proceeds in the following way:

case $c = 0$.

- If $(b_i, b'_i b''_i) \in \{(0, 0, 0), (0.1.1), (1, 0, 1)\}$ then set c to 0 and proceed.
- If $(b_i, b'_i b''_i) = (0, 0, 1)$ then set c to 1 and proceed.
- In every other case reject.

case $c = 1$.

- If $(b_i, b'_i b''_i) = (1, 1, 0)$ then set c to 0 and proceed.
- If $(b_i, b'_i b''_i) \in \{(1, 0, 0), (0.1.0), (1, 1, 1)\}$ then set c to 1 and proceed.
- In every other case reject.

At last $\mathcal{A}'$ accepts an input iff he never rejects it.

Now observe that by bringing terms on the other side of the equation, we can transform every proposition $a + b = c$ with $a, b, c \in \mathbb{R}$ in a unique way up to commutativity and associativity to an equivalent proposition of the form $a' + b' = c'$ or of the form $a' + b' + c' = 0$ with $a', b', c' \in \{a, -a, b, -b, c, -c\} \cap \mathbb{R}_0^+$ and this form only depends on the signs of a, b and c. Since R'_+ is automatic and 0 is definable in this presentation it is easy to see that

$$R''_+ := \{(u, v, w) \in L^3 : \pi(+u) + \pi(+v) + \pi(+w) = 0\}$$

is also automatic and can therefore be recognized by an automaton $\mathcal{A}''$. We can now construct an automaton $\mathcal{A}$ that recognizes R_+.

On input $\langle \sigma_0 w_0, \sigma_1 w_1, \sigma_2 w_2 \rangle$ This automaton simply reads the signs in the first letter and decides which proposition to check, chooses the corresponding $\hat{\mathcal{A}} = \mathcal{A}'$ or $\hat{\mathcal{A}} = \mathcal{A}''$ and the corresponding permutation $\rho : \{0, 1, 2\} \to \{0, 1, 2\}$ of the input parameters. Then he behaves like $\hat{\mathcal{A}}$ on $\langle w_{\rho(0)}, w_{\rho(1)}, w_{\rho(2)} \rangle$.

Note that this presentation is not injective, but we can construct an injective presentation $\mathcal{L}' = (L', R_+^{\mathcal{L}'})$ with $L' := L - (\{-\left(\begin{smallmatrix}0\\0\end{smallmatrix}\right)^\omega\} \cup \Sigma^* \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)^\omega)$ and $R_+^{\mathcal{L}'} = R_+ \cap L'^3$. Since the set $\{-\left(\begin{smallmatrix}0\\0\end{smallmatrix}\right)^\omega\} \cup \Sigma^* \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)^\omega$ is regular it follows that $\mathcal{L}'$ is an injective automatic presentation of $(\mathbb{R}, +)$. □

Lemma 7.1.2. $(\mathbb{R}, \cdot)$ is automatic.

Proof. First observe that $(\mathbb{R}^+, \cdot)$ is automatic, since it is isomorphic to $(\mathbb{R}, +)$. This is because of the fact that $\log_2$ is a bijection between $\mathbb{R}^+$ and $\mathbb{R}$ and $\log_2(2^x \cdot 2^y) = x + y = \log_2(2^x) + \log_2(2^y)$. This means every automatic presentation of $(\mathbb{R}, +)$ with labeling function $\pi(x)$ is also an automatic presentation of $(\mathbb{R}^+, \cdot)$ with labeling function $2^{\pi(x)}$.

Now let $\mathcal{L} = (L, \approx^{\mathcal{L}}, R_{\cdot}^{\mathcal{L}}), \pi$ be an automatic presentation of $(\mathbb{R}^+, \cdot)$ over the alphabet Σ. We construct an automatic presentation $\mathcal{L}' = (L', \approx^{\mathcal{L}}, \cdot^{\mathcal{L}}), \pi'$ of $(\mathbb{R}, \cdot)$ over the alphabet $\Sigma \uplus \{-1, 0, 1\}$. Intuitively we use the fact that every $x \in \mathbb{R}$ can be written as ex' with $e \in \{-1, 0, 1\}, x' \in \mathbb{R}^+$ and $ex \cdot e'x' = ee' \cdot xx'$. We define $\mathcal{L}'$ as follows:

- $L' := \{-1, 0, 1\}L$
- $\approx^{\mathcal{L}'} := \{(eu, ev) : (e \in \{-1, 1\} \wedge u \approx^{\mathcal{L}} v) \vee e = 0\}$
- $R_{\cdot}^{\mathcal{L}'} := \{(e_1 u, e_2 v, (e_1 \cdot e_2) w) : (e \in \{-1, 1\} \wedge (u, v, w) \in R_{\cdot}^{\mathcal{L}}) \vee e_1 \cdot e_2 = 0\}$
- $\pi^{\mathcal{L}'}(eu) := e \cdot \pi^{\mathcal{L}}(u)$

Since all these relations are regular, it follows that $\mathcal{L}'$ is an automatic presentation of $(\mathbb{R}, \cdot)$. □

Note that the fact that all reducts are automatic does not imply that $\mathfrak{R}$ is automatic. This is formulated in the following lemma.

Lemma 7.1.3 (Blumensath [2]). The are structures $\mathfrak{A}$ such that all reducts of $\mathfrak{A}$ are automatic but $\mathfrak{A}$ itself is not ω-automatic.

Proof. Consider the natural numbers with addition and squaring $\mathfrak{A} = (\mathbb{N}, +, ^2)$. The structure $\mathfrak{A}$ is not (ω-)automatic since the multiplication is first order definable in $\mathfrak{A}$.

We now show that every reduct of $\mathfrak{A}$ is automatic. The structure $(\mathbb{N})$ is clearly automatic and we have already seen that $(\mathbb{N}, +)$ is automatic. To see that $(\mathbb{N}, ^2)$ is automatic observe that every $n \in \mathbb{N}$ can uniquely be written as m^{2^k} for some $m \in M := \mathbb{N} - \{k^2 : k \in \mathbb{N}\}$. Let $m_0, m_1, m_2, \ldots$ be an enumeration of M. Then $(\mathbb{N}, ^2)$ is isomorphic to the structure $(\mathbb{N} \times \mathbb{N}, s)$ with $s(i, j) = (i, j + 1)$ via the isomorphism $f(m_i^{2^j}) = (i, j)$. The structure $(\mathbb{N} \times \mathbb{N}, s)$ can be represented injectively over $\{0, 1\}^2$ in the following way:

- $L := \{\langle 1^n 0^\omega, 1^m 0^\omega \rangle : n, m \in \mathbb{N}\}$
- $s(\langle 1^n 0^\omega, 1^m 0^\omega \rangle) = \langle 1^n 0^\omega, 1^{m+1} 0^\omega \rangle$
- $\pi(\langle 1^n 0^\omega, 1^m 0^\omega \rangle) = (n, m)$

Since this presentation is automatic, it follows that $(\mathbb{N}, ^2)$ is automatic. □

7.2 Definable Reals

Another quite direct way to show that $\mathfrak{R}$ is not automatic would be to show that the set of natural numbers is definable in automatic presentations of $\mathfrak{R}$. Therefore we are interested in how the natural numbers are coded in such presentations. Since every $n \in \mathbb{N}$ is first order definable in $\mathfrak{R}$, it is definable in every automatic presentation. By Lemma 4.3.1 every such n has an ultimately periodic coding and hence we can investigate the coding length of the natural numbers in terms of the concepts introduced in section 4.3.

Lemma 7.2.1. For every automatic presentation $\mathcal{L} = (L, \approx, R_+, R_\cdot), \pi$ of $\mathfrak{R}$ there is a constant c such that for every $n \in \mathbb{N}$ it holds that $\lambda^\pi_{p+c}(2^{2^n}) \leq c^{n+1}$.

Proof. Since the squaring function $()^2$ is first order definable in $\mathfrak{R}$ there is a Büchi-automaton $\mathcal{A} = (Q, \Sigma^2, q_0, \Delta, F)$ that recognizes $R_{()^2}$. By Corollary 4.3.3 there is a c_0 such that for all $x \in \mathbb{R}$, that are definable in $\mathcal{L}, \pi$, it holds that $\lambda_{p+c}(x^2) \leq c_0 \cdot \lambda_{p+c}(x)$. We set $c := \max\{c_0, \lambda^\pi_{p+c}(2)\}$ and show the claim by induction over n.

For $n = 0$ we have $\lambda^\pi_{p+c}(2) \leq c$ by definition. Now if the proposition holds for some $n \in \mathbb{N}$, we get:

$$\begin{aligned}\lambda^\pi_{p+c}\left(2^{2^{n+1}}\right) &= \lambda^\pi_{p+c}\left(\left(2^{2^n}\right)^2\right)\\ &\leq \lambda^\pi_{p+c}\left(2^{2^n}\right) \cdot c\\ &\leq c^{n+1} \cdot c\\ &= c^{n+2}.\end{aligned}$$

□

From this starting point we can estimate the coding length of other natural numbers. Next we do this for general powers of two.

Lemma 7.2.2. For every automatic presentation $(L, \approx, R_+, R_\cdot), \pi$ of $\mathfrak{R}$ it holds that $\lambda^\pi_{p+c}(2^n) = n^{\mathcal{O}(\log n)}$ for every $n \in \mathbb{N}$.

Proof. For $n \in \mathbb{N}$ choose $I \subseteq \{0, \dots, \lfloor \log_2 n \rfloor\}$ such that $\sum_{i \in I} 2^i = n$ and c as in Lemma

7.2. We can now estimate $\lambda^\pi_{p+c}(2^n)$ as follows:

$$\begin{aligned}
\lambda^\pi_{p+c}\left(2^n\right) &= \lambda^\pi_{p+c}\left(2^{\sum_{i\in I} 2^i}\right) \\
&= \lambda^\pi_{p+c}\left(\prod_{i\in I} 2^{2^i}\right) \\
&\leq \left(\prod_{i\in I} \lambda^\pi_{p+c}\left(2^{2^i}\right)\right) c^{|I|} \\
&\leq \left(\prod_{i=0}^{\lfloor \log_2 n\rfloor} c^{i+1}\right) c^{\lfloor \log_2 n\rfloor+1} \\
&= c^{\frac{(\lfloor \log_2 n\rfloor+1)(\lfloor \log_2 n\rfloor+2)}{2}+\lfloor \log_2 n\rfloor+1} \\
&= c^{\mathcal{O}((\log n)^2)} = n^{\mathcal{O}(\log n)}
\end{aligned}$$

□

As a last step in this chain we will now investigate the minimal coding length of natural numbers in general.

Lemma 7.2.3. For every automatic presentation $(L, \approx, R_+, R_\cdot), \pi$ of $\mathfrak{R}$ there is a constant c such that $\lambda^\pi_{p+c}(n) = n^{\mathcal{O}((\log\log n)^2)}$ for every $n \in \mathbb{N}$.

Proof. Choose c and I as before and we get

$$\begin{aligned}
\lambda^\pi_{p+c}(n) &= \lambda^\pi_{p+c}\left(\sum_{i\in I} 2^i\right) \\
&\leq \left(\prod_{i\in I} \lambda^\pi_{p+c}(2^i)\right) c^{|I|} \\
&= \left(\prod_{i=0}^{\lfloor \log_2 n\rfloor} i^{\mathcal{O}((\log i))}\right) c^{\lfloor \log_2 n\rfloor} \\
&= \left(\prod_{i=0}^{\lfloor \log_2 n\rfloor} (\log n)^{\mathcal{O}(\log\log n)}\right) c^{\lfloor \log_2 n\rfloor} \\
&= (\log n)^{\mathcal{O}(\log n(\log\log n))} \\
&= n^{\mathcal{O}((\log\log n)^2)}.
\end{aligned}$$

□

Unfortunately these boundaries seem to be too large to be used in growth rate arguments for non automaticity.

Another approach that could be tried is the use of pairing functions. The easiest way to define a pairing function is to give a first order definition. Although no pairing function that is FO definable in $\mathfrak{R}$ is known to the author, there are first order definable functions $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ with $f \upharpoonright \mathbb{N}$ is a pairing function. For example the function $f(x, y) = \frac{(x+y)(x+y+1)}{2} + y$ is obviously first order definable and the restriction to $\mathbb{N}$ is the well known Cantor pairing function. We can use this fact to obtain some further properties of the codings of natural numbers.

Lemma 7.2.4. For every automatic presentation $(\mathcal{L}, \pi)$ of $\mathfrak{R}$ there is a constant c such that every $\sim_e$-class contains codings of at most c different natural numbers.

Proof. Let $f : \mathbb{R}^2 \to \mathbb{R}$ be the previously mentioned function. Since f is first order definable in $\mathfrak{R}$, the relation $\pi^{-1}(f)$ is automatic. Choose c to be the constant from Lemma 5.2.2. Suppose there is a $\sim_e$-class X such that $\pi(X) \cap \mathbb{N} > c$. Then there is also a $m \in \mathbb{N}$ and a $\sim_e^m$-class X' with $\pi(X') \cap \mathbb{N} > c$. Choose $X' \in L/_{\sim_e^m}$ such that $n = |\pi(X') \cap \mathbb{N}|$ is maximal and $Y \subseteq X' \cap \pi^{-1}(\mathbb{N})$ such that for every $x \in \pi(X') \cap \mathbb{N}$ there is exactly one $v \in Y$ with $\pi(v) = x$.

From Lemma 5.2.2 we can conclude that there are at least $\lceil \frac{n^2}{c} \rceil$ pairs $(v_{i_1}, v_{i_2}) \in Y^2$ such that the values $f(\pi(v_{i_1}), \pi(v_{i_2})) \in \mathbb{N}$ have an encoding from the same $\sim_e^m$-class. Since $\pi(Y) \subseteq \mathbb{N}$ and f restricted to the natural numbers is a pairing function it follows that there is a $\sim_e^m$-class that contains codings of at least $\lceil \frac{n^2}{c} \rceil > n$ different natural numbers. But this contradicts the choice of X'. □

We now want to shift our attention from $\mathbb{N}$ to $\mathbb{Q}$. As for the natural number every $q \in \mathbb{Q}$ is definable in every automatic presentation of $\mathfrak{R}$. Another interesting property of $\mathbb{Q}$ is that is dense in $\mathbb{R}$. This means that every ϵ-environment $d(x, \epsilon)$ around some $x \in \mathbb{R}$ contains a $q \in \mathbb{Q}$. This fact can be used to obtain some further structural properties.

Lemma 7.2.5. For every automatic presentation of $(\mathbb{R}, +, \cdot)$ with labeling function π it holds that for every $x \in \mathbb{R}$ and ϵ-environment $d(x, \epsilon)$ the set $\pi^{-1}(d(x, \epsilon))$ contains an ultimately periodic word.

Proof. This follows from the fact that $\mathbb{Q}$ is dense in $\mathbb{R}$ and every $q \in \mathbb{Q}$ is definable in every automatic presentation of $(\mathbb{R}, +, \cdot)$. □

Another direct consequence is that we can define arbitrary small intervals around every element of $\mathbb{R}$.

Lemma 7.2.6. For every automatic presentation $(\mathcal{L}, \pi)$ of $\mathfrak{R}$, $a \in \mathbb{R}$ and $\epsilon > 0$ there is an interval $(p, q) \subset \mathbb{R}$ with $a \in (p, q)$ and for all $b \in (p, q)$ it holds that $|a - b| \leq \epsilon$ and (p, q) is definable in $(\mathcal{L}, \pi)$.

Proof. The order $<$ is first order definable in $\mathfrak{R}$ by the formula

$$\varphi(x, y) = \exists z(z \neq 0 \wedge x + zz = y).$$

Therefore $<$ is definable in $(\mathcal{L}, \pi)$ and with that the relation defined by

$$Rxyz \Leftrightarrow x < y < z$$

is also definable in $(\mathcal{L}, \pi)$. Now choose some $p, q \in \mathbb{Q}$ with $p < a < q$ as well as $|p - a| \leq \epsilon$ and $|q - a| \leq \epsilon$. since p and q are definable in $(\mathcal{L}, \pi)$ the relation $Rpyq = (p, q)$ is also definable in $(\mathcal{L}, \pi)$. □

We have seen that the structural properties of $\mathfrak{R}$ allow us to ensure some structural properties of its presentations. However, it remains still an open question if $\mathfrak{R}$ is automatic.

It may be possible to combine the results presented in this work and to obtain further properties. For example the possibility to define an order on $\mathfrak{R}$ in combination with Theorem 6.3.1 seems promising. But for now, we have to delay this investigation to future research.

Bibliography

[1] Vince Bárány, Erich Grädel, and Sasha Rubin. Automata-based presentations of infinite structures. *Submitted for publication.*

[2] Achim Blumensath. Automatic structures. Diploma thesis, RWTH-Aachen, 1999.

[3] Patrick Cegielski and Denis Richard. Decidability of the theory of the natural integers with the cantor pairing function and the successor. *Theoretical Computer Science*, 257(1–2):51–77, 2001.

[4] Christian Delhommé. Automaticité des ordinaux et des graphes homogènes. *Comptes Rendus Mathematique*, 339(1):5 – 10, 2004.

[5] H.-D. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic.* Springer, second edition, 1994.

[6] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In *Complexity of computation (Proc. SIAM-AMS Sympos. Appl. Math., New York, 1973)*, pages 43–73. SIAM–AMS Proc., Vol. VII. Amer. Math. Soc., Providence, R.I., 1974.

[7] Leo Harrington and Saharon Shelah. Counting equivalence classes for co-κ-souslin equivalence relations. In *Logic Colloquium '80 (Prague, 1980)*, volume 108 of *Stud. Logic Foundations Math*, pages 147–152. North-Holland, Amsterdam-New York, 1982. eds. van Dalen, D., Lascar, D. and Smiley, T.J.

[8] G. Hjorth. Borel equivalence relations. In M. Foreman and A. Kanamori, editors, *Handbook of Set Theory.* Springer, Netherlands, 2010.

[9] Greg Hjorth, Bakh Khoussainov, Antonio Montalbán, and André Nies. From automatic structures to borel structures. In *Proceedings of the 2008 23rd Annual IEEE Symposium on Logic in Computer Science*, pages 431–441, Washington, DC, USA, 2008. IEEE Computer Society.

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

[11] Łukasz Kaiser, Sasha Rubin, and Vince Bárány. Cardinality and counting quantifiers on ω-automatic structures. In Susanne Albers and Pascal Weil, editors, *Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science, STACS 2008*, pages 385–396, 2008.

[12] Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In *LCC*, pages 367–392, 1994.

[13] Bakhadyr Khoussainov, André Nies, Sasha Rubin, and Frank Stephan. Automatic structures: Richness and limitations. *CoRR*, abs/cs/0703064, 2007.

[14] Dietrich Kuske. Is ramsey's theorem omega-automatic? In *STACS*, pages 537–548, 2010.

[15] Dietrich Kuske and Markus Lohrey. First-order and counting theories of ω-automatic structures. In *In FoSSaCS*, pages 322–336, 2006.

[16] Dominique Perrin and Jean-Eric Pin. Semigroups and automata on infinite words. In *Semigroups, formal languages and groups (York, 1993)*, pages 49–72. Kluwer Acad. Publ., Dordrecht, 1995.

[17] A L Semënov. Logical theories of one-place functions on the set of natural numbers. *Mathematics of the USSR-Izvestiya*, 22(3):587, 1984.

[18] Wolfgang Thomas. Handbook of theoretical computer science (vol. b). chapter Automata on infinite objects, pages 133–191. MIT Press, Cambridge, MA, USA, 1990.

[19] Wolfgang Thomas. *Languages, automata, and logic*, pages 389–455. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[20] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *STOC*, pages 137–146, 1982.

[21] Thomas Wilke. An algebraic theory for regular languages of finite and infinite words. *Intern. J. Algebra and Comput*, 3(4):447–489, 1993.