Automatic Structures

Achim Blumensath and Erich Grädel
Mathematische Grundlagen der Informatik
RWTH Aachen
{blume,graedel}@i7.informatik.rwth-aachen.de

Abstract

We study definability and complexity issues for automatic *and* ω-automatic *structures. These are, in general, infinite structures but they can be finitely presented by a collection of automata. Moreover, they admit effective (in fact automatic) evaluation of all first-order queries. Therefore, automatic structures provide an interesting framework for extending many algorithmic and logical methods from finite structures to infinite ones.*

We explain the notion of (ω-)automatic structures, give examples, and discuss the relationship to automatic groups. We determine the complexity of model checking and query evaluation on automatic structures for fragments of first-order logic. Further, we study closure properties and definability issues on automatic structures and present a technique for proving that a structure is not automatic. We give model-theoretic characterisations for automatic structures via interpretations. Finally we discuss the composition theory of automatic structures and prove that they are closed under finitary Feferman-Vaught-like products.

1. Introduction

The relationship between logical definability and computational complexity is an important issue in a number of different fields including finite model theory, databases, knowledge representation, and computer-aided verification. So far most of the research has been devoted to finite structures where the relationship between definability and complexity is by now fairly well understood (see e.g. [6, 18]) and has many applications in particular to database theory [1]. However, in many cases the limitation to finite structures is too restrictive. Therefore in most of the fields mentioned above, there have been considerable efforts to extend the methodology from finite structures to suitable classes of infinite ones. In particular, this is the case for databases and computer-aided verification where infinite structures (like constraint databases or systems with infinite state spaces) are of increasing importance.

From a more general theoretical point of view, one may ask what classes of infinite structures are suitable for such an extension. More specifically what conditions must be satisfied by a class $\mathcal{K}$ of not necessarily finite structures such that the approach and methods of finite model theory make sense. There are two obvious and fundamental conditions:

Finite representations. Every structure $\mathfrak{A} \in \mathcal{K}$ should be representable in a finite way.

Effective semantics (for a relevant logic L, e.g., first-order logic). Given any formula $\psi(\bar{x})$ of L and (a presentation of) a structure $\mathfrak{A} \in \mathcal{K}$, one can effectively produce a presentation of the set $\{\, \bar{a} : \mathfrak{A} \models \psi(\bar{a}) \,\}$.

Note that effective semantics means in particular that the L-theory of every $\mathfrak{A} \in \mathcal{K}$ is decidable. A class of infinite structures that have been studied quite intensively in model theory are *recursive structures*. There have recently been some papers proposing the study of recursive structures (e.g., recursive databases) for the issues just mentioned [14, 15, 22]. However, the class of recursive structures is too large since, in general, only the quantifier-free formulae admit effective evaluation algorithms. Other classes of infinite structures where the relationship of definability and complexity has been studied include metafinite structures [12] and constraint databases [20].

In this paper we consider *automatic structures*. While *automatic groups* have been studied rather intensively in computational group theory (see [9, 10]) a general notion of automatic structure has only been defined and investigated in a paper by Khoussainov and Nerode [19], and the theory of these structures is not well-developed yet. Informally, a relational structure $\mathfrak{A} = (A, R_1, \ldots, R_m)$ is automatic if its universe and its relations can be presented by finite automata. This means that we can find a regular language $L_\delta \subseteq \Sigma^*$ (which provides names for the elements of $\mathfrak{A}$) and a function $\nu : L_\delta \to A$ mapping every word $w \in L_\delta$ to the element of $\mathfrak{A}$ that it represents. The function ν must be surjective (every element of $\mathfrak{A}$ must be named) but need not be injective (elements can have more than one name). In addition it must be recognisable by finite automata whether two words in L_δ name the same elements, and, for each relation R_i of $\mathfrak{A}$, whether a tuple of words in L_δ names a tuple belonging to R_i. A more detailed definition will be given in the next section.

We believe that automatic structures are very promising for the approach sketched above. Not only do automatic structures admit finite presentations, there also are numerous interesting examples and a large body of methods that has been developed in five decades of automata theory. Further, contrary to the class of recursive structures, automatic structures admit effective (in fact, automatic) evaluation of all first-order queries and possess many other pleasant algorithmic properties.

The notion of an automatic structure can be modified and generalised in many directions, for instance by using automata over infinite words, or over finite or infinite trees. In this paper we study automatic and ω-automatic structures only. Many results can be extended to tree-automatic structures without much change (see [2]), but for lack of space, we do not mention them here.

Here is an outline of this paper. In Section 2 we define the notions of automatic and ω-automatic structures and mention some examples. For the purposes of this paper, our most important examples of automatic structures are the expansions $\mathfrak{N}_p = (\mathbb{N}, +, |_p)$ of Presburger Arithmetic by a restricted divisibility predicate and tree structures $Tree(p)$. Our fundamental examples of ω-automatic structures are $\mathfrak{R}_p$, the expansion of the additive real group $(\mathbb{R}, +)$ by order and restricted divisibility, and $Tree^\omega(p)$, a natural extension of $Tree(p)$. We will also explain in Section 2 the notion of an automatic group.

In Section 3 we show that first-order logic (and in fact its extension by the quantifier "there exist infinitely many") has effective semantics on (ω-)automatic structures. We also study complexity results for model-checking and query-evaluation for first-order logic and for some of its fragments.

In Section 4 we study definability properties of automatic structures and present a technique for proving that a structure is not automatic. As an application we prove, for instance, that neither Skolem Arithmetic $(\mathbb{N}, \cdot)$ nor the divisibility poset $(\mathbb{N}, |)$ are automatic.

In Section 5 we present model-theoretic characterisations of automatic and ω-automatic structures. We prove that a structure is automatic if and only if it is interpretable in $\mathfrak{N}_p$ or, equivalently, in $Tree(p)$ for some (and hence all) $p \geq 2$. Similarly, a structure is ω-automatic if and only if it is interpretable in $\mathfrak{R}_p$ or $Tree^\omega(p)$.

Finally in Section 6 this characterisation is used to study the composition theory of automatic structures. We prove that automatic structures are closed under finitary products, unions, and similar constructions.

The goal of this paper is not to make significant new contributions to automata theory. The main technical contributions of this paper are (1) an algorithm to evaluate the quantifier "there exists infinitely many", (2) the complexity results for low level fragments of first-order logic on automatic structures, (3) the proofs that certain interesting structures are not automatic and in particular, (4) the composition theorem for automatic structures. But the main purpose of this paper is *conceptual*: we want to explore to what extent automatic structures are a suitable framework for extending the methods of finite model theory to infinite structures. We believe that the model-theoretic characterizations of automatic and ω-automatic structures in terms of interpretability are particularly useful for this and also suggest a very general way for obtaining other interesting classes of infinite structures suitable for such an approach: Fix a structure $\mathfrak{A}$ with 'nice' (algorithmic and/or model-theoretic) properties, and consider the class of all structures that are (first-order) interpretable in $\mathfrak{A}$. Obviously each structure in this class is finitely presentable (by an interpretation). Further, since many 'nice' properties are preserved by first-order interpretations, every structure in the class inherits them from $\mathfrak{A}$. In particular, every class of queries that is effective on $\mathfrak{A}$ and closed under first-order operations is effective on the interpretation-closure of $\mathfrak{A}$.

2. Automatic structures and automatic groups

We assume that the reader is familiar with the basic notions of automata theory and regular languages. One slightly nonstandard aspect is that we need a notion of regularity not just for languages $L \subseteq \Sigma^*$ but also k-ary relations of words, for $k > 1$. The idea is that regular relations are defined by automata that take tuples $\bar{w} = (w_1, \ldots, w_k)$ of words as inputs and work synchronously on all k components of $\bar{w}$. To make this precise, we represent a tuple $\bar{w} \in (\Sigma^*)^k$ by a word $w_1 \otimes \cdots \otimes w_k$ over the alphabet $(\Sigma \cup \{\Box\})^k$, called the *convolution* of $w_1, \ldots, w_k$. Here $\Box$ is a padding symbol not belonging to Σ, which is appended to some of the words w_i to make sure that all components have the same length. More formally, for $w_1, \ldots, w_k \in \Sigma^*$, with $w_i = w_{i1} \cdots w_{i\ell_i}$ and $\ell = \max\{|w_1|, \ldots, |w_k|\}$,

$$w_1 \otimes \cdots \otimes w_k := \begin{bmatrix} w'_{11} \\ \vdots \\ w'_{k1} \end{bmatrix} \cdots \begin{bmatrix} w'_{1\ell} \\ \vdots \\ w'_{k\ell} \end{bmatrix} \in \left((\Sigma \cup \{\Box\})^k\right)^*$$

where $w'_{ij} = w_{ij}$ for $j \leq |w_i|$ and $w'_{ij} = \Box$ otherwise.

Now, a relation $R \subseteq (\Sigma^*)^k$ is called *automatic* or *regular*, if $\{\, w_1 \otimes \cdots \otimes w_k : (w_1, \ldots, w_r) \in R \,\}$ is a regular language. In the sequel we do not distinguish between a relation on words and its encoding as a language.

As usual in mathematical logic, we consider *structures* $\mathfrak{A} = (A, R_1, R_2, \dots, f_1, f_2, \dots)$ where A is a non-empty set, called the *universe* of $\mathfrak{A}$, where each $R_i \subseteq A^{r_i}$ is a relation on A and each $f_j : A^{s_j} \to A$ is a function on A. The names of the relations and functions of $\mathfrak{A}$, together with their arities, form the *vocabulary* of $\mathfrak{A}$. We consider constants as functions of arity 0. A *relational* structure is a structure without functions. We can associate with every structure $\mathfrak{A}$ its *relational variant* which is obtained by replacing each function $f : A^s \to A$ by its graph $G_f := \{ (\bar{a}, b) \in A^{s+1} : f(\bar{a}) = b \}$.

Definition 2.1. A relational structure $\mathfrak{A}$ is *automatic* if there exist a regular language $L_\delta \subseteq \Sigma^*$ and a surjective function $\nu : L_\delta \to A$ such that the relation

$$L_\varepsilon := \{ (w, w') \in L_\delta \times L_\delta : \nu w = \nu w' \} \subseteq \Sigma^* \times \Sigma^*$$

and, for all predicates $R \subseteq A^r$ of $\mathfrak{A}$, the relations

$$L_R := \{ \bar{w} \in (L_\delta)^r : (\nu w_1, \dots, \nu w_r) \in R \} \subseteq (\Sigma^*)^r$$

are regular. An arbitrary (not necessarily relational) structure is automatic if and only if its relational variant is.

By an *automatic presentation* of a τ-structure $\mathfrak{A}$ we either mean a pair $(\nu, \mathfrak{d})$ consisting of the function $\nu : L_\delta \to A$ and a collection $\mathfrak{d} = (M_\delta, M_\varepsilon, (M_R)_{R \in \tau})$ of finite automata that recognise L_δ, L_ε, and L_R for all relations R of $\mathfrak{A}$, or we mean just the collection $\mathfrak{d}$ alone. (Note that $\mathfrak{d}$ determines the structure that it presents up to isomorphism.) An automatic presentation is called *deterministic* if all its automata are, and it is called injective if $L_\varepsilon = \{ (u, u) : u \in L_\delta \}$ (which implies that $\nu : L_\delta \to A$ is injective). We write $\mathrm{AutStr}[\tau]$ for the class of all automatic structures of vocabulary τ.

Examples. (1) All finite structures are automatic.

(2) Important examples of automatic structures are Presburger Arithmetic $(\mathbb{N}, +)$ and its expansions $\mathfrak{N}_p := (\mathbb{N}, +, |_p)$ by the relation

$$x \,|_p\, y \;:\text{iff } x \text{ is a power of } p \text{ dividing } y.$$

Using p-ary encodings (starting with the least significant digit) it is not difficult to construct automata recognizing equality, addition and $|_p$.

(3) Natural candidates for automatic structures are those consisting of words. (But note that free monoids with at least two generators do *not* have automatic presentations.) Fix some alphabet Σ and consider the structure $\mathit{Tree}(\Sigma) := (\Sigma^*, (\sigma_a)_{a \in \Sigma}, \preceq, \mathrm{el})$ where

$$\begin{aligned} \sigma_a(x) &:= xa, \\ x \preceq y &:\text{iff } \exists z (xz = y), \\ \mathrm{el}(x, y) &:\text{iff } |x| = |y|. \end{aligned}$$

Obviously, this structure is automatic as well.

The following two observations are simple, but useful.

(1) Every automatic structure admits an automatic presentation with alphabet $\{0, 1\}$ [2].

(2) Every automatic structure admits an injective automatic presentation [19].

Automatic Groups. The class of automatic structures that have been studied most intensively are automatic groups. Let $(G, \cdot)$ be a group and $S = \{s_1, \dots, s_m\} \subseteq G$ a set of semigroup generators of G. This means that each $g \in G$ can be written as a product $s_{i_1} \cdots s_{i_r}$ of elements of S and hence the canonical homomorphism $\nu : S^* \to G$ is surjective. The *Cayley graph* $\Gamma(G, S)$ of G with respect to S is the graph $(G, S_1, \dots, S_m)$ whose vertices are the group elements and where S_i is the set of pairs (g, h) such that $g \cdot s_i = h$. By definition $(G, \cdot)$ is *automatic* if there is a finite set S of semigroup generators and a regular language $L_\delta \subseteq S^*$ such that the restriction of ν to L_δ is surjective and provides an automatic presentation of $\Gamma(G, S)$. (That is, the inverse image of equality, $L_\varepsilon = \{ (w, w') \in L_\delta \times L_\delta : \nu w = \nu w' \}$, and $\nu^{-1}(S_i)$ (for $i = 1, \dots, m$) are regular).

Note that it is not the group structure $(G, \cdot)$ itself that is automatic in the sense of Definition 2.1, but the Cayley graph. There are many natural examples of automatic groups (see [9, 10]). The importance of this notion in computational group theory comes from the fact that an automatic presentation of a group yields (efficient) algorithmic solutions for computational problems that are undecidable in the general case.

Remark. By definition, if G is an automatic group, then for some set S of semigroup generators, the Cayley graph $\Gamma(G, S)$ is an automatic structure. Contrary to a claim in [19] it is not clear whether the converse holds. Indeed the definition of an automatic group requires that the function $\nu : L_\delta \to G$ is the restriction of the canonical homomorphism from S^* to G. The mere condition that $\Gamma(G, S)$ is an automatic structure does not seem to imply this.

ω-automatic structures. The notion of an automatic structure can be modified and generalised in a number of different directions (see [2, 19]). In particular, we obtain the interesting class ω-AutStr of ω-automatic structures. The definition is analogous to the one for automatic structures except that the elements of an ω-automatic structure are named by infinite words from some regular ω-language and the relations of the structure are recognisable by Büchi automata.

Examples. (1) All automatic structures are ω-automatic.

(2) The real numbers with addition, $(\mathbb{R}, +)$, and indeed the expanded structure $\mathfrak{R}_p := (\mathbb{R}, +, \leq, |_p, 1)$ are ω-automatic, where

$$x \,|_p\, y \;:\text{iff } \exists n, k \in \mathbb{Z} : x = p^n \text{ and } y = kx.$$

(3) The tree automatic structures $Tree(\Sigma)$ extend in a natural way to the (uncountable) ω-automatic structures $Tree^{\omega}(\Sigma) = (\Sigma^{\leq\omega}, (\sigma_a)_{a\in\sigma}, \preceq, \mathrm{el})$.

3. Model-checking and query-evaluation

In this section we study decidability and complexity issues for automatic structures. For a structure $\mathfrak{A}$ and a formula $\varphi(\bar{x})$, let $\varphi^{\mathfrak{A}} := \{\, \bar{a} : \mathfrak{A} \models \varphi(\bar{a}) \,\}$ be the relation (or query) defined by φ on $\mathfrak{A}$. Two fundamental algorithmic problems are

Model-checking: Given a (presentation of a) structure $\mathfrak{A}$, a formula $\varphi(\bar{x})$, and a tuple of parameters $\bar{a}$ in $\mathfrak{A}$, decide whether $\mathfrak{A} \models \varphi(\bar{a})$.

Query-evaluation: Given a presentation of a structure $\mathfrak{A}$ and some formula $\varphi(\bar{x})$, compute a presentation of $(\mathfrak{A}, \varphi^{\mathfrak{A}})$. That is, given a pair $(\nu, \mathfrak{d})$ representing $\mathfrak{A}$, construct an automaton that recognises $\nu^{-1}(\varphi^{\mathfrak{A}})$.

We first observe that all first-order queries on automatic structures are effectively computable. In fact, this is the case not only for first-order logic but also for formulae containing the quantifier $\exists^{\omega}$ meaning "there are infinitely many."

Proposition 3.1. *Given an injective presentation* $(\nu, \mathfrak{d})$ *of an automatic or* ω*-automatic structure* $\mathfrak{A}$ *and a formula* $\varphi(\bar{x}) \in \mathrm{FO}(\exists^{\omega})$ *one can effectively construct an automaton recognising* $\nu^{-1}(\varphi^{\mathfrak{A}})$.

Proof. For FO-formulae this follows readily from classical results on the closure properties of regular (ω-)languages.

In case of automatic structures the quantifier $\exists^{\omega}$ can be handled using a pumping argument. Consider for simplicity the formula $\exists^{\omega} x \psi(x,y)$. There are infinitely many x satisfying ψ iff for any m there are infinitely many elements x whose encoding is at least m symbols longer than that of y. If we take m to be the number of states of the automaton for ψ then, by the Pumping Lemma, the last condition is equivalent to the existence of at least one such x. Thus $\exists^{\omega} x \psi(x,y) \equiv \exists x(\psi(x,y) \wedge$ "x is long enough") for which we can obviously construct an automaton. Note that the injectivity of $(\nu, \mathfrak{d})$ ensures that each of the infinitely many words encodes a different element of $\mathfrak{A}$.

For ω-automatic structures the proof is more involved. First we introduce some notation. By $v[i,k)$ we denote the factor $v_i \dots v_{k-1}$ of $v = v_0 v_1 \dots \in \Sigma^{\omega}$. Similarly, $v[i,\omega)$ is equal to $v_i v_{i+1} \dots$, and $v[i] := v[i,i+1)$.

Let M be a deterministic ω-automaton with s states recognising the language $L(M) \subseteq \Gamma^{\omega} \otimes \Sigma^{\omega}$. For $w \in \Gamma^{\omega}$ let $V(w) := \{\, v \in \Sigma^{\omega} \mid w \otimes v \in L(M) \,\}$.

Let $v, w \in \Sigma^{\omega}$ and define $v \approx^* w$ iff $v[n,\omega) = w[n,\omega)$ for some n. Let $[v]_* := \{\, v' \in V(w) \mid v' \approx^* v \,\}$ be the $\approx^*$-class of v in $V(w)$.

Claim. $V(w)$ is infinite if and only if there is some $v \in \Sigma^{\omega}$ such that $[v]_* \in V(w)/{\approx^*}$ is infinite.

Proof. $(\Leftarrow)$ is trivial and $(\Rightarrow)$ is proved by showing that $V/{\approx^*}$ contains at most s finite $\approx^*$-classes.

Assume there are $v_0, \dots, v_s \in V(w)$ belonging to different finite $\approx^*$-classes. Denote the run (sequence of states) of M on $w \otimes v_i$ by ρ_i. Define $I_{ij} := \{\, k < \omega \mid \rho_i[k] = \rho_j[k] \,\}$. Since there are only s states, for each $k < \omega$ there have to be indices i, j such that $k \in I_{ij}$, i.e., $\bigcup_{i,j} I_{ij} = \omega$. Thus, at least one I_{ij} is infinite. For each $[v_i]_*$ there is a position n_i such that $v[n_i,\omega) = v'[n_i,\omega)$ for all $v, v' \in [v_i]_*$. Let m be the maximum of $n_0, \dots, n_s$. Fix i, j such that I_{ij} is infinite. Since $v_i \not\approx^* v_j$ there is a position $m' > m$ such that $v_i[m,m') \neq v_j[m,m')$. Choose some $m'' \in I_{ij}$ with $m'' \geq m'$. Let $u := v_i[0,m) v_j[m,m'') v_i[m'',\omega)$. Then, $w \otimes v_i \in L(M)$ iff $w \otimes u \in L(M)$ which implies that $u \in [v_i]_*$. But $u[m,\omega) \neq v_i[m,\omega)$ in contradiction to the choice of m. □

To finish the proof let $\varphi(\bar{x}) := \exists^{\omega} y \psi(\bar{x},y)$ and $\mathfrak{A}$ ω-automatic. One can express that $[v]_*$ is finite by

$$\mathrm{finite}(\bar{x}, v) := \exists n \forall v' (\psi(\bar{x}, v') \wedge v \approx^* v' \rightarrow \mathrm{equal}(v, v', n)),$$

where

$$\mathrm{equal}(v, v', n) := n = 1^i 0^{\omega} \wedge v[i,\omega) = v'[i,\omega).$$

Clearly, $\approx^*$ and equal can be recognised by ω-automata. By the claim above,

$$\varphi(\bar{x}) \equiv \exists v (\psi(\bar{x}, v) \wedge \neg \mathrm{finite}(\bar{x}, v)).$$

Hence, we can construct an automaton recognising $\varphi^{\mathfrak{A}}$. □

Corollary 3.2. *The* $\mathrm{FO}(\exists^{\omega})$*-theory of any automatic structure and of any* ω*-automatic with injective presentation is decidable.*

As an immediate consequence we conclude that full arithmetic $(\mathbb{N}, +, \cdot)$ is neither automatic, nor ω-automatic. For most of the common extensions of first-order logic used in finite model theory, such as transitive closure logics, fixed point logics, monadic second-order logic, or first-order logic with counting, the model-checking problem on automatic structures becomes undecidable.

Complexity. We will consider the *structure*, *expression*, and *combined complexity* of model-checking and query-evaluation, where, respectively, the dependence on the input structure, formula, and of both parts is measured whereas the remaining parts of the input are considered fixed.

Of course, the complexity of these problems may very much depend on how automatic structures are presented. We focus here on presentations by *deterministic* automata because these admit boolean operations to be performed in polynomial time, whereas for nondeterministic automata, complementation may cause an exponential blowup.

In the following we always assume that the vocabulary of the given automatic structures and the alphabet of the automata we deal with are fixed. Furthermore the vocabulary is assumed to be relational when not stated otherwise. For a (deterministic) presentation $\mathfrak{d}$ of an automatic structure, we denote by $|\mathfrak{d}|$ the maximal size of the automata in $\mathfrak{d}$, and for an automatic presentation $(\nu, \mathfrak{d})$ of the structure $\mathfrak{A}$, we define $\lambda^{\mathfrak{d}} : A \to \mathbb{N}$ to be the function

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

mapping each element of $\mathfrak{A}$ to the length of its shortest encoding. Finally, let $\lambda^{\mathfrak{d}}(a_1, \dots, a_r)$ be an abbreviation for $\max\{\, \lambda^{\mathfrak{d}}(a_i) : i = 1, \dots, r \,\}$.

While we have seen above that query-evaluation and model-checking for first-order formulae are effective on AutStr, the complexity of these problems is non-elementary, i.e., it exceeds any fixed number of iterations of the exponential function. This follows immediately from the fact the the complexity of $\mathrm{Th}(\mathfrak{N}_p)$ is non-elementary (see [11]).

Proposition 3.3. *There exist automatic structures such that the expression complexity of the model-checking problem is non-elementary.*

It turns out that model-checking and query-evaluation for quantifier-free and existential formulae are still—to some extent—tractable. As usual, let Σ_0 and Σ_1 denote, respectively the class of quantifier-free and the class of existential first-order formulae.

Theorem 3.4. (i) *Given a presentation $\mathfrak{d}$ of a relational structure $\mathfrak{A} \in$ AutStr, a tuple $\bar{a}$ in $\mathfrak{A}$, and a quantifier-free formula $\varphi(\bar{x}) \in$ FO, the model-checking problem for $(\mathfrak{A}, \bar{a}, \varphi)$ is in*

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

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

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

Proof. (i) To decide whether $\mathfrak{A} \models \varphi(\bar{a})$ holds, we need to know the truth value of each atom appearing in φ. Then, all what remains is to evaluate a boolean formula which can be done in $\mathrm{DTIME}\big[O\big(|\varphi|\big)\big]$ and $\mathrm{ATIME}\big[O\big(\log|\varphi|\big)\big] \subseteq \mathrm{DSPACE}\big[O\big(\log|\varphi|\big)\big]$ (see [5]). The value of an atom $R\bar{x}$ can be calculated by simulating the corresponding automaton on those components of $\bar{a}$ which belong to the variables appearing in $\bar{x}$. The naïve algorithm to do so uses time $O\big(\lambda^{\mathfrak{d}}(\bar{a})\,|\mathfrak{d}|\log|\mathfrak{d}|\big)$ and space $O\big(\log|\mathfrak{d}| + \log\lambda^{\mathfrak{d}}(\bar{a})\big)$.

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

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

(ii) We present a reduction of the LOGSPACE-complete problem DETREACH, reachability by deterministic paths, (see e.g. [18]) to the model-checking problem. Given a graph $\mathfrak{G} = (V, E, s, t)$ we construct the automaton $M = (V, \{0\}, \Delta, s, \{t\})$ with

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

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

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

where $\mathfrak{A} = (B, P)$ is the structure with the presentation $(\{0\}^*, L(M))$. A closer inspection reveals that the above transformation can be defined in first-order logic.

(iii) Evaluation of boolean formulae is ALOGTIME-complete (see [5]). □

For most questions we can restrict attention to relational vocabularies and replace functions by their graphs at the expense of introducing additional quantifiers. When studying quantifier-free formulae we will not want do to this and hence need to consider the case of quantifier-free formulae with function symbols sparately. This class is denoted $\Sigma_0 + \mathrm{fun}$.

Lemma 3.5 (Epstein et al. [9])**.** *Given a tuple $\overline{w}$ of words over Σ, and an automaton $\mathfrak{A} = (Q, \Sigma, \delta, q_0, F)$ recognising the graph of a function f, the calculation of $f(\overline{w})$ is in*

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

Proof. Using the ideas of [9] one can implement an algorithms with those complexity bounds. See [2] for details. □

Theorem 3.6. (i) *Let τ be a vocabulary which may contain functions. Given the presentation $\mathfrak{d}$ of a structure*

$\mathfrak{A} \in \mathrm{AutStr}[\tau]$, *a tuple* $\bar{a}$ *in* $\mathfrak{A}$, *and a quantifier-free formula* $\varphi(\bar{x}) \in \mathrm{FO}[\tau]$, *the model-checking problem for* $(\mathfrak{A}, \bar{a}, \varphi)$ *is in*

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

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

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

Proof. (i) Our algorithm proceeds in two steps. First the values of all functions appearing in φ are calculated starting with the innermost one. Then all functions can be replaced by their values and a formula containing only relations remains which can be evaluated as above. We need to evaluate at most $|\varphi|$ functions. If they are nested the result can be of length $|\varphi|\,|\mathfrak{d}| + \lambda^{\mathfrak{d}}(\bar{a})$. This yields the bounds given above.

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

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

Remark. Theorem 3.6 says that, on any fixed automatic structure, quantifier-free formulae can be evaluated in quadratic time. This extends the result of [9] that the word problem for every automatic group is solvable in quadratic time. Indeed, for every automatic group G generated by $s_1, \dots, s_m$, the structure $(G, e, g \mapsto gs_1, \dots, g \mapsto gs_m)$ is just a functional way of presenting the Cayley graph and therefore automatic. Each instance of the word problem is described by a quantifier-free sentence (a term equation) on this structure.

Theorem 3.7. (i) *Given a presentation* $\mathfrak{d}$ *of a structure* $\mathfrak{A} \in \mathrm{AutStr}$, *a tuple* $\bar{a}$ *in* $\mathfrak{A}$, *and a formula* $\varphi(\bar{x}) \in \Sigma_1$, *the model-checking problem for* $(\mathfrak{A}, \bar{a}, \varphi)$ *is in*

$$\mathrm{NTIME}\big[O\big(|\varphi|\,|\mathfrak{d}|\,\lambda^{\mathfrak{d}}(\bar{a}) + |\mathfrak{d}|^{O(|\varphi|)}\big)\big] \text{ and}$$
$$\mathrm{NSPACE}\big[O\big(|\varphi|\,\big(|\mathfrak{d}| + \log|\varphi|\big) + \log\lambda^{\mathfrak{d}}(\bar{a})\big)\big].$$

(ii) *The structure complexity of model-checking for* Σ_1*-formulae is* NPTIME*-complete with respect to* $\leq_{\mathrm{tt}}^{\mathrm{p}}$*-reductions.*

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

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

The algorithm determines which atoms appear in φ and simulates the product automaton constructed from the automata for those relations. At each step the symbol for the quantified variables is guessed nondeterministically. Note that the values of those variables may be longer than the input so we have to continue the simulation after reaching its end for at most the cardinality of the state-space number of steps. Since this cardinality is $O\big(|\mathfrak{d}|^{|\varphi|}\big)$ a closer inspection of the algorithm yields the given bounds.

(ii) We reduce the NPTIME-complete non-universality problem for nondeterministic automata over a unary alphabet (see [21, 17]), given such an automaton check whether it does not recognise the language 0^*, to the given problem. This reduction is performed in two steps. First the automaton must be simplified and transformed into a deterministic one, then we construct an automatic structure and a formula $\varphi(x)$ such that $\varphi(a)$ holds for several values of a if and only if the original automaton recognises 0^*. As the model-checking has to be performed for more than one parameter this yields not a many-to-one but a truth-table reduction.

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

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

	Structure-Complexity	Expression-Complexity
Model-Checking		
Σ_0	LOGSPACE-complete	ALOGTIME-complete
$\Sigma_0 + \text{fun}$	NLOGSPACE	PTIME-complete
Σ_1	NPTIME-complete	PSPACE-complete
Query-Evaluation		
Σ_0	LOGSPACE	PSPACE
Σ_1	PSPACE	EXPSPACE

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

$$\begin{aligned} \mathfrak{B} \models \exists y R 0^{kn} y \quad &\text{iff} \quad 0^{kn} \otimes y \in L(M'') \\ &\text{iff} \quad 0^n \in L(M). \end{aligned}$$

It follows that

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

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

(iii) is shown by coding computations of Turing machines. The proof can be found in [2]. $\square$

We now turn to the query-evaluation problem for these formula classes.

Theorem 3.8. *Given a presentation $\mathfrak{d}$ of a structure $\mathfrak{A} \in$ AutStr and a formula $\varphi(\bar{x})$, an automaton representing $\varphi^{\mathfrak{A}}$ can be computed*

(i) *in time $O\big(|\mathfrak{d}|^{O(|\varphi|)}\big)$ and space $O\big(|\varphi| \log |\mathfrak{d}|\big)$ in the case of quantifier-free $\varphi(\bar{x})$, and*

(ii) *in time $O\big(2^{|\mathfrak{d}|^{O(|\varphi|)}}\big)$ and space $O\big(|\mathfrak{d}|^{O(|\varphi|)}\big)$ in the case of existential formulae $\varphi(\bar{x})$.*

In particular, the structure complexity of query-evaluation is in LOGSPACE *for quantifier-free formulae and in* PSPACE *for existential formulae. The expression complexity is in* PSPACE *for quantifier-free formulae and in* EXPSPACE *for existential formulae.*

Proof. Enumerate the state space of the product automaton and output the transition function. $\square$

4. Structures that are not automatic

To prove that a structure is automatic, we just have to find a suitable presentation. But how can we prove that a structure is not automatic? The main difficulty is that a priori, nothing is known about how elements of an automatic structure are named by words of the regular language.[1]

Besides the two obvious criteria, namely that automatic structures are countable and that their first-order theory is decidable, not much is known. The only non-trivial criterion that is available at present use growth rates for the length of the encodings of elements of definable sets.

Proposition 4.1 (Elgot and Mezei [8])**.** *Let $\mathfrak{A}$ be an automatic structure with injective presentation $(\nu, \mathfrak{d})$, and let $f : A^n \to A$ be a function of $\mathfrak{A}$. Then there is a constant m such that $\lambda^{\mathfrak{d}}(f(\bar{a})) \leq \lambda^{\mathfrak{d}}(\bar{a}) + m$ for all $\bar{a} \in A^n$.*

The same is true if we replace f by a relation R where for all $\bar{a}$ there are only finitely many values b such that $R\bar{a}b$ holds.

This result deals with a single application of a function or relation. In the remaining part of this section we will study the effect of applying functions iteratively, i.e., we will consider some definable subset of the universe and calculate upper bounds on the length of the encodings of elements in the substructure generated by it. First we need bounds for the (encodings of) elements of some definable subsets. The following lemma follows easily from classical results in automata theory (see, e.g., [7, Proposition V.1.1]).

Lemma 4.2. *Let $\mathfrak{A}$ be a structure in* AutStr *with presentation $\mathfrak{d}$, and let B be an* $\mathrm{FO}(\exists^{\omega})$*-definable subset of A. Then $\lambda^{\mathfrak{d}}(B)$ is a finite union of arithmetical progressions.*

In the process of generating a substructure we have to count the number of applications of functions.

Definition 4.3. Let $\mathfrak{A} \in$ AutStr with presentation $\mathfrak{d}$, let $f_1, \dots, f_r$ be finitely many operations of $\mathfrak{A}$ with arities $r_1, \dots, r_r$, respectively, and let $E = \{e_1, e_2, \dots\}$ be some subset of A with $\lambda^{\mathfrak{d}}(e_1) \leq \lambda^{\mathfrak{d}}(e_2) \leq \cdots$. Then $G_n(E)$, the n^{th} *generation* of E, is defined inductively by

$$\begin{aligned} G_1(E) &:= \{e_1\}, \\ G_n(E) &:= G_{n-1}(E) \cup \{e_n\} \\ &\qquad \cup \big\{ f_i(\bar{a}) \bigm| \bar{a} \in G_{n-1}^{r_i}(E),\, 1 \leq i \leq r \big\}. \end{aligned}$$

Putting everything together we obtain the following result. The case of finitely generated substructures already appeared in [19].

Proposition 4.4. *Let $\mathfrak{d}$ an injective presentation of an automatic structure $\mathfrak{A}$, let $f_1, \dots, f_r$ be finitely many definable operations on $\mathfrak{A}$ and let E be a definable subset of A. Then there is a constant m such that $\lambda^{\mathfrak{d}}(a) \leq mn$ for all $a \in G_n(E)$. In particular, $|G_n(E)| \leq |\Sigma|^{mn+1}$ where Σ is the alphabet of $\mathfrak{d}$.*

[1] In the case of automatic groups, where the naming function is fixed, more techniques are available such as the k-fellow traveller property, see [9].

The proof consists of a simple induction on n.

Theorem 4.5. *None of the following structures has an automatic presentation.*

(i) *Any trace monoid $\mathfrak{M} = (M, \cdot)$ with at least two non-commuting generators a and b.*

(ii) *Any structure $\mathfrak{A}$ in which a pairing function f can be defined.*

(iii) *The divisibility poset $(\mathbb{N}, |)$.*

(iv) *Skolem Arithmetic $(\mathbb{N}, \cdot)$.*

Proof. (i) We show that $\{a,b\}^{\leq 2^n} \subseteq G_{n+1}(a,b)$ by induction on n. We have $\{a,b\} \subseteq \{a,aa,b\} = G_2(a,b)$ for $n = 1$, and for $n > 1$

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

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

(ii) is analogous to (i), and (iv) immediately follows from (iii) as the divisibility relation is definable in $(\mathbb{N}, \cdot)$.

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

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

the set of powers of some prime

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

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

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

The least common multiple of two numbers is

$$\begin{aligned} \operatorname{lcm}(x,y) = z \;:\text{iff}\; & x \,|\, z \wedge y \,|\, z \\ & \wedge \neg \exists u (u \neq z \wedge x \,|\, u \wedge y \,|\, u \wedge u \,|\, z). \end{aligned}$$

For every $n \in \mathbb{N}$ there are only finitely many m with Snm. Therefore S satisfies the conditions of Proposition 4.1. Consider the set generated by P via S and lcm, and let $\gamma(n) := |G_n(P)|$ be the cardinality of $G_n(P)$. If $(\mathbb{N}, |)$ is in AutStr then $(\mathbb{N}, |, P, Q, S) \in$ AutStr and $\gamma(n) \in 2^{O(n)}$ by Proposition 4.4. Let $P = \{p_1, p_2, \dots\}$. For $n = 1$ we have $G_1(P) = \{p_1\}$. Generally, $G_n(P)$ consists of

(1) numbers of the form $p_1^{k_1}$,

(2) numbers of the form $p_2^{k_2} \cdots p_n^{k_n}$, and

(3) numbers of a mixed form.

In n steps we can create

(1) $p_1, \dots, p_1^n$ (via S),

(2) $\gamma(n-1)$ numbers with $k_1 = 0$, and

(3) $\gamma(n-2) - 1$ numbers of a mixed form for every $0 < k_1 < n$ (via lcm).

All in all we obtain

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

Contradiction. □

Remark. (1) Since it is easy to construct a *tree-automatic* presentation of Skolem Arithmetic this result implies that the class of structures with tree-automatic presentation strictly includes the class of automatic structures (see [2]).

(2) The structure $(\mathbb{N}, \perp)$ where $\perp$ stands for having no common divisor is automatic.

5. Characterising automatic structures via interpretations

Interpretations are important in mathematical logic, for model-theory in particular. They are used to define a copy of a structure inside another one, and thus permit to transfer definability, decidability, and complexity results among theories.

Definition 5.1. A *(k-dimensional) interpretation* of a relational σ-structure $\mathfrak{A} = (A, R_1, \dots, R_m)$ in a τ-structure $\mathfrak{B}$ is given by a sequence

$$I = \langle \delta(\bar{x}), \varepsilon(\bar{x}, \bar{y}), \varphi_{R_1}(\bar{x}_1, \dots, \bar{x}_{r_1}), \dots \rangle$$

of first-order formulae of vocabulary τ (where each tuple $\bar{x}$, $\bar{y}$, $\bar{x}_i$ consists of k variables), provided that there exists a surjective map $h : \delta^{\mathfrak{B}} \rightarrow A$, called the *coordinate map* of the interpretation such that the following hold:

(i) For all $\bar{b}, \bar{c} \in \delta^{\mathfrak{B}}$

$$\mathfrak{B} \models \varepsilon(\bar{b}, \bar{c}) \text{ iff } h(\bar{b}) = h(\bar{c}),$$

(ii) for every relation R_j of $\mathfrak{A}$ and all $\bar{b}_1, \dots, \bar{b}_{r_j} \in \delta^{\mathfrak{B}}$

$$\mathfrak{B} \models \varphi_R(\bar{b}_1, \dots, \bar{b}_{r_j}) \text{ iff } \big(h(\bar{b}_1), \dots, h(\bar{b}_r)\big) \in R.$$

That is, the formula $\varepsilon(\bar{x},\bar{y})$ defines a congruence on the structure $\left(\delta^{\mathfrak{B}},\varphi^{\mathfrak{B}}_{R_1},\dots,\varphi^{\mathfrak{B}}_{R_m}\right)$ such that h is an isomorphism from the quotient structure $\left(\delta^{\mathfrak{B}},\varphi^{\mathfrak{B}}_{R_1},\dots,\varphi^{\mathfrak{B}}_{R_m}\right)/\varepsilon^{\mathfrak{B}}$ to $\mathfrak{A}$. In the case that $\mathfrak{A}$ *is* this quotient structure itself (rather than just being isomorphic to it) we say that $\mathfrak{A}$ is *definable* in $\mathfrak{B}$. Obviously, $\mathfrak{A}$ is definable in $\mathfrak{B}$ if and only if there is an interpretation of $\mathfrak{A}$ in $\mathfrak{B}$ whose coordinate map is the canonical projection, mapping every tuple $\bar{b} \in \delta^{\mathfrak{B}}$ to its equivalence class $\bar{b}/\varepsilon$.

If $\mathfrak{A}$ is a structure including not only relations but also functions then, by definition, an interpretation of $\mathfrak{A}$ in $\mathfrak{B}$ is an interpretation of the relational variant of $\mathfrak{A}$ (where functions are replaced by their graphs) in $\mathfrak{B}$.

We write $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{B}$ to denote that there exists an interpretation of $\mathfrak{A}$ in $\mathfrak{B}$. If both $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{B}$ and $\mathfrak{B} \leq_{\mathrm{FO}} \mathfrak{A}$ we say $\mathfrak{A}$ and $\mathfrak{B}$ are *mutually interpretable*.

Examples. (1) Recall that we write $a\,|_p\, b$ to denote that a is a power of p dividing b. Let $V_p : \mathbb{N} \to \mathbb{N}$ be the function that maps each number to the largest power of p dividing it. It is very easy to see that the structures $(\mathbb{N},+,|_p)$ and $(\mathbb{N},+,V_p)$ are mutually interpretable. Indeed we can define the statement $x = V_p(y)$ in $(\mathbb{N},+,|_p)$ by the formula $x\,|_p\,y \wedge \forall z(z\,|_p\,y \to z\,|_p\,x)$. In the other direction, $V_p(x) = x \wedge \exists z(x+z = V_p(y))$ is a definition of $x\,|_p\,y$.

(2) For every $p \in \mathbb{N}$ we write *Tree*(p) for the tree structure *Tree*$(\{0,\dots,p-1\})$. The structures $\mathfrak{N}_p$ and *Tree*(p) are mutually interpretable, for each $p \geq 2$ (see [2, 11]).

Observe that Proposition 3.1 implies an interesting closure property for AutStr and ω-AutStr.

Proposition 5.2. *The classes of automatic and ω-automatic structures are closed under interpretations, i.e., if $\mathfrak{B}$ is (ω-)automatic and $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{B}$, then so is $\mathfrak{A}$.*

Corollary 5.3. *The classes of automatic, resp. ω-automatic, structures are closed under* (i) *extensions by definable relations,* (ii) *factorisations by definable congruences,* (iii) *substructures with definable universe, and* (iv) *finite powers.*

The model-theoretic characterisation of automatic structures is given in the following theorem. It states that the structure $\mathfrak{N}_p$ (and *Tree*(p)) is *complete* for AutStr, i.e., a structure $\mathfrak{A}$ belongs to AutStr if and only if $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{N}_p$.

Theorem 5.4. *For every structure $\mathfrak{A}$, the following are equivalent:*

(i) $\mathfrak{A}$ *is automatic.*

(ii) $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{N}_p$ *for some (and hence all)* $p \geq 2$.

(iii) $\mathfrak{A} \leq_{\mathrm{FO}}$ *Tree*(p) *for some (and hence all)* $p \geq 2$.

Proof. The facts that (ii) and (iii) are equivalent and that they imply (i) follow immediately from the mutual interpretability of $\mathfrak{N}_p$ and *Tree*(p), from the fact that these structures are automatic, and from the closure of automatic structures under interpretations.

It remains to show that every automatic structure is interpretable in $\mathfrak{N}_p$ (or *Tree*(p)). Suppose that $\mathfrak{d}$ is an automatic presentation of $\mathfrak{A}$ with alphabet $[p] := \{0,\dots,p-1\}$ for some $p \geq 2$ (without loss of generality, we could take $p = 2$). For every word $w \in [p]^*$, let $\mathrm{val}(w)$ be the natural number whose p-ary encoding is w, i.e., $\mathrm{val}(w) := \sum_{i<|w|} w_i p^i$. By a classical result, sometimes called the Büchi-Bruyère Theorem, a relation $R \subseteq \mathbb{N}^k$ is first-order definable in $(\mathbb{N},+,V_p)$ if and only if

$$\{\,(\mathrm{val}^{-1}(x_1),\dots,\mathrm{val}^{-1}(x_k)) : (x_1,\dots,x_k) \in R\,\}$$

is regular. (See [4] for a proof of this fact and for more information on the relationship between automata and definability in expansions of Presburger Arithmetic.) The formulae that define in this sense the regular language and the regular relations in an automatic presentation of $\mathfrak{A}$ provide an interpretation of $\mathfrak{A}$ in $(\mathbb{N},+,V_p)$. Hence also $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{N}_p$. $\square$

For automatic groups we are not free to change the coordinate map, therefore the arguments used above give a characterisation in terms of definability rather than interpretability.

Theorem 5.5. *$(G,\cdot)$ is an automatic group if and only if there exists a finite set $S \subseteq G$ of semigroup generators such that $\Gamma(G,S)$ is* FO*-definable in Tree*(S).

We now turn to ω-automatic structures. To provide a similar characterisation we can use an equivalent of the Büchi-Bruyère Theorem for encodings of ω-regular relations. One such result has been obtained recently by Boigelot, Rassart and Wolper [3]. Using natural translations between ω-words over $[p]$ and real numbers, they prove that a relation over $[p]^\omega$ can be recognised by a Büchi automaton if an only if its translation is first-order definable in the structure $(\mathbb{R},+,<,\mathbb{Z},X_p)$ where $X_p \subseteq \mathbb{R}^3$ is a relation that explicitly represents the translation between $[p]^\omega$ and $\mathbb{R}$. $X_p(x,y,z)$ holds iff there exists a representation of x by a word in $[p]^\omega$ such that the digit at the position specified by y is z. A somewhat unsatisfactory aspect of this result is the assumption that the encoding relation X_p must be given as a basic relation of the structure. It would be preferable if more natural expansions of the additive real group $(\mathbb{R},+)$ could be used instead.

We show here that this is indeed possible if, as in the case of $\mathfrak{N}_p$, we use a restricted variant of the divisibility relation. Recall that the structures $\mathfrak{R}_p$ and *Tree*$^\omega(p)$ (introduced at the end of Section 2) are ω-automatic. As a first step we show that the behaviour of Büchi automata recognising regular relations over $[p]^\omega$ can be simulated by first-order formulae in *Tree*$^\omega(p)$. Second we show that *Tree*$^\omega(p)$ and $\mathfrak{R}_p$ are mutually interpretable. As a result we obtain the following model-theoretic characterisation of ω-automatic structures.

Theorem 5.6. *For every structure $\mathfrak{A}$, the following are equivalent*

(i) $\mathfrak{A}$ *is* ω*-automatic.*

(ii) $\mathfrak{A} \leq_{\mathrm{FO}} \mathfrak{R}_p$ *for some (and hence all)* $p \geq 2$.

(iii) $\mathfrak{A} \leq_{\mathrm{FO}}$ *Tree*$^{\omega}(p)$ *for some (and hence all)* $p \geq 2$.

Proof. In order to construct interpretations of *Tree*$^{\omega}(p)$ in $\mathfrak{R}_p$ and vice versa we define formulae which allow to access the digits of, respectively, some number in $\mathfrak{R}_p$ and some word in *Tree*$^{\omega}(p)$. In the later case we set

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

which states that the digit of x at position $|y|$ is k. For $\mathfrak{R}_p$ the situation is more complicated as some real numbers admit two encodings. The following formula describes that there is one encoding of x such that the digit at position y is k. (This corresponds to the predicate X of [3].)

$$\begin{aligned}\mathrm{dig}_k(x,y) := \exists s \exists t(\,|x| = s + k\cdot y + t \wedge p\cdot y\,|_p\, s \\ \wedge\, 0 \leq s \wedge 0 \leq t \leq y)\end{aligned}$$

For $\mathfrak{R}_p \leq_{\mathrm{FO}}$ *Tree*$^{\omega}(p)$ we represent each number as a pair of words. The first one is finite and encodes the integer part, the other one is infinite and contains the fractional part. In the other direction we map finite words $a_1 \cdots a_r \in [p]^*$ to the interval $[2,3]$ via

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

Infinite words $a_1 a_2 \cdots \in [p]^{\omega}$ are mapped to two intervals $[-1,0]$ and $[0,1]$ via

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

This is necessary because some words, e.g., $0(p-1)^{\omega}$ and 10^{ω}, would be mapped to the same number otherwise. Now the desired interpretations can be constructed easily using the formulae dig_k defined above.

It remains to prove that if $R \subseteq ([p]^*)^n$ is ω-regular then it is definable in *Tree*$^{\omega}(p)$. Let $M = (Q,[p]^n,\Delta,q_0,F)$ be a Büchi-automaton for R. W.l.o.g. assume $Q = [p]^m$ for some m and $q_0 = (0,\dots,0)$. We prove the claim by constructing a formula $\psi_M(\bar{x}) \in \mathrm{FO}$ stating that there is a successful run of M on $x_1 \otimes \cdots \otimes x_n$. The run is encoded by a tuple $(q_1,\dots,q_m) \in ([p]^{\omega})^m$ of ω-words such that the symbols of $q_1,\dots,q_m$ at some position equal $k_1,\dots,k_m$ iff the automaton is in state $(k_1,\dots,k_m)$ when scanning the input symbol at that position. $\psi_M(\bar{x})$ has the form

$$\begin{aligned}\exists q_1 \cdots \exists q_m[\mathrm{ADM}(\bar{q},\bar{x}) \wedge \mathrm{START}(\bar{q},\bar{x}) \\ \wedge\, \mathrm{RUN}(\bar{q},\bar{x}) \wedge \mathrm{ACC}(\bar{q},\bar{x})]\end{aligned}$$

where the admissibility condition $\mathrm{ADM}(\bar{x},\bar{q})$ states that all components of $\bar{x}$ and $\bar{q}$ are infinite, $\mathrm{START}(\bar{x},\bar{q})$ says that the first state is $\bar{0}$, $\mathrm{ACC}(\bar{x},\bar{q})$ that some final state appears infinitely often, and $\mathrm{RUN}(\bar{x},\bar{q})$ ensures that all transitions are correct.

Define the following auxiliary formulae. To access the digits of a tuple of words at some position we define $\mathrm{Sym}_{\bar{a}}(\bar{x},z) := \bigwedge_i \mathrm{dig}_{a_i}(x_i,z)$, and to characterise the ω-words of $[p]^{\leq\omega}$ we set

$$\mathrm{Inf}(x) := \forall y(x \preceq y \rightarrow x = y).$$

ADM and START are defined as

$$\begin{aligned}\mathrm{ADM}(\bar{q},\bar{x}) &:= \bigwedge_{i=1}^{m} \mathrm{Inf}(q_i) \wedge \bigwedge_{i=1}^{n} \mathrm{Inf}(x_i),\\ \mathrm{START}(\bar{q},\bar{x}) &:= \mathrm{Sym}_{\bar{0}}(\bar{q},\varepsilon),\end{aligned}$$

RUN states that at every position a valid transition is used

$$\begin{aligned}&\mathrm{RUN}(\bar{q},\bar{x}) :=\\ &\quad \forall z \bigvee_{(\bar{k},\bar{a},\bar{k}')\in\Delta} \big(\mathrm{Sym}_{\bar{k}}(\bar{q},z) \wedge \mathrm{Sym}_{\bar{a}}(\bar{x},z)\\ &\qquad\qquad\qquad \wedge \mathrm{Sym}_{\bar{k}'}(\bar{q},\sigma_0 z)\big),\end{aligned}$$

and ACC says that there is one final state which appears infinitely often in $\bar{q}$

$$\mathrm{ACC}(\bar{q},\bar{x}) := \bigvee_{\bar{k}\in F} \forall z \exists z' \big(|z'| > |z| \wedge \mathrm{Sym}_{\bar{k}}(\bar{q},z')\big). \qquad \square$$

6. Composition of structures

The composition method developed by Feferman and Vaught, and by Shelah considers compositions (products and sums) of structures according to some index structure and allows one to compute—depending on the type of composition—the first-order or monadic second-order theory of the whole structure from the respective theories of its components and the monadic theory of the index structure.

The characterisation given in the previous section can be used to prove closure of automatic structures under such compositions of finitely many structures (see [23, 13, 16]). A generalised product—as it is defined below—is a generalisation of a direct product, a disjoint union, and an ordered sum. We will prove that given a finite sequence $(\mathfrak{A}_i)_i$ of structures which belong to some class $\mathcal{K}$ containing a complete structure, all their generalised products are members of $\mathcal{K}$ as well.

The definition of such a product is a bit technical. Its relations are defined in terms of the types of the components of its elements. The *atomic n-type* $\mathrm{atp}_{\mathfrak{A}}(\bar{a})$ of a tuple $(a_0,\dots,a_{n-1})$ in a structure $\mathfrak{A}$ is the conjunction of all atomic and negated atomic formulae $\varphi(\bar{x})$ such that $\varphi(\bar{a})$ holds in $\mathfrak{A}$.

Let us first look at how a direct product and an ordered sum can be defined using types.

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

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

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

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

Definition 6.1. Let $\tau = \{R_0, \dots, R_r\}$ be a finite relational vocabulary, r_j the arity of R_j, and $\hat{r} := \max\{r_0, \dots, r_r\}$. Let $(\mathfrak{A}_i)_{i \in I}$ be a sequence of τ-structures, and $\mathfrak{I}$ be an arbitrary relational σ-structure with universe I.

Fix for each $k \le \hat{r}$ an enumeration $\{\tau_0^k, \dots, \tau_{t_k}^k\}$ of the atomic k-types and set

$$\begin{aligned} \sigma_k := \sigma & \cup \{D_0, \dots, D_{k-1}\} \\ & \cup \{ T_l^m \mid m \le k, l \le t_m \}. \end{aligned}$$

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

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

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

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

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

Example. (continued)

(1) For the direct product of $\mathfrak{A}_0 \times \mathfrak{A}_1$ we would set $\mathfrak{I} := (I)$ with $I = \{0,1\}$, $D := \{(1,1)\}$, and

$$\beta := \bigvee_{l \in L} T_l^2 0 \wedge \bigvee_{l \in L} T_l^2 1,$$

where L is the set of atomic types containing the formula Rx_0x_1.

(2) In this case we would set $\mathfrak{I} := (I)$ with $I = \{0,1\}$, $D := \{(1,0),(0,1)\}$, and

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

where L is the set of atomic types containing the formula $x_0 < x_1$.

Theorem 6.2. *Let τ be a finite relational vocabulary, and $\mathcal{K}$ a class of τ-structures containing all finite τ-structures and a structure $\mathfrak{C}$ which is complete for $\mathcal{K}$ with regard to many-dimensional* FO*-interpretations.*

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

Proof. Let $\tau = \{R_0, \dots, R_r\}$. W.l.o.g. assume that $I = \{0, \dots, |I|-1\}$ and that $\mathfrak{C}$ contains constants 0 and 1. We have to construct an interpretation of $\mathfrak{A} := \mathcal{C}(\mathfrak{A}_i)_{i \in I}$ in $\mathfrak{C}$. Let r_j be the arity of R_j. Consider n_i-dimensional interpretations

$$\begin{aligned} I^i := \big\langle h^i, \delta^i(\bar{x}^i), \varepsilon^i(\bar{x}^i, \bar{y}^i), \varphi_0^i(\bar{x}_0^i, \dots, \bar{x}_{r_0-1}^i), \dots, \\ \varphi_r^i(\bar{x}_0^i, \dots, \bar{x}_{r_r-1}^i)\big\rangle \end{aligned}$$

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

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

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

$$\begin{aligned} I := \big\langle h, \delta(\bar{x}), \varepsilon(\bar{x}, \bar{y}), \varphi_0(\bar{x}_0, \dots, \bar{x}_{r_0-1}), \dots, \\ \varphi_r(\bar{x}_0, \dots, \bar{x}_{r_r-1})\big\rangle \end{aligned}$$

where

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

and

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

In order to define φ_j we consider an interpretation

$$\begin{aligned} I^I := \big\langle h^I, \delta^I(\bar{x}), \varepsilon^I(\bar{x}, \bar{y}), \varphi_0^I(\bar{x}_0, \dots, \bar{x}_{s_0-1}), \dots, \\ \varphi_s^I(\bar{x}_0, \dots, \bar{x}_{s_s-1})\big\rangle \end{aligned}$$

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

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

for $k < r_j$. D_l and T_l^m can be defined by

$$D_l i := (d_l)_i = 1 \text{ and}$$
$$T_l^m i := (\tau_l^m)^{I^i}(\bar{x}_0^i, \dots, \bar{x}_{r_j-1}^i).$$

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

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

Corollary 6.3. *Both,* AutStr *and* ω-AutStr *are effectively closed under finitary generalised products.*

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

Corollary 6.4. *Let* $\mathfrak{A}_0, \dots, \mathfrak{A}_{n-1} \in$ AutStr. *Then there exists automatic presentations of*

(i) *the direct product* $\prod_{i<n} \mathfrak{A}_i$,

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

(iii) *the* ω*-fold disjoint union* $\omega \cdot \mathfrak{A}_0$ *of* $\mathfrak{A}_0$.

Corollary 6.5. *Let* $\mathfrak{A}_0, \dots, \mathfrak{A}_{n-1} \in$ AutStr *be ordered structures. There exists automatic presentations of*

(i) *the ordered sum* $\sum_{i<n} \mathfrak{A}_i$ *and*

(ii) *the* ω*-fold ordered sum* $\sum_{i<\omega} \mathfrak{A}_0$ *of* $\mathfrak{A}_0$.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] A. Blumensath. *Automatic Structures*. Diploma Thesis, RWTH Aachen, 1999.

[3] B. Boigelot, S. Rassart, and P. Wolper. On the expressiveness of real and integer arithmetic automata. In *Proc. 25th Int. Colloq. on Automata, Languages and Programming*, volume 1443 of *LNCS*, pages 152–163, 1998.

[4] V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Math. Soc.*, 1:191–238, 1994.

[5] S. Buss. The boolean formula value problem is in ALOGTIME. In *Proc. 19th ACM Symp. on the Theory of Computing*, pages 123–131, 1987.

[6] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, 1995.

[7] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, New York, 1974.

[8] C. C. Elgot and J. E. Mezei. On relations defined by generalized finite automata. *IBM J. Res. Develop.*, 9:47–68, 1965.

[9] D. Epstein, J. Cannon, D. Holt, S. Levy, M. Paterson, and W. Thurston. *Word Processing in Groups*. Jones and Bartlett Publishers, Boston, 1992.

[10] B. Farb. Automatic groups: A guided tour. *L' Enseignement Mathématique*, 38:291–313, 1992.

[11] E. Grädel. Simple interpretations among complicated theories. *Inf. Process. Lett.*, 35:235–238, 1990.

[12] E. Grädel and Y. Gurevich. Metafinite model theory. *Information and Computation*, 140:26–81, 1998.

[13] Y. Gurevich. Monadic second-order theories. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, pages 479–506. Springer, 1985.

[14] D. Harel. Towards a theory of recursive structures. In *Proc. 23rd Int. Symp. on Mathematical Foundations of Computer Science*, volume 1450 of *LNCS*, pages 36–53. Springer, 1998.

[15] T. Hirst and D. Harel. More about recursive structures: Descriptive complexity and zero-one laws. In *Proc. 11th IEEE Symp. on Logic in Computer Science*, pages 334–348, 1996.

[16] W. Hodges. *Model Theory*. Cambridge University Press, 1993.

[17] H. Hunt III, D. Rosenkrantz, and T. Szymanski. On the equivalence, containment, and covering problems for the regular and context-free languages. *J. Comput. Syst. Sci.*, 12:222–268, 1976.

[18] N. Immerman. *Descriptive complexity*. Springer, 1998.

[19] B. Khoussainov and A. Nerode. Automatic presentations of structures. *LNCS*, 960:367–392, 1995.

[20] G. Kuper, L. Libkin, and J. Paredaens, editors. *Constraint Databases*. Springer, 1999.

[21] A. R. Meyer and L. J. Stockmeyer. Word problems requiring exponential time. In *Proc. 5th ACM Symp. on the Theory of Computing*, pages 1–9, 1973.

[22] A. Stolboushkin. Towards recursive model theory. *Lecture Notes in Logic*, 11:325–338, 1998.

[23] W. Thomas. Ehrenfeucht games, the composition method, and the monadic theory of ordinal words. *LNCS*, 1261:118–143, 1997.