CDMTCS Research Report Series

Some Thoughts On Automatic Structures

Bakhadyr Khoussainov

Sasha Rubin

Univeristy of Auckland
Auckland, New Zealand

CDMTCS-185
May 2002

Centre for Discrete Mathematics and
Theoretical Computer Science

Some Thoughts On Automatic Structures

Bakhadyr Khoussainov[1] and Sasha Rubin[2]

[1] Computer Science Department, The University of Auckland, New Zealand
bmk@cs.auckland.ac.nz
[2] Mathematics Department, The University of Auckland, New Zealand
rubin@math.auckland.ac.nz

1 Introduction

Our aim in writing this paper is twofold. On the one hand, we present the theory of automatic structures from the points of view of model theory, algebra, complexity theory, and automata theory. On the other hand, we survey basic results and present possible directions for future research in the area. The theory of automatic structures as such is relatively new though its roots go back to the beginning of the developments of finite automata theory. The term *automatic structure* in the sense presented in this paper appeared in [12] in 1995. One can view the theory of automatic structures as

Automata Theory + Model Theory + Universal Algebra + Complexity Theory

The fact that the foundations of the theory come from the interactions of well-established areas with their own techniques, methods, and problems places it on solid ground.

A remarkable point is that an interest in the area arose from investigations of research groups with different mathematical interests and motivations. One is the group of Cannon, Epstein and Thurston and their collaborators in the theory of hyperbolic manifolds. They realised that many computational problems about fundamental groups of 3-manifolds are related to finite automata theory. This resulted in the development of the theory of automatic groups, see [7]. The area has now been generalised with the introduction of automatic semigroups (see [1] for example). The other group of researchers in computable model theory and algebra (see the handbook [9] for the current state of the area) headed by Nerode and Ershov began the development of the theory of feasible structures with the goal of understanding the relationship between constructions in model theory and computations with bounded resources, e.g. polynomial time/space computations (see for example [17]). In this line of work the paper [12] by Khoussainov and Nerode initiated the study of automatic structures in a general setting. Finally, Blumensath and Grädel motivate the study of automatic structures as a natural generalisation of finite model theory that provides a framework for extending techniques and results from finite structures to infinite ones, see [4].

In the theory of formal languages finite automata are used to describe infinite objects, that is infinite languages. Then the study of infinite objects is done in terms of their finite descriptions. From an algebraic point of view, languages as such do not carry any nontrivial structural information because they are identified simply with sets of words.

The underlying idea in the theory of automatic structures is to describe not just infinite languages but also algebraic structures, such as graphs, orders, groups etc., using finite automata.

Here is a brief outline the paper. In the next section we give basic definitions and provide examples. Section 3 gives an overview of the area from a model theory point of view. Section 4 is devoted to the study of automatic structures from a point of view of algebra. In Section 5, we present results that relate the area to complexity theory. Finally in Section 6, we discuss topics related to automatic isomorphisms and domain dependency of algebraic properties of structures. Each section contains a list of open problems for possible future work in the area.

2 Basic Definitions and Examples

Fix a finite alphabet Σ. A *finite automaton (FA)* $\mathcal{A}$ over Σ is a tuple of sets (S, I, Δ, F), where S is a finite set of *states*, $I \subset S$ is the set of *initial states*, $\Delta \subset S \times \Sigma \times S$ is the *transition table* and $F \subset S$ is the set of *final states*. One naturally extends Δ to a relation on $S \times \Sigma^* \times S$, also denoted by Δ. We sometimes write Δ in functional notation. A *computation* of the automaton $\mathcal{A}$ on a word $\sigma_1\sigma_2 \dots \sigma_n$ $(\sigma_i \in \Sigma)$ is a sequence of states of $\mathcal{A}$, say $q_0, q_1, \dots, q_n$, such that $q_0 \in I$ and $(q_i, \sigma_{i+1}, q_{i+1}) \in \Delta$ for all $0 \leq i \leq n-1$. If $q_n \in F$, then the computation is *successful* and we say that automaton $\mathcal{A}$ *accepts* the word. The *language*, $\mathcal{L}(\mathcal{A}) \subset \Sigma^*$, accepted by the automaton $\mathcal{A}$ is the set of all words accepted by $\mathcal{A}$. In general, $D \subset \Sigma$ is *finite automaton (FA) recognisable*, or *regular*, if $D = \mathcal{L}(\mathcal{A})$ for some finite automaton $\mathcal{A}$. We assume that the reader is familiar with the basics of finite automata theory.

We now introduce automata that recognise n–ary relations on Σ^*. In this paper, these automata are *synchronous n–tape automata*. The following description is based on Eilenberg et al. [6] and is equivalent to the formal definition. A synchronous n–tape automaton can be thought of as a one-way Turing machine with n input tapes. Each tape is regarded as semi-infinite having a finite word in the alphabet Σ written on it followed by an infinite succession of blanks, $\diamond$ symbols. The automaton starts in an initial state, reads simultaneously the first symbol of each tape, changes state, reads simultaneously the second symbol of each tape, changes state, etc., until it reads a blank on each tape. The automaton then stops and accepts the n–tuple of words, or not, according to whether or not it is in a final state. The set of all n–tuples accepted by the automaton is the relation recognised by the automaton.

Definition 1. *Let $\Sigma_\diamond$ denote the alphabet $\Sigma \cup \{\diamond\}$ where $\diamond \notin \Sigma$. The* **convolution of a tuple** $(w_1, \cdots, w_n) \in (\Sigma^\star)^n$ *is the tuple $(w_1, \cdots, w_n)^\diamond \in (\Sigma_\diamond^n)^\star$ formed by concatenating the least number of blank symbols, $\diamond$, to the right ends of the w_i, $1 \leq i \leq n$, so that the resulting words have equal length. The* **convolution of a relation** $R \subset \Sigma^{*n}$ *is the relation $R^\diamond \subset (\Sigma_\diamond^n)^*$ formed as the set of convolutions of all the tuples in R.*

Definition 2. **An** n**–tape synchronous automaton** *(henceforth* n–tape automaton*) on* Σ *is a finite automaton over the alphabet* $(\Sigma_\diamond)^n$*. An* n*–ary relation* $R \subset \Sigma^{\star n}$ *is* **FA recognisable** *or* **regular** *if its convolution* $R^\diamond$ *is recognisable by an* n*–tape automaton.*

Thus, a 2–tape automaton takes as input a pair of the form (w_1, w_2) where $w_1, w_2 \in \Sigma_\diamond^\star$, the words w_1 and w_2 have the same length, and in each word no element of Σ follows a $\diamond$. At the kth step of a computation on (w_1, w_2), the automaton reads as input a pair of the form (σ_1, σ_2) where $\sigma_i \in \Sigma_\diamond$ is the kth symbol of w_i, $i = 1, 2$. Clearly 1–tape automata coincide with the usual finite automata.

We now relate n–tape automata to structures. A *structure* $\mathcal{A}$ consists of a set A called the *domain*, some constants and atomic relations and operations on A. We may assume that $\mathcal{A}$ only contains relational and constant predicates as the operations can be replaced with their graphs. We write $\mathcal{A} = (A, R_1^A, \ldots, R_k^A, c_0^A, \ldots, c_t^A)$ where R_i^A is an n_i–ary relation on $\mathcal{A}$ and c_j^A is a constant element of $\mathcal{A}$. Then $(R_1^{n_1}, \ldots, R_k^{n_k}, c_0, \ldots, c_t)$ is called the *signature* of $\mathcal{A}$. In the sequel, all structures are relational, have finite or countable domains and finite signatures.

Definition 3. *A structure* $\mathcal{A} = (A, R_1^A, \ldots, R_k^A, c_0^A, \ldots, c_t^A)$ *is* **automatic** *over* Σ *if its domain* $A \subset \Sigma^\star$ *and the relations* $R_i^A \subset \Sigma^{*n_i}$ *all are FA recognisable. A structure will be called* **automatic** *if it is automatic over some alphabet. An isomorphism from a structure* $\mathcal{B}$ *to an automatic structure* $\mathcal{A}$ *is an* **automatic presentation** *of* $\mathcal{B}$ *in which case* $\mathcal{B}$ *is called* automatically presentable *(over* Σ*) and* $\mathcal{A}$ *an* **automatic copy** *of* $\mathcal{B}$*.*

We often ease notation by removing the superscripts from R_i when no ambiguity arises.

In general, a given structure need not be automatic. For example, the structure $\mathcal{A} = (\{0,1\}^\star, P)$, where $P = \{0^n 1^n \mid n \in \omega\}$, is not automatic because the unary predicate P can not be recognised by a finite automaton. However, this structure possesses many automatic presentations; for example $\mathcal{B}_1 = (\{0\}^\star, P_1)$ and $\mathcal{B}_2 = (\{0,1\}^\star, P_2)$, where $P_1 = \{0^{2n} \mid n \in \omega\}$ and $P_2 = \{(001)^n \mid n \in \omega\}$. Clearly, both of these structures are isomorphic to $\mathcal{A}$ and are automatic. Therefore, these two are automatic copies of $\mathcal{A}$. Thus, one can think of an automatic copy $\mathcal{B}$ of a structure $\mathcal{A}$ as a finite automaton implementation of $\mathcal{A}$.

One can use somewhat different terminology. Let $\mathcal{A}$ be a structure and $\mathcal{B}$ be an automatic copy of $\mathcal{A}$; so the domain B is FA recognisable. Let $\nu : B \to A$ be a mapping that establishes an isomorphism from $\mathcal{B}$ onto $\mathcal{A}$. Now, one can think of the set B as the set of codes of elements of $\mathcal{A}$. Thus, if $a \in A$ and $\nu(w) = a$ then w is the code (under ν) of the element a. Under this coding the atomic relations of $\mathcal{A}$ are made to be finite automata recognisable. Thus, one can talk about algorithmic properties of $\mathcal{A}$ under the presentation $\mathcal{B}$. In other words, instead of working with the structure $\mathcal{A}$ itself (that may actually be impossible to do because $\mathcal{A}$ can be computationally inaccessible) one may identify $\mathcal{A}$ with $\mathcal{B}$ and work directly with $\mathcal{B}$.

Thus, the definitions above present the idea of using finite automata as a tool for describing structures, or more precisely the isomorphism types of structures. In this sense, automatic structures supersede finite automata recognisable languages, and connect two fundamental concepts of mathematics: algebraic structures and finite state machines.

Here is a list of structures that have automatic presentations. We invite the reader to find automatic presentations of these structures though some of the presentations will be explained later.

- Presburger arithmetic $(\omega, +, S)$.
- The group of integers $(Z, +)$.
- The linearly ordered sets ω^n, where n is finite.
- The Boolean algebra of finite and co-finite subsets of a countable set.
- Countable absolutely free unary algebras.
- Finitely generated abelian groups.
- The linearly ordered set of rational numbers.
- Infinite countable vector spaces over finite fields.
- The structure (ω, f), where f is a bijection on ω forming exactly one cycle of length $p(n)$ for any given n with p being any polynomial with coefficients from ω.
- The semirings $(\omega, +, min)$ and $(\omega, +, max)$.

Let $\mathcal{A} = (A, R_1, \ldots, R_n)$ be an automatic structure. Consider the list $pr(\mathcal{A}) = (M_0, M_1, \ldots, M_n)$ of finite automata such that each automaton M_i, $i > 0$, is deterministic and recognises R_i, and M_0 recognises the domain A. We set $|pr(\mathcal{A})|$ to be $max\{|M_i| \mid 0 \leq i \leq n\}$. This is the size of the representation of $\mathcal{A}$. Of course, there are infinitely many presentations of the structure $\mathcal{A}$.

3 A Model Theory Point of View

We refer the reader to the book [10] for the basic model theory concepts. Let T be a first order theory. In model theory primary objects of study are models of T. As our basic interest is in automatic structures, we would like to investigate those models of T that are automatic. A motivating question is this: what are the properties of the theory T that allow the theory T to admit automatic models? We do not have a complete answer to this question. Here are, however, some partial results.

Perhaps, a fundamental property that naturally makes automatic structures objects of investigation from a model theoretic as well as a complexity point of view is the following theorem, first stated explicitly in [12]:

Theorem 1. *For any automatic structure $\mathcal{A}$ there exists an algorithm that given a first order definition of any relation R in $\mathcal{A}$ produces a finite automaton that recognises the relation R. In particular, the first order theory of $\mathcal{A}$ is decidable.*

Proof (sketch). Let R be a relation and $\phi(x_1, \ldots, x_n)$ be a formula defining R. The proof proceeds by induction on the complexity of the formula ϕ and rests on the closure properties of FA recognisable relations. If ϕ is an atomic formula then the definition of automatic structures tells us that R is FA recognisable. The case for boolean connectives is taken care of by the fact that the FA recognisable relations are closed under the union and complementation operations. The existential quantifier corresponds to taking projections; and the FA recognisable relations are closed under this operation. $\square$

Thus, if T is a complete theory and has an automatic model then T is decidable. We note that it is not always true that every decidable complete theory has an automatic model. A good example is the absolutely free algebra generated by a constant whose language contains at least one binary function symbol (for the proof see [12]). Blumensath and Grädel extended [3] Theorem 1 by showing that the theorem holds even if one considers the first order logic extended by the quantifier "there exist infinitely many", denoted by $FO(\exists^\infty)$. In particular

Theorem 2. *For any automatic structure $\mathcal{A}$ the set of all sentences of $FO(\exists^\infty)$ logic true in the structure $\mathcal{A}$ is decidable.* $\square$

One of the important topics in model theory and logic is related to interpretations. A general question of interest is the following. Does there exist, for a given class of models, a universal structure in which all models in the class and only them can be interpreted? For example, the arithmetic $(\omega, 0, S, +, \times)$ is the universal structure in which all computably enumerable structures can be interpreted by existential formulas. Blumensath and Grädel studied this question for the class of automatic structures.

Definition 4. *Let $\mathcal{A}$ and $\mathcal{B}$ be structures of signatures L and K, respectively. An* **n-dimensional interpretation** *of $\mathcal{A}$ in $\mathcal{B}$ consists of*

1. *a $FO(\exists^\infty)$-formula $\delta(x_1, \cdots, x_n)$ of signature K,*
2. *for each predicate symbol R of L, a $FO(\exists^\infty)$-formula $\phi_R(\bar{x}_1, \cdots, \bar{x}_m)$ of K where each $\bar{x}_i$ is an n-tuple of distinct variables and m is the arity of R, and*
3. *a surjective map $f : \delta(B^n) \to A$*

such that for all $\bar{b}_i \in \delta(B^n)$,

$$\mathcal{B} \models \phi_R(\bar{b}_1, \cdots, \bar{b}_m) \iff (f\bar{b}_1, \cdots, f\bar{b}_m) \in R^{\mathcal{A}}.$$

In this case we say that A is interpretable in B.

The idea is that each element in A is named by an element of B^n via the formula $\delta(x_1, \cdots, x_n)$ and to each atomic operation $R^{\mathcal{A}}$ there corresponds a first order definable formula ϕ_R in $\mathcal{B}$. Here is a simple but important theorem whose proof can be deduced from Theorem 1.

Theorem 3. *[4] If $\mathcal{A}$ is interpretable in an automatic structure $\mathcal{B}$ then $\mathcal{A}$ is automatic.* $\square$

It is well known that any structure $\mathcal{A}$ can be coded into a graph $\mathcal{G}(A)$ so that these two structures are interpretable in each other (see for example [10]). This basically tells us that the study of model theoretic properties of structures can be reduced to the study of graphs. It turns out that a similar result holds for the class of automatic structures. In [11] the following theorem is proved:

Theorem 4. *For any automatic structure $\mathcal{A}$ there exists an automatic graph $\mathcal{G}(\mathcal{A})$ such that these two structures are interpretable in each other.*

The proof of the theorem is based on a careful checking that the transformation of $\mathcal{A}$ into $\mathcal{G}(\mathcal{A})$ provided in [10] preserves the automata-theoretic properties of the original structure in the graph $\mathcal{G}(\mathcal{A})$.

Let us consider the following two structures:

1. The first structure is a weak fragment of arithmetic, namely $\mathbb{N}_p = (\omega, +, |_p)$, where $p > 0$ is fixed, $+$ is the addition operation, and $x|_p y$ iff $x = p^n$ and $y = kx$ for some natural numbers k, n.
2. The second structure is a tree structure, namely $\mathcal{M}(\Sigma) = (\Sigma^\star, \sigma_a, \preceq, el)_{a \in \Sigma}$, where $|\Sigma| > 1$ and
   (a) $\sigma_a(x) = xa$ for all $a \in \Sigma$ and $x \in \Sigma^\star$,
   (b) $x \preceq y$ if x is a prefix of y, and
   (c) $el(x, y)$ if $|x| = |y|$.

It is not hard to see that both of these structures have automatic presentations. Grädel in [8] proved that these two structures are interpretable in each other. It turns out that each of these structures is universal for the class of all automatic structures.

Theorem 5. *[3] A structure $\mathcal{A}$ has an automatic presentation if and only if it can be interpreted in $\mathcal{M}(\Sigma)$.*

Proof (sketch). Since $\mathcal{M}(\Sigma)$ is automatic, so is any $\mathcal{A}$ that is interpretable in $\mathcal{M}(\Sigma)$. Now, assume that $\mathcal{A}$ is automatic and consider an atomic relation R of $\mathcal{A}$. By the definition of automaticity of $\mathcal{A}$, the relation R is FA recognisable. It suffices to prove that R is definable in $\mathbb{N}_p$ by the result of Grädel mentioned above. Definability of R in $\mathbb{N}_p$ can be done by coding the successfull computations of the automaton recognising the relation R into a formula of the language of $\mathbb{N}_p$. This was first proved by Büchi in the sixties. We refer the reader to [3] for a detailed proof. $\square$

There are many questions of a model-theoretic nature that could be asked about automatic structures. As mentioned at the beginning of this section the class of models of a given complete first order theory T is a primary object of study. For example, model theory provides necessary and sufficient conditions for a theory to have prime and saturated models. The prime models are the ones that can be elementarily embedded into all models of T, while saturated models are the ones that realise all possible types of T in expansions by constants. Informally, prime models are the smallest models of T

and the saturated models are the biggest models of T. We also mention that computable model theory characterises theories that have decidable prime or saturated models (see [9]). Thus a natural step in this direction consists of investigating the following problems:

Problem 1. 1. Give a characterisation of first order complete theories that have automatic (prime, saturated or homogeneous) models.
2. Characterise those first order complete theories T such that every model of T has an automatic presentation.

For example, consider the first order theory $Th(\omega)$ of the linear ordering ω. It has an automatically presentable prime model ω, an automatically presentable saturated model $\omega + \mathbb{Z} \cdot Q$, and a model with no automatic presentation, namely $\omega + \mathbb{Z} \cdot \omega^{\omega}$. The first problem should be seen in light of Theorem 1 or Theorem 2 and the classical computable model theory result stating that a decidable theory has a computable model in which the satisfaction predicate is computable.

In model theory an important concept is the notion of categoricity. Assume that we are given a cardinal α. A complete theory T is **α-categorical** if all models of T that have cardinality α are isomorphic. There are characterisations of α-categorical theories, such as the Ryll-Nardzewski theorem, and one of the well-known facts is the result of Morley stating that $\aleph_1$-categoricity is equivalent to β-categoricity for all $\beta \geq \aleph_1$. We pose the following problem:

Problem 2. Give a characterisation of $\aleph_0$-categorical or $\aleph_1$-categorical theories for which every model has a FA presentation.

For example, the theory of the linear order of rational numbers is $\aleph_0$-categorical and has a FA presentation. Similarly, the theory T of the model $(\mathbb{Z}, S)$ is $\aleph_1$-categorical and all models of T have automatic presentations.

4 An Algebraic Point of View

This section discusses two closely related issues. One is devoted to understanding the isomorphism types of automatic structures. The other is concerned with the question as whether or not a given particular structure of interest possesses an automatic presentation.

We begin with the first issue. From an algebraic point of view, a basic goal in the study of a classes of algebraic structures, such as groups, Boolean algebras, linear orders, lattices, rings etc., consists of finding certain invariants that characterise the structures up to isomorphism. Automaticity is quite a restrictive requirement to place on an algebraic structure. Therefore one could hope to have satisfying characterisations of isomorphism types of automatic structures or of classes of automatic structures. It turns out that, in general, the situation here is complicated for some classes of structures and well understood for others. To illustrate this, we consider two types of structures whose algebraic nature and classical isomorphism types are very well understood.

Permutation Structures

We consider algebras of the type (A, f), where $f : A \to A$ is a bijection. These structures are called **permutation structures**. For $a \in A$, the set $\{f^i(a) \mid i \in \omega\}$ is called an **orbit of** f. Define the following two isomorphism invariants

$$I_1(\mathcal{A}) = \{n \mid \text{there is an orbit of size } n\} \quad \text{and}$$
$$I_2(\mathcal{A}) = \{(n, m) \mid \text{there are exactly } m \text{ orbits of size } n\},$$

where $n, m \leq \omega$. Note that $I_2(\mathcal{A})$ is a full isomorphism invariant. In other words permutation structures $\mathcal{A}$ and $\mathcal{B}$ are isomorphic if and only if $I_2(\mathcal{A}) = I_2(\mathcal{B})$, while $I_1(\cdot)$ is merely preserved under isomorphism. Our goal is to characterise automatic permutation structures in terms of the invariant sets I_1 and I_2. The results below show that the situation here is not as desirable as one wants. We consider two cases.

Case 1. Assume we are restricted to those automatic structures that have automatic presentations over a unary alphabet. The following theorem gives an explicit characterisation of the isomorphism types of automatic permutation structures in this case.

Theorem 6. *[3] [14] A permutation structure $\mathcal{A}$ has an automatic presentation over a* unary *alphabet if and only if $I_1(\mathcal{A})$ is finite and there are finitely many infinite orbits.*

Case 2. This is the general case, $|\Sigma| = 2$. In the following theorem we show how to construct automatic permutation structures from a given FA recognisable language $L \subset \Sigma^\star$. This construction builds an automatic permutation structure whose isomorphism invariant I_1 exhibits nontrivial behaviour.

Theorem 7. *For any function f which is either a polynomial p whose coefficients are positive integers or an exponential function k^{an+b}, where $k \geq 2$ and a, b are fixed positive integers, there exists an automatic permutation structure $\mathcal{A}$ such that $I_1(\mathcal{A}) = \{f(n) \mid n \geq 1\}$ and $I_2(\mathcal{A}) = \{(f(n), 1) \mid n \geq 1\}$.*

Proof (sketch). Let L be a language over Σ. The growth of L is the function g_L defined as $g_L(n) = |\Sigma^n \cap L|$ for $n \in \omega$. Thus, $g_L(n)$ is the number of strings of length n that belong to L. For the given function f there exists a FA recognisable language L whose growth function is exactly f (this is proved in [11]). Let $\leq$ be an automatic well order of type ω on $\Sigma^\star$. For each $x \in L$ we proceed as follows. If x has length n and is not the maximal element among all elements in L of length n then $f(x)$ is the minimal $y \geq x$ in L whose length is n. Otherwise, $f(x)$ is the minimal element of length n that belongs to L. Clearly, (L, f) is the desired permutation structure. $\square$

In the theorem above the invariant I_2 is a function whose range is finite, in fact a subset of $\{0, 1\}$. The next example shows that the range of I_2 can be infinite. Indeed, consider an automatic permutation structure $\mathcal{A}$ so that $I_2(\mathcal{A}) = \{(n, 1) \mid n > 0\}$. Consider the automatic permutation structure $\mathcal{B} = \mathcal{A} \times \mathcal{A}$. Then it is not hard to see that the range of $I_2(\mathcal{B})$ is an infinite set.

To further illustrate the complexity of the general case, we briefly show a connection between the invariant I_1 and the running times of Turing machines. Let T be a TM and $C(T)$ be the graph consisting of all configurations of T, with an edge from configuration c to d if T can move from c to d in a single transition. The configuration graph $\mathcal{C}(T)$ is an automatic graph. A TM T is **reversible** if every vertex in $C(T)$ has indegree and outdegree at most one. Bennett [2] showed that any deterministic TM T can be simulated by a reversible TM R. Furthermore, running times of these machines differ by a constant factor.

Let $Time_T(w)$ be the number of steps T takes to halt on w. We assume that the unique initial state of T is not a final one, and that for any non-final configuration c there is a d such that (c, d) is an edge in $C(T)$.

Theorem 8. *For every reversible TM T, there is an automatic permutation structure $\mathcal{A}(T)$ for which $I_1(\mathcal{A}(T)) = \{Time_T(w) \mid w \in \Sigma^\star\}$.*

Proof (sketch). Transform the configuration graph of T into a permutation structure in such a way that T halts on w if and only if the initial configuration containing w belongs to a finite orbit whose length is $Time_T(w)$. It is possible to guarantee that these are the only finite orbits of the permutation structure. $\square$

As a corollary of the theorem one obtains the following undecidability result that indicates that characterisation of isomorphism types of automatic structures is a hard problem:

Theorem 9. *It is undecidable whether or not two automatic structures are isomorphic.*

Proof. For a deterministic TM T', construct an equivalent reversible TM T and the structure $\mathcal{A}(T)$. T halts on no word iff $\mathcal{A}$ is isomorphic to the permutation structure consisting only of infinitely many infinite orbits of type $(\mathbb{Z}, f)$. $\square$

Problem 3. Find the precise complexity of the isomorphism problem for automatic structures in the arithmetical hierarchy or show that the isomorphism problem is not arithmetical.

Well Ordered Sets

We now turn our interest to well-ordered sets – that is, linear orderings satisfying the property that every non-empty subset has a minimum element. We refer the reader to Rosenstein [16] for the relevant background. We would like to understand which well ordered sets have automatic presentations and which do not. Clearly ω is an automatic well-order. It is not hard to see that the product of any two automatic well-ordered sets is automatic. Therefore, the well ordered set isomorphic to ω^n has an automatic presentation. In [12] Khoussainov and Nerode asked the question of finding the minimal well-ordered set without an automatic presentation. This has been answered by Delhomme, Goranko and Knapik [5]:

Theorem 10. *The well-ordered set ω^ω is the minimal one that does not possess an automatic presentation.* $\square$

The proof of this theorem is beautiful and greatly exploits the finite state transitions of the automata that represent a well-ordered set. It seems that the ideas of the proof could be very useful in studying questions related to the existence of automatic presentations of structures of a particular interest.

We list several problems which we think are of interest, at least from an algebraic point of view:

Problem 4. 1. Characterise Boolean algebras that possess automatic presentations.
2. Characterise Abelian groups (or groups in general) that possess automatic presentations.
3. Characterise linearly ordered sets that possess automatic presentations.

We turn to a brief discussion on the second issue mentioned at the beginning of this section. Assume we are given a structure $\mathcal{A}$. Does $\mathcal{A}$ have an automatic presentation? Of course, not every structure $\mathcal{A}$ has an automatic presentation. A necessary condition, which we have mentioned is not sufficient, is that the theory of $\mathcal{A}$ be decidable. We think that we still lack some good techniques and methods for proving or disproving that certain structures admit automatic presentations. We do not even know whether or not some fundamental structures common to computer science and mathematics have automatic presentations. Here is a list of a sample of such structures with which we end this section:

Problem 5. 1. Does there exist an automatic presentation of the additive group of rational numbers?
2. Does there exist an automatic presentation of the random graph?
3. Does there exist an automatic presentation of the atomless Boolean algebra?
4. Does there exist an infinite automatic field?

We note that none of the above questions have positive answers in case when one is restricted to a unary alphabet. We also expect that answers to some of these questions could be obtained with not much difficulty.

5 A Complexity Theory Point of View

In light of Theorem 1, automatic structures are natural objects from a complexity theory point of view. A fundamental problem is the **the query evaluation problem**: given an automatic structure $\mathcal{A}$ and a formula $\phi(\bar{x})$ compute a finite representation of all $\bar{a}$ in the structure such that $\phi(\bar{a})$ holds in $\mathcal{A}$. This amounts to constructing an automaton recognising the set $\phi(\mathcal{A})$ of all the tuples that make the formula $\phi(\bar{x})$ true in $\mathcal{A}$. A particular case of the query evaluation problem is the **model checking problem**: given an automatic structure $\mathcal{A}$, a formula $\phi(\bar{x})$ and a tuple $\bar{a} \in A$, decide whether or not $\phi(\bar{a})$ holds in the structure $\mathcal{A}$.

Each of these problems can be measured from three points of view as discussed in [3]:

1. (Structural complexity) Given a formula ϕ, what is the time or space complexity when one varies the automatic structure $\mathcal{A}$?
2. (Expression complexity) Given an automatic structure $\mathcal{A}$, what is the time or space complexity one varies the formula ϕ?
3. (Combined complexity) What is the time or space complexity when one varies ϕ and $\mathcal{A}$?

Note that the complexity here is defined in terms of the size of the deterministic automata representing $\mathcal{A}$. It seems natural that the complexities of questions related to query evaluation or model checking problems can be reduced to known questions about finite automata. We provide some illustrative results. For simplicity we restrict ourselves to pure relational signatures and quantifier free formulas. The case when signatures have function symbols should be examined more carefully. This is because the graphs of the atomic functions are FA recognisable, and hence the values of the functions on a given input must be computed to evaluate a given formula containing the function symbol.

Here is a theorem that tells us that model checking problem for quantifier free formulas can be computed efficiently, while the query evaluation problem needs more time and space resources.

Theorem 11. *[4]*

1. *Given an automatic structure $\mathcal{A}$, a quantifier free formula $\phi(\bar{x})$, and a tuple $\bar{a} \in A$ checking whether or not $\phi(\bar{a})$ is true in $\mathcal{A}$ can be computed in $O(|\phi||\bar{a}||pr(\mathcal{A})|log(|pr(\mathcal{A})|))$-time and in $O(log(|\phi|) + log(|pr(\mathcal{A})| + log(|\bar{a}|)))$-space.*
2. *Given an automatic structure $\mathcal{A}$, a quantifier free formula $\phi(\bar{x})$, constructing an automaton that computes the set $\phi(\mathcal{A})$ takes $O(|pr(\mathcal{A})|^{O(|\phi|)})$–time and $O(|\phi|log(|pr(\mathcal{A})|))$–space.*

Proof (sketch). The proof of the first part basically follows from the fact that given an automaton $\mathcal{M}$ and a string w checking whether or not $\mathcal{A}$ accepts w can be computed in $O(|w||\mathcal{M}|log(|\mathcal{M}|))$-time and $O(log(|\mathcal{M}|) + log(|w|))$-space.

For the second part, for each atomic formula ϕ_i, $i \leq |\phi|$, that occurs in ϕ consider the automaton $\mathcal{M}_i$ that recognises $\phi_i(\mathcal{A})$. Then the automaton $\mathcal{M}$ recognising $\phi(\mathcal{A})$ is constructed in such a way that the state set of $\mathcal{M}$ is the cartesian product of state sets of $\mathcal{M}_i$. Hence constructing $\mathcal{M}$ takes $O(|pr(\mathcal{A})|^{O(|\phi|)})$-time. □

We refer the reader to [3] for a good exposition of complexities of query evaluation problems on automatic structures.

The next result, first noted by [4], seems rather discouraging, however, it manifests the richness of the class of automatic structures. A function is called **non-elementary** if it can not be bounded above by means of any of the functions from the list 2^n, 2^{2^n}, ...

Theorem 12. *There exists an automatic structure in which the expression complexity of the model checking problem is non-elementary.*

A structure that satisfies this theorem is in fact $\mathbb{N}_p$ (Grädel [8]). This result by itself leads to many open and natural questions that one could investigate. Here is one.

Problem 6. Give natural examples of classes of automatic structures for which the model checking problem is solvable in polynomial time.

In relation to this problem we note that Blumensath [3] shows that for any automatic structure that can be presented over a unary alphabet the model checking problem is solvable in polynomial time.

6 Automatic Isomorphisms and Domain Dependency

Classically, isomorphic structures are considered identical. In general isomorphic computable structures may exhibit distinct computability-theoretic features. Therefore computable model theory and algebra treat computably isomorphic structures as identical. As we study structures from the automata theory point of view, it is natural to identify those automatic structures that are automatically isomorphic. We formalise this in the following definition.

Definition 5. *An automatic structure $\mathcal{A}$ is* **automatically isomorphic** *to an automatic structure $\mathcal{B}$ if there is an isomorphism from $\mathcal{A}$ into $\mathcal{B}$ that is FA recognisable.*

Thus, if $\mathcal{A}$ and $\mathcal{B}$ are automatically isomorphic then there is an automaton that exhibits an isomorphism between $\mathcal{A}$ and $\mathcal{B}$. It is not hard to see that the automatic isomorphism relation between automatic structures is reflexive, symmetric and transitive. Therefore, we say that $\mathcal{A}$ and $\mathcal{B}$ have the same **automatic isomorphism** type if they are automatically isomorphic.

An immediate question that arises consists of finding those structures that are invariant with respect to automatic presentations; that is those structures for which any two automatic copies are automatically isomorphic. This is answered in [12].

Theorem 13. *Any two automatic presentations of a structure $\mathcal{A}$ are automatically isomorphic if and only if $\mathcal{A}$ is finite.*

Proof (sketch). Assume that $\mathcal{A}$ is automatic and infinite. Take a new symbol σ, and replace every element $w = a_0 \dots a_n$ in the domain of $\mathcal{A}$ with a new word $\sigma(w) = a_0\sigma a_1\sigma \dots a_{n-1}\sigma a_n$. Now consider the new structure whose domain is $\sigma(A) = \{\sigma(w) \mid w \in A\}$. Induce the atomic relations on A into $\sigma(A)$ in a natural way thus obtaining a new automatic copy $\sigma(\mathcal{A})$ of $\mathcal{A}$. These two automatic structures are isomorphic but not automatically isomorphic. □

A basic point that can be deduced from this theorem is that we would like to preserve the codes naming the elements of a structure. That is, we fix domains and then consider automatic isomorphisms between structures whose domains coincide. As we think that this is an interesting point of study we give the following definition:

Definition 6. *An automatic structure $\mathcal{A}$ is* **automatically D-isomorphic** *to an automatic structure $\mathcal{B}$ if A and B have the same domain D and there is an isomorphism from $\mathcal{A}$ into $\mathcal{B}$ that is FA recognisable.*

One can now study the question as to which structures are invariant with respect to automatic presentations over the same domain. Here we give an example of a class of such structures. Recall that an equivalence structure is of the type $(E, \sim)$, where $\sim$ is an equivalence relation on E.

Proposition 1. *Let $\mathcal{E} = (\{1\}^\star, \sim)$ be an automatic equivalence structure. Then any two automatic presentations of $\mathcal{E}$ over $\{1\}^\star$ are automatically $\{1\}^\star$–isomorphic.*

Proof (sketch). It is shown in [13] that $\mathcal{E}$ has finitely many infinite equivalence classes, and there exists n such that the size of any finite equivalence class is at most n. Let $\mathcal{E}_1$ be an automatic copy of $\mathcal{E}$ over the domain $\{1\}^\star$. Then using a characterisation of FA recognisable languages over a unary alphabet (these are basically finite unions of arithmetic progressions) it is possible to establish an automatic isomorphism between $\mathcal{E}$ and $\mathcal{E}_1$. □

Another interesting problem is to find isomorphic automatic structures over the same domain which are not automatically isomorphic. We present an example.

Proposition 2. *Consider the equivalence structure $\mathcal{E}$ that consists of infinitely many classes of size 1 and size 2. The structure $\mathcal{E}$ has two automatic presentations over the domain $\{0,1\}^\star$ that are not automatically isomorphic.*

Proof (sketch). Here are two presentations $\mathcal{E}_1$ and $\mathcal{E}_2$ whose domains are $\{0,1\}^\star$. In $\mathcal{E}_1$ the two element equivalence classes are $\{0^{2n}, 0^{2n+1}\}$, where $n \in \omega$. In $\mathcal{E}_2$ the sets $\{(01)^{2n}, (01)^{2n+1}\}$, where $n \geq 1$, form the two element equivalence classes. It can be proved that these two automatic structures are not automatically isomorphic. □

The problem of a general character is now the following:

Problem 7. Let D be a regular language. Characterise those automatic structures for which any two automatic presentations with domain D are automatically isomorphic.

Finally, we note that many concepts and results of this paper can naturally be generalized by considering tree automata presentations or Büchi automata presentations (see [3]). The former properly extends the class of finite automata presentable structures and the latter allows one to extend the results to uncountable structures. One can also study those presentations where finite automata are restricted by certain language-theoretic or algebraic properties. We leave all these for further development of the area.

References

1. C.M. Campbell, E.F. Robertson, N. Ruškuc, R.M. Thomas, Automatic Semigroups. *Theoretical Computer Science* **250 1-2** (2001), 365–391.
2. C. Bennett, Logical Reversibility of Computation. *IBM Journal of Res. Develop.*, **17** (1973), 525–532.
3. A. Blumensath, Automatic Structures. Diploma Thesis, RWTH Aachen, 1999.
4. A. Blumensath, E. Grädel, Automatic Structures. In: *Proc. 15th IEEE Symp. on Logic in Computer Science*, 2000, 51–62.
5. C. Delhomme, V. Goranko, T. Knapik Automatic Linear Orderings. personal communication.
6. S. Eilenberg, C.C. Elgot, J.C. Shepherdson, Sets Recognised by n–Tape Automata. *Journal of Algebra* **13 4** (1969), 447–464.
7. D. Epstein, J. Cannon, D. Holt, S. Levy, M. Paterson, W. Thurston, *Word Processing in Groups.* Jones and Bartlett, Boston MA, 1992.
8. E. Grädel Simple Interpretations among Complicated Theories. *Information Processing Letters* **35** (1990), 235–238.
9. Y. Ershov, S. Goncharov, A. Nerode, J. Remmel (eds.) Victor Marek (assoc.ed.): *Handbook of Recursive Mathematics. Volume 1-2.* North-Holland, 1999.
10. W. Hodges *Model Theory.* Encyclopedia of Mathematics and its Applications **42**, Cambridge University Press, 1993.
11. H. Ishihara, B. Khoussainov, S. Rubin Some results on automatic structures. LICS 2002.
12. B. Khoussainov, A. Nerode, Automatic Presentations of Structures, In: *Logic and Computational Complexity (Indianapolis, IN, 1994).* LNCS **960**, Springer, Berlin, 1995, 367–392.
13. B. Khoussainov, S. Rubin, Graphs with Automatic Presentations over a Unary Alphabet. DCAGRS, London, Ontario, 2000. *Journal of Automata, Languages and Combinatorics* **6 4**, 2001.
14. S. Rubin, Finite Automata and Well Ordered Sets. *New Zealand Journal of Computing* **7 2** (1999), 39–46.
15. S. Rubin, Automatic Linear Orderings. Unpublished.
16. J.G. Rosenstein, *Linear Orderings.* Academic Press, 1982.
17. A. Nerode, J. Remmel, Complexity-theoretic Algebra: Vector Space Bases. In: *Feasible mathematics (Ithaca, NY, 1989).* Progr. Comput. Sci. Appl. Logic **9** Birkhuser Boston, Boston, MA, 1990, 293–319.