CDMTCS Research Report Series

On Isomorphism Invariants of Some Automatic Structures

Hajime Ishihara
School of Information Sciences
JAIST, Japan

Bakhadyr Khoussainov
Department of Computer Science
University of Auckland
Auckland, New Zealand

Sasha Rubin
Department of Mathematics
University of Auckland
Auckland, New Zealand

CDMTCS-172
January 2002

Centre for Discrete Mathematics and
Theoretical Computer Science

On Isomorphism Invariants of Some Automatic Structures

Hajime Ishihara   Bakhadyr Khoussainov   Sasha Rubin

Abstract

In this paper we study structures defined by finite automata, called automatic structures. We provide a method that reduces the study of automatic structures to the study of automatic graphs. We investigate isomorphism invariants of automatic structures with an emphasis to equivalence relation structures, linearly ordered sets, and permutation structures.

1 Introduction

In this paper we investigate those structures that can be defined, in a certain precise sense, by means of finite automata. The general idea is to code elements of a given structure in such away that all the atomic first order queries about the structure can be decided by finite automata. We call these structures automatic structures. In this paper we will be interested in the classes of automatic graphs, equivalence structures, linear orderings and permutation structures. Our basic motivation lies in trying to characterise, in an appropriate terminology, isomorphism invariants of automatic structures. In [5] Blumensath and Grädel characterised automatic structures in terms of an important concept of logic, namely interpretability. They proved that a structure $\mathcal{A}$ is automatic if and only if it is first order interpretable in the structure $N_p = (N, +, |_p)$, where $x|_p y$ iff $x = p^n$ and $y = kx$ for some $n, k \in N$. However, it seems that the problem of characterising the isomorphism invariants of automatic structures is a challenging task. We will show that even for simple cases such as equivalence structures and permutation structures the situation is quite complex.

There are several reasons to be interested in understanding isomorphism invariants of automatic structures. One is that we would like to understand the interplay between automata-theoretic and model-theoretic (or algebraic) concepts, e.g. recognizability and definability. The other reason is of complexity-theoretic nature. It is known that the first order theory of any automatic structure is decidable [12]. A natural question arises as to which automatic structures are feasible and which are not. Blumensath and Grädel [3] [5] show that there are automatic structures whose theories are non-elementary. In other words, the expression complexity of the model checking problem in automatic structures can be an intractable problem. On the other hand, there are examples of automatic structures, e.g. structures presented by finite automata over unary alphabet or the rational numbers with the natural ordering, for which the expression complexity of the model checking is polynomial.

These and other results on automatic structures implicitly tell us that there are intimate interactions between studying isomorphism invariants of automatic structures and the expression complexity of the model checking problem. The more we know about isomorphism invariants of automatic structures, the more the theory of this structure is computationally accessible.

We now give an overview of this paper. The next section is an introductory section where we give basic definitions and state some known results in the area. The section on automatic graphs is devoted to showing that there is a functor from the class of all automatic structures into the class of all automatic graphs. We will show that this functor preserves not only model-theoretic but also automata-theoretic properties of structures. The section on automatic equivalence structures is devoted to constructing automatic equivalence relations with different types of isomorphism invariants. The main goal is to show that some isomorphism invariants of automatic equivalence structures possess complicated complexity-theoretic and algebraic behaviour. The next section reduces the study of automatic equivalence relations to that of automatic linear orderings. Finally, in the last section we study automatic permutations. We show how to construct automatic permutations from automatic equivalence structures. The basic result of the section concerns the relationship between permutation structures and the running times of reversible Turing machines. As a consequence of this we will prove that the isomorphism problem for permutation structures is undecidable.

Here are some notes on related literature. A systematic study of interactions between automata and algebraic structures began from the work of Cannon and Thurston on automatic groups [9]. This was generalised by Khoussainov and Nerode in [12] but motivated from a point of view of computable model theory. A significant work in understanding of automatic structures has been done by Blumensath and Grädel [3] [5]. A recent paper by Benedikt and et al. [1] investigates model-theoretic properties of automatic structures, e.g. questions related to quantifier elimination. In recent work Delhomme, et al. [6] show that the minimal ordinal without automatic presentations is ω^ω.

2 Basic Notions

A *finite automaton (FA)* $\mathcal{A}$ over an alphabet Σ is a tuple (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*. A *computation* of $\mathcal{A}$ on a word $\sigma_1\sigma_2\ldots\sigma_n$ ($\sigma_i \in \Sigma$) is a sequence of states $q_0, q_1, \ldots, q_n$ 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 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 $\mathcal{A}$. A set $D \subset \Sigma^\star$ 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 in finite automata theory.

Automata that recognise n–ary relations are *synchronous n–tape automata*. The following description is based on Eilenberg et al. [7]. A synchronous n–tape automaton is a one-way Turing machine with n input tapes. Each tape is regarded as semi-infinite having written on it a word in the alphabet Σ 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 if 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 *Consider the alphabet $\Sigma_\diamond^n$, with $\Sigma_\diamond = \Sigma \cup \{\diamond\}$, $\diamond \notin \Sigma$. The* convolution *of a tuple $(w_1, \cdots, w_n) \in \Sigma^{*n}$ is the word $(w_1, \cdots, w_n)^\diamond \in (\Sigma_\diamond^n)^\star$ formed by concatenating the least number of $\diamond$ symbol to the right ends of the w_i, $1 \le i \le n$, so that the resulting words have equal length. The* convolution of a relation *$R \subset \Sigma^{*n}$ is the language $R^\diamond \subset (\Sigma_\diamond^n)^\star$ formed as the set of convolutions of the tuples in R.*

Definition 2 An 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 *if its convolution $R^\diamond$ is recognisable by an n–tape automaton.*

We now relate n–tape automata to structures. A *structure* $\mathcal{A}$ consists of a set A called the *domain* and some constants, 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. 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). A structure will be called* automatic *if it is automatic over some alphabet.*

The following result makes automatic structures objects of study from a complexity-theoretic point of view:

Theorem 1 *[12] There exists an algorithm that given an automatic structure $\mathcal{A}$ and a first order definition of a relation R in $\mathcal{A}$ produces a finite automaton that recognises R. In particular, the first order theory of $\mathcal{A}$ is decidable.* □

Blumensath and Grädel extended this result in [3] by showing that the theorem holds even if one considers the first order logic extended by the quantifier "there exist infinitely many", denoted by $FO(\exists^\infty)$, and combined this with an important concept in model theory, – interpretability.

Definition 4 *Let $\mathcal{A}$ and $\mathcal{B}$ be structures of signatures L and K, respectively. An* **n-dimensional interpretation** *Γ of $\mathcal{A}$ in $\mathcal{B}$ consists of a $FO(\exists^\infty)$-formula $\delta(x_1, \cdots, x_n)$ of K, for each symbol S of L, a $FO(\exists^\infty)$-formula $\phi_S(\bar{x}_1, \cdots, \bar{x}_m)$ of K where each $\bar{x}_i$ is an n-tuple of distinct variables and m is the arity of S, and a surjective map $f : \delta(B^n) \to A$ such that for all $\bar{b}_i \in \delta(B^n)$, $\mathcal{B} \models \phi_S(\bar{b}_1, \cdots, \bar{b}_m) \iff (f\bar{b}_1, \cdots, f\bar{b}_m) \in S^{\mathcal{A}}$.*

We give an example from [3]. For a non-unary alphabet Σ with p symbolsconsider the structures $\mathcal{W}(\Sigma) = (\Sigma^\star, (\sigma_a)_{a\in\Sigma}, \preceq, \mathrm{el})$ and $N_p = (N, +, |_p)$, where $\sigma_a(x) = xa$, $x \preceq y$ if x is a prefix of y, $el(x, y)$ if x and y have the same length, $x|_p y$ if x divides y and x is a power of p, and $+$ is addition. Then N_p and $\mathcal{W}(\Sigma)$ are mutually interpretable.

Theorem 2 *[5] If $\mathcal{B}$ is automatic and $\mathcal{A}$ is interpretable in $\mathcal{B}$ then $\mathcal{A}$ is automatic.*

3 On Automatic Graphs

In this section we provide a procedure that given an automatic structure $\mathcal{A}$ produces an automatic graph $\mathcal{G}(\mathcal{A}) = (V(\mathcal{A}), E(\mathcal{A}))$, with set of vertcies $V(\mathcal{A})$ and edges $E(\mathcal{A})$, so that $\mathcal{A}$ and $\mathcal{G}(\mathcal{A})$ can be recovered from each other. The transformation of $\mathcal{A}$ into $\mathcal{G}(\mathcal{A})$, denoted by Γ, is described in Hodges [11] (Theorem 5.5.1) and possesses natural algebraic and model-theoretic properties. To investigate properties of Γ we need to explicitly define it with an eye towards automata-theoretic considerations.

An n-**tag**, where $n > 1$, is a symmetric graph isomorphic to the graph $\{0, 1, \ldots, n, c\}, E)$, where the set E of edges consists of all pairs $\{i, i+1\}$ for $0 < i < n$, $\{n, 1\}$ and $\{2, c\}$. The vertex 0 is the **start** of the n-tag. The element c is needed to make the tag rigid, that is a structure without nontrivial automorphisms. Further, c will not be mentioned explicitly.

Let v be a new symbol. With each element $a \in A$ we associate a 5-tag denoted by $T(a)$ so that the vertices of $T(a)$ are the words $va, va1, \ldots, va5$ and the edges are $\{va, va1\}, \{vai, vai+1\}, \{va5, va1\}$, where $i = 1, \ldots, 5$. Thus, the start vertex va of the tag $T(a)$ can be associated with the element a of the structure $\mathcal{A}$.

We now code the predicate P_i. Firstly, with each tuple $\bar{a} = (a_1, \ldots, a_{m_i})$ for which $P_i(\bar{a})$ we associate a $(5+i)$-tag $T(i, \bar{a})$ with vertices $v\bar{a}$, $v\bar{a}1$, …, $v\bar{a}(i+5)$ and edges $\{v\bar{a}, v\bar{a}1\}$, $\{v\bar{a}k, v\bar{a}(k+1)\}$, $\{v\bar{a}(i+5), v\bar{a}\}$, where $k = 1, \ldots, i+4$. Secondly, with the tuple $\bar{a} = (a_1, \ldots, a_{m_i})$ and the kth element of this tuple a_k we associate the graph $L(i, \bar{a}, k)$ consisting of the k vertices va_k, $v\bar{a}k1$, $\bar{a}k2$, $\bar{a}k3$, …, $\bar{a}k(k-2)$, $v\bar{a}$ and edges appearing between any consecutive pair in this list. Thus, $L(i, \bar{a}, k)$ establishes a path of length k between va_k and $v\bar{a}$ in case a_k is indeed the kth element of the tuple $\bar{a}$. The proof of the following lemma is left to the reader.

Lemma 1 *If the domain A and the predicate P_i of the structure $\mathcal{A}$ are regular languages then:*

1. *The language $T(A) = \bigcup_{a\in A} T(a)$ and the binary relation $E_1(A) = \{(x, y) \mid$ there is an $a \in A$ so that $\{x, y\}$ is an edge in $T(a)\}$ are regular.*

2. *The language $T(P_i) = \bigcup_{\bar{a}\in P_i} T(i, \bar{a})$ and the binary relation $E_2(P_i) = \{(x, y) \mid$ there is an $\bar{a} \in P_i$ so that $\{x, y\}$ is an edge in $T(i, \bar{a})\}$ are regular.*

3. *The language $L(i) = \bigcup_{\bar{a}\in P_i, 1\le k\le m_i} L(i, \bar{a}, k)$ and the binary relation $E_3(P_i) = \{(x, y) \mid \{x, y\}$ is an edge in some $L(i, \bar{a}, k)\}$ are regular.* □

Define $\mathcal{G}(\mathcal{A}) = (V(\mathcal{A}), E(\mathcal{A}))$, where $V(\mathcal{A})$ and $E(\mathcal{A})$ respectively are:

$$T(A) \cup \bigcup_{1\le i\le n} T(P_i) \cup \bigcup_{1\le i\le n} L(i) \quad \text{and} \quad E_1(A) \cup \bigcup_{1\le i\le n} E_2(P_i) \cup \bigcup_{1\le i\le n} E_3(P_i).$$

Theorem 3 *For the structure $\mathcal{A}$ and the graph $\mathcal{G}(\mathcal{A})$ the following are true:*

1. *$\mathcal{A}$ is automatic if and only if $\mathcal{G}(\mathcal{A})$ is automatic.*
2. *There is an isomorphism α between the group $Aut(\mathcal{A})$ of automorphisms of $\mathcal{A}$ and the group $Aut(\mathcal{G}(\mathcal{A}))$ of automorphisms of $\mathcal{G}(\mathcal{A})$. Moreover, $f \in Aut(\mathcal{A})$ is automatic if and only if $\alpha(f)$ is automatic.*
3. *A substructure $\mathcal{B}$ of $\mathcal{A}$ is automatic if and only if the subgraph $\mathcal{G}(\mathcal{B})$ of $\mathcal{G}(\mathcal{A})$ is automatic.*
4. *A structure $\mathcal{B}$ is automatically isomorphic to $\mathcal{A}$ if and only if the graph $\mathcal{G}(\mathcal{B})$ is automatically isomorphic to $\mathcal{G}(\mathcal{A})$.*
5. *From any automatic presentation of $\mathcal{A}$ an automatic presentation of $\mathcal{G}(\mathcal{A})$ can be constructed in linear time.*

Proof. Part 1). Lemma 1 shows that if $\mathcal{A}$ is automatic then so is $\mathcal{G}(\mathcal{A})$. Assume that $\mathcal{G}(\mathcal{A})$ is automatic. The set $D = \{x \mid x \text{ is the start of a 5-tag}\}$ is FA recognisable because it is FO-definable in $\mathcal{G}(A)$. For each i, $i = 1, \ldots, n$, consider the relation $R_i = \{(x_1, \ldots, x_n) \mid$ there is an x such that the distance between x_k and x is k and x is the start of a $5+i$–tag $\}$. This relation is FA recognisable. From the construction of $\mathcal{G}(\mathcal{A})$ we see that $\mathcal{A}$ and $(D, R_1, \ldots, R_n)$ are isomorphic.

Part 2). Let f be an automorphism of $\mathcal{A}$. Define $\alpha(f) : \mathcal{G}(\mathcal{A}) \to \mathcal{G}(\mathcal{A})$ as follows. If $f(a) = b$ then set $\alpha(f)(va) = vb$. Take a tuple $\bar{a} = (a_1, \ldots, a_n)$ so that $P_i(\bar{a})$ is true. Let $\bar{b} = (f(a_1), \ldots, f(a_{m_i}))$. Set $\alpha(f)(v\bar{a}) = v\bar{b}$. Now extend this partial map to an automorphism $\alpha(f)$ of $\mathcal{G}(\mathcal{A})$. This automorphism is unique. The fact that α is an isomorphism can be checked by using the definition of $\mathcal{G}(\mathcal{A})$. Assume that α is an automatic isomorphism. We want to show that $f(\alpha)$ is an automatic automorphism of $\mathcal{G}(\mathcal{A})$. Take an $x \in V(\mathcal{A})$. Then either $x \in T(a)$ or $T(i, \bar{a})$ or $x \in L(i, \bar{a}, k)$ for appropriate a, $\bar{a}$, i and k. Say, for instance $x \in T(a)$ and hence $x = vai$ for some $i = 0, \ldots, 5$ (in case $i = 0$ we assume that $va0$ is va). From the definition of $\alpha(f)$ we see that $\alpha(f)(x) = y$ iff $y \in T(\alpha(a))$ and $y = v\alpha(a)i$. This can be can be recognised by a finite automaton since α is automatic. We leave the other cases and the rest of the proof to the reader.

Part 3) follows from Part 1) and Part 4) from Part 2).

The sizes of the automata that recognise the languages $T(A)$ and $T(P_i)$ are proportional to the sizes of the automata recognising A and P_i. To recognise the language $L(i)$ we need to recognise words on $L(i, \bar{a}, k)$ paths, use the automaton recognizing P_i, and use the automaton that tells us if any given b is equal to the kth coordinate of $\bar{a}$. The size of the automaton that recognises $L(i)$ is thus proportional to the sizes of the automata presenting $\mathcal{A}$. Similarly, the size of the automaton recognising $E(\mathcal{A})$ is linear in the size of the presentation of $\mathcal{A}$. □

Note: Results of this section can be obtained from the fact that all automatic structures are interpretable in N_p [3]. We, however, provided a direct method of transforming structures into graphs rather than doing this indirectly by using interpretations in N_p.

4 On Automatic Equivalence Relations

Here we study automatic equivalence structures and provide several methods of constructing automatic equivalence structures with different types of isomorphism invariants. An **equivalence structure** is $\mathcal{E} = (E, \rho)$, where ρ is an equivalence relation on E. For $\mathcal{E}$ define the following two isomorphism invariants: $I_1(\mathcal{E}) = \{n \mid$ there is an equivalence class of size $n\}$, and $I_2(\mathcal{E}) = \{(n, m) \mid$ there are exactly m equivalence classes of size $n\}$. Clearly, $I_2(\mathcal{E})$ is a full isomorphism invariant in the sense that $\mathcal{E}$ and $\mathcal{E}'$ are isomorphic iff $I_2(\mathcal{E}) = I_2(\mathcal{E}')$. Also, $I_1(\mathcal{E})$ can be expressed in terms of $I_2(\mathcal{E})$. Our goal is to understand how these invariants behave in case $\mathcal{E}$ is an automatic structure. In [13] and [3] it is shown that $\mathcal{E}$ has an automatic presentation over a *unary* alphabet iff $I_1(\mathcal{E})$ is finite and there are finitely many infinite equivalence classes. The situation in general non-unary case is complex as the results of this section show.

For an equivalence structure $\mathcal{E}$ we define $\mathcal{E}_\omega$ and $\mathcal{E}_f$ as the restriction of $\mathcal{E}$ to all elements in infinite and finite equivalence classes, respectively.

Lemma 2 *If the equivalence structure $\mathcal{E}$ is automatic then so are $\mathcal{E}_f$ and $\mathcal{E}_\omega$. Moreover, $\mathcal{E}$ has an automatic presentation iff $\mathcal{E}_f$ does.*

Proof. Follows from the fact that $\mathcal{E}_f$ is definable by a $FO(\exists^\infty)$-formula and that $\mathcal{E}_\omega$ has always an automatic presentation. □

Thus, in characterising automatic equivalence structures $\mathcal{E}$, we can always assume that each equivalence class is finite. Therefore, from now on $I_1(\mathcal{E})$ does not contain ω, and if $(n, m) \in I_2(\mathcal{E})$ then n is finite.

Lemma 3 *If $\mathcal{E}$ is an automatic then it has an automatic presentation satisfying the property that if $(x, y) \in \rho$ then $|x| = |y|$.*

Proof. Suppose $\mathcal{E}$ is automatic over Σ. Consider an automatic linear order $\leq$ on E of type ω so that if $x \leq y$ then $|x| \leq |y|$. The set $\{x \mid x$ is the longest element in its equivalence class $\}$ is regular. Define a new domain E' over $((\Sigma \cup \{1\})^\star)^2$ as the set of pairs $(x, 1^n)$ where x is in the domain of $\mathcal{E}$ and n is the longest word in the ρ-equivalence class containing x. The set E' is FA-recognisable. Define the equivalence relation ρ' containing pairs $((x, 1^n), (y, 1^m))$ iff $(x, y) \in \rho$ and $n = m$. Then (E', ρ') is a desired automatic equivalence relation isomorphic to $\mathcal{E}$. □

Corollary 1 *(also see [3]) Let $\mathcal{E}$ be an infinite automatic equivalence relation where $|\Sigma| \geq 2$, and n_i be an increasing enumeration of the sizes of its equivalence classes. Then $n_i \leq 2^{0(i)}$.* □

Next we build equivalence structure from languages. Let L be a language. Define an equivalence structure $\mathcal{E}(L) = (E, \sim_L)$ with $E = L$. Two strings x and y are $\sim_L$-equivalent if $|x| = |y|$ and $x, y \in L$. Here is an easy lemma:

Lemma 4 *If L is regular then $\mathcal{E}(L)$ is an automatic structure.* □

The next series of results provide several examples and constructions for building automatic equivalence structures whose isomorphism invariant $I_1(\mathcal{E})$ exhibits non-trivial behaviour.

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 $n \in \omega$. The following is implicit in [15].

Lemma 5 *For any polynomial function p whose coefficients are positive integers there is a regular language L_p whose growth function is p.*

Proof. Note that if L_1 and L_2 have growth rates p_1 and p_2, respectively, and $L_1 \bigcap L_2 = \emptyset$ then their union has growth rate $p_1 + p_2$. So it is sufficient to exhibit for each $k \in N$ a language L_{n^k} with growth rate n^k.

For $w \in \Sigma^\star$, write w^+ for $ww^\star$. Note that $A_k = 0^+1^+ \cdots k^+$ has growth $\binom{n-1}{k}$. Consider the languages $B_k = 0^+1^+ \cdots (k- = 1)^+k^*$. Then $B_k = A_{k-1} \cup A_k$. Hence the growth of B_k is $\binom{n-1}{k} + \binom{n-1}{k-1} = \binom{n}{k}$. Consider the languages C_k defined as the disjoint union of $k!$ copies of B_k. Then C_k has growth $n(n-1)\cdots(n-k+1)$ which we write as $n^{\underline{k}}$. We now make use of the standard identity $x^k = \Sigma_{i=0}^{k} S(k,i) x^{\underline{i}}$ where the $S(k,i)$ are Stirling numbers of the first kind; that is the number of ways of partitioning a set of size k into i non-empty subsets. So $L_{n^k} = \cup_{k=0}^{n} \cup_{S(n,k)} C_k$, where the unions are taken to be disjoint, has the required growth. □

Lemma 6 *For any exponential function $e(n)$ of the form k^{an+b}, where $2 \leq k$ and a, b are positive integers, there exists a regular language whose growth function is exactly e.*

Proof. Let $\Sigma = \{1, 2, \cdots, k^a\}$. Then $L = \Sigma^*$ has growth k^{an}. The disjoint union of k^b many copies of L has growth k^{an+b}. □

It is worth to note that the growth level of any relular language is bounded by either a polynomial or an exponential (see [1]).

Theorem 4 *For any function f which is either a polynomial p whose coefficients are positive integers or exponential function k^{an+b}, where $k \geq 2$ and a, b are fixed positive integers, there exists an automatic equivalence relation $\mathcal{E}$ such that $I_1(\mathcal{E}) = \{f(n) \mid n \geq 1\}$ and $I_2(\mathcal{E}) = \{(f(n), c) \mid n \geq 1\}$, with $c \leq \omega$ being a constant.*

Proof. From Lemma 5 and Lemma 6 there exists a regular language L whose growth function is identical to f. By Lemma 4 the automatic equivalence structure $\mathcal{E}(L)$ is a desired one. The theorem for case $c = 1$ is proved. Now note that disjoint union of automatic equivalence structures are automatic. □

The next result shows that the second invariant $I_2(\mathcal{E})$ of automatic equivalence structures can also exhibit a complex behaviour. The invariant $I_2(\mathcal{E})$ defines the

height function $h_{\mathcal{E}}$ as follows: $h_{\mathcal{E}}(n) = m$ if and only if $(n, m) \in I_2(\mathcal{E})$. Finally, for two functions f, g with domains N, their **Dirichlet convolution** is $(f \star g)(n) = \sum_{ab=n} f(a)g(b)$.

Proposition 1 *Let $\mathcal{H}$ be the class of height functions of automatic equivalence structures. Then $\mathcal{H}$ is closed under addition and Dirichlet convolution.*

Proof. For addition consider the disjoint union, and for convolution the direct product $\mathcal{E}_1 \times \mathcal{E}_2 = (E, \rho)$, where $E = E_1 \times E_2$ and $\rho = \{((x, y), (a, b)) \mid (x, a) \in \rho_1 \;\&\; (y, b) \in \rho_2\}$. □

Thus, for instance there is an automatic equivalence structure $\mathcal{E}$ so that $I_1(\mathcal{E}) = \omega \setminus \{0\}$ and $h_{\mathcal{E}}(n)$ is the number of all pairs (i, j) such that $i \cdot j = n$.

We now give an automata-theoretic characterisation of automatic equivalence structures (with finite equivalence classes).

Definition 5 *An automatic binary relation R is a* **regular enumeration** *of a family $\mathcal{F}$ of regular sets if $\mathcal{F} = \{R_x \mid x \in dom(R)\}$, where R_x is the projection $\{u \mid (x, u) \in R\}$.*

We think of R as a mapping from $\mathrm{dom}(R)$ onto $\mathcal{F}$. If R is a regular enumeration then one can always construct a regular one to one enumeration of $\mathcal{F}$ since the relation $\{(x, y) \mid R_x = R_y\}$ is FA-recognisable.

Let R be a one to one regular enumeration of $\mathcal{F}$ such that $R_x \bigcap R_y = \emptyset$ for $x \neq y$. Consider the structure $\mathcal{E}(R)$ with domain $\bigcup_{x \in \mathrm{dom}(R)} R_x$ and binary relation $\{(u, v) \mid \exists x \in \mathrm{dom}(R) : u, v \in R_x \;\&\; |u| = |v|\}$. Then $\mathcal{E}(R)$ is an equivalence structure. The proof of the following is immediate.

Proposition 2 *The structure $\mathcal{E}(R)$ is an automatic equivalence structure.* □

Example 1 *Let X and Y be nonempty regular languages such that no two words in Y are prefixes of each other. Consider the family $\mathcal{F} = \{yX \mid y \in Y\}$. The mapping $R : y \to yX$ is a one to one and regular enumeration of $\mathcal{F}$. Hence $\mathcal{E}(R)$ is automatic.*

The construction of $\mathcal{E}(R)$ is as general as possible because one can reverse the construction as follows. Let $\mathcal{E} = (E, \rho)$ be an automatic equivalence structure. We may assume that $(u, v) \in \rho$ implies that $|u| = |v|$. Form the set W of all the minimal elements (with respect to an automatic order $\leq$ of type ω on the set of all words) from each equivalence class. Consider $R = \{(w, v) \mid w \in W \text{ and } (w, v) \in \rho\}$. Clearly, R is a regular one to one enumeration of ρ-equivalence classes and $\mathcal{E}(R)$ is isomorphic to $\mathcal{E}$.

5 On Automatic Linearly Ordered Sets

Here we explain how to convert automatic equivalence structures $\mathcal{E}$ into a certain type of linearly ordered (lo) sets $\mathcal{L}_{\mathcal{E}}$ so that $\mathcal{L}_{\mathcal{E}}$ and $\mathcal{E}$ can be recovered from each other. This will show that the study of automatic lo sets is at least as complex as automatic equivalence structures.

Let $\mathcal{L} = (L, \leq)$ be a lo set. For $x, y \in L$ define the **interval** $[x, y] = \{z \mid x \leq z \leq y\}$ if $x \leq y$ and $[x, y] = \{z \mid y \leq z \leq x\}$ if $y < x$. We say that the elements $x, y \in L$ are **in the same block** if $[x, y]$ is finite, and we write B for this equivalence relation on L. Having the relation B, define a new lo set $\mathcal{L}_B$ by factorizing L by B as follows. The elements of $\mathcal{L}_B$ are the equivalence classes; and $x_B \leq_B y_B$ if $x \leq y$, where x_B is the equivalence class containing x.

Lemma 7 *If $\mathcal{L}$ is an automatic lo set then the block relation B is FA recognizable. Hence the factor $\mathcal{L}_B$ is an automatic linear order.* □

The idea of factorization suggests relating automatic equivalence relations with lo sets. We need a definition.

Definition 6 *We denote Q the type of the lo set of rationals. We say that a lo set $\mathcal{L}$ has Q-rank 1 if $\mathcal{L}_B$ is isomorphic to either Q or $1 + Q$ or $Q + 1$, where $1 + Q$ ($Q + 1$) is the the linearly ordered set of rationals with the least (greatest) element.*

Let $\mathcal{L}$ be a linearly ordered set of Q-rank 1. Define the set $I(\mathcal{L}) = \{(n, m) \mid \mathcal{L}$ has m blocks of size $n \geq 2\}$. Write $D(x)$ for the unary relation on L stating that x is in some dense interval. Define $\mathcal{E}(\mathcal{L})$ as the equivalence structure with domain L and relation $B \cup (D \times D)$. Thus, any lo set $\mathcal{L}$ of Q-rank 1 naturally induces an equivalence structure $\mathcal{E}(\mathcal{L})$. The following proposition thus follows:

Proposition 3 *If $\mathcal{L}$ is an automatic lo set of Q-rank 1 then $\mathcal{E}(\mathcal{L})$ is an automatic equivalence structure.* □

The next theorem shows how to construct automatic lo sets of Q-rank 1 given an automatic equivalence structure thus conversing the proposition above.

Theorem 5 *From any automatic equivalence structure $\mathcal{E}$ it is possible to construct an automatic lo set $\mathcal{L}_{\mathcal{E}}$ of Q-rank 1 so that $I(\mathcal{L}_{\mathcal{E}}) = I_2(\mathcal{E})$.*

Proof. Let $\mathcal{E} = (E, \rho)$ be automatic, and $\prec$ be an automatic well order of type ω on E. Write $\prec_A$ for $\prec$ restricted to set A. Order the equivalence classes of $\mathcal{E}$ by $\prec'$ as follows. We write $x_\rho \prec' y_\rho$ iff the $\prec$–minimal element in x_ρ (the equivalence class containing x) is less than the $\prec$–minimal element in y_ρ. List the equivalence classes of $\mathcal{E}$ as $\{B_i\}$ for $i \in \omega$ where $i \leq j$ iff $B_i \prec' B_j$. The required linear ordering $\mathcal{L}_{\mathcal{E}}$ is then $\Sigma_i(\mathcal{B}_i + \mathcal{D})$, where $\mathcal{B}_i = (B_i, \prec_{B_i})$ and $\mathcal{D}$ is a linear ordering of type Q of rationals. The lo set $\mathcal{L}$ has Q-rank 1. This lo set possesses an automatic presentation which can be shown by using the fact that the linearly ordered set of rationals has one [12]. □

Corollary 2 *For any function g which is either a polynomial p whose coefficients are positive integers or exponential function k^{an+b}, where $k \geq 2$ and a, b are fixed positive integers, there exists an automatic linear order $\mathcal{L}$ of Q-rank 1 such that $I(\mathcal{L}) = \{(g(n), 1) \mid n < \omega\} \cup \{(\omega, 1)\}$.* □

6 On Automatic Permutation Structures

A permutation structure is $\mathcal{A} = (A, f)$, where f is a bijection on A. For an $a \in A$, the set $\{f^i(a) \mid i \in \omega\}$ is an **orbit of** f. As for the equivalence structures, define two isomorphism invariants $I_1(\mathcal{A}) = \{n \mid \text{there is an orbit of size } n\}$, and $I_2(\mathcal{A}) = \{(n, m) \mid \text{there are exactly } m \text{ orbits of size } n\}$. Then I_2 is a full isomorphism invariant. In [14] and in [3] it is shown that $\mathcal{A}$ 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.

Any automatic equivalence structure $\mathcal{E}$ over $\Sigma^\star$ can be turned into an automatic permutation structure $\mathcal{A}(\mathcal{E})$ as follows. Let $\leq$ be an automatic well order of type ω on $\Sigma^\star$. For each $x \in E$ we proceed as follows. If x is in $\mathcal{E}_f$ and is not the maximal element in its equivalence class then $f(x)$ is the minimal $y \geq x$ which is ρ-equivalent to x. Otherwise, $f(x)$ is the minimal element in the equivalence class containing x. If $x \in \mathcal{E}_\omega$ then f transforms the equivalence class containing x into $\mathbb{Z}$-type chain, namely the structure isomorphic to $(\mathbb{Z}, S)$ where S is the successor function on $\mathbb{Z}$. Note that $I_2(\mathcal{E}) = I_2(\mathcal{A}(\mathcal{E}))$. Hence, the result similar to Theorem 4 holds true for automatic permutation structures.

We now show that the isomorphism invariants $I_1(\mathcal{A})$ of automatic permutations can be related to the running times of Turing machines (TMs). 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.

Lemma 8 *For any TM T the configuration graph $\mathcal{C}(T)$ is automatic. Further, the set of all vertices with with outdegree (indegree) 0 is FA-recognisable.* □

Definition 7 *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. For the sake of completeness, we sketch the proof.

A transition of T is a quintuple $(\sigma, q, \delta, d, s) \in \Delta$ where $\sigma, \delta \in \Sigma$, $q, s \in Q$ and $d \in \{L, R\}$. On input w, R runs as T would, but also saves each of T's transitions on a separate 'history' tape. Once the simulated T has halted, R copies the output to another tape. It then retraces the steps that T took, in reverse, deleting the saved transitions one at a time, resulting in R having the original input w printed on one tape, a blank 'history' tape, and the output $T(w)$ on the third tape. This three tape TM R is itself simulated by a single tape machine. So, the reason that R is reversible is that if a configuration c of T has indegree greater than 1, then the transitions corresponding to each edge into c are distinct. Since the corresponding configuration of R codes these transitions, the particular configuration of T which preceded c is uniquely determined.

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 6 *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. Let Q be the set of states of T. Consider the set $\tilde{Q} = \{\tilde{q} \mid q \in Q\}$. We may assume that $Q \cap \tilde{Q} = \emptyset$. Let the configuration graph of T be $\mathcal{C}(T) = (C, E)$. Consider $X = \{c \mid c$ is not initial and there is no d for which $(d, c) \in E\}$. Clearly X is FA recognizable. For each $x \in X$ consider the set $\{xo^i \mid i \in \omega\}$, where o is a new symbol.

Now we define a new graph $\mathcal{G}_1 = (V_1, E_1)$, with $V_1 = C(T) \cup \{xo^i \mid i \in \omega, x \in X\}$, and $E_1 = E \cup \{(xo^{i+1}, xo^i \mid x \in X, i \in \omega\}$. This graph is clearly automatic. For each $c \in V_1$ let $\tilde{c}$ be obtained by replacing the state q appearing in c with $\tilde{q}$. Define the following set $\tilde{V_1} = \{\tilde{c} \mid c \in V_1\}$.

On $V_1 \cup \tilde{V_1}$ we now define the following permutation g. If $(c, d) \in E_1$ then $g(c) = d$ and $\tilde{g}(\tilde{d}) = \tilde{c}$. If c is a final configuration then $g(c) = \tilde{c}$. If c is an initial configuration then $g(\tilde{c}) = c$. Thus, $(V_1 \cup \tilde{V_1}, g)$ is an automatic permutation structure. Moreover, $T(w) = n$ if and only if the structure $(V_1 \cup \tilde{V_1}, g)$ has an orbit of size $2n$. Let $f = g^2$. Then it is not hard to check that the structure $(V_1 \cup \tilde{V_1}, f)$ is a desired automatic permutation structure. □

As a corollary of the theorem we obtain the following undecidability result.

Theorem 7 *It is undecidable whether two automatic permutation 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 with only infinitely many infinite chains of type $\mathbb{Z}$. □

Blumensath [4] also proved undecidability of the isomorphism problem for automatic structures by an implicit construction of reversible TMs.

7 Conclusion

We would like to have a characterisation of natural isomorphism invariants of automatic structures. Ideally, we would like these characterizations to give us some useful information about the complexity-theoretic nature of the structures from logical and algebraic points of view. When the structures are automatic over a unary alphabet, characterization for some common structures are known [3],[13]. These characterizations imply that theories of these structures are computationally accessible and show the algebraic nature of the structures. In this paper our aim was to show difficulties involved in the non-unary case. Theorem 3 reduces the study of automatic structures to automatic graphs. Theorem 4 and Theorem 5 are initial steps in understanding the isomorphism invariants of some simple structures. Finally, Theorem 6 exhibits a nontrivial relationship between running times of TMs and automatic structures. Clearly, more work remains to be done in understanding automatic structures and their complexities. We deal with some of them in upcoming papers.

References

[1] M. Benedikt, L. Libkin, T. Schwentick, L. Segoufin A Model-theoretic Approach to Regular String Relations. In: *Proc. 16th IEEE Symp. on Logic in Computer Science*, 2001, 431–44= 0.

[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, personal communication.

[5] A. Blumensath, E. Grädel, Automatic Structures. In: *Proc. 15th IEEE Symp. on Logic in Computer Science*, 2000, 51–62.

[6] C. Delhomme, V. Goranko, T. Knapik Automatic Linear Orderings. personal communication.

[7] S. Eilenberg, C.C. Elgot, J.C. Shepherdson, Sets Recognised by n–Tape Automata. *Journal of Algebra* **13 4** (1969), 447–464.

[8] C.C Elgot, J.E. Mezei, On Relations Defined by Generalised Finite Automata. *IBM J. Res. Develop.* **9** (1965), 47–68.

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

[10] E. Grädel Simple Interpretations among Complicated Theories. *Information Processing Letters* **35** (1990), 235–238.

[11] W. Hodges *Model Theory*. Encyclopedia of Mathematics and its Applications **42**, Cambridge University Press, 1993.

[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*, to appear.

[14] S. Rubin, Finite Automata and Well Ordered Sets. *New Zealand Journal of Computing* **7 2** (1999), 39–46.

[15] J. Shallit, A. Szilard, S. Yu, K. Zhang Characterising Regular Languages with Polynomial Densities In: Proceedings 17th International Symposium Mathematical Foundations of Computer Science, *Lecture Notes in Computer Science* **629** (1992), 494–503.