Some results on automatic structures

Hajime Ishihara
School of Information Sciences
JAIST, Japan
ishihara@jaist.ac.jp

Bakhadyr Khoussainov
Computer Science Department
The University of Auckland, New Zealand
bmk@cs.auckland.ac.nz

Sasha Rubin
Mathematics Department
The University of Auckland, New Zealand
rubin@math.auckland.ac.nz

Abstract

We study the class of countable structures which can be presented by synchronous finite automata. We reduce the problem of existence of an automatic presentation of a structure to that for a graph. We exhibit a series of properties of automatic equivalence structures, linearly ordered sets and permutation structures. These serve as a first step in producing practical descriptions of some automatic structures or illuminating the complexity of doing so for others.

1. Introduction

In this paper we investigate those countable structures that can be presented, 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. Here by isomorphism invariant of a structure we mean a property of the structure expressible in a certain formalism – say, first or second order logic. If such a property is preserved under isomorphisms of the structure, we call it an isomorphism invariant. For example if the structure in question is an equivalence relation, then the number of equivalence classes of any given size is an isomorphism invariant. 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 $N_p = (\mathbb{N}, +, |_p)$, where $x|_p y$ iff $x = p^n$ and $y = kx$ for some $n, k \in \mathbb{N}$. 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. recognisability and definability. The other reason is of complexity-theoretic nature. It is known that the first order theory of any automatic structure is decidable [11]. A natural question arises as to which automatic structures are feasible and which are not. Blumensath and Grädel [4] [5] [9] 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 – that is, the problem of checking whether a formula is satisfied in a given structure – can be intractable. On the other hand, there are examples of automatic structures, e.g. structures presented by finite automata over unary alphabets or the rational numbers with the natural ordering, for which the expression complexity of the model checking problem 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 defini-

Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS'02)
1043-6871/02 $17.00 © 2002 **IEEE**

tions and state some known results in the area. The section on automatic graphs is devoted to exhibiting a functor from the class of all automatic structures into the class of all automatic graphs. We prove 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 construct automatic permutations from automatic equivalence structures. We also show the relationship between permutation structures and the running times of reversible Turing machines. As a consequence of this relationship we will prove that the isomorphism problem for permutation structures is undecidable.

We briefly note related work. A systematic study of interactions between automata and algebraic structures began from the work of Cannon and Thurston on automatic groups [8]. This was generalised by Khoussainov and Nerode in [11] but motivated from a point of view of computable model theory. A significant work in the understanding of automatic structures is due to Blumensath and Grädel [4] [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 unpublished work Delhomme, et al. [6] show that the minimal ordinal without an automatic presentation 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 $\sigma_1\sigma_2 \dots \sigma_n$ ($\sigma_i \in \Sigma$) is a sequence of states, 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 $0 \le i \le n-1$. 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, $D \subset \Sigma^\star$ 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 standard basics in finite automata theory.

We now introduce *synchronous n–tape automata* which recognise n–ary relations. The following description is based on Eilenberg et al. [7]. 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 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 *Let $\Sigma_\diamond$ be $\Sigma \cup \{\diamond\}$ where $\diamond \notin \Sigma$. The* convolution of a tuple *$(w_1, \cdots, w_n) \in \Sigma^{*n}$ is the tuple $(w_1, \cdots, w_n)^\diamond \in (\Sigma_\diamond)^{*n}$ formed by concatenating the least number of blank symbols, $\diamond$, 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 relation $R^\diamond \subset (\Sigma_\diamond)^{*n}$ formed as the set of convolutions of all 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 $\mathcal{A} = (A, R_1^A, \dots, R_k^A, c_0^A, \dots, c_t^A)$ 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 $(R_1^{n_1}, \dots, R_k^{n_k}, c_0, \dots, c_t)$ 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 $\mathcal{A} = (A, R_1^A, \dots, R_k^A, c_0^A, \dots, c_t^A)$ 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 motivates studying automatic structures from a complexity-theoretic point of view.

Theorem 1 *[11] 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.* $\square$

Blumensath and Grädel extended this result in [4] by showing that the theorem holds even if one considers the

Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS'02)
1043-6871/02 $17.00 © 2002 **IEEE**

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, namely 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, and 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. For a non-unary alphabet Σ consider the structures $\mathcal{W}(\Sigma) = (\Sigma^\star, (\sigma_a)_{a\in\Sigma}, \preceq, \mathrm{el})$ and $N_p = (\mathbb{N}, +, |_p)$, 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 $N_{|\Sigma|}$ and $\mathcal{W}(\Sigma)$ are mutually interpretable. Here is an important theorem [5]:

Theorem 2 *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 vertices $V(\mathcal{A})$ 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 ([10] Theorem 5.5.1) and possesses natural algebraic and automata-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 $0 \le i < n$, $\{n, 1\}$ 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. Furthermore 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 edges $\{va, va1\}$, $\{vak, va(k+1)\}$ for $1 \le k \le 4$, and $\{va5, va1\}$. Here va is the start vertex of the tag $T(a)$, which we think of as representing the element a of the structure $\mathcal{A}$.

For $1 \le i \le n$, we code the predicate R_i of arity n_i as follows. Firstly, with each tuple $\bar{a} = (a_1,\ldots,a_{n_i})$ for which $R_i(\bar{a})$ is true we associate a $(5+i)$-tag $T(i,\bar{a})$ with vertices $v\bar{a}$, $v\bar{a}1$, …, $v\bar{a}(5+i)$ and edges $\{v\bar{a}, v\bar{a}1\}$, $\{v\bar{a}k, v\bar{a}(k+1)\}$ for $1 \le k \le i+4$, and $\{v\bar{a}(5+i), v\bar{a}\}$. Secondly, with each tuple $\bar{a} = (a_1,\ldots,a_{n_i})$ for which $R_i(\bar{a})$ is true and where the kth element of this tuple is a_k, we associate the graph $L(i,\bar{a},k)$ consisting of the k vertices va_k, $v\bar{a}k1$, $v\bar{a}k2$, $v\bar{a}k3$, …, $v\bar{a}kk$, $v\bar{a}$ 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 R_i of the structure $\mathcal{A}$ are regular language 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 $T(R_i) = \bigcup_{\bar{a}\in R_i} T(i,\bar{a})$ and the binary relation $E_2(R_i) = \{(x,y) \mid$ there is an $\bar{a} \in R_i$ so that $\{x,y\}$ is an edge in $T(i,\bar{a})\}$ are regular.*
3. *The language $L(i) = \bigcup_{\bar{a}\in R_i, 1\le k\le n_i} L(i,\bar{a},k)$ and the binary relation $E_3(R_i) = \{(x,y) \mid \{x,y\}$ is an edge in some $L(i,\bar{a},k)\}$ are regular.* $\square$

Now define $\mathcal{G}(\mathcal{A}) = (V(\mathcal{A}), E(\mathcal{A}))$, where $V(\mathcal{A})$ is

$$T(A) \cup \bigcup_{1\le i\le n} T(R_i) \cup \bigcup_{1\le i\le n} L(i)$$

and $E(\mathcal{A})$ is

$$E_1(A) \cup \bigcup_{1\le i\le n} E_2(R_i) \cup \bigcup_{1\le i\le n} E_3(R_i).$$

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.*

Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS'02)
1043-6871/02 $17.00 © 2002 **IEEE**

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 every i, $i = 1, \ldots, n$, consider the relation $R_i = \{(x_1, \ldots, x_n) \mid \text{there is an } x$ such that the distance between x_k and x is k and x is the start of a $5 + i$–tag $\}$. This relation is FO-definable and hence 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. Hence $\mathcal{A}$ is automatic.

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 $R_i(\bar{a})$ is true. Let $\bar{b} = (f(a_1), \ldots, f(a_{n_i}))$. 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 f is an automatic automorphism of $\mathcal{A}$. We want to show that $\alpha(f)$ is an automatic automorphism of $\mathcal{G}(\mathcal{A})$. Take an $x \in V(\mathcal{A})$. Then either $x \in T(a)$ or $x \in T(i, \bar{a})$ 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 $y \in T(f(a))$ and $y = vf(a)i$. This can be can be recognised by a finite automaton since f 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 $T(R_i)$ are proportional to the sizes of the automata recognising A and R_i. To recognise the language $L(i)$ we need to recognise words on $L(i, \bar{a}, k)$ paths, use the automaton for R_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 $\mathcal{A}$. $\Box$

Note: Results of this section can be obtained from the fact that all automatic structures are interpretable in N_p [4]. 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

In this section we study automatic equivalence structures. We provide several methods of constructing automatic equivalence structures with different types of isomorphism invariants. An **equivalence structure** $\mathcal{E}$ is (E, ρ) where ρ is an equivalence relation on domain E. For $\mathcal{E}$ define the following two isomorphism invariants: $I_1(\mathcal{E}) = \{n \leq \omega \mid \text{there is an equivalence class of size } n\}$, and $I_2(\mathcal{E}) = \{(n, m) \mid \text{there are exactly } m \text{ equivalence classes of size } n\}$. 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 automatic. In [12] and [4] 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 results of this section show that the situation in the general case is complex.

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 $\mathcal{E}_\omega$ always has an automatic presentation. $\Box$

Thus, in characterising automatic equivalence structures $\mathcal{E}$, we can assume that each equivalence class is finite. Therefore, from now on we always assume that $I_1(\mathcal{E})$ 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 equivalence structure then it has an automatic presentation (E', ρ') 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 the type ω so that if $x \leq y$ then $|x| \leq |y|$. We remark that one can always extend an automatic structure by such an order. The set $\{x \mid x \text{ is the } \leq\text{-longest element in its equivalence class }\}$ is regular. Define a new domain E' over alphabet $((\Sigma \cup \{1\})^\star)^2$ as the set of pairs $(x, 1^n)$ where x is in the domain of $\mathcal{E}$ and n is the length of the $\leq$-longest word in the ρ-equivalence class containing x. Note that E' is FA-recognisable. Define a new equivalence relation ρ' containing pairs $((x, 1^n), (y, 1^m))$ iff $(x, y) \in \rho$ and $n = m$. Then (E', ρ') is an automatic equivalence relation isomorphic to $\mathcal{E}$. $\Box$

Corollary 1 *(also see [4]) 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 $n_i \leq 2^{O(i)}$.* $\Box$

Next we build equivalence structures from languages $L \subset \Sigma^*$. 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.* $\Box$

One would like to characterise automatic equivalence structures in terms of $I_1(\mathcal{E})$ and $I_2(\mathcal{E})$. The next series of results provide several examples and standard constructions for building automatic equivalence structures whose isomorphism invariant $I_1(\mathcal{E})$ exhibits nontrivial 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 $n < \omega$. The following is implicit in [14].

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 $k \in \mathbb{N}$ 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 $L_{n^k} = \cup_{i=0}^k \cup_{S(k,i)} C_i$, where the unions are taken to be disjoint, has the required growth. $\Box$

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}. $\Box$

We note that for any regular language L its growth level is either bounded by a polynomial or is bounded by 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 the desired one. The theorem for case $c = 1$ is proved. Now note that for $c \leq \omega$, the c-fold disjoint union of automatic equivalence structures is automatic. $\Box$

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 $(n, m) \in I_2(\mathcal{E})$. Consider two functions f, g with domains $\mathbb{N}$. Their **Dirichlet convolution** is $(f \star g)(n) = \sum_{ab=n} f(a)g(b)$, and their **Cauchy product** is $(f\#g)(n) = \sum_{a+b=n} f(a)g(b)$.

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

Proof. Let $\mathcal{E}_i = (E_i, \rho_i)$ for $i = 1, 2$ be two automatic equivalence structures with height functions f and g respectively. Without loss of generality, assume that the domains E_1 and E_2 are disjoint. Define the automatic equivalence structure $\mathcal{E}_{E_1 \cup E_2}$ as their disjoint union; that is, we assume that E_1 and E_2 are disjoint, and define the domain as $E_1 \cup E_2$ and the relation as $\rho = \rho_1 \cup \rho_2$. Then the height function of $E_{E_1 \cup E_2}$ is $f + g$.

For the Dirichlet convolution define the direct product $\mathcal{E}_1 \times \mathcal{E}_2$ as the equivalence structure with domain $E_1 \times E_2$. Define two pairs (x_1, y_1) and (x_2, y_2) to be related if $(x_1, x_2) \in \rho_1$ and $(y_1, y_2) \in \rho_2$. Then this equivalence structure is automatic and has has height $f * g$.

For the Cauchy product, let $T_i \subset E_i$ be the unary predicate that picks out the lexicographically least element from each equivalence class of $\mathcal{E}_i$. Define an equivalence structure with domain $(T_1 \times E_2) \cup (T_2 \times E_1)$. Define two pairs (x_1, y_1) and (x_2, y_2) to be related if either $[(y_1, y_2) \in \rho_{E_1 \cup E_2}$ and $x_1 = x_2]$ or $\{(x_1, y_2), (x_2, y_1)\} \subset \rho_{E_1 \cup E_2}$. Then this equivalence structure is automatic and has height $f\#g$. $\Box$

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 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 $\bigcup_{x \in \text{dom}(R)} R_x$ and binary relation $\{(u,v) \mid \exists x \in \text{dom}(R) : u,v \in R_x \ \& \ |u| = |v|\}$. Then $\mathcal{E}(R)$ is an equivalence structure. The proof of the following is immediate.

Proposition 2 *An equivalence structure is automatic if and only if it is of the form $\mathcal{E}(R)$ for a regular enumeration R.*

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 $R = \{(w,v) \mid w \in W, v \in V, (w,v) \in \rho\}$. Clearly, R is a regular one to one enumeration of the 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 certain types 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 factorising 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 recognisable. Hence the factor $\mathcal{L}_B$ is also automatic.* $\Box$

The idea of factorisation 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 lo set of rationals with the least (greatest) element.*

Let $\mathcal{L}$ be a lo set of Q-rank 1. Define the set $I(\mathcal{L}) = \{(n,m) \mid \mathcal{L}$ has m blocks of size $n\}$. Write $U(x)$ 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 $B \cup (U \times U)$.

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*

$$\begin{aligned} I_2(\mathcal{E}(\mathcal{L}_{\mathcal{E}})) &= \{(n,m) \mid (n,m) \in I_2(\mathcal{E}), n < \omega\} \\ &\cup \{(\omega, m+1) \mid (\omega, m) \in I_2(\mathcal{E})\}. \end{aligned}$$

Proof. Suppose that $\mathcal{E} = (E, \rho)$ has an automatic presentation. Let $\prec$ be an automatic well ordering on the set E. Write $\prec_A$ for $\prec$ restricted to domain A. Order the equivalence classes of $\mathcal{E}$ by $\prec$ as follows. We write $x_\rho \prec_E y_\rho$ iff the $\prec$–minimal element in x_ρ is $\prec$ the $\prec$–minimal element in y_ρ. List the equivalence classes of $\mathcal{E}$ as $\{B_i\}$ for $i < \omega$ where $i \leq j$ iff $B_i \prec_E B_j$. The required lo set $\mathcal{L}_{\mathcal{E}}$ is then $\Sigma_i(\mathcal{B}_i + \mathcal{D})$, where $\mathcal{B}_i = (B_i, \prec_{B_i})$ and $\mathcal{D}$ is an automatic lo of type Q over a new alphabet. The lo set $\mathcal{L}$ has an automatic presentation. To this end we note that if x_i is the $\prec$–minimal element of B_i, then we order the set $x_i D = \{x_i d \mid d \in D\}$ as $x_i d_1$ is less than $x_i d_2$ iff $d_1 \leq_D d_2$. $\Box$

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 lo set $\mathcal{L}$ of Q-rank 1 such that $I(\mathcal{L}) = \{(g(n), 1) \mid n < \omega\} \cup \{(\omega, 1)\}$.* $\Box$

6. On Automatic Permutation Structures

A permutation structure $\mathcal{A}$ is (A, f) where f is a bijection on A. For an $a \in A$, the set $\{f^i(a) \mid i < \omega\}$ is an **orbit of** f. As for the equivalence structures, define two isomorphism invariants $I_1(\mathcal{A}) = \{n \leq \omega \mid$ there is an orbit of size $n\}$, and $I_2(\mathcal{A}) = \{(n,m) \mid$ there are exactly m orbits of size $n\}$. Then I_2 is a full isomorphism invariant. In [13] and in [4] 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 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 into $\mathbb{Z}$-type chain, namely the structure isomorphic to $(\mathbb{Z}, S)$

where S is the successor function on the integers. 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 want to show that the isomorphism invariants $I_1(\mathcal{A})$ of automatic permutation structures can be related to the running times of Turing machines (TMs). Let T be a TM over input alphabet Σ. Its configuration graph $C(T)$ consists of the set of all configurations of T, with an edge from c to d if T can move from c to d in a single transition. Here is a simple lemma:

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.* $\Box$

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, so that $Time_T(w) = \omega$ if T does not halt on w.

Theorem 6 *For every reversible TM T, there corresponds an automatic permutation structure $\mathcal{A}(T)$ for which $I_1(\mathcal{A}(T)) = \{Time_T(w) \mid w \in \Sigma^\star\}$.*

Proof. Let the configuration graph of T be $\mathcal{C}(T) = (C, E)$. Assume that the unique initial state of T is not a final state. Then Time$(w) > 0$ for all w. Let $I, O \subset C$ respectively be the set of configurations with indegree 0 and outdegree 0. For each $c \in C$, let $\tilde{c}$ be a symbol not in C. Let $\tilde{C} = \{\tilde{c} \mid c \in C\}$. Define a permutation function f on domain $C \cup \tilde{C}$ as follows. If $(c, d) \in E$ then $f(c) = d$ and $f(\tilde{d}) = \tilde{c}$; If $c \in I$ and $(c, d) \in E$ then $f(\tilde{d}) = \tilde{c}$; If $d \in O$ and $(c, d) \in E$ then $f(\tilde{d}) = \tilde{c}$.

Consider the structure $(C \cup \tilde{C}, E, I, O, \sim)$ where $\sim$ is the function with domain C mapping c to $\tilde{c}$. It is automatic over alphabet $\Sigma \cup \tilde{\Sigma}$. We conclude that the structure $(C \cup \tilde{C}, f)$ is also automatic. Factor this structure by the equivalence relation satisfying pairs of the form $(c, \tilde{c})$ and $(\tilde{c}, c)$ for $c \in I \cup O$. Write (D, g) for the resulting automatic permutation structure. If $Time_T(w) = n$ then the (D, g) has an orbit of length $2n$. So, define the desired structure $\mathcal{A}(T) = (D, h)$ where $h = g \circ g$. $\Box$

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 $\mathbb{Z}$. $\Box$

Blumensath [3] 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 characterisations 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, characterisation for some common structures are known [4],[12]. These characterisations 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. The paper shows that more work remains to be done in understanding automatic structures and their complexities. We deal with some of them in upcoming papers.

We thank the referees for their general comments. In particular, one referee suggested including the Cauchy product in Proposition 1.

References

[1] M. Benedikt, L. Libkin, T. Schwentick, and L. Segoufin. A Model-Theoretic approach to regular string relations. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS-01)*, pages 431–440, June 16–19 2001.

[2] C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, Nov. 1973.

[3] A. Blumensath. personal communication.

[4] A. Blumensath. *Automatic Structures*. Diploma thesis, RWTH Aachen, 1999.

[5] A. Blumensath and E. Grädel. Automatic structures. In *15th Symposium on Logic in Computer Science (LICS' 00)*, pages 51–62, June 2000.

[6] C. Delhomme, V.Goranko, and T.Knapik. Automatic linear orderings. personal communication.

[7] S. Eilenberg, C. Elgot, and J. Shepherdson. Sets recognised by n–tape automata. *Journal of Algebra*, 13(4):447–464, 1969.

[8] D. B. H. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. *Word processing in groups*. Jones and Bartlett, 1992.

[9] E. Grädel. Simple interpretations among complicated theories. *Information Processing Letters*, 35(5):235–238, 1990.

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

[11] B. Khoussainov and A. Nerode. Automatic presentations of structures. *Lecture Notes in Computer Science*, 960:367–392, 1995.

[12] B. Khoussainov and S. Rubin. Graphs with automatic presentations over a unary alphabet. *Journal of Automata, Languages and Combinatorics*, 6(4), 2001.

[13] S. Rubin. Finite automata and well ordered sets. *New Zealand Journal of Computing*, 7(2):39–46, 1999.

[14] A. Szilard, S. Yu, K. Zhang, and J. Shallit. Characterizing regular languages with polynomial densities. In I. Havel and V. Koubek, editors, *Mathematical Foundations of Computer Science 1992, 17th International Symposium*, volume 629 of *lncs*, pages 494–503, 24–28 Aug. 1992.

Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS'02)
1043-6871/02 $17.00 © 2002 **IEEE**