CDMTCS
Research
Report
Series

Abstracts for the Workshop on Automata, Structures and Logic

Department of Computer Science
University of Auckland
Auckland, New Zealand

CDMTCS-257
December 2004

Centre for Discrete Mathematics and
Theoretical Computer Science

WORKSHOP ON AUTOMATA, STRUCTURES AND LOGIC

Program and Abstracts

Auckland, New Zealand 2004

The workshop is partially supported by

Computer Science Department, University of Auckland
New Zealand Institute of Mathematics and Applications
Marsden Fund of Royal Society of New Zealand
The Association for Symbolic Logic (ASL)

Graph transformations and Logic

Bruno Courcelle
LaBRI, Bordeaux 1 University and CNRS,
France
http://www.labri.fr/~courcell

By a *graph transformation*, we mean a (possibly multivalued) mapping from graphs to graphs, specified in a finitary way that makes possible to prove decidability results. There exist graph transformations of many different types.

In this lecture, we will consider two types of transformations of graphs, hypergraphs, logical structures and even combinatorial structures like matroids. The term « graph » stands for all these objects.

First we will consider transformations specified by formulas of Monadic Second-Order logic (MS logic). We call them MS *transductions*, to stress the analogy with rational transductions and tree-transducers. They define reductions between graph problems specified by MS formulas. Reducing a model-checking problem from graphs to trees is useful because many properties, in particular those expressible in MS logic are verifiable in linear time on finite trees (by means of a translation of logical formulas into finite deterministic automata).

We will also consider the proper subclass of *Quantifier-Free transductions*. Combined with disjoint union, they define graph operations which generalize the concatenation of words. Using them, one obtains extensions to finite graphs of basic notions of language theory, like that of a context-free grammar (it is more convenient to handle them in terms of recursive set equations than of graph rewriting rules), and that of a recognizable set, where recognizability is defined in terms of finite congruences (and not in terms of finite automata, because graph automata do not exist, except in very particular cases).

Furthermore, MS logic is a convenient language for specifying recognizable sets of graphs and many graph problems. The class of context-free sets of graphs (or structures) is closed under MS transductions. The inverse image of a recognizable set of graphs under an MS transduction is recognizable (a new result by A. Blumensath and B. Courcelle). We obtain thus robust extensions of the basic notions of language theory, at least for describing sets of finite objects.

The lecture will also survey some recent results concerning the following questions.

i) Can one find graph operations beyond quantifier-free definable ones that have still a good behaviour ?

ii) D. Seese conjectured that if a set of graphs has a decidable satisfiability problem for MS formulas, then it is the image of a set of trees under an MS transduction. Many particular cases of validity of this conjecture are known. Its slight weakening where the satisfiability is assumed decidable for MS formulas written with a set predicate expressing even cardinality has been proved by B.Courcelle and S. Oum. The proof uses results akin to that of Robertson and Seymour on planar forbidden minors and tree-width.

iii) One can classify sets of graphs in terms of the existence of MS transductions defining one in terms of the other. For example, one can construct by an MS transduction strings from trees but not vice versa. The tools developped for ii) yield a classification of sets of graphs in a five level hierarchy.

On the Automata Size for Presburger Arithmetic

Felix Klaedtke

Presburger arithmetic is the first-order theory with addition and the ordering over the integers. Its decidability was first proved independently by Presburger and Skolem by the method of quantifier elimination. Recently, it has become popular to use automata for deciding Presburger arithmetic; a point that was already made by Büchi in 1960: Integers are represented as words (e.g., using the 2's complement representation), and the automata are recursively constructed from the formulas. The constructed automaton for a formula precisely accepts the words representing integers which make the formula true.

On the one hand, a crude complexity analysis on the automata-based approach to Presburger arithmetic leads to a non-elementary worst-case complexity. On the other hand, empirical results show that automata-based decision procedures are competitive to other methods. In this talk we analyze this automata-based approach. Our analysis provides a triple exponential upper bound on the size of the minimal deterministic automaton for a Presburger arithmetic formula. Our analysis is based on a comparison of automata and quantifier-free formulas that are obtained by an improvement of Cooper's quantifier elimination method for Presburger arithmetic. Moreover, we give an example showing that this triple exponential upper bound on the automata size is tight (even for nondeterministic automata). From this it follows a double exponential upper bound for alternating automata.

Towards Automatic Model Theory of Modal Logic

Valentin Goranko and Govert van Drimmelen
Rand Afrikaans University, Johannesburg, South Africa
vfg,gcvd@rau.ac.za

The domain and the distinguished relations in an *automatic structure* are represented by regular languages or, equivalently, finite state automata. Such (possibly infinite) structures thus have finite presentations. Notably, the first-order theory (even extended with the quantifier 'there exists infinitely many') of an automatic structure is decidable.

We investigate applications of automatic structures to modal logic, and more specifically, model-theoretic aspects of modal logics using automatic Kripke frames and models.

In the first line of investigation we introduce and study the *automatic model property*: every satisfiable formula of the logic is satisfiable in an automatic model. It provides a method for proving decidability of finitely axiomatizable modal logics without the finite model property.

A second line of research is to consider *automatically approximable* non-automatic Kripke models, such that for every *n* there is an effectively obtainable automatic model bisimilar to the given one up to depth *n* and hence satisfying the same modal formulae up to modal depth *n*. Such models still have a decidable modal theory, by referring the model checking in the model to model checking in the appropriately approximating automatic models.

Finally we study the modal logics of automatic frames. Being fragments of universal monadic second-order logic, their decidability does not follow from the general decidability result for automatic structures mentioned above.

Most of the ideas, and some of the results, of this study generalize to first-order logic and other logical theories, as well as to □-automatic and tree-automatic models.

Ground term rewriting graphs

Christof Loeding

Ground term rewriting graphs are the transition graphs of ground term rewriting systems. The vertices of these graphs are terms over finite ranked alphabet and the edges are generated by a finite set of ground term rewriting rules. We consider a generalized form of rewriting rules, namely "T rewrites to T'" where T and T' are regular sets of terms and where an application allows to replace some t in T by some t' in T'. The transition graphs of such systems are called regular ground term rewriting graphs (RGTR graphs). Besides this explicit representation by rewriting systems, RGTR graphs can also be defined by equational systems over operators such as union and asynchronous product of graphs, and insertion of edges.

This talk will give an overview on algorithmic and structural properties of the class of RGTR graphs, and its relation to other classes of infinite graphs. We present decidability results on several reachability problems and model checking problems for different logics, including first-order logic and fragments of temporal logic. The structural analysis is based on the notion of tree-width of graphs. Using this notion one obtains an exact characterization of the well known classes of pushdown graphs and HR-equational graphs inside the class of RGTR graphs.

Cascade products and temporal logics on finite trees

Z. Ésik

Dept. of Computer Science, University of Szeged, P.O.B. 652, 6701 Szeged, Hungary

The cascade product of finite automata and its semigroup theoretic variants have been extremely useful in the characterization of the expressive power of several logics on finite words, cf. e.g., [1, 3, 11, 12]. In this paper we provide an algebraic characterization of the expressive power of a wide class of temporal logics on finite trees (terms) using the cascade product [8] of tree automata.

Suppose that R is a finite subset of the naturals containing 0. We consider (finite) ranked alphabets Σ such that the set Σ_n of letters of rank n is non-empty iff $n \in R$. We assume that each ranked alphabet comes with a fixed lexicographic order. Finite (ground) Σ-trees, or terms, are defined as usual. We denote the set of Σ-trees by T_Σ.

Syntax. For a ranked alphabet Σ, the set of formulas over Σ is the least set containing the symbol p_σ, for all $\sigma \in \Sigma$, closed with respect to the boolean operations $\vee$ (disjunction) and $\neg$ (negation), as well as the following construct. Suppose that $L \subseteq T_\Delta$, and for each $\delta \in \Delta$, φ_δ is a formula over Σ. Then

$$L(\delta \mapsto \varphi_\delta)_{\delta \in \Delta} \tag{1}$$

is a formula over Σ.

Semantics. Suppose that φ is a formula over Σ and $t \in T_\Sigma$. We say that t satisfies φ, in notation $t \models \varphi$, if

- $\varphi = p_\sigma$, for some $\sigma \in \Sigma$, and the root of t is labeled σ, or
- $\varphi = \varphi' \vee \varphi''$ and $t \models \varphi'$ or $t \models \varphi''$, or
- $\varphi = \neg\varphi'$ and it is not the case that $t \models \varphi'$, or
- $\varphi = L(\delta \mapsto \varphi_\delta)_{\delta \in \Delta}$, and the characteristic tree $\widehat{t} \in T_\Delta$ determined by t and the family $(\varphi_\delta)_{\delta \in \Delta}$ belongs to L. Here, $\widehat{t}$ has the same underlying directed graph as t, and a vertex v is labeled $\delta \in \Delta_n$ in $\widehat{t}$ iff v is labeled by some $\sigma \in \Sigma_n$ in the tree t, moreover, δ is the first letter in lexicographic order on Δ_n such that the subtree of t rooted at v satisfies φ_δ. If no such letter exists, then δ is the last letter in the lexicographic order on Δ_n.

For any formula φ of over Σ, we let L_φ denote the language defined by φ:

$$L_\varphi = \{t \in T_\Sigma : t \models \varphi\}.$$

We say that formulas φ and ψ (over Σ) are equivalent exactly when $L_\varphi = L_\psi$.

Example Let $R = \{0, 2\}$, say, moreover, let $\Delta_0 = \{\uparrow_0, \downarrow_0\}$, $\Delta_2 = \{\uparrow_2, \downarrow_2\}$ with lexicographic order such that $\uparrow_i < \downarrow_i$, $i = 0, 2$. Let L consist of those Δ-trees that contain at least one vertex labeled $\uparrow_0$ or $\uparrow_2$. Given formulas φ and φ' over Σ, consider the formula $\psi = L(\uparrow_i \mapsto \varphi, \downarrow_i \mapsto \varphi')_{i=0,2}$. Then a tree $t \in T_\Sigma$ satisfies ψ iff some subtree of t satisfies φ. Thus, the modal operator (1) associated with L corresponds to the (non-strict) EF modality of CTL [9]. Similarly, when L' is the set of those Δ-trees containing at least one $\uparrow_0$ or $\uparrow_2$ on the second level, then $\psi = L'(\uparrow_i \mapsto \varphi, \downarrow_i \mapsto \varphi')_{i=0,2}$ corresponds to the formula EXφ of CTL. One can derive all the usual CTL modalities by this pattern.

We will consider subsets of formulas associated with classes of tree languages. When $\mathcal{L}$ is a class of tree languages, we let FTL($\mathcal{L}$) denote the collection of formulas all of whose subformulas of the form (1) above are such that L belongs to $\mathcal{L}$. We denote by $\mathbf{FTL}(\mathcal{L})$ the class of all tree languages definable by the formulas in $\mathrm{FTL}(\mathcal{L})$.

We will use tree automata to characterize the expressive power of logics $\mathbf{FTL}(\mathcal{L})$, when $\mathcal{L}$ is a class of regular tree languages. Suppose that Σ is a ranked alphabet. Since we are considering only ground trees, we define a Σ-tree automaton to be a finite Σ-algebra which has no proper subalgebras, i.e., which is generated by the elements corresponding to the letters in Σ_0. Each Σ-tree automaton equipped with a specified subset of its underlying carrier defines a regular language $L \subseteq T_\Sigma$, cf. [5].

When $\mathbb{A}$ is a Σ-tree automaton with carrier A, $\mathbb{B}$ is a Δ-tree automaton with carrier B, and α is a family of functions $\alpha_n : A^n \times \Sigma_n \to \Delta_n$, $n \in R$, the cascade product $A \times_\alpha B$ is the minimal subalgebra of the Σ-algebra with carrier $A \times B$ and operations

$$\sigma((a_1, b_1), \ldots, (a_n, b_n)) = (\sigma(a_1, \ldots, a_n), \delta(b_1, \ldots, b_n)),$$

where $\delta = \alpha(a_1, \ldots, a_n, \sigma)$, for all $(a_1, b_1), \ldots, (a_n, b_n) \in A \times B$, $\sigma \in \Sigma_n$, $n \in R$.

Below we will say that quotients are expressible in $\mathrm{FTL}(\mathcal{L})$ if for any quotient $t^{-1}L = \{s \in T_\Sigma : t(s) \in L\}$ of a language $L \subseteq T_\Sigma$ in $\mathcal{L}$, where t is any Σ-tree with a "hole", and for any formulas $\varphi_\delta \in \mathrm{FTL}(\mathcal{L})$, $\delta \in \Delta$, there is a $\mathrm{FTL}(\mathcal{L})$-formula equivalent to $(t^{-1}L)(\delta \mapsto \varphi_\delta)_{\delta \in \Delta}$. Moreover, we will say that the next modalities are expressible in $\mathrm{FTL}(\mathcal{L})$ if for each Σ and each i such that $1 \leq i \leq n$ for some $n \in R$, and for each formula φ in $\mathrm{FTL}(\mathcal{L})$, there exits a $\mathrm{FTL}(L)$-formula $\mathrm{X}_i\varphi$ such that for any tree $t \in T_\Sigma$, $t \models \mathrm{X}_i\varphi$ iff the root of t is labeled by a letter of rank $\geq i$ and the i-th subtree of t satisfies φ. We can easily show that this condition is equivalent to the condition that $\mathbf{FTL}(\mathcal{L})$ contains all definite tree languages [6, 7]. Both conditions hold for most natural temporal logics on trees.

Our main contribution is the following general result:

Theorem *Suppose that $\mathcal{L}$ is a class of regular tree languages such that quotients and the next modalities are expressible in* $\mathrm{FTL}(\mathcal{L})$. *Then a tree language belongs to* $\mathbf{FTL}(\mathcal{L})$ *iff its minimal tree automaton belongs to the least class of tree automata containing the minimal tree automata of the languages in $\mathcal{L}$, closed with respect to the cascade composition and quotients.*

An immediate corollary of the above theorem is the fact that if $\mathcal{L}$ is a class of regular tree languages, then $\mathbf{FTL}(\mathcal{L})$ consists of regular languages. Of course this fact also follows from the obvious observation that when L consists of regular languages, then $\mathrm{FTL}(\mathcal{L})$ can be embedded in the monadic second order logic of [13] on finite trees. As a further corollary of the main result, we show that the lattice of those classes $\mathbf{V}$ of tree automata containing the definite tree automata [6, 7], closed with respect to the cascade product and homomorphic images, is isomorphic to the lattice of all classes of regular tree languages of the form $\mathcal{V} = \mathbf{FTL}(\mathcal{L})$, closed with respect to quotients and containing the definite tree languages. An isomorphism is given by the Eilenberg correspondence (c.f. [4]): Given $\mathbf{V}$, map $\mathbf{V}$ to the class of tree languages $\mathcal{V}$ whose minimal automaton belongs to $\mathbf{V}$, or equivalently, which can be accepted by a tree automaton in $\mathbf{V}$.

Along the way of proving the above theorem, we establish several useful properties of the tree language classes $\mathbf{FTL}(\mathcal{L})$ and the operator $\mathbf{FTL}$. For example, $\mathbf{FTL}(\mathcal{L})$ is always closed under the boolean operations (trivial) and "inverse literal tree homomorphisms" (almost trivial), and is closed under quotients iff any quotient of each tree language in $\mathcal{L}$ belongs to $\mathbf{FTL}(\mathcal{L})$ iff quotients are expressible in $\mathrm{FTL}(\mathcal{L})$. Thus, when these latter conditions hold and L consists of regular languages, then $\mathbf{FTL}(\mathcal{L})$ is a "literal tree language variety", which are closely related to the tree language varieties of [10]. We also prove that $\mathbf{FTL}$ is a closure operator on (regular) tree language classes and establish an Eilenberg Variety Theorem for literal varieties.

Suppose $\mathbf{V} \leftrightarrow \mathbf{FTL}(\mathcal{L})$ under the above Eilenberg correspondence. Then a regular tree language L belongs to $\mathbf{FTL}(\mathcal{L})$ iff its minimal tree automaton belongs to $\mathbf{V}$. Thus, when $\mathbf{V}$ is decidable, then there results an effective characterization of the expressive power of the logic **FTL**($\mathcal{L}$). In the lecture, we will apply this approach to derive effective characterizations of certain fragments of CTL, both on the finite trees considered here and on the usual unordered tree models of CTL, complementing the results obtained in [2]. We expect that our method will lead to a characterization of the expressive power of full CTL.

References

1. A. Baziramwabo, P. McKenzie and D. Therien, Modular temporal logic, in: *Proc. 1999 IEEE Conference LICS, Trento, Italy.*
2. M. Bojanczyk and I. Walukiewicz, Characterising EF and EX tree logics, *proc. CONCUR 2004*, to appear.
3. J. Cohen, J.-E. Pin and D. Perrin, On the expressive power of temporal logic, *J. Computer and System Sciences*, 46(1993), 271–294.
4. S. Eilenberg, *Automata, Languages, and Machines*, vol. A and B, Academic Press, 1974 and 1976.
5. F. Gécseg and M. Steinby, *Tree Automata*, Akadémiai Kiadó, 1984.
6. U. Heuter, Definite tree languages, *Bulletin of the EATCS*, 35(1988), 137–142.
7. M. Nivat and A. Podelski, Definite tree languages (cont'd), *Bulletin of the EATCS*, 38(1989), 186–190.
8. G. Ricci, Cascades of tree-automata and computations in universal algebras. *Math. Systems Theory*, 7(1973), 201–218.
9. K. Schneider, *Verification of Reactive Systems*, Springer, 2004.
10. M. Steinby, General varieties of tree languages, *Theoret. Comput. Sci.*, 205(1998), 1–43.
11. H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity*, Birkhauser, 1994.
12. H. Straubing, D. Therien and W. Thomas, Regular languages defined with generalized quantifiers, *Information and Computation*, 118(1995), 289–301.
13. J. W. Thatcher and J. B. Wright, Generalized finite automata theory with an application to a decision problem of second-order logic. *Math. Systems Theory*, 2(1968), 57–81.
14. P. Wolper, Temporal logic can be more expressive, *Information and Control*, 56(1983), 72–99.

Amorphous Automata and Bisimulation Quantifiers

Tim French

The University of Western Australia
tim@csse.uwa.edu.au

Modal logic is a powerful tool for specifying properties of transitions structures labeled with propositions (Kripke structures). Recently there has been much interest in increasing the expressive power of modal logic via extensions to the language. Examples of this are fixed-point operators (μ-CTL), hybrid operators, dynamic operators (PDL) and propositional quantification (QPTL). Of these extensions, propositional quantification is the most powerful. However such power frequently leads to undecidable languages.

We investigate interpreting propositional quantification with respect to bisimulations of a given Kripke structure. Given a Kripke structure, M, and some atomic proposition, x, we say an x-variant of M is any Kripke structure that is identical to M, except in its interpretation of the proposition x. Then given any formula α, we say $M \models \exists x \alpha$ if and only if there is some model K that is bisimilar to M, and some model N that is an x-variant of K such that $N \models \alpha$. Thus propositional quantification is interpreted *modulo bisimulation*. We examine the temporal logic, CTL, augmented with bisimulation quantification (QCTL).

We find that such semantics are natural in the context of modal logic, remain highly expressive (at least as expressive as the μ-calculus) and in many cases are decidable. Bisimulation quantifiers allowing us to retain the basic structure and simplicity of modal logics, and gain the expressive power of monadic second-order logic.

Decidability for such logics can be shown via a reduction to *amorphous automata* [1]. Note that amorphous automata were originally introduced in [3] as μ-automata and applied to monadic second-order logics. Subsequently the definition of μ-automata appears to have been somewhat generalized. The most important property of such automata is that acceptance of transition structures is bisimulation invariant. Amorphous Automata are similar to binary tree automata with a Rabin (or Streett, or parity) acceptance condition [2]. However, rather than acting on only binary trees, amorphous automata act on the bisimulation class of a transition structure. To achieve this the transition function of the automata is given by some function, $\lambda : Q \times \Sigma \longrightarrow \wp(\wp(Q))$, where Q is the set of states, and Σ is the alphabet that is being read. If the automata reads the letter a whilst in the state q, then the set of successor states must be taken from $\lambda(q, a)$. Since we are considering all bisimilar structures, a single automaton state can label more than one node, and a single node can be labeled by more than one state. We have found the following complexity results for amorphous automata with a parity acceptance condition:

1. The emptiness problem can be solved in time exponential to the size of the automaton. It is not known whether this is an optimal result.
2. Given an amorphous automaton of size n we can construct an automaton that accepts the complementary language of size order n^n. This result relies on a slight extension to Safra's determinization construction so that it applies to parity ω-automata.
3. Similarly we can construct amorphous automata that accept disjunction of languages, projections of languages (modulo bisimulation), and reachable languages of linear size.
4. Amorphous automata can be translated to and from alternating automata in exponential time.

Given a formula of a basic modal logic extended with bisimulation quantifiers (eg QCTL, the extension of CTL) we can use the constructions mentioned above to define an amorphous automata that accepts exactly the structures that satisfy the given formula.

Furthermore the expressivity of QCTL is such that we can embed many other multi-modal logics into QCTL to derive further decidability results. We interpret bisimulation quantification in modal logics by restricting the set of bisimulations to those that preserve the required frame conditions (eg the transitivity of the relation in the modal logic $K4$). This allows us to generalize many extensions and combinations of classical modal logics.

Due to the exponential complementation construction, the decidability process can be shown to be non-elementary. While the logic QCTL is equally as expressive as μ-CTL, the non-elementary decidability is off-set by the fact that QCTL is non-elementarily more succinct than μ-CTL.

Thus amorphous automata and bisimulation quantifiers allow us to define a very expressive, decidable and succinct language that generalizes many of the current extensions to modal logic.

References

1. T. French. Decidability of quantified propositional branching time logics. In *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence*, pages 165–176, 2001.
2. E. Gradel, W. Thomas, and T. Wilke (Eds.). *Automata, Logics and Infinite Games.* Springer, 2002.
3. D. Janin and I. Walukiewicz. Automata for the μ-calculus and related results. In *Proceedings 20th Int. Symp. on Mathematical Foundations of Computer Science*, volume 969, 1995.

On rational and automatic monoids

Jacques Sakarovitch
CNRS/ENST — Paris, France

It is possible to study and classify the multiplicative "structure" of monoids by means of functions on words: those which are defined by the choice of a set of normal forms for a given set of generators. To any "reasonable" complexity hierarchy of functions — the classes have to be closed under composition with inverse morphisms — corresponds a hierarchy of monoids: every monoid is given the least complexity of functions that can compute a set of representatives for the map equivalence of a surjective morphism from a free monoid onto that monoid (*cf.* [1]).

In any hierarchy of word functions, the lowest class will be the one of *rational functions*: those computed by finite automata. The corresponding monoids can thus be called *rational monoids*. These monoids generalize at the same time the finite and the free monoids. As far as the rational sets and relations are concerned, they are not distinguishable from the free monoids. In particular, Kleene's theorem holds in any rational monoid (*cf.* [2]). By means of rather tedious constructions (non rational extensions of rational monoids) it is proved that the converse is not true, that is there exist monoids which are not rational and in which Kleene's theorem holds ([3]).

The families of rational and of automatic monoids are not comparable: the only rational groups are the finite ones and there are rational monoids that are not automatic (in the sense of [4]). And to put it plainly, the notion of rational monoid is not as powerful as the one of automaticity that will be at the center of attention in this workshop. My purpose however is to present its main features with the idea that the two can be usefully combined.

[1] J. S. Description des monoïdes de type fini, *E.I.K.* **17**, 1981, 417–434.

[2] J. S. Easy multiplications I, The realm of Kleene's theorem, *Information and Computation* **74**, 1987, 173–197.

[3] M. Pelletier and J. S. Easy multiplications II, Extensions of rational semigroups, *Information and Computation* **88**, 1990, 18–59.

[4] C. M. Campbell *et al.* Automatic semigroups, *Theoretical Computer Science* **365**, 2001, 365–391.

On the complexity of infinite computations

Damian Niwinski

Classical complexity theory deals with decision problems over integers, and does not easily adapt to properties of infinite computations, typically characterized by sets of infinite words or infinite trees. The latter can usually be captured by suitable kind of automata with infinitary acceptance criteria, which gives rise to numerous hierarchies of infinitary languages. The talk will show how these hierarchies relate between themselves, and how they interplay with the hierarchies offered by classical descriptive set theory, such as the Borel hierarchy and its Wadge refinement.

In the model-checking methodology, the complexity of a property to be verified may indicate us a suitable algorithmic technique. So an important question is to determine this complexity effectively. The talk will present the state-of-the-art of this largely unsolved problem, including the recent results of I.Walukiewicz and the author on deciding the level of deterministic tree languages in nondeterministic hierarchy.

Which structures are automatic?

Frank Stephan

Automatic Structures are a natural extension of regular languages into algebra. The present talk gives an overview on fundamental and also on recent results with respect to the question, which basic algebraic structures have an automatic presentation and which not. Furthermore, important properties and limitations of various automatic structures are given.

Automatic trees have finite Cantor-Bendixson rank. Furthermore, they have a regular infinite path whenever they have an infinite path at all. If the number of infinite paths is countable, then every path is regular.

The topological complexity of a linear order is measured with the Finite Condensation rank. This rank is finite for every automatic linear order and therefore, only well-orderings corresponding to ordinals $\square < \square^{\square}$ have an automatic presentation.

Furthermore, Boolean algebras have an automatic presentation iff they are isomorphic to finite products of the Boolean algebra of finite and co-finite subsets of the natural numbers.

Although the additive groups of the integers and the dyadic numbers have automatic presentations, this is already unknown for the additive group of rationals. The free Abelian group of infinite rank does not have an automatic presentation.

Further negative results provide that the free Abelian group of infinite rank, the countably infinite random graph, the universal partial order and any infinite integral domain do not have automatic presentations.

Equational presentations of tree-automatic structures

Thomas Colcombet

Abstract

We investigate in this abstract various possibilities for representing tree-automatic structures. In particular we show that the classical presentation of prefix-recognizable structures as least solutions of equational systems admit a natural extension for tree-automatic structures. We also show that the first-order logic extended with counting quantifiers remains decidable for the solutions of some infinite equational systems. This extends the decidability results known for tree-automatic structures.

1 Introduction

Tree automatic structures were introduced in [2] though the underlying idea can be traced back to the work of Dauchet and Tison [8]. Tree automatic structures are a natural extension of automatic structures [9] over an universe made of trees. More precisely, in a tree-automatic structure, the universe is a rational set of finite terms, and the interpretation of each relational symbol is automatic in the meaning that it is described by a finite state automaton. By extension, we say (in this abstract) that a structure is tree-automatic whenever it is isomorphic to a tree-automatic structure. The definition of automaticity of a relation is such that automatic relations are closed by the boolean connectives as well as projection and cylindrification. This closure properties result in a decidable first-order theory for tree-automatic structures. In fact this result can be strenghtens to capture first-order logic extended with counting quantifiers[1].

Automatic structures belong to a more general topic, the study of the classes of structures admitting a finite presentation, i.e. that each (possibly infinite) structure can be described by a finite object. A widely studied class of this kind is the one of prefix-recognizable structures [4]. It happens that each structure in this class has different presentations. Let us cite three such presentations. The first one is the *internal presentation*, in which an exact description of the universe and of the relation is given: for prefix-recognizable structures, the universe is a rational set of words and each relation is described by a regular set of prefix rewriting rules[4]. The second way is *transformational* as each structure is described by a transformation applied to a given known structure; in this sens, the prefix-recognizable structures are the monadic (second-order) interpretations of the infinite complete binary tree[1, 3]. The third presentation is *equational*; in this sens, the prefix-recognizable structures are the least solutions of finite equational systems over a given set of fixed operators called VR (standing for vertex replacement) [1].

In this abstract, we present similar results concerning the class of tree-automatic structures. Those structures classically admit an internal presentation. We mainly contribute here with an equational presentation: we provide a set of operators such that a structure is tree-automatic iff it is isomorphic to the solution of an finite equational system using those operators (Theorem 1). This approach unifies tree-automatic structures with prefix-recognizable ones since the operators we use are the ones defining the prefix recognizable structures, but enhanced with a product operator. It also allows us to use tools specially designed for the treatment of equational systems, such as tree transducers with lookahead [6, 5] for the study of tree-automatic structures. Finally, it is natural in this context to step outside the case of finite equational systems and consider infinite such systems. Theorem 2 shows that in this case, some representation equivalences remain true. Corollary 3 then provides decidability results for those "extended tree-automatic structures".

Detailed proofs can be found in [5].

[1] The counting quantifiers are $\exists^\omega x$ and $\exists^{m[n]} x$ meaning respectively "there exists infinitely many x's such that ..." and "there exists m-many x's modulo n such that ... ".

2 Equational presentation: the VRC operators

The core of the VRC operaptors that we are about to introduce is the positive quantifier free definable (pqfd) interpretation. Formally a pqfd interpretation $\mathcal{I}$ is an operation which transforms relational structures into relational structures and is described by a tuple of formulas $(\delta, \phi_{R_1}, \ldots, \phi_{R_n})$ (where $R_1, \ldots, R_n$ are the relational symbols of the resulting structure). In this case, each formula is made only of predicates applied to first-order variables, conjonctions, disjonctions and the constants true and false (i.e. first-order without quantification and without negation). As usually, the formula δ has a single free variable and is used to define the universe of the resulting structure, and each formula ϕ_R has $\mathbf{arity}(R)$ free variables and defines the new interpretation of the relational symbol R.

There is four kind of *VRC operators*: the constant structure with a singleton universe and no relation; a binary operator performing the *disjoint union* of structures; a binary operator performing the *cartesian product* of structures; and finally, a unary operator corresponding to each *pqfd interpretation*.

All these operators have the property of being continuous (in the meaning e.g. of ω-complete partial orders), and thus by the theorem of Knaster-Tarski, each equational system using them admits a unique least solution. In particular, it is known from [1] that the solution of finite equational systems using all VRC operators but the product, are isomorphic to prefix-recognizable structures. The following theorem establishes a similar result for tree-automatic structures.

Theorem 1 *A structure is tree-automatic iff it is isomorphic to the least solution of a finite equational system over the VRC operators.*

3 Powerset monadic-interpretation

We provide here an transformational presentation suitable for representing the solutions of the (possibly infinite) equational systems over the VRC operators.

A *powerset monadic-interpretation* $\mathcal{I}^{\mathcal{P}}$ is described by a tuple of formulas $(\delta, \phi_{R_1}, \ldots, \phi_{R_n})$, where this time the formulas are monadic and the free variables are *monadic*. Applied to a structure, this interpretation produces a new structure where the elements of the universe are the subsets of the first universe satisfying δ. Similarly, the interpretation of the relations is defined via the ϕ formulas. It is easy to see that if the original structure has a deciable monadic theory, the one resulting of a powerset interpretation has a decidable first-order theory. In fact, one has the following refinement: when the original structure is a deterministic tree, then the resulting structure has a decidable first-order theory with counting quantifiers.

The following theorem shows that solving an equational system over the VRC operators is equivalent to applying a powerset monadic-interpretation to an infinite determinisic tree. In this theorem, we see equational systems as deterministic graphs containing, correctly encoded, all the relevant information.

Theorem 2 *The following class of structures are equivalent,*

- *the least solutions of equational systems using a finite subset of the VRC operators,*
- *the countable powerset monadic-interpretations of the unravelling of a deterministic graph.*

And this equivalence is effective in the meaning that the graphs and the equational systems are linked by parameterless MSO-definable transductions (see [7]).

Combined with the decidability remarks mentionned above, we obtain the following corollary.

Corollary 3 *If the monadic theory of an equational system using a finite subset of the VRC operators is decidable then the first-order theory with counting quantifiers of its least solution is decidable.*

Let us notice that in this corollary, the classical decidability result for tree-automatic structures simply corresponds to the particular case of finite equational systems.

References

[1] K. Barthelmann. On equational simple graphs. Technical Report 9, Universität Mainz, Institut für Informatik, 1997.

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

[3] A. Blumensath. Prefix-recognisable graphs and monadic second-order logic. Technical Report AIB-06-2001, RWTH Aachen, May 2001.

[4] D. Caucal. On infinite transition graphs having a decidable monadic theory. In *ICALP'96*, volume 1099 of *LNCS*, pages 194–205, 1996.

[5] T. Colcombet. *Propri?t?s et repr?sentation de structures infinies.* PhD thesis, Universit? Rennes I, March 2004.

[6] T. Colcombet and C. Löding. On the expressive power of deterministic transducers over infinite trees. In *STACS'04*, LNCS. Springer Verlag, 2004.

[7] B. Courcelle. Monadic second-order graph transductions: A survey. In *Theoretical Computer Science*, volume 126, pages 53–75. Elsevier, 1994.

[8] M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *LICS'90*, pages 242–248. IEEE, 1990.

[9] B. Khoussainov and A. Nerode. Automatic presentations of structures. In D. Leivant, editor, *International Workshop in Logic and Computational Complexity*, volume 960 of *LNCS*, pages 367–392, 1994.

A Pumping Lemma for Higher-Order Pushdown Automata

Achim Blumensath

Higher-order pushdown automata were introduced by Maslov in [7]. Damm and Goerdt [4] used them to characterise the so-called OI-hierarchy which consists of the solutions of higher-order lambda schemes. Recently, this work has received renewed attention in the study of hierarchies of trees or graphs with decidable monadic theories (see, e.g., [6]).

The *Caucal hierarchy* is obtained by alternated applications of monadic second-order interpretations and the Muchnik construction (see [8, 9, 1]) starting with the class of all finite structures. Since these operations preserve decidability of MSO-theories it follows that every structure in this hierarchy has a decidable monadic theory. Originally, Caucal [3] defined the hierarchy only for graphs where the above operations can be replaced by, respectively, inverse rational mappings and unravellings.

The lowest level of the Caucal hierarchy consists of the class of *prefix-recognisable* (also called *tree-interpretable*) structures. Restricted to graphs this is the class of all graphs that can be obtained from the configuration graph of some pushdown automaton by contracting all ε-transitions. Recently, Carayol and Wöhrle [2] have extended this characterisation to the whole hierarchy: A graph belongs to the n-th level of the Caucal hierarchy if and only if it can be obtained by contracting ε-transitions from the configuration graph of some higher-order pushdown automaton of level n.

Naturally, the question arises of which structures are contained in the Caucal hierarchy and at what level they do appear. One way to answer this question consists in classifying the configuration graphs of higher-order pushdown automata.

This is the motivation of the results presented in this article. A pumping lemma for higher-order pushdown automata can be used to derive bounds on the length of paths in configuration graphs thereby providing a tool for proving that some graphs do not belong to a certain level of the hierarchy. Unfortunately, the bounds we are able to prove are far from optimal and much to high to be of use for separating the various levels. Nevertheless, we hope that the technical tools developed in the present article will be of aid in future work to derive sharper bounds.

For indexed grammars (which correspond to pushdown automata of level 2), a pumping lemma was proved by Hayashi [5]. The proof of the following result owes much to this paper.

Theorem 1 (Pumping Lemma). *Let r be a run of size $k := |\mathrm{dom}(r)|$ with first element w. Set*

$$b := \max\{\, \kappa_1(x) \mid x \in \mathrm{dom}(r) \,\},$$

and $$m := b(n+1)^{n+1} |\Gamma|^{n+1} \big(|Q| + 2\|\pi_0 r(w)\|\big)^{n+1} 2^{n+g(1,n-1,b)}.$$

If $k \geq \beth_n(m)$ then there exists a run r^* of length

$$k < |\mathrm{dom}(r^*)| \leq k^{(bk+4)2^{n-1}}.$$

such that $r^*(\varepsilon) = r(\varepsilon)$ and r^* ends in the same state as r.

References

[1] D. Berwanger and A. Blumensath, *The Monadic Theory of Tree-like Structures*, in Automata, Logic, and Infinite Games, E. Grädel, W. Thomas, and T. Wilke, eds., LNCS 2500, Springer Verlag, 2002, pp. 285–301.

[2] A. Carayol and S. Wöhrle, *The Caucal Hierarchy of Infinite Graphs in Terms of Logic and Higher-order Pushdown Automata*, in Proc. 23rd Conf. on Foundations of Software Technology and Theoretical Computer Science, FSTTCS, 2004.

[3] D. Caucal, *On Infinite Terms Having a Decidable Monadic Theory*, in Proc. 27th Int. Symp. on Mathematical Foundations of Computer Science, MFCS, 2002, pp. 165–176.

[4] W. Damm and A. Goerdt, *An Automata-Theoretical Characterization of the OI-Hierarchy*, Information and Control, 71 (1986), pp. 1–32.

[5] T. Hayashi, *On derivation trees of indexed grammars: An extension of the uvwxy-theorem*, Publ. RIMS, Kyoto Univ., 9 (1973), pp. 61–92.

[6] T. Knapik, D. Niwiński, and P. Urzyczyn, *Higher-order pushdown trees are easy*, 5th Int. Conf. on Foundations of Software Science and Computation Structures, FoSSaCS 2002, LNCS, 2303 (2002), pp. 205–222.

[7] A. N. Maslov, *Multilevel stack automata*, Problemy Peredachi Informatsii, 12 (1976), pp. 55–62.

[8] A. L. Semenov, *Decidability of monadic theories*, LNCS, 176 (1984), pp. 162–175.

[9] I. Walukiewicz, *Monadic second-order logic on tree-like structures*, Theoretical Computer Science, 275 (2002), pp. 311–346.

The hyperalgebraic hierarchy: an infinite hierarchy of infinite structures

Teodor Knapik, University of New Caledonia

Hyperagebraic structures arise as least solutions of higher-order systems of equations over graph-building operations. Depending on the choice of these operations, two main infinite hierarchies of infinite structures with decidable MSO theory may be considered. The greater of these hierarchies has at least 2 other characterizations: in terms transition graphs of higher-order pushdown automata, or, in terms of alternate sequences of unfoldings and MSO-interpretations applied on a finite graph. When restricted to graphs, level 1 of the grater hierarchy consists of prefix-recognizable graphs whereas the same level of the smaller hierarchy consists of HR-equational graphs. At level 2 the unknowns carry parameters and at level 3 these parameters may be of function type. Starting from the latter level a syntactic condition of "safety" is meaningful. The structures that are least solutions of safe systems have a decidable MSO theory. When restricted to bounded-branching trees, they are accepted by higher order pushdown automata and, may also be obtained by iterated substitutions applied on a regular tree.

Tree-automatic gaps

Christian Delhomme

Handling certain ordinal parameters for automatic or tree-automatic structures, such as the height for binary relations, may be eased under some transitivity assumptions, related to the fact that the transitive closure of an automatic or tree-automatic relation may fail to be so. We investigate the relevance of such kind of assumptions.

Numerical predicates on words, trees, and traces

Dietrich Kuske
(joint work with Markus Lohrey)

Using automatic structures, it is easily shown that the first-order theory of the complete binary tree together with the "equal-length-predicate" is decidable (this is already implicite in the work of Rabin from 1969). Here, two questions arise

1. Which other numerical predicates can be adjoined to the binary tree without loosing decidability of the first-order theory?

2. What structures other than trees can be (naturally) equipped with numerical predicates resulting in decidable theories?

We recall Seese's technique to prove that "most" numerical predicates on the binary tree give rise to undecidable theories (e.g., the "double-length-predicate" has such a severe effect). This allows to identify rather small undecidable fragments of the corresponding theories.

To answer the second question, we turn attention to free partially commutative monoids (also known as Mazurkiewicz traces). The automaticity of the partial order of all finite traces is demonstrated thereby showing the decidability of the corresponding theory. This simplifies a proof by Madhusudan (LICS 2003). While the partial order of traces remedies many properties of the tree, we show that the extension of this partial order by the equal-length-predicate gives rise to an undecidable theory.

Title: Finitely generated groups with automatic presentations

Speaker: Graham Oliver
Department of Computer Science,
University of Leicester, UK.
gpo1@mcs.le.ac.uk

A structure is said to be computable if its domain can represented by a set which is accepted by a Turing machine and if there are decision-making Turing machines for each of its relations. Khoussainov and Nerode have introduced [5] a very interesting restriction of this general idea, to *automatic structures*, i.e. those structures whose domain and relations can be checked by finite automata as opposed to Turing machines. A structure isomorphic to an automatic structure is said to have an *automatic presentation*.

The theory of automatic presentations was, in part, inspired by the theory of automatic groups [2]; however, the definitions are somewhat different. One point is that automatic groups are necessarily finitely generated whilst this is not the case for groups with an automatic presentation. We compare the two theories by giving a complete classification of the finitely generated groups with automatic presentations. In what follows, we will assume that all groups referred to are finitely generated.

One interesting result [5] is that all abelian groups have automatic presentations. One can readily extend this to *virtually abelian* groups, i.e. groups with an abelian subgroup of finite index. However, we show that the converse also holds, so that the virtually abelian groups are precisely the groups with automatic presentations.

The proof that these are the only such groups proceeds in three steps. We consider a group G with a finite generating set A which is closed under taking inverses, and we let $\delta(g)$ denote the minimum length of a word in A^* representing the element g. For $n \geqslant 1$ we let $\gamma(n)$ denote the number of elements g in G with $\delta(g) \leqslant n$. Restrictions on the growth of the lengths of codes of elements of G in an automatic presentation show that $\gamma(n)$ is bounded above by a polynomial function. This condition on γ is independent of the choice of finite generating set and G is said to have *polynomial growth*.

A theorem of Gromov [4] classifies groups with polynomial growth as being those that contain a nilpotent subgroup of finite index, i.e. the *virtually nilpotent* groups. Finally, it is known that a virtually nilpotent group with decidable first order theory is virtually abelian (see [3, 6, 7]), and this completes the proof.

It is known that virtually abelian groups are automatic [2] but that there are many examples of automatic groups that do not have automatic presentations (for example, free groups); so the class of groups with automatic presentations is a proper subclass of the class of automatic groups. Now let G be a group with generators $\{a_1, \ldots, a_n\}$, and let $C = (G, R_1, \ldots, R_n)$ be a new structure where $R_i(g, g')$ if $ga_i = g'$; C is called the *Cayley graph* of G. By definition, if G is automatic, then C has an automatic presentation. However, the Heisenberg group (the group of nonsingular 3×3 upper-triangular matrices over $\mathbb{Z}$), although not automatic, does have a Cayley graph with an automatic presentation [1].

To summarise, let `AutoPres` be the class of groups with automatic presentations, `Automatic` be the class of automatic groups, and let `CayleyAutoPres` be the class of groups whose Cayley graph has an automatic presentation; then we have $\texttt{AutoPres} \subsetneq \texttt{Automatic} \subsetneq \texttt{CayleyAutoPres}$.

References

[1] A. Blumensath and E. Grädel, Finite presentations of infinite structures: automata and interpretations, *in* G. Gottlob, M. Hermann & M. Rusinowitch (eds.), *Proceedings of the 2nd International Workshop on Complexity in Automated Deduction* (2002).

[2] D. B. A. 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).

[3] Ju. L. Eršov, Elementary group theories, *Dokl. Akad. Nauk SSR* **203** (1972), 1240–1243.

[4] M. Gromov, Groups of polynomial growth and expanding maps, *Publications Mathématique d'IHÉS* **53** (1981), 53–78.

[5] B. Khoussainov and A. Nerode, Automatic presentations of structures, *in* D. Leivant (ed.), *Logic and Computational Complexity* (Lecture Notes in Comp. Sci. **960**, Springer-Verlag, 1995), 367–392.

[6] G. A. Noskov, The elementary theory of a finitely generated almost solvable group, *Izv. Akad. Nauk SSSR Ser. Mat.* **47** (1983), 498–517.

[7] N. S. Romanovskiĭ, On the elementary theory of an almost polycyclic group, *Math. Sb.* **111** (1980), 135–143.

DPDA EQUIVALENCE AND INFINITE GRAPHS

COLIN STIRLING
EDINBURGH UNIVERSITY

The question whether language equivalence between deterministic context-free languages is decidable was solved positively by Senizergues in 1997. His proof, which is very algebraic, involves two semi-decision procedures without a complexity upper bound. By viewing the problem as an equivalence problem on infinite graphs, we provide a simpler deterministic decision procedure with a primitive recursive complexity upper bound.

Logic, Databases, and Complexity

Rod Downey (Wellington New Zealand)

There has been a lot of work in using algorithms generated by logic and topological graph theory as a paradigm for algorithm design. I will describe recent work towards our understanding of the usefulness of this, particularly in the context of parameterized complexity.

Hybrid control and Automata

Anil Nerode
Goldwin Smith Professor of Mathematics and Computer Science
Cornell University
Ithaca, NY 14850
USA
Anerode1@twcny.rr.com

Hybrid systems are interacting reactive networks of continuous physical processes and finite automata. The subject has advanced rapidly in the last decade. There are many meetings on the subject in mathematics, computer science, and engineering meetings. This is because hybrid systems theory promises to show how to control systems such as computer networks, air traffic control, chemical process control, enterprise supply chain control, etc. (Kohn-Nerode founded a company, Clearsight, which pursues such goals.) The Kohn-Nerode version of the subject starts by noting that in optimal control, optimal control functions may have to be measure–valued and may not be physically realizable. But in engineering only epsilon optimal control is needed, for an end user defined epsilon. For any such epsilon, one can actually construct, via differential geometry and differential equations, Finsler Manifolds, Bellman equations, Pontryagin maximum principle, etc., epsilon approximations to optimal control, which are physically realized by finite control automata, yielding a controlled hybrid system. More or less conversely, when a physical system has both digital and discrete elements, we can continualize the discrete elements, then construct approximate optimal control by finite automata of the resulting purely continuous system to achieve epsilon optimal control of the original system.

One can conceptualize control systems as differential automata, which can be approximated by finite automata. We discuss directions of research and the current state of the subject. Our approach emphasizes the underlying differential geometry. We are interested in the development of differential automata theory on manifolds, designed for the applications mentioned.

Title: Automatic Groups and Semigroups

Speaker: Rick Thomas

Department of Computer Science
University of Leicester, UK
rmt@mcs.le.ac.uk

Abstract

There have been some intriguing interactions in recent years between group theory and theoretical computer science. One area which has proved to be very fruitful in providing interesting and useful results is that of *automatic groups* (in the sense of Epstein et al. [3]). For example, it is known that any automatic group must be finitely presented and the word problem of an automatic group can be solved in quadratic time. For an overview of automatic groups, see [1] and [3].

The purpose of this talk will be to introduce the notion of an automatic group, give some examples and mention some results. Our main motivation is to indicate how the definition can be extended to semigroups and give an outline of some recent work establishing a theory of automatic semigroups. We find that some of the results in automatic groups do generalize to semigroups (such as the solution of the word problem in quadratic time) whereas others do not (for example, there exist automatic semigroups that are not finitely presented).

For an introduction to automatic semigroups, see [2].

References

[1] G. Baumslag, S. M. Gersten, M. Shapiro & H. Short, Automatic groups and amalgams, *Journal of Pure and Applied Algebra* **76** (1991), 229–316.

[2] C. M. Campbell, E. F. Robertson, N. Ruškuc & R. M. Thomas, Automatic semigroups, *Theoretical Computer Science* **365** (2001), 365–391.

[3] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. Levy, M. S. Paterson & W. Thurston, *Word Processing in Groups* (Jones and Barlett, 1992).

Intrinsically regular relations in automatic structures.

Sasha Rubin

An automatic structure *A* is one whose domain A and atomic relations are finite automaton (FA) recognisable. A structure isomorphic to *A* is called automatically presentable.

Suppose *R* is an FA recognisable relation on *A*. This paper concerns questions of the following type. When is *R* definable in *A* ? For which automatic presentations of *A* is (the image of) *R* also FA recognisable ? In other words we are concerned with the relationship between FA recognisability and definability of relations in automatic structures. To this end we say that *R* is *intrinsically regular* in a structure *A* if it is FA recognisable in every automatic presentation of the structure. For example, in every automatic structure all relations definable in first order logic with additional quantifiers 'there exists infinitely many' and 'there exists a multiple of k many' are intrinsically regular.

We mention results characterising the intrinsically regular relations in a sample of structures. More specifically we investigate whether or not intrinsically regular relations are definable. For example, on the one hand, the set of even numbers of the structure (N,□), although not definable in this structure, is intrinsically regular. On the other, there exists an automatic presentation of (*N*, S) in which the set of even numbers is not regular. In particular, a unary relation in (*N*, S) is intrinsically regular if and only if it is first order definable in (*N*, S).

Theories of automatic structures and their computational complexity *

Markus Lohrey

Lehrstuhl für Informatik I, RWTH Aachen, Germany
lohrey@i1.informatik.rwth-aachen.de

Automatic structures were introduced in [10, 14]. The idea goes back to the concept of automatic groups [7]. Roughly speaking, a relational structure is called automatic if the elements of the universe can be represented by words from a regular language and every relation of the structure can be recognized by a synchronized 2-tape automaton. Automatic structures received increasing interest during the last years [1, 3, 13, 15–17]. One of the main motivations for investigating automatic structures is the fact that every automatic structure has a decidable first-order theory. On the other hand, Blumensath and Grädel presented an example of an automatic structure $\mathcal{A}$ with a nonelementary first-order theory [3], i.e., the running time of any algorithm for deciding the truth of a first-order formula in $\mathcal{A}$ exceeds any tower of exponents of fixed size. This motivates the search for subclasses of automatic structures for which the first-order theory becomes elementary decidable. In [20], the author identified such a subclass: *automatic structures of bounded degree.* A relational structure has bounded degree if for some fixed constant c, for every element x of the structure there exist at most c elements y such that x and y belong to some tuple in some relation. Equivalently, the Gaifman-graph [9] of the structure has bounded degree. Using a method of Ferrante and Rackoff [8], it was shown that for every automatic structure of bounded degree the first-order theory can be decided in triply exponential alternating time with a linear number of alternations [20, Thm. 3]. We are currently not able to match this upper bound by a sharp lower bound. But it is possible to construct an automatic structure of bounded degree for which the first-order theory has a lower bound of doubly exponential alternating time with a linear number of alternations [20, Thm. 5]. The proof for this lower bound uses the interpretation method of Compton and Henson [5].

Our upper bound technique for automatic structures of bounded degree can be also extended to *tree automatic structures* [2]. Tree automatic structures generalize automatic structures by representing elements via trees and using tree automata for recognizing the universe as well as the relations of the structure. The basic idea goes back to the work of Dauchet and Tison [6] on ground tree rewriting systems. A typical example of a tree automatic structure that is not automatic is the set of natural numbers with multiplication [2]. For a tree automatic structure of bounded degree, the first-order theory is decidable in fourfold exponential time with a linear number of alternations [20, Thm. 6]. Currently,

* This work was partly done while the author was at FMI, University of Stuttgart, Germany.

the best known lower bound for tree automatic structures of bounded degree is the same as for (word) automatic structures.

As mentioned in the beginning, the basic idea of automatic structures goes back to the definition of *automatic groups*, which have attracted a lot of attention in combinatorial group theory during the last 15 years, see e.g. the textbook [7]. Roughly speaking, a finitely generated group $\mathcal{G}$, generated by the finite set Γ, is automatic, if the elements of $\mathcal{G}$ can be represented by words from a regular language over Γ, and the right multiplication with a generator can be recognized by a synchronized 2-tape automaton. This concept easily yields a quadratic time algorithm for the word problem of an automatic group. It is straight forward to extend the definition of an automatic group to the monoid case; this leads to the class of *automatic monoids*, see e.g. [4, 11, 12, 25]. Analogously to the group case, it is easy to show that for every automatic monoid the word problem can be solved in quadratic time.

From the definition of an automatic structure it follows immediately that the *Cayley-graph* of an automatic monoid is an automatic structure. The Cayley-graph of a finitely generated monoid $\mathcal{M}$ with respect to a finite generating set Γ is a Γ-labeled directed graph with node set $\mathcal{M}$ and an a-labeled edge from a node x to a node y if $y = xa$ in $\mathcal{M}$. Cayley-graphs of groups are a fundamental tool in combinatorial group theory [22] and serve as a link to other fields like topology, graph theory, and automata theory, see, e.g., [23, 24]. Results on the geometric structure of Cayley-graphs of automatic monoids can be found in [26, 27]. Since the Cayley-graph of an automatic monoid is an automatic structure, its first-order theory is decidable. This allows to verify non-trivial properties for automatic monoids, like for instance right-cancellativity. On the other, the author has constructed an automatic monoid for which the first-order theory is not elementary decidable [21]. One should note that the Cayley-graph of a *right-cancellative* automatic monoid is an automatic structure of bounded degree; hence, by the upper bound from [20], its first-order theory can be decided in triply exponential alternating time with a linear number of alternations. For related results on the logical properties of Cayley-graphs of groups and monoids see [18, 19].

Let us end this abstract with a few open questions:

1. Between our upper bound for the complexity of the first-order theory of an automatic structure of bounded degree (triply exponential alternating time with a linear number of alternations) and the best lower bound (doubly exponential alternating time with a linear number of alternations) there is a gap. What is the precise complexity ? For tree automatic structures this gap is even larger.
2. Is there an alternative natural characterization of the class of (tree) automatic structures of bounded degree?
3. Are there other natural classes of automatic structures for which first-order logic is elementary decidable?

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'2001)*, pages 431–440. IEEE Computer Society Press, 2001.
2. A. Blumensath. Automatic structures. Diploma thesis, RWTH Aachen, 1999.
3. A. Blumensath and E. Grädel. Automatic structures. In *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science (LICS'2000)*, pages 51–62. IEEE Computer Society Press, 2000.
4. C. M. Campbell, E. F. Robertson, N. Ruškuc, and R. M. Thomas. Automatic semigroups. *Theoretical Computer Science*, 250(1-2):365–391, 2001.
5. K. J. Compton and C. W. Henson. A uniform method for proving lower bounds on the computational complexity of logical theories. *Annals of Pure and Applied Logic*, 48:1–79, 1990.
6. M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *Proceedings of the 5th Annual IEEE Symposium on Logic in Computer Science (LICS '90)*, pages 242–256. IEEE Computer Society Press, 1990.
7. D. B. A. 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, Boston, 1992.
8. J. Ferrante and C. Rackoff. *The Computational Complexity of Logical Theories.* Number 718 in Lecture Notes in Mathematics. Springer, 1979.
9. H. Gaifman. On local and nonlocal properties. In J. Stern, editor, *Logic Colloquium '81*, pages 105–135. North Holland, 1982.
10. B. R. Hodgson. On direct products of automaton decidable theories. *Theoretical Computer Science*, 19:331–335, 1982.
11. M. Hoffmann. *Automatic semigroups.* PhD thesis, University of Leicester, Department of Mathematics and Computer Science, 2000.
12. J. F. P. Hudson. Regular rewrite systems and automatic structures. In *Semigroups, Automata and Languages*, pages 145–152. World Scientific, 1998.
13. H. Ishihara, B. Khoussainov, and S. Rubin. Some results on automatic structures. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS'2002)*, pages 235–244. IEEE Computer Society Press, 2002.
14. B. Khoussainov and A. Nerode. Automatic presentations of structures. In *LCC: International Workshop on Logic and Computational Complexity*, number 960 in Lecture Notes in Computer Science, pages 367–392, 1994.
15. B. Khoussainov and S. Rubin. Graphs with automatic presentations over a unary alphabet. *Journal of Automata, Languages and Combinatorics*, 6(4):467–480, 2001.
16. B. Khoussainov, S. Rubin, and F. Stephan. Automatic partial orders. To appear in *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS'2003)*, 2003.
17. D. Kuske. Is Cantor's Theorem automatic. In *Proceedings of the 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2003), Almaty (Kazakhstan)*, 2003. this volume.
18. D. Kuske and M. Lohrey. Logical aspects of Cayley-graphs: the group case. to appear in *Annals of Pure and Applied Logic.*
19. D. Kuske and M. Lohrey. Decidable theories of Cayley-graphs. In H. Alt and M. Habib, editors, *Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2003), Berlin (Germany)*, number 2607 in Lecture Notes in Computer Science, pages 463–474. Springer, 2003.

20. M. Lohrey. Automatic structures of bounded degree. In *Proceedings of the 10th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2003), Almaty (Kazakhstan)*, number 2850 in Lecture Notes in Artificial Intelligence, pages 344–358, 2003.
21. M. Lohrey. Decidability and Complexity in Automatic Monoids. submitted, 2004.
22. R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory.* Springer, 1977.
23. D. E. Muller and P. E. Schupp. Groups, the theory of ends, and context-free languages. *Journal of Computer and System Sciences*, 26:295–310, 1983.
24. D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37(1):51–75, 1985.
25. F. Otto and N. Ruškuc. Confluent monadic string-rewriting systems and automatic structures. *Journal of Automata, Languages and Combinatorics*, 6(3):375–388, 2001.
26. P. V. Silva and B. Steinberg. A geometric characterization of automatic monoids. to appear in *Quarterly Journal of Mathematics.*
27. P. V. Silva and B. Steinberg. Extensions and submonoids of automatic monoids. *Theoretical Computer Science*, 289:727–754, 2002.

FA-presentable structures: richness and limitations

Andre Nies

Abstract: To be finite automaton (FA)-presentable seems to be a rather strong restriction on a countable structure in a finite signature: the domain can be represented by a regular set, in a way that finite automata can also verify that the atomic relations hold (where the strings representing elements are written one below the other, using symbols in a power alphabet).
For instance, the additive group of integers is FA-presentable: numbers are represented in binary, and the correctness of the usual carry bit addition algorithm can be verified by a finite automaton. This notion was introduced by Khoussainov and Nerode, 1995. (There and elsewhere, FA-presentable structures are called automatic structures, however, for groups, the term ``automatic" refers to a different concept.)

For each FA-presentable structure one can also give an FA-representation of all definable relations, effectively in the defining formula. So FA-presentable structures are closed under interpretations, and each one has a decidable theory.

When considering an elementary class of effectively given structures, a good measure for its richness is the complexity of the isomorphism problem. For instance, Khoussainov, Nies, Rubin and Stephan (LICS 2004) have proved that for FA-presentable graphs or even successor trees, isomorphism is as complex as possible ($Sigma^1_1$-complete), so the class is quite rich after all. With little effort, the result carries over to partial orders, lattices, and commutative semigroups.

One the other hand, the class of FA-presentable infinite Boolean algebras is very limited: Khoussainov, Nies, Rubin and Stephan have shown that they are just the finite powers of the Boolean algebra of finite or cofinite subsets of a countable set.

The general program is to locate each interesting class somewhere between those two possibilities. I discuss some new results for abelian groups and groups in general, mostly on the limiting side.

Groups: Examples of non-abelian FA-presentable groups include infinite direct powers of a finite non-abelian group, and GL_n(R) for some FA-presentable ring, e.g. the Boolean algebra mentioned before, viewed as a ring. Each finitely generated FA-presentable group is abelian-by-finite (Oliver and Thomas). A recent improvement of this (Nies and Thomas): if G is ANY FA-presentable group, then each f.g. subgroup of G is abelian-by-finite. Using similar methods, each FA- presentable commutative ring with 1 is locally finite.

Abelian groups: Known examples are (Z,+), the Prufer groups, and Z[1/p] for a prime p, and the obvious derived examples (like finite direct products). While the multiplicative group of rationals is non-automatic (Khoussainov e.a.), it is a major open question if the additive rationals are FA-presentable.

$Sigma^1_1$-completeness seems unlikely for groups, so the eventual goal might be a characterization of FA-presentable (abelian) groups.