Tree-automatic scattered linear orders

Sanjay Jain*, Bakhadyr Khoussainov**, Philipp Schlicht, and Frank Stephan***

[1] Department of Computer Science,
National University of Singapore, Singapore 117543, Republic of Singapore
sanjay@comp.nus.edu.sg
[2] Department of Computer Science,
The University of Auckland, Auckland, New Zealand
bmk@cs.auckland.ac.nz
[3] Mathematical Institute,
University of Bonn, Endenicher Allee 60, 53115 Bonn, Germany.
schlicht@math.uni-bonn.de
[4] Department of Mathematics and Department of Computer Science,
National University of Singapore, Singapore 117543, Republic of Singapore
fstephan@comp.nus.edu.sg

Abstract. We study tree-automatic linear orders on regular tree languages. We first show that there is no tree-automatic scattered linear order, and in particular no well-order, on the set of all finite labeled trees. This also follows from results of Gurevich-Shelah [8] and Carayol-Löding [4]. We then show that a regular tree language admits a tree-automatic scattered linear order if and only if all trees are included in a subtree of the full binary tree with finite tree-rank. As a consequence of this characterization, we obtain an algorithm which, given a regular tree language, decides if the tree language can be well-ordered by a tree automaton. Finally, we connect tree automata with automata on ordinals and determine sharp lower and upper bounds for tree-automatic well-orders on natural examples of regular tree languages. Our proofs use elementary techniques of automata theory.

1 Introduction

The aim of this paper is to study tree-automatic linear orders on regular tree languages. More precisely, given a regular tree language A, we would like to know whether A can be ordered by a tree-automatic scattered or well-founded linear order. This is a part of a larger theme where the goal is to classify tree and word-automatic structures. Much work has already been done on the classification of automatic structures in certain classes such as linear orders, Boolean algebras, abelian groups [6] [13] [15] [16] [21] [26], [3]. Recent results by Kuske, Lohrey

* Supported in part by NUS grants R252-000-420-112 and C252-000-087-001.
** Supported in part by the Marsden Fund of New Zealand
*** Supported in part by NUS grants R146-000-114-112 and R252-000-420-112.

and Liu indicate that there is no complete characterization of those linear orders that can be presented by tree automata [17] [18] [19]. Therefore, in this paper we restrict the classification question by considering tree-automatic structures with domain being a fixed regular tree language A. This is formalized in Definition 4. Our hope is to derive algebraic properties of tree-automatic structures with the given domain A, and to obtain some algorithmic consequences from this simple yet natural restriction. Delhommé in [6] proved one of the first important characterization results on tree-automatic structures. Namely, a well-ordered set has a tree-automatic presentation if and only if it is a proper initial segment of the ordinal $\omega^{\omega^{\omega}}$. Our question can also be seen as a refinement of the work of Delhommé. As a consequence of our study, we give an alternative proof of Delhommé's result (See Theorem 22).

The first result of this paper (Theorem 10) shows that there is no tree-automatic scattered linear order on the set $T(\Sigma)$ of all finite binary trees labeled by symbols from a finite alphabet Σ. As a consequence one obtains that there is no tree-automatic well-order on the set $T(\Sigma)$. We note that this consequence can also be derived from Gurevich and Shelah's theorem stating that no monadic second-order definable choice function exists on the infinite binary tree T_2 [8] and finite-set-interpretablity of tree-automatic structures on T_2 [5]. We also note that it is possible to prove Theorem 10 through the theorem of Gurevich and Shelah, and the result of Colcombet and Löding showing that for certain tree-automatic equivalence relation there is a tree-automatic selector function. Our proof of Theorem 10, however, uses basics of tree automata, mainly the Pumping lemma. In this sense, our proof is simpler and elementary. The second result of our paper (Theorem 17) characterizes all tree-automatic languages that admit tree-automatic scattered linear orders. Namely, we prove that a regular tree language has a tree-automatic well-order if and only if the language has a finite tree-rank. Roughly, the language has tree rank k if the maximal height of the full binary finite tree embedded into the language is k. For instance, the language $T(\Sigma)$ has no finite rank. This theorem has an algorithmic consequence. Namely, there exists an algorithm that given a regular tree language decides if the language can be well-ordered by a tree-automaton. The third contribution of this paper is the connection between certain types of tree-automatic structures and finite automata on ordinals. This connection is then used to give an alternative proof of Delhommé's theorem showing that $\omega^{\omega^{\omega}}$ is the smallest ordinal that has no tree-automatic presentation. Finally, we give examples of regular tree languages and describe the spectra (e.g. lower and upper bounds) of tree-automatic well-orders on them.

2 Preliminaries

We start with some basic notions needed for this paper and some background. By a structure $\mathcal{A}$ we mean a tuple $(A, R_1, \ldots, R_n)$, where A is the *domain (or the universe)* of the structure and R_1, ..., R_n are *atomic relations* on A. The main structures of our study will be linearly ordered sets. A structure $\mathcal{A} = (A, \leq)$ is a *linearly ordered set* if $\leq$ is a partial order on A such that for all $x, y \in A$ we have either $x \leq y$ or $y \leq x$. A linearly ordered set $\mathcal{A} = (A, \leq)$ is a *well-order* if every non-empty subset of A has a $\leq$-minimal element. Well-ordered sets are also called *ordinals*. A linearly ordered set $\mathcal{A} = (A, \leq)$ is called *scattered* if no embedding exists from the natural order of the set of all rational numbers into $\mathcal{A}$. There is an equivalent definition of scatteredness in terms of Cantor-Bendixson ranks that we will give at the end of Section 2. To define (word) tree-automatic structures we recall the following definitions from automata theory.

A finite alphabet is denoted by Σ. As always, $\Sigma^\star$ denotes the set of all finite words over Σ. A *finite automaton* is a tuple $\mathcal{M} = (S, \iota, \Delta, F)$, where S is the set of *states* and $\iota \in S$ is the *initial state*, $\Delta \subseteq S \times \Sigma \times S$ is the *transition table*, and $F \subset S$ is the set of *final states*. A *run* of $\mathcal{M}$ on word $w = \sigma_1\sigma_2 \ldots \sigma_n \in \Sigma^\star$ is a sequence of states $q_0, q_1, \ldots, q_n$ such that $q_0 = \iota$ and $(q_i, \sigma_{i+1}, q_{i+1}) \in \Delta$ for all $i \in \{0, 1, \ldots, n-1\}$. If $q_n \in F$, for some run of $\mathcal{M}$ on w, then the automaton $\mathcal{M}$ *accepts* w. The *language* of $\mathcal{M}$ is $L(\mathcal{M}) = \{w \mid w \text{ is accepted by } \mathcal{M}\}$. These languages are called *regular* or *finite automaton recognizable*.

A Σ–tree is a mapping $t : dom(t) \to \Sigma$ such that the domain $dom(t)$ is a finite prefix-closed subset of the binary tree $\{0, 1\}^\star$ such that for every non-leaf node $v \in dom(t)$ we have $v0, v1 \in dom(t)$. The symbol λ denotes the root of $dom(t)$. Every node v in the domain of any Σ–tree is binary string over $\{0, 1\}$. The *boundary* of $dom(t)$ is the set $\partial dom(t) = \{xb \mid x \text{ is a leaf of } dom(t) \text{ and } b = 0 \text{ or } b = 1\}$. The set of all Σ-trees is denoted by $T(\Sigma)$. Sometimes we refer to Σ-trees simply as trees.

Definition 1. A *tree automaton* is a tuple $\mathcal{M} = (S, \iota, \Delta, F)$, where S is the set of *states* and $\iota \in S$ is the *initial state*, $\Delta \subseteq S \times \Sigma \times (S \times S)$ is the *transition table*, and $F \subseteq S$ is the set of *final states*.

A *run* of the tree automaton $\mathcal{M}$ on tree t is a mapping $r : dom(t) \cup \partial dom(t) \to S$ such that $r(\lambda) = \iota$ and $(r(x), t(x0, r(x0), r(x1)) \in \Delta$ for all $x \in dom(t)$. If for every leaf $x \in dom(t)$ we have both $r(x0) \in F$ and $r(x1) \in F$, then the run r is said to be accepting. Automaton $\mathcal{M}$ *accepts* the tree t, if there is a run of $\mathcal{M}$ on t which is accepting. The *tree language* of $\mathcal{M}$ is $L(\mathcal{M}) = \{t \in T(\Sigma) \mid t \text{ is accepted by } \mathcal{M}\}$. These tree languages are called *regular* or *tree-automatic*.

To define tree-automatic and word-automatic structures, we need one technical notion. Let $t_0, \ldots, t_{n-1}$ be trees. For $x \in dom(t_0) \cup \ldots \cup dom(t_{n-1})$ and $i < n$, we set $t'_i(x) = t_i(x)$ if $x \in dom(t_i)$, and $t'_i(x) = \Box$ if $x \notin dom(t_i)$. *The convolution* of the trees $t_0, \ldots, t_{n-1}$ is then the tree given by a mapping $conv(t_0, \ldots, t_{n-1})$ from $dom(t_0) \cup \ldots \cup dom(t_{n-1})$ to $(\Sigma \cup \{\Box\})^n$ which satisfies for all $x \in dom(t_0) \cup \ldots \cup dom(t_{n-1})$ that $conv(t_0, \ldots, t_{n-1})(x) = (t'_0(x), \ldots, t'_{n-1}(x))$. We say that an n-ary relation R on $T(\Sigma)$ is *tree-automatic* if the convolution $conv(R) = \{conv(t_0, \ldots, t_{n-1}) \mid (t_0, \ldots, t_{n-1}) \in R\}$ is a regular tree language. One can easily modify the convolution operation for finite strings, and hence define automatic relations on the set $\Sigma^\star$. These now allow us to give the following definition.

Definition 2. A structure $\mathcal{A} = (A, R_1, \ldots, R_n)$ is called *tree-automatic (word-automatic)* if domain A and the atomic relations R_1, …, R_n are all tree-automatic (automatic). For a structure $\mathcal{B}$, if $\mathcal{B}$ is isomorphic to the structure $\mathcal{A}$ then we say that $\mathcal{A}$ is a *tree-automatic (word-automatic) presentation* of $\mathcal{B}$. We often refer to tree and word-automatic structures and presentations as automatic structures and presentations, respectively.

Automatic presentations $\mathcal{A}$ of a structure $\mathcal{B}$ can be identified with finite collections of automata for the domain and atomic relations of $\mathcal{A}$. Moreover, the definition of automata presentability is a Σ^1_1-definition in arithmetic since automata presentability of $\mathcal{B}$ requires a search for an isomorphism from automatic structures $\mathcal{A}$ to $\mathcal{B}$. However, we often abuse our definition and refer to automata presentable structures as automatic structures. Examples of (tree-)automatic structures are Presburger arithmetic $(\mathbb{N}; +)$ which is word-automatic and Skolem arithmetic $(\mathbb{N}; \times)$ which is tree-automatic but not word-automatic [14]. Other examples of (tree-)automatic structures include term algebras, configuration spaces of Turing machines and the countable atomless Boolean algebra.

The decidability of the emptiness problem and closure properties of regular languages give us the following *uniform decidability theorem*:

Theorem 3 ([2, 10–12]). *Let $FO + \exists^\omega$ be the extension of the first-order logic with the $\exists^\omega$ (there are infinitely many) quantifier. There exists an algorithm, that given an automatic presentation of $\mathcal{A}$ and a formula $\phi(x_1, \ldots, x_n)$ in $FO + \exists^\omega$–logic, produces an automaton that recognises all tuples $(a_1, \ldots, a_n)$ from the structure that make the formula true. In particular the first-order theory of any automatic structure is decidable.*

We use this theorem in this paper often without referencing the theorem directly. However, we will often use a proof of the fact that tree-automatic relations are closed under $\exists^\omega$–quantifier. Therefore, for completeness of the paper, we provide a simple proof of this fact in the next section.

Blumensath and Grädel [2] address automaticity of structures in terms of interpretability. They proved that there are specific automatic structures that encompass all automatic structures in the first-order logic. For instance, a structure is word-automatic if and only if it is first-order interpretable in the following extension of Presburger arithmetic $(\omega; +, |_2)$, where $x|_2 y$ iff x is a power of 2 and y is a multiple of x. In this sense, automaticity is equivalent to first-order interpretability. There are other logical characterisations of automaticity, e.g. through finite set interpretations as in [5], and the reader is referred to [5] [23].

The definition below is central to this paper and refines the definition of automaticity by placing the emphasis on automatic domains. For the definition, we fix a class of structures K, where structures are identified up to isomorphism. For instance, K can be the class of well-ordered sets, undirected graphs of bounded degree, trees, Abelian groups and so on.

Definition 4. For a regular tree language X, *the algebraic spectrum of X with respect to the class K*, denoted by $AlgSpec_K(X)$, is the class of all structures

$\mathcal{B} \in K$ such that there exists a tree-automatic structure $\mathcal{A}$ with domain X isomorphic to $\mathcal{B}$. If $\mathcal{B} \in AlgSpec_K(X)$ then we say that *the set X admits (the isomorphism type of) the structure $\mathcal{B}$*. The spectrum $AlgSpec_K(X)$ for word-automatic languages X is defined similarly.

For example, no tree-automatic (or word-automatic) language admits a structure with undecidable first-order theory. The results in [24] show that $0^\star$ admits a well-order α if and only if $\alpha < \omega^2$. In [24] it is proven that if X is regular and X admits an ordinal α then $\alpha < \omega^\omega$. Generally, Delhommé [6] showed that if X is regular and X admits a well-founded partial order $\mathcal{A}$ then the height of $\mathcal{A}$ is below ω^ω (see also [13]). Also, as mentioned above, Delhommé [6] shows that no regular tree language admits ordinals greater or equal to ω^{ω^ω}. Another nice example is the result by Tsankov in [26] showing that no regular language admits the additive group of rational numbers $(\mathbb{Q};+)$. We stress that Definition 4 calls for a refined analysis of automaticity, and hence interpretability, of structures. Proving that a certain structure (e.g. the ordinal ω^n) is not admitted by a given regular or regular tree language requires a deep analysis of underlying automata and understanding algebraic or model-theoretic properties of underlying structures of interest. In this paper the class K as in Definition 4 will be the class of linearly ordered sets.

For this paper, we need to define Cantor-Bendixson ranks (CB-ranks) of linearly ordered sets. These are ordinals assigned to linearly ordered sets $\mathcal{L} = (L, \leq)$. Elements $x, y \in L$ are $\sim$–equivalent if the interval between x and y is finite. The relation $\sim$ is an equivalence relation. The order $\leq$ naturally induces a linear order on the quotient set $\mathcal{L}/\sim$. Denote the resulting order by $\mathcal{L}'$. This new linearly ordered set $\mathcal{L}'$ is sometimes called *the derivative of* $\mathcal{L}$. We iterate this process and produce the sequence of derivatives as follows: $\mathcal{L}_0 = \mathcal{L}$, $\mathcal{L}_1 = \mathcal{L}_0'$, $\mathcal{L}_{\alpha+1} = {\mathcal{L}_\alpha}'$ and $\mathcal{L}_\beta$ is the quotient of $\mathcal{L}$ by the union of the equivalence relations for $\mathcal{L}_\alpha$ for $\alpha < \beta$ if β is a limit ordinal.

Definition 5. We say that a linearly ordered set $\mathcal{L}$ is *scattered* if there exists an α such that $\mathcal{L}_\alpha$ is a finite linearly ordered set. The least ordinal α for which $\mathcal{L}_\alpha$ is finite is called the *Cantor-Bendixson rank* of $\mathcal{L}$. We denote it by *CB-rank($\mathcal{L}$)*.

It is well-known that $\mathcal{L}$ is scattered if and only if there is no embedding of the order of the rational numbers into $\mathcal{L}$ [22]. This justifies our original definition of scatteredness given at the beginning of the introduction. Examples of scattered linearly ordered sets are the order of integers, well-ordered sets and their finite sums and products.

3 Basic results

In this section we assume that the cardinality of Σ is at least 2. Our first result shows that the set $T(\Sigma)$ of all Σ–trees admits a tree-automatic dense linear order without end-points, that is the order of the rational numbers.

Proposition 6. *The language $T(\Sigma)$ admits the order of the rational numbers.*

Proof. Without loss of generality we assume that $\Sigma = \{a, b\}$ and we declare $a < b$. For any given two trees $p, q \in T(\Sigma)$ such that $p \neq q$ consider the left-most node $x_{(p,q)}$ in the convolution tree $conv(p, q)$ for which $p'(x_{(p,q)}) \neq q'(x_{(p,q)})$, where p' and q' are defined in the definition of convolution operation for trees. Here the left-most node is taken with respect to the pre-order on the nodes of the tree $conv(p, q)$. Now we define the relation $\sqsubseteq$ on $T(\Sigma)$ as follows. For trees $p, q \in T(\Sigma)$ declare $p \sqsubseteq q$ if and only if either $p = q$ or $conv(p, q)(x_{(p,q)}) \in \{(a, b), (\Box, b), (a, \Box)\}$.

We now claim that the relation $\sqsubseteq$ is the desired one. Note that the relation is tree-automatic. A tree automaton recognising this relation can be described as follows. On input $conv(p, q)$ the automaton non-deterministically selects a path leading to $x_{(p,q)}$. At all nodes v left of $x_{(p,q)}$ the automaton verifies that $p(v) = q(v)$. Once the node $x_{(p,q)}$ is reached the automaton accepts the tree. If node $x_{(p,q)}$ does not exist then the automaton fails along the non-deterministically chosen path that searches for $x_{(p,q)}$.

It is not hard to verify that the relation $\sqsubseteq$ is a linear order on $T(\Sigma)$. We need to show that $\sqsubseteq$ is dense and has no end-points. Indeed, take a tree $p \in T(\Sigma)$. Let v be any leaf of p; thus $x_{(p,q)}$ is a prefix of v . We now extend p to p_1 such that $p_1(v0) = a$ and $p_1(v1) = b$, and we extend p to p_2 such that $p_2(v0) = b$ and $p_2(v1) = b$. In this way we have $p_1 \sqsubseteq p \sqsubseteq p_2$. Hence, $\sqsubseteq$ is a linear order without end-points.

Let p, q be such that $p \neq q$ and $p \sqsubseteq q$. Consider $x_{(p,q)}$. Assume that $p(x_{(p,q)}) = a$ and either $q(x_{(p,q)}) = b$ or $q(x_{(p,q)}) = \Box$. Let v be a leaf of p above $x_{(p,q)}$. Extend p to p_2 (as above) using v. Then $p \sqsubseteq p_2 \sqsubseteq q$. Assume that $p(x_{(p,q)}) = \Box$ and $q(x_{(p,q)}) = b$. Let w be a leaf of q above $x_{(p,q)}$. Extend q to q_1 using the node w. Then $p \sqsubseteq q_1 \sqsubseteq q$. Hence the linear order $\sqsubseteq$ is dense. ∎

The next two propositions prove two known results. The first shows that tree-automatic structures are closed under taking the quotients with respect to tree-automatic congruence relations. The second shows that tree-automatic relations are closed under the *there are infinitely many quantifier operation* $\exists^{\omega}$. We provide the proofs of these two facts since we will use the ideas of their proofs further in this paper. Note that the proof of the first fact in [1] contains unrecoverable error. Colcombet and Löding in [5] give a correct proof of the the first fact in more general ω-tree automata setting. Our proof is more direct and considers the simpler finite case.

Proposition 7. *Suppose $\sim$ is a tree-automatic equivalence relation on a tree-automatic set A. There is a tree-automatic function f picking representatives from the $\sim$–equivalence classes.*

Proof. For each $p \in A$, set $t(p)$ to be the Σ–tree such that the domain of $t(p)$ is the set $\cap_{q \sim p} dom(q)$ and $t(p)$ labels every node of its domain by some default value. The tree $t(p)$ does not need to be in A. There is a constant c such that for all $p \in A$ there is a $q \in A$ for which $q \sim p$ and every node in q is at distance at most c from $dom(t(p))$. To prove this, we list all leaves $u_0, u_1, \ldots, u_{n-1}$ in $dom(t(p))$. We start with $q_0 = p$, and for each $m < n$ proceed, inductively, as follows. There is a Σ–tree $t \sim q_m$ which does not contain u_m0 and u_m1. By the pumping lemma, there is a $q_{m+1} \sim t$ such that q_{m+1} coincides with q_m on all nodes below u_m and the height of the subtree of q_{m+1} above u_m is bounded by a constant. By transitivity we have $q_m \sim q_{m+1}$. Doing this with all nodes $u_0, \ldots, u_{n-1}$, we produce a tree $q_n \sim p$. The tree q_n is the desired tree q. Consider the set $S(p) = \{q \in A \mid p \sim q$ and each $x \in dom(q)$ is at most c edges away from $dom(t(p))\}$. The relation $q \in S(p)$ is tree-automatic. We define $f(p)$ as follows. Restrict the order $\sqsubseteq$ from Proposition 6 to $S(p)$ and take $f(p)$ to be the least element of $S(p)$ with respect to the order $\sqsubseteq$. ∎

Proposition 8. *Let $R(x, y, \bar{a})$ be a tree-automatic binary relation in variables x and y and fixed parameters $\bar{a} = (a_1, \ldots, a_n) \in T(\Sigma)^n$. Consider the following set $S = \{p \in T(\Sigma) \mid$ there exist at most finitely many q such that $R(p, q, \bar{a})\}$. The relation S is tree-automatic.*

Proof. For trees $p, q \in T(\Sigma)$ we write $p \subseteq_d q$ if and only if $dom(p) \subseteq dom(q)$. It is clear that $\subseteq_d$ is tree-automatic. Consider the following set S':

$$S' = \{p \mid \exists q \in T(\Sigma) \forall t (R(p, t, \bar{a}) \rightarrow t \subseteq_d q)\}.$$

The relation S' is first-order definable from tree-automatic relations $\subseteq_d$ and R. Therefore the relation S' is regular. It is now not hard to verify that $S = S'$. ∎

The next corollary can be viewed as a geometric interpretation of the proposition stated above.

Corollary 9. *Consider the relations R and S as in Proposition 8. For every $p \in S$ define the following Σ–tree $\phi(p)$:*

(a) $dom(\phi(p)) = \cup_{R(p,q)} dom(q)$.
(b) $\phi(p)$ labels every node $v \in dom(\phi(p))$ by a default value, say by $a \in \Sigma$.

The function $\phi : p \rightarrow \phi(p)$, where $p \in S$, is tree-automatic. Hence, there exists a constant c such that every node $v \in dom(\phi(p))$ is at most c edges away from a node in the domain of p.

4 Non-scatteredness

In this section we prove that the set $T(\Sigma)$ of all Σ-trees does not admit a scattered tree-automatic linear order. As a corollary one obtains that there is no tree-automatic well-order on the set $T(\Sigma)$. Our proof uses the Pumping lemma for tree automata and the uniform decidability theorem. In this sense our proof is elementary and self-contained (yet a bit technical).

There are alternative ways to prove this. For instance, the corollary can be proved using Gurevich and Shelah's theorem stating that no monadic second-order (MSO) definable choice function exists on the infinite binary tree T_2 [8] and the fact that a structure is tree-automatic iff it is finite set-interpretable in T_2 [5]. Another possible way to prove the theorem is the following. First, show that every MSO definable scattered linear order on $L \subseteq \{0,1\}^\star$ can be used to define an MSO definable well-order on L. Secondly, show that if $T(\Sigma)$ has a tree-automatic scattered linear order then the set of all *slim* trees has an MSO definable scattered order. A tree is *slim* if its branching nodes form a chain. The slim trees determine an MSO definable scattered linear order, and hence also a well-order, on $\{0,1\}^\star$. This contradicts the theorem of Gurevich and Shelah. We mention the result of A. Carayol and C. Löding in [4] for an automata-theoretic proof of Gurevich-Shelah's theorem as opposed to set-theoretic techniques in [8].

Theorem 10. *$T(\Sigma)$ does not admit a tree-automatic scattered linear order.*

Proof. Suppose by way of contradiction that $\leq$ is a tree-automatic scattered linear order on $T(\Sigma)$. Consider the set $A = \{t \in T(\Sigma) \mid \text{for all } v \in dom(t)(t(v) = a)\}$, where $a \in \Sigma$ is fixed. We write $x \in t$ instead of $x \in dom(t)$. The restriction of $\leq$ to A is scattered tree-automatic linear order on A. For rest of the proof we consider trees from A.

A node in t is branching if it has exactly two successors. A node $\tilde{x}$ *generates* a tree x iff $\tilde{x}$ is a branching node in x and every branching node $\tilde{y} \in x$ is a prefix of $\tilde{x}$. Define $B = \{x \in A \mid x \text{ is generated by some node } \tilde{x}\}$. We use x, y, z and U, S, T to refer to trees in B and A, respectively. For $x \in B$, $\tilde{x}$ denotes the node that generates x. For each $x \in B$ let σ_x be the the path from the root λ to $\tilde{x}$. Notice that σ_x uniquely determines x for $x \in B$. The set $Strings = \{\sigma_x \mid x \in B\}$ admits the order $\sqsubseteq$, where $\sigma_x \sqsubseteq \sigma_y$ iff $x \leq y$. The word-automatic linear order $(Strings, \sqsubseteq)$ is isomorphic to $(B, \leq)$. The CB-rank r of $(Strings, \sqsubseteq)$ is finite as proved in [16] [24].

Consider the sequence of derivatives of B defined just before Definition 5: $B_0, B_1, \ldots, B_r$, where $B_0 = B$. Let $n = |B_r|$ and m be the number of infinite $\sim$ equivalence classes in B_{r-1}. We can assume that the order $\leq$ on $T(\Sigma)$ is chosen such that the triple (r, n, m), called the *extended rank of* $(B, \leq)$, is the smallest possible with respect to lexicographical ordering of triples. Note that $r \geq 1$.

We now give an inductive analysis of the sequence $B_0, B_1, \ldots, B_r$, define the relations $\leq_{m+1}$, $\sim_{m+1}$ and C_{m+1}, constants c_{m+1} and d_{m+1}, and functions t_{m+1}, f_{m+1} and rep_{m+1}, where $0 \leq m < r$. For $m = 0$ we have: $B_0 = B$, $\sim_0 = \{(x,x) \mid x \in B\}$, and $\leq_0 = \leq$. So, assume that for m, we have already defined B_m, $\sim_m$ and $\leq_m$ on B_m.

For $x, y \in B_m$, set $C_{m+1}(x,y)$ be the interval $[x,y]$ if $x \leq_m y$ and $[y,x]$ if $y \leq_m x$. Write $x \sim_{m+1} y$, where $x, y \in B_m$, if the interval $C_{m+1}(x,y)$ is finite. Note that $x \sim_{m+1} y$ iff there is a tree $U \in A$ such that for all $z \in C_{m+1}(x,y)$ we have $dom(z) \subseteq dom(U)$. Hence $\sim_{m+1}$ is recognised by a tree automaton.

We recast the proof of Proposition 7 to extract the constant c_{m+1} and the function rep_{m+1} that selects representatives from the $\sim_{m+1}$-classes. For $x \in B_m$, let $t_{m+1}(x)$ be the intersection of all the trees $y \in B_m$ with $x \sim_{m+1} y$. There exists a constant c_{m+1} independent of x such that for some $y \in B_m$ we have $C_{m+1}(x,y)$ is finite and y is at most c_{m+1} edges away from $t_{m+1}(x)$. Then $rep_{m+1}(x)$ is the length-lexicographically least y such that $C_{m+1}(x,y)$ is finite and y is at most c_{m+1} edges away from $t_{m+1}(x)$. Now set $B_{m+1} = \{rep_{m+1}(x) \mid x \in B_m\}$ and $\leq_{m+1}$ be the $\leq$ restricted to B_{m+1}. Thus, we have the following:

Claim 11. *There is a descending sequence $B_0, B_1, \ldots, B_r$ of subsets of B such that $B_0 = B$ and, for $m = 0, 1, \ldots, r$*

(a) Each of B_0, B_1, …, B_r is tree-automatic.
(b) Each tree-automatic linearly ordered set $(B_m, \leq_m)$, $m = 0, \ldots, r$, is isomorphic to the mth derivative of $(B, \leq)$.
(c) For each B_m, $m = 0, \ldots, r$, for each $x \in B_m$ there is exactly one $y \in B_{m+1}$, where $y = rep_{m+1}(x)$, such that $C_{m+1}(x, y)$ is finite.
(d) The function $x \to rep_{m+1}(x)$, where $x \in B_m$, is tree-automatic. Moreover, there exists a constant c_{m+1} such that for all $x \in B_m$ we have $rep_{m+1}(x)$ is at most c_{m+1} edges away from $t_{m+1}(x)$.

The next claim defines the constant d_{m+1} and the function f_{m+1}. The proof follows from the first-order definability and the Pumping lemma.

Claim 12. *For each $x \in B_m$ let $f_{m+1}(x)$ be the tree $U \in A$ such that U is the union of all the domains of $y \in B_m$ such that $y \in C_{m+1}(x, rep_{m+1}(x))$. Then the mapping $x \to f_{m+1}(x)$, where $x \in B_m$, is tree-automatic and there is a constant d_{m+1} such that every node in $f_{m+1}(x)$ is at distance at most d_{m+1} from a branching node in x.*

Without loss of generality, we assume that $d_m \geq c_m$, $m = 1, \ldots, r$.

Claim 13. *There exists a tree $z \in B$ such that $(\{u \in B \mid \tilde{z}$ is a branching node in $u\}, <)$ has extended rank (r', n', k') with $(r', n', k') <_{lex} (r, n, k)$.*

Proof. Start with a tree x in B_r such that the $\sim_r$–equivalence class of x in B_{r-1} is infinite. There are $v, w \in B_{r-1}$ satisfying the following conditions:

- $v \leq x \leq w$;
- $v \sim_r w$;
- either $v = \min\{u \in B_{r-1} : u \sim_r x\}$ or $|\tilde{v}| > |\tilde{x}| + d_1 + \ldots + d_r + 2$;
- either $w = \max\{u \in B_{r-1} : u \sim_r x\}$ or $|\tilde{w}| > |\tilde{x}| + d_1 + \ldots + d_r + 2$.

We fix v and w chosen above. Take any sequence $x_r, x_{r-1}, \ldots, x_0$ that satisfies the following conditions:

1. $x_r = x$ and $x_m \in B_m$, where $m = 0, 1, \ldots, r$.
2. $x_m \sim_{m+1} x_{m+1}$, where $m \in \{0, 1, \ldots, r-1\}$.
3. Either $x_{r-1} < v$ or $x_{r-1} > w$.

Assume first that $x_{r-1} < v$. Consider now the following sequence of nodes

$$\tilde{v}_r, \tilde{v}_{r-1}, \ldots, \tilde{v}_1, \tilde{v}_0$$

where $\tilde{v}_r = \tilde{v}$, and each $\tilde{v}_m$ is obtained from $\tilde{v}_{m+1}$ by omitting the top d_{m+1} edges of $\tilde{v}_{m+1}$ for $m = r-1, \ldots, 1, 0$.

Now, by inverse induction, using the conditions put on the constants and functions f_r, one can prove that $\tilde{v}_m$ is a prefix of $\tilde{x}_{m+1}$ for $m = r-1, r-2, \ldots, 1, 0$.

Due to the the length of v, we have $|\tilde{v}_0| \geq 2$. Again using induction, one can show that if $\tilde{x}_{r-1} > w$ then $\tilde{x}_0$ extends the prefix $\tilde{w}_0$ of w of length 2.

Now choose z such that $\tilde{z}$ has length 2 and $\tilde{z}$ is different from $\tilde{v}_0, \tilde{w}_0$. Hence $\tilde{x}_0$ cannot be a node which extends $\tilde{z}$. For $u \in B$, let $REP_0(u) = u$ and define $REP_{i+1}(u)$ as follows: let $u_0 = u$, $u_{i+1} = rep_{i+1}(u_i)$; $REP_{i+1}(u) = u_{i+1}$. Now, all $u \in B$ containing $\tilde{z}$ as a branching node satisfy: $REP_r(u) \neq x$, or $REP_{r-1}(u) = y$ for some $y \in B_{r-1}$ with $v \leq y \leq w$.

Now consider the set: $Z = \{u \in B \mid \tilde{z} \text{ is a branching node in } u\}$. As, for all $u \in Z$, $REP_r(u) \neq x$, or $REP_{r-1}(u) = y$ for some $y \in B_{r-1}$ with $v \leq y \leq w$, we immediately have that extended rank of $(Z, \leq)$ is less than (r, n, m). There are now two cases to consider.

Case 1. For all the $u \in B$ containing $\tilde{z}$ as a branching node, $REP_{r-1}(u) \not\sim_r x$. In this case the linearly ordered set $(\{u \in B \mid \tilde{z} \text{ is a branching node in } u\}, <)$ has rank (r', n', m') such that either $r' < r$ or $r' = r$ and $n' < n$. Hence, z is the desired tree.

Case 2. There exists a tree $u \in B$ containing $\tilde{z}$ as a branching node such that $REP_{r-1}(u) = y$ for some $y \in B_{r-1}$ with $v \leq y \leq w$. In this case the linearly ordered set $(\{u \in B \mid \tilde{z} \text{ is a branching node in } u\}, <)$ has a smaller rank (r', n', m') such that if $r' = r$ and $n' = n$ then $m' < m$. Hence, z is again the desired tree. ∎

Claim 14. *There exists, in contradiction to the assumption, a scattered tree-automatic linear order $(T(\Sigma), \leq')$ such that the extended rank of $(B, \leq')$ is strictly less than (r, n, k).*

Indeed, take the tree z from the previous claim. Consider the set A' all Σ-trees which contain $\tilde{z}$ as a branching node and which do not have branching nodes incomparable to $\tilde{z}$. Let $\leq'$ be the restriction of $\leq$ to A'. Thus, $\mathcal{A}' = (A', \leq')$ can be viewed as a tree-automatic scattered linear order of the set $T(\Sigma)$. The linearly ordered set $(B, \leq')$ has extended rank (r', n', m') smaller than the extended rank of $(B, \leq)$. This contradicts the choice of $\leq$. ∎

Corollary 15. *The set $T(\Sigma)$ of all Σ–trees does not admit any well-ordering.*

Note that almost all parts of the proof of Theorem 10 worked only on the trees in B of the special form which are generated by nodes. A straightforward generalization gives the following corollary.

Corollary 16. *Let $\alpha, \beta, \gamma, \delta$ be strings such that β and γ are different but of the same length. Let $A = \{x : \text{some node } \tilde{x} \in \alpha(\beta \cup \gamma)^* \delta \text{ generates the tree } x\}$. Then there is no tree-automatic scattered linear order on A.*

5 A characterisation result

The techniques of the previous section can be applied to characterize regular tree languages that admit scattered linear orders. For this, we need to introduce the concept of tree rank. We say that a tree t (labeled or not labeled) *embeds* into tree q if there exists an injective map h from t into q that preserves the prefix relation. By n-th full binary tree we mean the finite tree t_n of height n in which every node at height $< n$ has exactly two children in t_n. We say that an infinite tree T has *tree-rank* n, written $tr(T) = n$, if t_n embeds into T but t_{n+1} does not embed into T.

Theorem 17. *Let A be an infinite regular tree language. Then the following three conditions are equivalent.*

(1) There exists a tree-automatic linear ordering $\leq$ such that $(A, \leq)$ is a scattered linearly ordered set.
(2) There exists a tree-automatic linear ordering $\sqsubseteq$ such that $(A, \sqsubseteq)$ is a well-ordered set.
(3) The infinite tree $\mathcal{A} = \bigcup\{dom(t) : t \in A\}$ forms a tree of finite tree-rank.

Proof. The implication (2)$\rightarrow$ (1) is obvious. We prove the implication (3)$\rightarrow$ (2). Assume that (3) is true. One way to prove this is by induction on the rank. But below we provide a more explicit proof. There exists a finite binary tree t such that t cannot be embedded into the tree $\mathcal{A}$. Below the ranks are with respect to the tree $\mathcal{A}$; that is, the tree-rank of a node $\tilde{x} \in \mathcal{A}$ is the tree-rank of the subtree of $\mathcal{A}$ above $\tilde{x}$. Hence, for each branching node $\tilde{x}$ in $\mathcal{A}$ and its immediate successors $\tilde{x}0$ and $\tilde{x}1$ we have one of the following:

– The tree-ranks of $\tilde{x}0$ and $\tilde{x}1$ are both below that of $\tilde{x}$.
– One of the immediate successors has tree-rank that is equal to the tree-rank of $\tilde{x}$ and the other immediate successor has strictly smaller tree-rank.

Now one introduces a naming-function Λ from the nodes of A to strings where the name of the root is the empty string and for every node $\tilde{x}$ for which Λ has already been defined and which satisfies $\tilde{x}, \tilde{x}0, \tilde{x}1 \in \mathcal{A}$, the value of Λ is defined on the successors of $\tilde{x}$ as follows:

– $\Lambda(\tilde{x}0) = \Lambda(\tilde{x})a$ and $\Lambda(\tilde{x}1) = \Lambda(\tilde{x})b$ iff the tree-rank of $\tilde{x}0$ is strictly below the tree-rank of $\tilde{x}$;
– $\Lambda(\tilde{x}0) = \Lambda(\tilde{x})b$ and $\Lambda(\tilde{x}1) = \Lambda(\tilde{x})a$ iff the tree-rank of $\tilde{x}0$ equals to the tree-rank of $\tilde{x}$ and (therefore) the tree-rank of $\tilde{x}1$ is strictly below the tree-rank of $\tilde{x}$.

From the definition of Λ we see that whenever $\tilde{x}, \tilde{y} \in \mathcal{A}$ have a common prefix of length n then the first n symbols of $\Lambda(\tilde{x})$ and $\Lambda(\tilde{y})$ are the same.

As we have already noted the tree-rank of at least one of the nodes $\tilde{x}0$ and $\tilde{x}1$ must be strictly below the tree-rank of $\tilde{x}$. Furthermore, the tree-rank is always one of the numbers $0, 1, \ldots, c$, where c is a some constant. For each of the possible tree-ranks d there is a first-order formula which tells us whether the tree-rank of a given node is at least d. Therefore, by the uniform decidability Theorem 3, the mapping Λ is tree-automatic. Moreover, there exists a tree-automatic predicate which checks whether $\Lambda(\tilde{x}) <_{lex} \Lambda(\tilde{y})$. We note that as the tree-rank of the root is at most c, there is no node $\tilde{x}$ in A for which $\Lambda(\tilde{x})$ contains more than c a's.

Now we define the relation $\sqsubseteq$ on the given regular set A of trees as follows. For trees $s, t \in A$, we set $s \sqsubseteq t$ if and only if either $s = t$ or the domains of s, t are different and the node $\tilde{x}$ in the symmetric difference (of the domains) with the lexicographically largest value $\Lambda(\tilde{x})$ satisfies $\tilde{x} \in t - s$ or the domains are equal and the lexicographically maximal $\tilde{x}$ for which the labels are different satisfies that the label of s is strictly before the label of t in some predefined ordering of the labels. This relation $\sqsubseteq$ is definable from tree-automatic relations. Hence $\sqsubseteq$ is a tree-automatic relation well-order.

What remains to show is (1) implies (3). For this we show that if (3) is false then (1) is false.

So assume that the tree-rank of the union $\mathcal{A}$ of the domains of trees in A is infinite. Obviously $\mathcal{A}$ is a regular set of strings. There is a string α where the automaton accepting $\mathcal{A}$ takes the state after processing α also again after processing $\alpha\beta$ and $\alpha\gamma$ for two incomparable extensions — if such an α would not exist, the tree-rank would be finite. Without loss of generality these extensions are of the same length — otherwise replace β and γ by $\beta^{|\gamma|}$ and $\gamma^{|\beta|}$, respectively. There is a string δ which permits to go from the state after α into an accepting state. Hence $\alpha(\beta \cup \gamma)^*\delta \subseteq \mathcal{A}$.

For each $\tilde{x} \in \alpha(\beta \cup \gamma)^*\delta$, we assign besides the tree x another tree, denoted by τ_x, from the language A as follows. As A is recognised by a tree automaton there is for each $\tilde{x}$ in $\mathcal{A}$ a tree in A containing $\tilde{x}$ which is recognised by the automaton for A. One can choose this tree such that the automaton recognising the tree does not repeat any state after leaving the domain of x. Hence there is a constant c such that all nodes in such a tree are at a distance at most c from the domain of x. Among all the trees of the language A which contain the node $\tilde{x}$ and have this distance property there is a lexicographically least one. We set τ_x be this tree. The mapping from x to τ_x is tree-automatic. Furthermore, we can without loss of generality assume that the mapping is one-one, as otherwise we might just replace β and γ by β^{c+1} and γ^{c+1}, respectively. If now A would admit a tree-automatic scattered linear order $\leq$ then the set $B = \{x : \tilde{x} \in \alpha(\beta \cup \gamma)^*\delta\}$ would also admit a tree-automatic scattered linear order $\leq'$ by

$$x \leq' y \Leftrightarrow \tau_x \leq \tau_y.$$

However, such an order cannot exist by Corollary 16. Therefore the negation of (3) implies the negation of (1). ∎

The tree-rank of $\mathcal{A} = \bigcup\{dom(t) : t \in A\}$ is either infinite or bounded by the number of states of a deterministic tree-automaton accepting the nodes in $\mathcal{A}$. Therefore one can determine this upper bound and then use the first-order formulas described in Theorem 17 and apply Theorem 3 to check if the tree-rank of $\mathcal{A}$ is properly above this bound. This decides whether $\mathcal{A}$ has finite tree-rank and thus we have

Corollary 18. *It is decidable if a given regular tree language can be well-ordered by a tree automaton.*

The next theorem shows that the set $\Sigma^\star$ of all strings admits all word-automatic scattered linear orders. This stands in sharp contrast to Theorem 10. We need the following simple lemma.

Lemma 19. *If $\mathcal{L} = (L, \leq)$ is a word-automatic linearly ordered set with at least one infinite $\sim$-class then the set $\Sigma^\star$ admits $\mathcal{L}$.*

The proof is easy. Recall that $x \sim y$ if there are finitely many elements in the interval determined by x and y. Consider an infinite $\sim$-equivalence class $[x] = \{y \mid x \sim y \text{ in } \mathcal{L}\}$. This class $[x]$ is a regular language. Consider the following regular language:

$$C = [x] \cup (\Sigma^\star \setminus L).$$

The linearly ordered set $([x], \leq)$, where $\leq$ is the order in $\mathcal{L}$, is isomorphic to either the positive integers or the negative integers or all integers. Assume, without loss of generality, that $([x], \leq)$ is isomorphic to the positive integers, that is, to the ordinal ω. We now write $\Sigma^\star$ as follows:

$$L_{[x]} \cup C \cup R_{[x]},$$

where $L_{[x]} = \{z \in L \mid z < x \text{ and } z \notin [x]\}$ and $R_{[x]} = \{z \in L \mid z > x \text{ and } z \notin [x]\}$. Define the following linear order $\leq_{new}$. The order $\leq_{new}$ preserves the old order $\leq$ on the sets $L_{[x]}$ and $R_{[x]}$, orders the strings in $[x] \cup (\Sigma^\star \setminus L)$ length-lexicographically, and declares all the elements in C be greater than all elements in $L_{[x]}$, and all the elements in C be less than all elements in $R_{[x]}$. The linear order $\leq_{new}$ is clearly word-automatic.

It is easy to see that $\leq_{new}$ is a linear order on $\Sigma^\star$. In addition, the original word-automatic linearly ordered set $\mathcal{L}$ is isomorphic to $(\Sigma^\star, \leq_{new})$. Hence, $\Sigma^\star$ admits $\mathcal{L}$. This proves the lemma. ∎

Theorem 20. *The set $\Sigma^\star$ admits every word-automatic infinite scattered linear order.*

Proof. Every word-automatic infinite scattered linearly ordered set has at least one infinite $\sim$-class. So the theorem follows from the following lemma. ∎

We do not know if the theorem above is true for all infinite word-automatic linear orders.

6 Lower and upper bounds

Let A be an infinite tree-automatic set of trees which has a tree-automatic well-ordering. Now let $minord(A)$ and $maxord(A)$ be the minimum and the supremum, respectively, of the ordinals α such that A admits a tree-automatic well-ordering of type α. In this section we study the possible range of values $minord$ and $maxord$ can take. If all branching nodes in the trees in A are of the form 1^n and Σ is unary, then $minord(A) = \omega$ and $maxord(A) = \omega^2$. If $|\Sigma| > 1$ then then $minord(A) = \omega$ and $maxord(A) = \omega^\omega$ (see for instance [24]).

We will generalize the situation above to the case when the tree-rank of $\bigcup\{dom(t) : t \in A\}$ is at most k. For this, we connect tree-automatic structures with structures recognized by automata running along an ordinal of the form ω^k. Suppose γ is an ordinal and Σ is a finite alphabet. A (deterministic) γ-*automaton* is a finite automaton together with a limit transition function $\mathcal{P}(S) \to S$, where S is the set of states. The letters of a finite input word $w : \gamma \to \Sigma \cup \{\Box\}$ (i.e. all but finitely many of its letters are $\Box$) are read successively. At every limit time $\lambda \leq \gamma$ the state is determined by the limit transition function applied to the set of states appearing unboundedly often before λ, as in Muller automata. The input is accepted if the state at γ is accepting.

Definition 21. Suppose γ is an ordinal and A is a set of finite γ-words. A structure $\mathcal{A} = (A, R_1, ..., R_n)$ is (finite word) γ-automatic if the domain and the relations are recognizable by (deterministic) γ-automata.

If A is a set of Σ-trees, we write $T_A = \{dom(t) \colon t \in A\}$.

Theorem 22. *Let C_k denote the set of binary trees such that each branching node contains at most k 0s. The following conditions are equivalent for (relational) structures $\mathcal{A}$.*

(1) $\mathcal{A}$ is isomorphic to a tree-automatic structure with domain $B \subseteq C_k$.
(2) $\mathcal{A}$ is isomorphic to a tree-automatic structure with domain B such that $tr(T_B) \leq k+1$.
(3) $\mathcal{A}$ is isomorphic to an ω^{k+1}-automatic structure.

Proof. The implication (1)→(2) holds since $tr(T_{C_k}) = k+1$.

Let us prove the implication (2)→(3). We would like to simulate the run of the tree automaton on a tape of length ω^{k+1}. Let $T = T_B$. Let $T_\sigma := \{\tau \in T : \tau \subseteq \sigma \vee \sigma \subseteq \tau\}$ and $tr(\sigma) := tr(T_\sigma)$ for $\sigma \in T$. Consider the following injection $F : T \to \omega^{k+1}$. Let $F(\lambda) = 0$. If $tr(\sigma 0) \neq tr(\sigma 1)$, let $F(\sigma 0) = F(\sigma) + \omega^{tr(\sigma 0)}$ and $F(\sigma 1) = F(\sigma) + \omega^{tr(\sigma 1)}$. If $tr(\sigma 0) = tr(\sigma 1) < tr(\sigma)$, let $F(\sigma 0) = F(\sigma) + \omega^{tr(\sigma)}$ and $F(\sigma 1) = F(\sigma) + \omega^{tr(\sigma 1)}$. Now every Σ-tree t defines a finite ω^{k+1}-word w with $w(\alpha) = t(s)$ if $F(s) = \alpha$. We simulate the run of the tree automaton on input t by a nondeterministic ω^{k+1}-automaton on input w as follows. While the tree automaton runs simultaneously along several branches, the ω^{k+1}-automaton will run the various computations one after the other, in the order of the F-images of the nodes. Once we are at a limit, we need to recall the state at a previous time. The T-predecessor of s with $F(s) = \omega^{n_0} + \omega^{n_1} + ... + \omega^{n_l}$ and $n_0 \geq ... \geq n_l$ is the unique $r \in T$ with $F(r) = \omega^{n_0} + \omega^{n_1} + ... + \omega^{n_{l-1}}$. Hence at any time $\alpha = \omega^k m_k + \omega^{k-1} m_{k-1} + ... + m_0 < \omega^{k+1}$ we need to remember only the states at the ordinals $\omega^k m_k + \omega^{k-1} m_{k-1} + ... + \omega^i m_i$ for $i \leq k$. If the tree automaton has m states, we can simulate it with an ω^{k+1}-automaton with m^{k+1} states. We may finally replace the nondeterministic ω^{k+1}-automaton by a deterministic ω^{k+1}-automaton (see [20, Theorem 7]).

Let us prove the implication (3)→(1). We consider the bijection $G : \omega^{k+1} \to T_{C_k}$ defined by $G(\omega^k m_k + \omega^{k-1} m_{k-1} + ... + m_0) = 1^{m_k} 0 1^{m_{k-1}} 0 ... 1^{m_0}$. Let $F =$

G^{-1}. We simulate the deterministic ω^{k+1}-automaton by a nondeterministic tree automaton. For each $\sigma \in T_{C_k}$, the tree automaton guesses the state s at time $F(\sigma 1)$. The following run of the tree automaton above $\sigma 1$ is based on s. The run of the tree automaton above $\sigma 0$ simulates the ω^{k+1}-automaton between $F(\sigma 0)$ and $F(\sigma 1)$ and checks whether the state at time $F(\sigma 1)$ is s. ∎

Note that a similar connection has been studied by Finkel and Todorcevic in [7, Proposition 3.4]. Through this correspondence we obtain by [25, Proposition 16]

Corollary 23. *The rank of every scattered tree-automatic linear order is below ω^ω. In particular, every tree-automatic ordinal is below ω^{ω^ω} (Delhommé [6]).*

We now give examples of natural regular tree languages and determine *minord* and *maxord*.

Theorem 24. *For $k \geq 1$ we define the following tree-automatic languages:*

- *A_k consists of all unlabeled trees where each branching node is of the form $0^m 1^n$ for some $m \in \{0, 1, \dots, k-1\}$ and $n \in \{0, 1, 2, \dots\}$.*
- *C_k consists of all trees where each branching node contains at most k 0s.*

Then the following statements hold:

(a) $minord(A_k) = \omega^k$, $maxord(A_k) = \omega^{k+1}$,
(b) $minord(C_k) = \omega^{k+1}$, and $maxord(C_k) = \omega^{\omega^{k+1}}$.

Proof. (a) We first show that $minord(A_k) \leq \omega^k$. We can identify each tree $t \in A_k$ with the tuple $(a_0, a_1, \dots, a_{k-1})$, where a_i is largest such that $0^i 1^{a_i}$ is a branching node in t. This gives us a one to one correspondence between N^k and A_k. The lexicographical order on N^k gives us the ordinal ω^k. This order also induces a tree-automatic order on A_k. Hence $minord(A_k) \leq \omega^k$.

Consider any automatic ordering $\leq$ for A_k. We use our identification of trees $t \in A_k$ with tuples $(a_0, a_1, \dots, a_{k-1})$ in N^k. Using an argument similar to the proof of pumping lemma we have, for a large enough constant c, the following statements:

(α) For $a_0, a_1, \dots, a_{k-1} \geq 1$, if $(ca_0, ca_1, \dots, ca_i, \dots, ca_{k-1}) \leq (ca'_0, ca'_1, \dots, ca'_i, \dots, ca'_{k-1})$, then $(ca_0, ca_1, \dots, c(a_i+1), \dots, ca_{k-1}) \leq (ca'_0, ca'_1, \dots, c(a'_i+1), \dots, ca'_{k-1})$.

(β) For $a_0, a_1, \dots, a_{k-1} \geq 1$, $(ca_0, ca_1, \dots, ca_i, \dots, ca_{k-1}) \leq (ca_0, ca_1, \dots, c(a_i + 1), \dots, ca_{k-1})$. Otherwise, the ordering $\leq$ is not a well order, by (α).

(γ) Without loss of generality we may assume the following statement (γ). For i with $0 \leq i < k-1$, $(ca_0, ca_1, \dots, ca_{k-1}) \leq (ca'_0, ca'_1, \dots, ca'_{k-1})$, when $[a'_i = 2$, $a_{i+1} = 2$, $(a_j = 1$, for j with $0 \leq j \leq i$ or $i+1 < j \leq k-1)$, $(a'_j = 1$, for j with $0 \leq j < i$ or $i < j \leq k-1)]$; otherwise, one can just reorder the coordinates to get the above property.

(δ) Statements (α), (β) and (γ) together with the reverse induction on i give the next statement (δ). $(ca_0, ca_1, \dots, ca_i, \dots, ca_{k-1}) \leq (ca'_0, ca'_1, \dots, ca'_i, \dots, ca'_{k-1})$, whenever for some i with $0 \leq i \leq k-1$, $[(a_j = a'_j$ for $j < i)$, $a_i < a'_i$ and $(a_j, a'_j \geq 1$, for j with $0 \leq j \leq k-1)]$.

Thus, the set $\{(ca_0, ca_1, \ldots, ca_{k-1}) : a_i > 0$, for $0 \le i \le k-1\}$, under the ordering $\le$, has order type ω^k. We conclude that $minord(A_k) = \omega^k$.

We now prove that $maxord(A_k) = \omega^{k+1}$ by induction on k. This holds for $k = 1$ by [24]. So assume that it holds for k, and consider an ordering $\le$ on A_{k+1}. Then there exists a constant c such that for $b_0, b_1, \ldots, b_k \le c$, $A_{b_0,b_1,\ldots,b_k} = \{(ca_0 + b_0, ca_1 + b_1, \ldots, ca_k + b_k) : a_i > 0$, for i with $0 \le i \le k\}$ has order type ω^{k+1}.

For all $b < c$, let $C_b^i = \{(a_0, a_1, \ldots, a_k) : a_i = b\}$. Then, by induction order type of C_b^i is at most $\omega^k \times m_b^i$, for some constant m_b^i. This implies that the order type of A_{k+1} is at most ω^{k+2}. This proves part (a).

(b) It follows from [25] and the correspondence in Theorem 22 that $maxord(C_k) = \omega^{\omega^{k+1}}$ and $minord(C_k) \le \omega^{k+1}$. To show that $minord(C_k) \ge \omega^{k+1}$, it is sufficient to consider a regular tree language B_k such that a node $\tilde{x}$ is a branching node in some tree in B_k iff $\tilde{x}$ has at most k 0s and arbitrary many 1s. Note that this property does no uniquely determine B_k.

Consider the subset of B_k consisting of the trees in which there is one main branch, and all offshoots have depth at most c', for some constant c'. This subset also satisfies the hypothesis of the theorem and is automatic. By coding the offshoots (as they are of bounded depth) into the labels, one can furthermore assume that all the branching nodes of the trees are on the main branch.

Thus, it is sufficient to prove the theorem for the languages B_k where, each tree in B_k is a labeled tree and all the branching nodes are on the same branch. Now using the pumping lemma, one has that, for some constant c, when one considers the subset B'_k of B_k, in which a branching node $\alpha 1^{2c+d}$ (if it exists) has the same label as the branching node $\alpha 1^{c+d}$, for all d, then this subclass also satisfies the hypothesis of the theorem and thus it is sufficient to prove the theorem for such classes B_k.

As there are only finitely many possibilities for labels on nodes $\alpha 1^{c+d}$, for $d \le c$, we can essentially ignore the labels for the purposes of proving the theorem. Now, any tree in B'_k can be represented using $(a_0, a_1, \ldots, a_r)$, where $r \le k$ and $1^{a_0}01^{a_1}0\ldots01^{a_r}$ is the maximal branching node in the tree. Then, essentially using the proof of Part (a), we can achieve the desired result. This proves part (b). ∎

Theorem 25. *Let A be an infinite regular tree language. Then $minord(A)$ and $maxord(A)$ are of the form ω^β, and $maxord(A)$ is not attained.*

Proof. Let $m = \omega^{\alpha_0}n_0 + ... + \omega^{\alpha_k}n_k$ with $\alpha_0 > ... > \alpha_k$ and $n_i > 0$ for all $i \le k$. Suppose $m = minord(A)$. Since $\omega^{\alpha_0}n_0 = (\omega^{\alpha_1}n_1 + ... + \omega^{\alpha_k}n_k) + \omega^{\alpha_0}n_0$, we obtain a tree-automatic well-order L on A of type $\omega^{\alpha_0}n_0$ by swapping the two parts. Let $otp_L(a)$ denote the order type of a in L. Then $A_i := \{a \in A : \omega^{\alpha_0}i \le otp(a) \le \omega^{\alpha_0}(i+1)\}$ is tree-automatic for each $i < n_0$. The sets A_i can be intertwined into a tree-automatic well-order on A of type ω^{α_0}.

Suppose $m = maxord(A)$. Suppose there is a tree-automatic well-order on A of type m. If $a \in A$ is its least element, we can define a well-order of type $m+1$ on A by $x \le y$ if $(x \le y \wedge x \ne a \wedge y \ne a) \vee (y = a)$. This contradicts the definition of m and hence m is not attained.

Suppose $k \geq 1$. Then there is a tree-automatic well-order L on A of type above $\omega^{\alpha_0} n_0$ and some $a \in A$ with $otp_L(a) = \omega^{\alpha_0} n_0$. Since $(\omega^{\alpha_1} n_1 + ... + \omega^{\alpha_k} n_k) + \omega^{\alpha_0} n_0 = \omega^{\alpha_0} n_0$, we obtain a tree-automatic well-order of type m on A by swapping the two parts.

Suppose $k = 0$ and $\alpha_0 \geq \omega$. Let us consider a tree-automatic well-order L on A of type at least $\omega^{\alpha_0}(n_0 - 1)$. We let A_0 be the set of successors in A with $otp_L(a) < \omega^{\alpha_0}$ and A_1 the set of limits in A with $otp_L(a) < \omega^{\alpha_0}$. Then A_0 and A_1 are tree-automatic and both have type ω^{α_0}. We obtain a tree-automatic well-order of type m on A by attaching A_0 on top of $A \setminus A_0$.

Suppose $k = 0$ and $1 \leq \alpha_0 < \omega$. Let us assume that $n_0 \geq 2$. We consider a tree-automatic well-order L on A of type $\omega^{\alpha_0}(n_0 - 1)$. We define B as the set of all $a \in A$ with $otp_L(a) = \omega^{\alpha_0 - 1} 2i + \delta$ for some $i \in \mathbb{N}$ and some $\delta < \omega^{\alpha_0 - 1}$. This set is first-order definable in $(A, \leq)$ and hence tree-automatic. Then B has type ω^{α_0} and $A \setminus B$ has type $\omega^{\alpha_0}(n_0 - 1)$. We obtain a tree-automatic well-order of type $m = \omega^{\alpha_0} n_0$ on A by attaching B on top of $A \setminus B$. ∎

References

1. Achim Blumensath. *Automatic structures.* Diploma thesis, RWTH Aachen, 1999.
2. Achim Blumensath and Erich Grädel. Automatic structures. In *15th Annual IEEE Symposium on Logic in Computer Science (Santa Barbara, CA, 2000)*, pages 51–62. IEEE Computer Society Press, Los Alamitos, CA, 2000.
3. Gábor Braun and Lutz Strüngmann. Breaking up finite automata presentable torsion-free abelian groups. International Journal of Algebra and Computation 21:8, 1463–1472, 2011.
4. Arnaud Carayol and Christof Löding. MSO on the Infinite Binary Tree: Choice and Order. CSL 2007: 161-176
5. Thomas Colcombet and Christof Löding. Transforming structures by set interpretations. *Logical Methods in Computer Science*, 3:2:4:1–36, 2007.
6. Christian Delhommé. Automaticité des ordinaux et des graphes homogènes. C. R. Math. Acad. Sci. Paris 339(1), 510, 2004.
7. Olivier Finkel and Stevo Todorcevic. A hierarchy of tree-automatic structures. J. Symbolic Logic 77(1), 350-368, 2012.
8. Yuri Gurevich and Sharon Shelah. Rabins uniformization problem. J. Symb. Log. 48(4), 11051119, 1983.
9. Wilfrid Hodges. *Model Theory.* Cambridge University Press, 1993.
10. Bernard R. Hodgson. On direct products of automaton decidable theories. *Theoretical Computer Science*, 19:331–335, 1982.
11. Bernard R. Hodgson. Décidabilité par automate fini. *Annales des sciences mathématiques du Québec*, 7(1):39–57, 1983.
12. Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. *Logic and computational complexity (Indianapolis, IN, 1994)*, volume 960 of *Lecture Notes in Computer Science*, pages 367–392. Springer, Berlin, 1995.
13. Bakhadyr Khoussainov and Mia Minnes. Model-theoretic complexity of automatic structures. *Annals of Pure and Applied Logic*, 161(3):416–426, 2009.
14. Bakhadyr Khoussainov and Mia Minnes. Three lectures on automatic structures. *Proceedings of Logic Colloquium 2007. Lecture Notes in Logic*, 35:132–176, 2010.

15. Bakhadyr Khoussainov, André Nies, Sasha Rubin and Frank Stephan. Automatic structures: richness and limitations. *Logical Methods in Computer Science*, volume 3, number 2, 2007. *19th IEEE Symposium on Logic in Computer Science, LICS 2004*, 14-17 July 2004, Turku, Finland, Proceedings; IEEE Computer Society, pages 44–53, 2004.
16. Bakhadyr Khoussainov, Sasha Rubin, and Frank Stephan: Automatic linear orders and trees. ACM Trans. Comput. Log. 6(4): 675-700 (2005)
17. D. Kuske, J. Liu, M. Lohrey. The isomorphism problem for ω-automatic trees, in Proc. of CSL 2010: 396-410. LNCS 6247, Springer, 2010.
18. D. Kuske, J. Liu, M. Lohrey. The isomorphism problem for classes of automatic structures, in Proc. of LICS 2010: 160-169. IEEE Computer Society, 2010.
19. D. Kuske, J. Liu, M. Lohrey. The isomorphism problem on classes of automatic structures with transitive relations. To appear in Transactions of American Math Society.
20. Itay Neeman. Monadic theories of wellorders. In Logic, Methodology and Philosophy of Science Proceedings of the Thirteenth International Congress (Glymour, Wei, Westerstahl, eds.), pp. 108121, College Publications, 2009.
21. André Nies and Pavel Semukhin. Finite automata presentable Abelian groups. *Annals of Pure and Applied Logic*, 161:458–467, 2009.
22. Joseph G. Rosenstein. *Linear Orderings.* Academic Press, New York, 1982.
23. Sasha Rubin. Automata presenting structures: a survey of the finite string case. *The Bulletin of Symbolic Logic*, 14(2):169–209, 2008.
24. Sasha Rubin. Automatic Structures. *PhD thesis*, 2004, The University of Auckland.
25. Philipp Schlicht and Frank Stephan. Automata on ordinals and automaticity of linear orders. To appear in Annals of Pure and Applied Logic.
26. Todor Tsankov. The additive group of the rationals does not have an automatic presentation. The Journal of Symbolic Logic, 76:1341-1351, 2011.