# Tree-automatic scattered linear orders

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

[1] Department of Computer Science,
National University of Singapore, Singapore 117417, 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.** Tree-automatic linear orders on regular tree languages are studied. It is shown that there is no tree-automatic scattered linear order, and therefore no tree-automatic well-order, on the set of all finite labeled trees, and that a regular tree language admits a tree-automatic scattered linear order if and only if for some $n$, no binary tree of height $n$ can be embedded into the union of the domains of its trees. Hence the problem whether a given regular tree language can be ordered by a scattered linear order or a well-order is decidable. Moreover, sharp bounds for tree-automatic well-orders on some regular tree languages are computed by connecting tree automata with automata on ordinals. The proofs use elementary techniques of automata theory.

**Keywords.** Tree automata, linear orders, automatic structures.

## 1 Introduction

The aim of this paper is to study tree-automatic linear orders on regular tree languages, and more precisely, we ask whether a given regular tree language can be ordered by a tree-automatic scattered or well-founded linear order. This is a part of a larger theme 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 and Abelian groups [4, 9, 19, 21, 23, 30, 36]. Recent results by Kuske, Lohrey and Liu indicate that there is no complete characterisation of the linear orders presentable by tree automata [25–27]. Therefore we restrict the classification question by considering tree-automatic structures whose domain is a fixed

⋆ Supported in part by NUS grants R252-000-420-112, R146-000-181-112 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, R146-000-181-112 and R252-000-420-112.

regular tree language. Our goal is to derive algebraic properties of tree-automatic structures with a given domain and algorithmic consequences. Delhommé [9] proved one of the first important characterisation 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 approach can be understood as a refinement of the work of Delhommé leading to an alternative proof of his result in Theorem 26.

In Theorem 12 we show that there is no tree-automatic scattered linear order, and therefore no well-order, on the set $T(\Sigma)$ of all finite binary trees labeled by symbols from a finite alphabet $\Sigma$. 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 $T_2$ [11]. We mention that Carayol and Löding [7, Theorem 1] provide a simple proof of the mentioned result of Gurevich and Shelah. In addition, they prove undecidability of the MSO theory of the full binary tree with any well-order. The last fact also implies the non-existence of a tree automatic well-order on the full binary tree.

A tree language has tree-rank $k$ if $k$ is maximal such that the full binary finite tree of height $k$ can be embedded into the union of all domains of $\Sigma$-trees in the tree language. For instance, the language $T(\Sigma)$ does not have a finite tree-rank. In Theorem 19 we show that a regular tree language allows a tree-automatic well-order if and only if the tree language has finite tree-rank. From the proof we obtain an algorithm which, given a regular tree language, decides if the language can be well-ordered by a tree automaton.

We further connect certain tree-automatic structures with finite automata on ordinals, which implies Delhommé's theorem that $\omega^{\omega^\omega}$ is the smallest ordinal with no tree-automatic presentation. Finally, we give examples of regular tree languages and describe the spectra and the lower and upper bounds of tree-automatic well-orders on them.

## 2 Preliminaries

Let us first collect several definitions and background facts. By a structure $\mathcal{A}$ we mean a tuple of the form $(A; R_1, \ldots, R_n)$, where $A$ is the *domain* or the *universe* of the structure and $R_1, \ldots, R_n$ are the *atomic relations* on $A$. We will mostly consider linearly ordered sets. A linear order is a *well-order* if every nonempty subset of its domain has a least element. The order types of well-orders are the *ordinals*. A linearly ordered set is *scattered* if there is no suborder isomorphic to the ordering $(\mathbb{Q}, \leq)$ of the rationals. Examples of scattered orders are the integers, (reverse) well-orders and lexicographic sums of scattered linear orders along (reverse) well-orders. Let us define the *Cantor-Bendixson rank (CB-rank)* of a linearly ordered set $\mathcal{L} = (L, \leq)$. For $x, y \in L$, let $x \sim_0 y$ be the identity relation. Let $\sim_{\alpha+1}$ denote the *derivative* of $\sim_\alpha$, that is, $x \sim_{\alpha+1} y$ if there are only finitely many equivalence classes of $\sim_\alpha$ between $x$ and $y$. For limit ordinals $\beta$, let $\sim_\beta = \bigcup_{\alpha<\beta} \sim_\alpha$. Then each relation $\sim_\alpha$ is an equivalence relation and the linear order $\leq$ induces a linear order on the quotient $\mathcal{L}/\sim_\alpha$, which we call the $\alpha$-th derivative of $\mathcal{L}$.

**Theorem 1 (Hausdorff, see [32]).** *A linear order $\mathcal{L}$ is scattered if and only if there is some $\alpha$ such that $\mathcal{L}/{\sim_\alpha}$ is finite.*

The least ordinal $\alpha$ for which $\mathcal{L}/{\sim_\alpha}$ is finite is called the *Cantor-Bendixson rank* (CB-rank) of $\mathcal{L}$ and is denoted by *CB-rank($\mathcal{L}$)*.

To define word-automatic and tree-automatic structures, recall the following definitions from automata theory. A finite alphabet is denoted by $\Sigma$ and $\Sigma^\star$ denotes the set of all finite strings (finite words) over $\Sigma$. Let $|\sigma|$ denote the length of a string $\sigma$. Let $\lambda$ denote the empty string. Let $\sigma \preceq \tau$ denote that string $\sigma$ is a prefix of string $\tau$.

A *finite automaton* over the alphabet $\Sigma$ is a tuple $\mathcal{M} = (S, \iota, \Delta, F)$, where $S$ is a finite set of *states*, $\iota \in S$ is the *initial state*, $\Delta \subseteq S \times \Sigma \times S$ is the *transition table*, and $F \subseteq S$ is the set of *final states*. A *run* of $\mathcal{M}$ on a word $w = a_1 a_2 \dots a_n$, (where $a_1, a_2 \dots, a_n$ are members of $\Sigma$) is a sequence of states $q_0, q_1, \dots, q_n$ such that $q_0 = \iota$ and $(q_i, a_{i+1}, q_{i+1}) \in \Delta$ for all $i \in \{0, 1, \dots, n-1\}$. 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*, *word-automatic*, or *finite automaton recognizable*.

We quickly review the definition of tree automata. A *tree* (also called *binary tree*) is a possibly infinite prefix-closed subset of $\{0,1\}^\star$. Members of a tree $T$ are called nodes of $T$. We say that $\sigma$ is a *leaf* of a tree $T$ if $\sigma$ belongs to $T$ but no proper extension of $\sigma$ belongs to $T$. Similarly, $\sigma$ is an *internal node* of $T$ if $\sigma$ as well as some proper extension of $\sigma$ belongs to $T$. If $\sigma$ and $\sigma a$ both belong to $T$ (where $a \in \{0,1\}$), then $\sigma a$ is called a *child* of $\sigma$ and $\sigma$ is called the *parent* of $\sigma a$. A node $\sigma$ is a *branching node* of $T$ if $\sigma$ as well as $\sigma 0$ and $\sigma 1$ belong to $T$. The distance between a node $u$ and node $v$ in a tree is the number of edges between them. That is, let $w$ be the longest common prefix of $u$ and $v$; then, the distance between $u$ and $v$ is $(|v| - |w|) + (|u| - |w|)$. We say that a finite tree is a full binary tree of height $n$ iff it consists of all the binary strings up to length $n$ with those of length $n$ being the leaves and the shorter ones being the branching nodes of the tree. A full binary tree (without specification of any height) contains all the binary strings.

A *labeled tree* is a tree $T$ together with a function from $T$ into a finite alphabet $\Sigma$. We say that a (labeled or unlabeled) tree $S$ *embeds* into a tree $T$ if there is an injective map $h\colon S \to T$ with $\sigma \preceq \tau \Leftrightarrow h(\sigma) \preceq h(\tau)$ for all $\sigma, \tau \in S$. We say that a tree $T$ has *tree-rank* $n$, written $tr(T) = n$, if $n$ is the maximal number such that a full binary tree of height $n$ can be embedded into $T$; in the case that such a maximal $n$ does not exist and all finite binary trees can be embedded into $T$, we say that $tr(T) = \infty$.

A *$\Sigma$–tree* is a mapping $t : dom(t) \to \Sigma$ such that the domain $dom(t)$ is a finite tree such that for every non-leaf node $v \in dom(t)$ we have $v0, v1 \in dom(t)$.[5] The *boundary* of $dom(t)$ is the set $\partial dom(t) = \{xb \mid x \text{ is a leaf of } dom(t) \text{ and } b \in \{0,1\}\}$. The set of all $\Sigma$-trees is denoted by $T(\Sigma)$. A *slim $\Sigma$-tree* is a $\Sigma$-tree $T$ such that the branching nodes in $T$ are all pairwise comparable with respect to the prefix relation. A slim $\Sigma$-tree $x$ is *generated* by a string $\tilde{x}$ iff $\tilde{x}$ is a branching node in $dom(x)$ and every branching node in $dom(x)$ is a prefix of $\tilde{x}$.

Note that $\Sigma$-trees are essentially labeled trees where each internal node is a branching node. Thus, often we refer to $\Sigma$-trees simply as trees with $\Sigma$ being implicit.

[5] There are various alternative definitions of $\Sigma$-trees which lead to the same class of tree automatic presentable structures. This specific definition has the advantage that the correspondence with ordinal automata in Section 7 is easier to state.

**Definition 2.** A *tree automaton* over the alphabet $\Sigma$ is a tuple $\mathcal{M} = (S, \iota, \Delta, F)$, where $S$ is a finite 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 $\mathcal{M} = (S, \iota, \Delta, F)$ on a tree $t$ is a map $r : dom(t) \cup \partial dom(t) \to S$ such that $r(\lambda) = \iota$ and $(r(x), t(x), r(x0), r(x1)) \in \Delta$ for all $x \in dom(t)$. If both $r(x0) \in F$ and $r(x1) \in F$ for every leaf $x \in dom(t)$, then the run $r$ is said to be *accepting*, and the tree automaton $\mathcal{M}$ *accepts* the tree $t$ if there is an accepting run of $\mathcal{M}$ on $t$. The *tree language* $L(\mathcal{M})$ of $\mathcal{M}$ is defined as the set of all finite trees $t \in T(\Sigma)$ accepted by $\mathcal{M}$. These tree languages are called *regular tree languages*, *tree-automatic*, or *tree automaton recognizable languages*.

In order to define tree-automatic structures, we convolute finitely many $\Sigma$-trees $t_0, \ldots, t_{n-1}$ into a single $\Sigma$-tree as follows. Let $\Box$ be a special symbol not in $\Sigma$. For $x \in dom(t_0) \cup \ldots \cup dom(t_{n-1})$ and $i < n$, let $t'_i(x) = t_i(x)$ 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 the tree $conv(t_0, \ldots, t_{n-1})$ with $conv(t_0, \ldots, t_{n-1})(x) = (t'_0(x), \ldots, t'_{n-1}(x))$ for all $x \in \bigcup_{i<n} dom(t_i)$. 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. Word-automatic relations are defined according to the analogous convolution of $\Sigma$-words.

**Definition 3.** A structure $\mathcal{A} = (A; R_1, \ldots, R_n)$ is called *tree-automatic* or *tree-automata recognizable (word-automatic* or *word-automata recognizable)* if the domain $A$ and the atomic relations $R_1, \ldots, R_n$ are all tree-automatic (word-automatic). The structure $\mathcal{A}$ is a *tree-automatic presentation* (word-automatic presentation) of a structure $\mathcal{B}$ if $\mathcal{A}$ is isomorphic to $\mathcal{B}$. If there exists a tree-automatic presentation (word-automatic presentation) of $\mathcal{B}$, then $\mathcal{B}$ is said to be *tree-automatic presentable* (word-automatic presentable).

We will sometimes write *automatic* instead of tree-automatic or word-automatic if the meaning is clear from the context. Automatic presentations $\mathcal{A}$ of a structure $\mathcal{B}$ can be identified with finite sequences of automata for the domain and the atomic relations of $\mathcal{A}$. The definition of automatic presentability is a $\Sigma^1_1$-definition in arithmetic since automatic presentability of $\mathcal{B}$ requires a search for an isomorphism from automatic structures $\mathcal{A}$ to $\mathcal{B}$. We will sometimes identify tree-automatic and word-automatic presentable structures with their presentations.

For example, Presburger arithmetic $(\mathbb{N}; +)$ and configuration spaces of Turing machines are word-automatic, while Skolem arithmetic $(\mathbb{N}; \times)$ is tree-automatic but not word-automatic [2].

The countable atomless Boolean algebra has the following tree-automatic presentation. Note that the closed-open subsets of the Cantor space $\{0,1\}^\omega$ form a countable atomless Boolean algebra. Every such set is a union of finitely many cones $v_i\{0,1\}^\omega$, $i \leq n$, where $\{v_0, ..., v_n\}$ is prefix-free. Each such set can be represented by a $\Sigma$-tree for $\Sigma = \{0\}$ whose branching nodes are exactly the prefixes of the words $v_i$. Then the Boolean operations are tree-automatic.

Let us describe basic properties of tree-automatic structures. Let $FO + \exists^\omega$ denote the extension of first-order logic with the $\exists^\omega$ quantifier (there are infinitely many). The decidability of the emptiness problem and closure properties of regular tree languages imply that the $FO + \exists^\omega$ theories of tree-automatic and word-automatic structures are uniformly decidable.

**Theorem 4 (Blumensath and Grädel [3], Hodgson [13, 14], Khoussainov and Nerode[18]).** *There is an algorithm whose input is the finite sequence of automata of a tree-automatic or word-automatic presentation of a structure $\mathcal{A}$ and a formula $\phi(x_1, \ldots, x_n)$ in $FO+\exists^\omega$, and whose output is an automaton which recognises all tuples $(a_1, \ldots, a_n)$ from the structure for which the formula holds.*

Note that the structure can be extended by any tree-automatic relation in the given presentation. Therefore, one can express that the domain of one tree is contained in the domain of another tree and use this to express $\exists^\omega$ as in Proposition 10.

In a way similar to the pumping lemma for regular languages over strings, the following pumping lemma for tree-automatic languages can be proven.

**Lemma 5.** *Fix any tree-automaton $\mathcal{M}$. There exists a constant $c$ such that for any $\Sigma$-tree $t$ accepted by $\mathcal{M}$ and any $\alpha'\beta' \in dom(t)$ such that $|\beta'| > c$, there exists $\alpha, \beta$ such that $0 < |\beta| \leq c$, $\alpha' \preceq \alpha \prec \alpha\beta \preceq \alpha'\beta'$ and the following holds: for all $i \geq 0$, there exsits a $\Sigma$-tree $t'$ acccepted by $\mathcal{M}$ such that:*

(i) $dom(t') = [dom(t) - \{\alpha\gamma \mid \gamma \in \{0,1\}^+\}] \cup \{\alpha\beta^i\gamma \mid \gamma \in \{0,1\}^+$ *and* $\alpha\beta\gamma \in dom(t)\} \cup \{\alpha\beta^r\gamma \mid r < i, \alpha\beta$ *is not a proper prefix of* $\alpha\gamma$ *and* $\alpha\gamma \in dom(t)\}$,
(ii) *for all* $\gamma \in \{0,1\}^\star$ *such that* $\gamma \in dom(t)$ *and* $\alpha$ *is not a proper prefix of* $\gamma$: $t'(\gamma) = t(\gamma)$,
(iii) *for all* $r < i$, *for all* $\gamma \in \{0,1\}^\star$ *such that* $\alpha\gamma \in dom(t)$ *and* $\alpha\beta$ *is not a proper prefix of* $\alpha\gamma$: $t'(\alpha\beta^r\gamma) = t(\alpha\gamma)$, *and*
(iv) *for all* $\gamma \in \{0,1\}^+$ *such that* $\alpha\beta\gamma \in dom(t)$: $t'(\alpha\beta^i\gamma) = t(\alpha\beta\gamma)$.

Intuitively, the constant $c$ in the above lemma can be taken as any number strictly above $s|\Sigma|$, where $s$ is the number of states of the tree-automaton. Intuitively, the portion of $t$ in $\beta$ part can be replicated as many times as needed. We obtain the following corollary.

**Corollary 6.** *Suppose $\mathcal{M}$ is a tree-automaton that accepts a subset of $\{conv(t, t') \mid t, t'$ are $\Sigma$-trees$\}$. Then there exists a constant $c$ such that,*

(a) *for each $\Sigma$-tree $t$ for which $\mathcal{M}$ accepts at least one $conv(t, t')$, $t'$ being a $\Sigma$-tree, there exists a $\Sigma$-tree $t''$ such that $\mathcal{M}$ accepts $conv(t, t'')$ and all nodes of $dom(t'')$ are within distance $c$ from some node in $dom(t)$.*
(b) *for each $\Sigma$-tree $t$ for which $\mathcal{M}$ accepts at least one $conv(t, t')$, $t'$ being a $\Sigma$-tree, and for each leaf $v$ of $dom(t)$, there exists a $\Sigma$-tree $t''$ such that $\mathcal{M}$ accepts $conv(t, t'')$ and*
   (i) *if $v$ is not a proper prefix of $u$ then $u$ is in $dom(t'')$ iff $v$ is in $dom(t')$.*
   (ii) *if $v$ is a proper prefix of $u$ and $u$ is in $dom(t'')$, then $u$ is at a distance at most $c$ from $v$.*

Blumensath and Grädel [3] address automaticity of structures in terms of interpretability. They proved that there are specific automatic structures that encompass all automatic structures in 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 $x|_2y$ 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, for example, Colcombet and Löding [8] study characterisations through finite set interpretations; further characterisations are given by Rubin [33].

The next definition refines the notion of tree-automaticity by placing a focus on the domain. Let $K$ denote a class of structures closed under isomorphism, for example the linearly ordered sets, well-ordered sets, trees or Abelian groups.

**Definition 7.** Let $X$ be a given regular tree language. The *spectrum* $Spec_K(X)$ of $X$ with respect to the class $K$ consists of all $\mathcal{B} \in K$ isomorphic to a tree-automatic structure $\mathcal{A}$ with domain $X$. If some $\mathcal{B} \in K$ is isomorphic to a tree-automatic structure with domain $X$ then we say that $X$ *admits* $\mathcal{B}$. The definition of the spectrum $Spec_K(X)$ for regular languages $X$ is analogous.

For example, no tree-automatic or word-automatic (tree) language admits a structure with undecidable first-order theory. Let us collect some more examples which restrict the spectra. Rubin [34] showed that $0^\star$ admits a well-order $\alpha$ if and only if $\alpha < \omega^2$. Delhommé [9] proved that if $X$ is regular and $X$ admits a well-founded partial order, then its height is strictly below $\omega^\omega$ (see also [19]) and that no regular tree language admits the ordinal $\omega^{\omega^\omega}$. Tsankov [36] proved that no regular language admits the additive group of rational numbers $(\mathbb{Q}; +)$ from deep results in group theory.

Definition 7 calls for a refined analysis of automaticity, and hence interpretability, of structures; proving that a certain structure, for example the ordinal $\omega^n$, is not admitted by a given regular or regular tree language requires an analysis of the automata and the algebraic and model theoretic properties of the underlying structures. In this paper, the class $K$ will be the class of linearly ordered sets.

## 3 Basic results

The first results give general properties of $T(\Sigma)$.

**Proposition 8.** *Assume $|\Sigma| \geq 2$. Then, the language $T(\Sigma)$ admits the order of the rational numbers.*

**Proof.** We give the proof for $\Sigma = \{a, b\}$ only and declare $a < b$. For any two given trees $p, q \in T(\Sigma)$ such that $p \neq q$ consider the lexicographically least 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 as in the definition of the convolution operation for trees. 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 $v$ lexicographically less than $x_{(p,q)}$ the automaton verifies that $p(v) = q(v)$. Once the node $x_{(p,q)}$ is reached the automaton accepts the tree depending on whether or not its label belongs to $\{(a,b), (\Box,b), (a,\Box)\}$. If the automaton chose along some path a wrong location for $x_{(p,q)}$, then the automaton fails along this non-deterministically chosen path that searches for $x_{(p,q)}$ and the corresponding run will not be accepting.

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 $p \in T(\Sigma)$ and let $v$ be any leaf of $p$. We 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 $p_1 \sqsubset p \sqsubset p_2$. Hence, $\sqsubseteq$ is a linear order without end-points.

Let $p, q$ be such that $p \neq q$ and $p \sqsubseteq q$. First consider the case that $p'(x_{(p,q)}) = a$ and either $q'(x_{(p,q)}) = b$ or $q'(x_{(p,q)}) = \Box$. Let $v$ be the lexicographically largest leaf of $p$ above $x_{(p,q)}$ (if $x_{(p,q)}$ is a leaf of $p$, then we take $v$ to be $x_{p,q}$). Note that in our terminology trees grow upwards, that is, a node $v$ is above a node $u$ if $u$ is closer to the root and $u, v$ are on a common branch. Extend $p$ to $p_2$ using $v$ as above (by adding nodes $p_2(v0) = b$ and $p_2(v1) = b$). Then $p \sqsubset p_2 \sqsubset q$. Next consider the case that $p'(x_{(p,q)}) = \Box$ and $q'(x_{(p,q)}) = b$. Let $w$ be the lexicographically largest leaf of $q$ above $x_{(p,q)}$ (if $x_{(p,q)}$ is a leaf of $q$, then we take $w$ to be $x_{p,q}$). Extend $q$ to $q_1$ using the node $w$ (by adding nodes $q_1(w0) = a$ and $q_1(w1) = b$ to $q$). Then $p \sqsubset q_1 \sqsubset q$. Hence the linear order $\sqsubseteq$ is dense. $\Box$

The next two results show that tree-automatic structures are closed under $\exists^\omega$ and under quotients by tree-automatic congruence relations. Although these facts are not new, we will give proofs since the proof ideas are used later. The first fact was claimed by Blumensath [2], proved by Colcombet and Löding [8] and complexity-theoretically analysed by Kuske and Weidner [28].

**Proposition 9.** *Suppose that $\equiv$ is a tree-automatic equivalence relation on a tree-automatic set $A$. There is a tree-automatic function $f$ picking representatives from equivalence classes of $\equiv$; that is, $f : A \to A$ and, for all $p, q \in A$ with $p \equiv q$, $f(p) \equiv p$ and $f(p) = f(q)$.*

**Proof.** For each $p \in A$, set $t(p)$ to be the $\Sigma$–tree such that the domain of $t(p)$ is the set $\bigcap_{q \equiv p} dom(q)$ 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 $q \equiv p$ and every node in $q$ is at a distance at most $c$ from a node in $t(p)$. To prove this, we list all leaves $u_0, u_1, \ldots, u_{n-1}$ in $dom(t(p))$, and consider the automaton which recognises the relation $\equiv$. Let $c$ bound the number of states of this automaton. We start with $q_0 = p$, and for each $m < n$ proceed, inductively, as follows. There is a $\Sigma$–tree $t \equiv q_m$ which does not contain $u_m0$ and $u_m1$. By Corollary 6(b), there is a $q_{m+1} \equiv t$ such that $q_{m+1}$ coincides with $q_m$ on all nodes except those above $u_m$ and the height of the subtree of $q_{m+1}$ above $u_m$ is bounded by the constant $c$. By transitivity we have $q_m \equiv q_{m+1}$. Doing this with all nodes $u_0, \ldots, u_{n-1}$, we produce a tree $q_n \equiv p$. The tree $q_n$ is the desired tree $q$.

Consider the set $S(p) = \{q \in A \mid p \equiv q$ and each node of $q$ is at most $c$ distance away from a node of $t(p)\}$. The relation $\{(p,q) \mid q \in S(p)\}$ is tree-automatic: Given the convolution of two trees, one can check with a tree automaton whether a leaf of one tree is more than $c$ levels above a leaf of another tree, due to nondeterministically choosing the path in this leaf and counting up to $c+1$. Furthermore, the relation $\equiv$ is tree-automatic, and intersections and complements of tree-automatic relations are tree-automatic.

We now define $f(p)$ as follows. Restrict the order $\sqsubseteq$ from Proposition 8 to $S(p)$ and take $f(p)$ to be the least element of $S(p)$ with respect to the order $\sqsubseteq$. $\square$

**Proposition 10.** *From a tree-automaton accepting a $(n+2)$-ary relation $R$, one can compute a tree-automaton accepting $S = \{(p, a_1, a_2, \dots, a_n) \in T(\Sigma)^{n+1} \mid \neg\exists^\omega q\ R(p, q, a_1, a_2, \dots, a_n)\}$.*

**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 set

$$S' = \{(p, a_1, a_2, \dots, a_n) \in T(\Sigma)^{n+1} \mid \exists q \in T(\Sigma)\ \forall t (R(p, t, a_1, a_2, \dots, a_n) \Rightarrow t \subseteq_d q)\}.$$

Since $S'$ is first-order definable from the tree-automatic relations $\subseteq_d$ and $R$, it is effectively regular (effective in the tree-automaton accepting $R$). It is not hard to verify that $S = S'$. $\square$

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

**Corollary 11.** *Consider the relations $R$ and $S$ as in Proposition 10 (with $n = 0$). For every $p \in S$ define the following $\Sigma$–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 $\phi : S \to T(\Sigma)$ is tree-automatic. Hence, there exists a constant $c$ such that every node $v$ of $\phi(p)$ is at most $c$ distance away from some node of $p$.*

**Proof.** Note that, for each $p \in S$, the number of $q$ which satisfy $R(p, q)$ is finite and thus the tree is finite. The corollary now follows from Proposition 10. $\square$

## 4 On the nonexistence of scattered orders

In this section we show that the set of all $\Sigma$-trees does not admit a scattered tree-automatic linear order. As a corollary there is no tree-automatic well-order on the set $T(\Sigma)$. The anonymous referees pointed out that the results in this section can also be obtained by using a connection of tree-automatic relations to MSO-definable relations; these connections and the resulting proofs are explained in Section 6.

**Theorem 12.** *$T(\Sigma)$ 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 $T(\Sigma)$. For any nonempty alphabet $\Xi \subseteq \Sigma$, the restriction of $\leq$ to $T(\Xi)$ is again a tree-automatic scattered linear order. Hence we can assume without loss of generality that $\Sigma$ is a singleton.

Let $B$ be the set of slim $\Sigma$-trees. We use $x, y, z$ and $U, S, T$ to refer to trees in $B$ and $T(\Sigma)$, respectively. For $x \in B$, $\tilde{x}$ denotes the node that generates $x$. The set $Strings = \{\tilde{x} \mid x \in B\}$

admits the order $\sqsubseteq$, where $\tilde{x} \sqsubseteq \tilde{y}$ 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 [23, 34].

Consider the sequence of derivatives $\sim_0, \sim_1, \ldots, \sim_r$, for $(L, \leq) = (B, \leq)$ defined just before Theorem 1, where $r$ is the CB-rank of $(B, \leq)$. Below we will define a set $B_i$ using representatives of equivalence classes of $B$ with respect to $\sim_i$ (for ease of presentation, we define and consider $\sim_{i+1}$ restricted to $B_i$ only; it can be extended to the whole of $B$ by additionally including $\sim_i$ in $\sim_{i+1}$). Let $n = |B_r|$ and $k$ be the number of infinite $\sim_r$ equivalence classes of $B_{r-1}$.

Among all the possible scattered linear orders on $T(\Sigma)$, choose the order $\leq$ on $T(\Sigma)$ such that the corresponding triple $(r, n, k)$, called the *extended rank of* $(B, \leq)$, is the smallest possible with respect to the lexicographical ordering of triples. Note that $r \geq 1$.

The idea now is to show the existence of a $\tilde{z}$ such that the set $B_{new}$ of all trees in $B$ having $\tilde{z}$ as a branching node has a lower extended rank (Claim 15). As one can convert the scattered ordering $\leq$ on $B_{new}$ to a scattered ordering $\leq'$ on $B$, this would give a contradiction (Claim 16). The following notions and definitions and corresponding Claim 13 and Claim 14 about them allow us to show the above mentioned Claim 15 and Claim 16.

We now give an inductive analysis of the sequence $B_0, B_1, \ldots, B_r$, formally define the relations $\leq_{m+1}$, $\sim_{m+1}$, sets $C_{m+1}$, 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 $x, y \in B_m$, let $C_{m+1}(x, y)$ denote the interval $[x, y]$ if $x \leq_m y$ and $[y, x]$ if $y \leq_m x$ (that is, $C_{m+1}(x, y)$ consists of the trees $z \in B_m$, such that $x \leq_m z \leq_m y$ in case $x \leq_m y$ ($y \leq_m z \leq_m x$, in case $y \leq_m x$)). We 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 $\Sigma$-tree $U$ 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 9 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 every node of $y$ is at most $c_{m+1}$ distance away from some node of $t_{m+1}(x)$. Then let $rep_{m+1}(x)$ be the $y$ which has $\tilde{y}$ length-lexicographically least, among all $y$ such that $C_{m+1}(x, y)$ is finite and every node of $y$ is at most $c_{m+1}$ distance away from some node of $t_{m+1}(x)$. Now set $B_{m+1} = \{rep_{m+1}(x) \mid x \in B_m\}$ and $\leq_{m+1}$ be $\leq$ restricted to $B_{m+1}$. Thus, we have the following claim.

**Claim 13.** *There is a descending sequence $B_0, B_1, \ldots, B_r$ of subsets of $B$ such that $B_0 = B$ and, for $m = 0, 1, \ldots, r$, the following conditions hold:*

(a) *$B_m$ is tree-automatic;*
(b) *The tree-automatic linearly ordered set $(B_m, \leq_m)$ is isomorphic to the $m$-th derivative of $(B, \leq)$;*
(c) *If $m < r$, for each $x \in B_m$ there is exactly one $y \in B_{m+1}$, denoted as $y = rep_{m+1}(x)$, such that $C_{m+1}(x, y)$ is finite;*

(d) *If $m < r$, the function $x \mapsto rep_{m+1}(x)$ is tree-automatic, has domain $B_m$ and there exists a constant $c_{m+1}$ such that, for all $x \in B_m$, every node of $rep_{m+1}(x)$ is at most $c_{m+1}$ distance away from some node of $t_{m+1}(x)$.*

The next claim defines the constant $d_{m+1}$ and the function $f_{m+1}$. The proof follows from the first-order definability, Corollary 6, and Corollary 11.

**Claim 14.** *For each $x \in B_m$ let $f_{m+1}(x)$ be the $\Sigma$-tree $U$ 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 $x \mapsto f_{m+1}(x)$, 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 a 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 15.** *There exists a $\Sigma$-tree $z \in B$ such that $(\{u \in B \mid \tilde{z}$ is a branching node in $u\}, \leq)$ has extended rank $(r', n', k')$ with $(r', n', k') <_{lex} (r, n, k)$.*

For a proof of the claim, start with a $\Sigma$-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 $v = \min\{u \in B_{r-1} \mid u \sim_r x\}$ or $|\tilde{v}| > |\tilde{x}| + d_1 + \ldots + d_r + 2$;
- either $w = \max\{u \in B_{r-1} \mid u \sim_r x\}$ or $|\tilde{w}| > |\tilde{x}| + d_1 + \ldots + d_r + 2$.

Note that such a $v$ exists as either the number of members of $B_{r-1}$ which are $\sim_r x$ but below $x$ in $B_{r-1}$ is finite or all but finitely many of them satisfy $|\tilde{v}| > |\tilde{x}| + d_1 + \ldots + d_r + 2$. Similar argument holds for existance of $w$.

We fix $v$ and $w$ chosen above. Take any sequence $x_r$, $x_{r-1}$, …, $x_0$ 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 reverse induction on $m = r-1, \ldots, 0$, we claim that $\tilde{v}_m$ is a prefix of $\tilde{x}_m$ for $m = r-1, r-2, \ldots, 1, 0$. To see this, first note that as $x_{r-1} < v \leq x_r$, by definition of $d_r$ and $f_r$, every node of $v$ is at most a distance $d_r$ from a branching node of $x_{r-1}$. Thus, $\tilde{v}_{r-1}$ obtained by omitting the top $d_r$ edges from $\tilde{v}_r$ is a prefix of $\tilde{x}_{r-1}$. By induction, suppose that $\tilde{v}_{m+1}$ is a prefix of $\tilde{x}_{m+1}$. Then, as every node in $x_{m+1}$ is at most a distance $d_{m+1}$ from a branching node in $x_m$ (by definition of $d_{m+1}$ and $f_{m+1}$), we have that $\tilde{v'}_m$ obtained by removing top $d_{m+1}$ edges from $\tilde{x}_{m+1}$ is a prefix of $\tilde{x}_m$. Now, as $\tilde{v}_{m+1}$ is a prefix of $\tilde{x}_{m+1}$, we have that $\tilde{v}_m$ is a prefix of $\tilde{v'}_m$. Thus by induction we have that $\tilde{v}_m$ is a prefix of $\tilde{x}_m$ for $m = r-1, r-2, \ldots, 1, 0$.

Due to the length of $v$, we have $|\tilde{v}_0| \geq 2$. Similarly, one can show that if $x_{r-1} > w$ 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 $Rep_{i+1}(u)$ (for $i \leq r-1$) 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 the extended rank of $(Z, \leq)$ is less than $(r, n, k)$. Thus, $z$ is the desired $\Sigma$-tree. This proves the claim.

**Claim 16.** *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 $A'$ of all $\Sigma$-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 $A'$. Now, $\mathcal{A}' = (A', \leq')$ can be converted into a tree-automatic scattered linear order of the set $B$: for $t_1, t_2 \in B$ generated by $\tilde{t}_1, \tilde{t}_2$ respectively, let $t_1 \leq'' t_2$ iff $t'_1 \leq' t'_2$, where $t'_1$ and $t'_2$ are generated by $\tilde{z}\tilde{t}_1$ and $\tilde{z}\tilde{t}_2$ respectively. The linearly ordered set $(B, \leq'')$ has extended rank $(r', n', k')$ which is smaller than the extended rank of $(B, \leq)$. This contradicts the choice of $\leq$. This completes the proof of Theorem 12. $\square$

**Corollary 17.** *The set* $T(\Sigma)$ *of all* $\Sigma$*–trees does not admit any well-ordering.*

Note that almost all parts of the proof of Theorem 12 worked only on the trees in $B$ of the special form which are generated by nodes. A straightforward generalisation gives the following corollary.

**Corollary 18.** *Let* $a, b, c, d$ *be strings such that* $b$ *and* $c$ *are different but of the same length. Let* $A = \{x \mid \text{some node } \tilde{x} \in a(b \cup c)^{\star}d \text{ generates the tree } x\}$. *Then there is no tree-automatic scattered linear order on* $A$.

## 5 A characterisation

The techniques of the previous section can be applied to characterise regular tree languages that admit scattered linear orders.

**Theorem 19.** *Let* $A$ *be a regular tree language. Then the following three conditions are equivalent:*

(1) *There is a tree-automatic scattered linear order* $\leq$ *on* $A$*;*

(2) *There is a tree-automatic well-order* $\sqsubseteq$ *on* $A$*;*
(3) *The tree* $T = \bigcup\{dom(t) \mid t \in A\}$ *has finite tree-rank.*

**Proof.** For finite $A$, theorem can easily be seen to hold. So assume $A$ is infinite.

It is clear that (2) implies (1). We now prove that (3) implies (2). We will give a proof in which we define the well-order. Let us define the tree-rank $tr(s)$ of a node $s \in T$ as the tree-rank of the subtree $T$ above $s$, that is, the subtree with root $s$. Then $tr(s)$ is a natural number by the assumption. For every branching node $s \in T$ either

- $tr(s0) < tr(s)$ and $tr(s1) < tr(s)$, or
- $tr(si) = tr(s)$ and $tr(s(1-i)) < tr(s)$ for $i = 0$ or $i = 1$.

Let $\Sigma = \{a, b\}$ and define $\Lambda\colon T \to \Sigma^\star$ as follows. Let $\Lambda(\lambda) = \lambda$. Suppose that $\Lambda(s)$ is defined and $s0, s1 \in T$. Now the following statements hold:

- $\Lambda(s0) = \Lambda(s)a$, $\Lambda(s1) = \Lambda(s)b$ if $tr(s0) < tr(s)$;
- $\Lambda(s0) = \Lambda(s)b$, $\Lambda(s1) = \Lambda(s)a$ if $tr(s0) = tr(s)$ and $tr(s1) < tr(s)$.

The first $n$ symbols of $\Lambda(s)$ and $\Lambda(t)$ are equal if $s, t \in T$ have a common prefix of length $n$. Note that $\Lambda(s)$ contains $a$ at most $tr(T)$ times for all nodes $s \in T$. Let $s \sqsubseteq t$ for trees $s, t \in A$ if one of the following three conditions holds:

- $s = t$ or
- $dom(s) \neq dom(t)$ and $u \in dom(t)$ for the unique $u \in (dom(s) - dom(t)) \cup (dom(t) - dom(s))$ with lexicographically largest $\Lambda(u)$ or
- $dom(s) = dom(t)$ and $s(u) < t(u)$ for the lexicographically largest $u \in dom(s)$ such that $s(u) \neq t(u)$.

Since $tr(T)$ is finite, the lexicographical order on $range(\Lambda)$ is a well-order, and hence $\sqsubseteq$ is a well-order on $A$.

We consider the extended language with a relation for slim $\Sigma$-trees. Any relation on $T$ defined by a first-order statement about $A$ in the extended language is decidable by a tree automaton by the proof of the uniform decidability Theorem 4. It is easy to verify that for each $d < tr(T)$, there is a first-order formula in this language which decides whether the tree-rank of a given node is at least $d$. It follows that $\sqsubseteq$ is tree-automatic.

It remains to show that (1) implies (3). Suppose by way of contradiction that the tree $T = \bigcup\{dom(t) \mid t \in A\}$ does not have a finite tree-rank. Since $T$ is regular and does not have a finite tree-rank, there is a word $\alpha$ such that the automaton accepting $T$ takes the same state $s$ after reading $\alpha$ and again after reading $\alpha\beta$ and $\alpha\gamma$ for two incomparable extensions. We can assume that $\beta$ and $\gamma$ have the same length, otherwise we replace them with $\beta^{|\gamma|}$ and $\gamma^{|\beta|}$, respectively. Let $\delta$ denote a word such that the state after reading $\alpha\delta$ is accepting. Then $\alpha(\beta \cup \gamma)^\star\delta \subseteq T$.

Let us assign a $\Sigma$-tree $t_s \in A$ to each $s \in \alpha(\beta \cup \gamma)^\star\delta$ as follows. There is a $\Sigma$-tree $t \in A$, containing $s$ as a node, such that

(i) $t$ is recognised by the automaton accepting $A$ and

(ii) for any node $s' = s''s'''$ in $t$, where $s''$ is the largest common prefix of $s'$ and $s$, in the accepting run of the automaton on $t$, no state of the automaton is repeated in the suffix $s'''$ of $s'$.

Hence for some constant $c$, all nodes of $dom(t)$ have distance $\leq c$ from some prefix of $s$. We can now choose the unique $\Sigma$-tree $t_s \in A$ satisfying (i) and (ii) above such that for all $t \in A$ satisfying the above requirements (i) and (ii) and one of the following three conditions:

- $t_s = t$ or
- $dom(t_s) \neq dom(t)$ and $u \in dom(t)$ for the unique $u \in (dom(t_s) - dom(t)) \cup (dom(t) - dom(t_s))$ with lexicographically largest $u$ or
- $dom(t_s) = dom(t)$ and $t_s(u) < t(u)$ for the lexicographically largest $u \in dom(t_s)$ such that $t_s(u) \neq t(u)$.

As in the previous part of the proof, the assignment of $t_s$ to $s \in \alpha(\beta \cup \gamma)^\star \delta$ is tree-automatic. We can assume that it is injective, since we can replace $\beta, \gamma$ with $\beta^{c+1}, \gamma^{c+1}$ otherwise (note that this will imply injectivity as $t_s$ contains $s$ as a node and none of the nodes in $t_s$ are at a distance more than $c$ from a prefix of $s$, whereas different members of $\alpha(\beta^{c+1} \cup \gamma^{c+1})^\star \delta$ are at least a distance $c+1$ apart).

Recall from Section 2 that a tree $x$ is generated by a string $\tilde{x}$ if $\tilde{x}$ is a branching node of the tree $x$ and every branching node of $x$ is a prefix of $\tilde{x}$. If $A$ admits a tree-automatic scattered linear order $\leq_A$, then $B = \{x \mid \tilde{x} \in \alpha(\beta \cup \gamma)^\star \delta$ and $x$ is generated by $\tilde{x}\}$ admits a tree-automatic scattered linear order $\leq_B$ defined by $x \leq_B y$ if $t_{\tilde{x}} \leq_A t_{\tilde{y}}$. However, this contradicts Corollary 18. $\square$

The tree-rank of $T = \bigcup\{dom(t) \mid t \in A\}$ is either infinite or bounded by the number of states of a deterministic automaton accepting the nodes in $T$. Therefore we can compute this upper bound and then apply Theorem 4 to determine whether the tree-rank of $T$ is properly above this bound; this decides whether $T$ has finite tree-rank and thus we have the following corollary.

**Corollary 20.** *It is decidable if a given regular tree language can be well-ordered by a tree automaton.*

In contrast to Theorem 12 we have the following result.

**Theorem 21.** *The set $\Sigma^\star$ admits every infinite word-automatic scattered linear order.*

**Proof.** Suppose that $\mathcal{L} = (L, \leq)$ is an infinite word-automatic scattered linear order. Recall the definition of $\sim_i$ defined before Theorem 1.

Suppose first that every $\sim_1$-equivalence class is finite. Then $\sim_\alpha = \sim_1$ for all $\alpha \geq 1$. Hence $\mathcal{L}/\sim_1$ is finite by Theorem 1. Thus $\mathcal{L}$ is finite, contradicting the assumption.

Suppose that $[x] = \{y \mid x \sim_1 y \text{ in } \mathcal{L}\}$ is an infinite $\sim_1$-equivalence class. Thus $[x]$ and $C = [x] \cup (\Sigma^\star \setminus L)$ are regular. Hence, $([x], \leq)$ is isomorphic to either the positive integers, or the negative integers, or the integers. We first suppose that $([x], \leq)$ is isomorphic to $(\omega, <)$. Let $L_{[x]} = \{z \in L \mid z < x, z \notin [x]\}$ and $R_{[x]} = \{z \in L \mid z > x \text{ and } z \notin [x]\}$ so that $\Sigma^\star = L_{[x]} \cup C \cup R_{[x]}$.

We define a linear order $\leq'$ extending the old order $\leq$ on $L_{[x]}$ and $R_{[x]}$ such that the words in $[x] \cup (\Sigma^\star \setminus L)$ are ordered length-lexicographically, all the elements in $C$ are strictly greater than all elements in $L_{[x]}$, and all elements in $C$ are strictly smaller than all elements in $R_{[x]}$. Then $\leq'$ is a word-automatic linear order on $\Sigma^\star$ isomorphic to $\mathcal{L}$. The cases for $([x], \leq)$ being isomorphic to the negative integers and the integers, respectively, are similiar, since every infinite regular set allows both of these order types. $\square$

We do not know if the above result holds for all infinite word-automatic linear orders.

## 6 Alternative Proofs

Büchi proved that the $\omega$-regular languages of $\omega$-words and those definable by monadic second order logic are the same. As the referees pointed out, this connection also holds for the tree-variant of regularity. Recall that a scattered tree-automatic linear ordering on the $\Sigma$-trees induces a scattered tree-automatic linear ordering on the slim $\Sigma$-trees. This ordering is $MSO$-definable and in turn induces an $MSO$-definable scattered linear ordering on the nodes generating the slim $\Sigma$-trees. Thus one has a scattered linear ordering on all the nodes of the full binary tree. Now, the next step is the following result which was sketched by an anonymous referee.

**Lemma 22.** *Let $A \subseteq \{0,1\}^\star$ be regular and let $\leq$ be a scattered linear order on $A$ which is $MSO$-definable in the full binary tree. Then there is a well-order on $A$ which is $MSO$-definable in the full binary tree.*

**Proof.** Let $B$ be the set of slim $\Sigma$-trees (for unary $\Sigma$) generated by elements of $A$, i.e. $t \in B$ if there are $a_1 a_2 \dots a_n \in A$ with each $a_i \in \{0,1\}$, $dom(t) = \{a_1 a_2 ... a_i b \mid b \in \{0,1\}, i \leq n\}$ and $t(x)$ is the unique value in $\Sigma$, for all $x \in dom(t)$. We will denote the linear order on $B$ corresponding to $\leq$ by $\leq$ as well. We now follow the first part of the proof of Theorem 12 and use the notation from this proof. The only difference is that in the proof of Theorem 12, we used $\sim_{i+1}$ to denote the restriction to $B_i$, while here we will again use $\sim_{i+1}$ for the relation on $B$ as defined just before Theorem 1. As noted in the proof of Theorem 12, the relation $\sim_i$ used in the current proof can be obtained by taking the union of $\sim_j$, for $j \leq i$, used in the proof of Theorem 12.

As before, the linear order $(B, \leq)$ is automatic and therefore has finite CB-rank $r$. Then $r$ is minimal with $\sim_{r+1} = B^2$. As before, all equivalence relations $\sim_i$ are definable (using $FO + \exists^\omega$) in $(B, \leq)$, and there are tree-automatic functions $f_i = rep_i$ choosing representatives from every $\sim_i$-equivalence class. We can additionally assume that for all $i$, the range of $f_{i+1}$ is contained in the range of $f_i$. We use these functions to define a well-order $\sqsubseteq$ on $B$ by placing the $\sim_i$ classes greater that the representative given by $f_{i+1}$ below (with respect to $\sqsubseteq$) the $\sim_i$ classes less than the representative given by $f_{i+1}$ (where the ordering is inverted). More precisely, let $x \sqsubseteq y$ if $f_i(x) \leq x \leq y$, $y \leq x \leq f_i(x)$ or $y < f_i(x) \leq x$, where $i$ is minimal with $x \sim_i y$. This relation is tree-automatic, since it is definable in the tree-automatic structure $(B, \leq, (\sim_i, f_i)_{i \leq r})$). Therefore its copy on $A$ is $MSO$-definable in the full binary tree. The lemma now follows from the two claims below which verify that $\sqsubseteq$ is a well-order.

**Claim 23.** $\sqsubseteq$ *is a linear order on* $B$.

For a proof of the claim, it is easy to see that $\sqsubseteq$ is reflexive, antisymmetric and total. We prove that $\sqsubseteq$ is transitive. Let $i_{x,y}$ denote the least $i \leq r$ with $x \sim_i y$, for $x, y \in B$. We write $x \sqsubset y$ if $x \sqsubseteq y$ and $x \neq y$. Suppose that $x \sqsubset y$ and $y \sqsubset z$. We prove that $x \sqsubset z$. Note that in $\{i_{x,y}, i_{x,z}, i_{y,z}\}$ there are at least two equal values and these are maximal among these three values. We distinguish four cases.

First suppose that $i = i_{x,y} < j = i_{y,z} = i_{x,z}$.

- Suppose $y < f_j(y)$. Then, $z < y < f_j(y) = f_j(x)$. Hence either $z < f_j(x) \leq x$ or $z < x < f_j(x)$ (as $f_i(x) = f_i(y) \neq f_i(z)$). In both cases $x \sqsubset z$.
- Suppose $f_j(y) \leq y$. Since $y \sqsubset z$, either $f_j(y) \leq y < z$ or $z < f_j(y) \leq y$. If $f_i(y) = f_j(y)$, then $f_i(x) = f_i(y) = f_j(y)$. Since $x \sqsubset y$ and $f_i(x) \leq y$, we have $f_i(x) \leq x < y$. Hence either $f_j(x) \leq x < y < z$ or $z < f_j(x) \leq x < y$. In both cases $x \sqsubset z$. On the other hand, if $f_i(y) \neq f_j(y)$, then $x$ is not $\sim_i$-equivalent to $f_j(y)$. Since $f_j(y) \leq y$, we have $f_j(x) = f_j(y) < f_i(x) = f_i(y)$ and $f_j(y) < x$. Note that $x$ and $y$ are $\sim_i$-equivalent, but $x$ and $z$ are not $\sim_i$-equivalent. If $f_j(y) \leq y < z$, then $f_j(x) < x < z$. If $z < f_j(y) \leq y$, then $z < f_j(x) < x$. In both cases $x \sqsubset z$.

Second, suppose that $i = i_{x,z} < j = i_{x,y} = i_{y,z}$. Then $x \sim_i z$ but $x \not\sim_i y$ and $y \not\sim_i z$. Since $y \sqsubset z$, we have the following options.

- Suppose $f_j(y) \leq y < z$. Then $f_j(y) \leq y < x$, contradicting the assumption that $x \sqsubset y$.
- Suppose $z < y < f_j(y)$. Then $x < y < f_j(y)$, again contradicting the assumption that $x \sqsubset y$.
- Suppose $z < f_j(y) \leq y$. Then $x < y$. Since $x \sqsubset y$, we have $f_j(y) = f_j(x) \leq x < y$. Hence, $f_j(x) = f_i(x)$ (as $z < f_j(x) \leq x$ and thus $f_j(x)$ must be same as $f_i(x)$). Then $z < f_i(x) = f_j(x) \leq x$ and hence $x \sqsubset z$.

Third, suppose that $i = i_{y,z} < j = i_{x,y} = i_{x,z}$. Since $x \sim_j y$, $f_j(x) = f_j(y)$.

- Suppose $f_j(x) \leq x$. Since $x \sqsubset y$, $y < f_j(x) \leq x$ or $f_j(x) \leq x < y$. Since $y \sim_i z$ and $x \not\sim_i y$, $z < f_j(x) \leq x$ or $f_j(x) \leq x < z$, so $x \sqsubset z$.
- Suppose $x < f_j(x)$. Since $x \sqsubset y$, $y < x \leq f_j(x)$. Since $y \sim_i z$ and $x \not\sim_i y$, $z < x \leq f_j(x)$. Hence $x \sqsubset z$.

Fourth, suppose that $i = i_{x,y} = i_{x,z} = i_{y,z}$.

- Suppose $x < f_i(x)$. Then, $y < x < f_i(x)$ as otherwise $y \sqsubset x$. Furthermore, as $y \sqsubset z$, we have $z < y < x < f_i(x)$. Hence we have $x \sqsubset z$.
- Suppose $f_i(x) \leq x$. Thus, either $f_i(x) \leq x < y$ or $y < f_i(x) \leq x$. If $f_i(x) \leq x < y$ then either $f_i(x) \leq x < y < z$ or $z < f_i(x) \leq x < y$, and in both cases, $x \sqsubset z$. On the other hand, if $y < f_i(x) \leq x$, then $z < y < f_i(x) \leq x$, as otherwise $z \sqsubset y$. Thus, we have $x \sqsubset z$.

**Claim 24.** $\sqsubseteq$ *is well-founded.*

For a proof of the claim, suppose that there exists an infinite strictly decreasing sequence in $(B, \sqsubseteq)$. By Ramsey's theorem, there is some $i \leq r$ and an infinite subsequence $(x_n)_{n \in \{0,1,2,\ldots\}}$

such that $x_{m+1} \sim_{i+1} x_m$ and $x_{m+1} \not\sim_i x_m$, and $x_{m+1} \sqsubset x_m$ for all $m$. If there are infinitely many $m$ such that $x_m \leq f_{i+1}(x_0)$, then there exist $m_1, m_2$ such that $m_1 < m_2$ and $x_{m_1} \sqsubset x_{m_2}$, contradicting the choice of $x_i$'s. Thus, for all but finitely many $m$, $f_{i+1}(x_0) \leq x_m$. But then for the least $m$ such that $f_{i+1}(x_0) \leq x_m$, we have that $f_{i+1}(x_m) = f_{i+1}(x_0) \not\sim_{i+1} x_m$, as there are infinitely many $m'$ such that $f_{i+1}(x_m) = f_{i+1}(x_0) < x_{m'} < x_m$. This contradicts the assumption on $f_{i+1}$ that $f_{i+1}(x_m) \sim_{m+1} x_m$. This completes the proof of the claim and also of Lemma 22. $\square$

**Alternative Proof for Theorems 12 and 19.** The alternative proof is for the following statement:

> If a regular tree-language $R$ satisfies that the union $T$ of the domains of the $\Sigma$-trees in $R$ has infinite tree-rank, then there is neither a tree-automatic scattered linear order on $R$ nor a tree-automatic well-order on $R$.

So assume that $R$ and $T$ are given and that $T$ has infinite tree-rank. There is a regular subset $S$ of $T$ satisfying the following two conditions:

- For every branching node $\sigma \in S$ there are exactly two branching nodes $\tau \in S$ such that $\tau$ properly extends $\sigma$ and there is no branching node of $S$ between $\sigma$ and $\tau$;
- For every branching node $\sigma \in S$ there is a tree $t \in R$ such that $t$ contains a branching node $\tau \in S$ iff $\tau \preceq \sigma$.

To see that such an $S$ exists, first let $S'$ be a prefix-closed regular subset of $T$ such that for every branching node $\sigma$ of $S'$, there exist branching node of $S'$ which extends $\sigma 0$ and another branching node of $S'$ which extends $\sigma 1$. Note that, using first-order definability of slim $\Sigma$-trees one can first-order define whether tree-rank of a tree is greater than any fixed given constant $d$. Furthermore, by the pumping lemma (Lemma 5, for an automaton for accepting $T$), tree-rank of a subtree of $T$ is finite if only if it is bounded by some constant $c$ (depending on the automaton accepting $T$). Hence, one can construct an automaton which accepts a node in $T$ iff the subtree rooted at it has at least tree-rank $c+1$. The so-defined regular set $S'$ consists therefore of all nodes in $T$ which are roots of a sub-tree of $T$ with infinite tree-rank. Now given any node $v$ of $S'$, one can automatically find a tree $t_v$ in $R$ which contain the node (that is, the mapping from $v$ to $t_v$ is automatic). As this mapping is automatic, we have a constant $c'$ such that every node of $t_v$ is at a distance at most $c'$ from a prefix of $v$. Now take $S$ as a regular subset of $S'$ for which (i) the root of $S'$ belongs to $S$, (ii) between any two branching nodes of $S$ there are exactly $2c'+1$ branching nodes of $S'$, (iii) for every branching node $\sigma$ of $S$, there are branching nodes extending $\sigma 0$ and $\sigma 1$ in $S$, and (iv) the branching nodes of $S'$ which are not branching nodes of $S$ have only the left child of $S'$ as a member of $S$. This $S$ satisfies the requirements given above.

Now there is a tree-automatic function $F$ which, for each branching node $\sigma \in S$, picks $t \in R$ containing exactly those branching nodes $\tau \in S$ which satisfy $\tau \preceq \sigma$; by assumption on $S$ this function is one-one. Hence one can deduce from the scattered linear ordering on $R$ a scattered linear ordering on the branching nodes of $S$: $\sigma \sqsubset_S \tau \Leftrightarrow F(\sigma) \sqsubset_R F(\tau)$ where $\sqsubset_S, \sqsubset_R$ refer to the two linear orders; as $\sqsubset_R$ is scattered so is $\sqsubset_S$. Furthermore, by Lemma 22, there is now a well-ordering on $S$ which is $MSO$-definable. However, Carayol and Löding [7, Theorem 7] proved that the theory of any well-ordering on the nodes of a full binary tree is undecidable. As the theory of any $MSO$-definable ordering on $S$ is decidable, this gives a contradiction and so the given tree-language must satisfy that the union $T$ of its domains does have finite tree-rank. $\square$

The just cited result is an automata-theoretic version of the Theorem of Gurevich and Shelah [11] who proved that there is no $MSO$-definable choice-function on the full binary tree. The undecidability result of Carayol and Löding [7] is stronger and proven with purely automata-theoretic tools.

## 7 Lower and upper bounds

Suppose that $A$ is an infinite regular tree language with a tree-automatic well-ordering. Let $minord(A)$ and $maxord(A)$ denote the minimum and the supremum of the ordinals $\alpha$ such that $A$ admits a tree-automatic well-ordering of type $\alpha$. In this section we study the possible values which $minord(A)$ and $maxord(A)$ might take. It is known that $minord(A) = \omega$ and $maxord(A) = \omega^2$ if all branching nodes in the trees in $A$ are of the form $1^n$ and $\Sigma$ is unary, and $minord(A) = \omega$ and $maxord(A) = \omega^\omega$ if $|\Sigma| > 1$ (see for instance [34]). We generalise this to the case where the tree-rank of $T = \bigcup\{dom(t) \mid t \in A\}$ is at most $k$.

Let us define the $\omega^k$-automata connected to this question. The definition and basic results on ordinal automata can be found for instance in [35, 10, 17]. Let $\mathcal{P}(S)$ denote the powerset of $S$. Suppose that $\gamma$ is an ordinal and $\Sigma$ is a finite alphabet. A (deterministic) *$\gamma$-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 $w : \gamma \to \Sigma \cup \{\square\}$ (that is, all but finitely many of its letters are $\square$) are read successively. Here, we let $dom(w) = \{\alpha \mid w(\alpha) \neq \square\}$. At every limit time $\gamma' \leq \gamma$ the state is determined by the limit transition function applied to the set of states appearing unboundedly often before $\gamma'$, as in Büchi and Muller automata [5, 6, 1, 31]. At time $\gamma$ we check whether the state is an accepting state.

**Definition 25.** Suppose $\gamma$ is an ordinal and $A$ is a set of finite $\gamma$-words. A structure $\mathcal{A} = (A; R_1, ..., R_n)$ is finite word $\gamma$-automatic if the domain and the relations are recognizable by deterministic $\gamma$-automata.

Let $C_k^\Sigma$ denote the set of $\Sigma$-trees such that each branching node contains at most $k$ 0's.

**Theorem 26.** *The following conditions are equivalent for relational structures $\mathcal{A}$:*

(1) *$\mathcal{A}$ is isomorphic to a tree-automatic structure with domain $B \subseteq C_k^\Sigma$ for some finite alphabet $\Sigma$;*
(2) *$\mathcal{A}$ is isomorphic to a tree-automatic structure with domain $B$ such that $tr(\bigcup_{t\in B} dom(t)) \leq k+1$;*
(3) *$\mathcal{A}$ is isomorphic to an $\omega^{k+1}$-automatic structure.*

**Proof.** The implication from (1) to (2) holds since $tr(T) = k+1$ for $T = \bigcup_{t\in C_k^\Sigma} dom(t)$. The implication from (2) to (3) can easily be proven in a way similar to the proof of Theorem 19, part (3)⇒(2), and using the result of Neeman [29, Theorem 7] that any nondeterministic $\omega^{k+1}$-automaton can be replaced by a deterministic $\omega^{k+1}$-automaton. Since the argument is analogous to the proof in Theorem 19, we only sketch the idea here. Using the mapping $\Lambda$ given in the proof Theorem 19, one can map each node $s$ in $T$ to an ordinal $val(s) < \omega^{k+1}$ as follows: for a node $s$ in $T$, if $\Lambda(s) = b^{c_k}ab^{c_{k-1}}a \dots b^{c_i}a$, where $i \geq 0$, then

$$val(s) = \omega^k(2c_k + 1) + \omega^{k-1}(2c_{k-1} + 1) + \dots + \omega^i(2c_i + 1);$$

if $\Lambda(s) = b^{c_k}ab^{c_{k-1}}a \dots b^{c_i}$, where $i \geq 0$, then

$$val(s) = \omega^k(2c_k + 1) + \omega^{k-1}(2c_{k-1} + 1) + \dots + \omega^i(2c_i).$$

Then, each $\Sigma$-tree $t \in C_k^\Sigma$ can be mapped to a $\omega^{k+1}$-word $w_t$ with domain $\{val(s) \mid s \in dom(t)\}$ and $w_t(val(s)) = t(s)$. Now the different $\Sigma$-trees $t$ and $t'$ in $C_k^\Sigma$ can be compared in a way similar to Theorem 19 by defining $t \leq t'$ iff (i) $w_t = w_{t'}$ or (ii) $dom(w_t) \neq dom(w_{t'})$ and for the largest $u \in (dom(w_{t'}) - dom(w_t)) \cup (dom(w_t) - dom(w_{t'}))$, $u$ belongs to $dom(w_{t'})$ or (iii) $dom(w_t) = dom(w_{t'})$ and $w_t(u) < w_{t'}(u)$ for the largest $u$ in $dom(w_t)$ such that $w_t(u) \neq w_{t'}(u)$. The rest of the argument is straightforward and similar to the proof of Theorem 19 and is therefore omitted.

To prove the implication from (3) to (1), let $T = \bigcup_{t \in C_k^\Sigma} dom(t)$. We consider the bijection $g : \omega^{k+1} \to T$ defined by $g(\omega^k m_k + \omega^{k-1} m_{k-1} + \dots + m_0) = 1^{m_k}01^{m_{k-1}}0 \dots 1^{m_0}$ and its inverse $f : T \to \omega^{k+1}$ (note that $f$ is partial).

We consider for each finite $\omega^{k+1}$-word $w$ over a finite alphabet $\Sigma$ the $\Sigma \cup \{\diamond\}$-tree $t$ such that the branching nodes of $dom(t)$ are the prefixes of elements of $g[dom(w)]$, $t(u) = w(f(u))$ for all $u \in g[dom(w)]$, and $t(u) = \diamond$ otherwise. This defines a bijection from the set of finite $\omega^{k+1}$-words to a tree regular subset of $C_k^{\Sigma \cup \{\diamond\}}$.

We simulate the deterministic $\omega^{k+1}$-automaton $Q$ by a nondeterministic tree automaton as follows. The states of the tree automaton are of the form $(s_1, s_2, X_1, X_2, i)$. Intuitively, the tree-automaton being at state $(s_1, s_2, X_1, X_2, i)$ at a tree-node means that if the automaton $Q$ starts at state $s_1$, with the states in $X_1$ having been already visited infinitely often, then after processing the corresponding section (above the tree-node having the state $s_1$) of the input word, the automaton $Q$ ends up in state $s_2$ having visited states in $X_2$ infinitely often; here $i$ denotes the number of 0-branches having been taken from the root to reach the current tree-node. Thus, at the root, the starting state of the tree-automaton is $(s_1, s_2, X_1, X_2, 0)$, where $s_1$ is the starting state of the automaton $Q$, $X_1 = \emptyset$, $s_2$ is the guessed accepting state after processing the whole word and $X_2$ is the guessed set of states which are visited infinitely often.

If at a tree-node the state is $(s_1, s_2, X_1, X_2, i)$, then on the 0-branch child, the state is $(s_1, s_2', X_1, X_2', i+1)$ and on the 1-branch the state is $(s_2', s_2, X_2', X_2, i)$, where $s_2'$ is a guessed state after having read the subword of the input corresponding to the 0-child. At a leaf node with state $(s_1, s_2, X_1, X_2, i)$, the tree-automaton accepts iff either $i = k + 1$ and $s_1 = s_2$, $X_1 = X_2$, or $i \leq k$, and the automaton $Q$ starting in state $s_1$, after seeing $\omega^{k+1-i}$ $\square$'s would end up in state $s_2$ and $X_2 =$ ($X_1$ unioned with the set of states visited infinitely often in the above process).

It is easy to verify that the above tree-automaton will be able to simulate the $\omega^{k+1}$-automaton $Q$. $\square$

Note that a similar connection has been studied by Finkel and Todorcevic in [10, Proposition 3.4]. Through this correspondence we obtain by [35, Proposition 16] the following corollary.

**Corollary 27.** *The CB-rank of every scattered tree-automatic linear order is below $\omega^\omega$. In particular, every tree-automatic ordinal is below $\omega^{\omega^\omega}$ (see [9]).*

In fact the CB-rank of all tree-automatic linear orders is below $\omega^\omega$ [15].

We now determine *minord* and *maxord* for some regular tree languages. For each $k \geq 1$, let $A_k$ denote the regular tree language consisting of all $\{0\}$-trees where each branching node is of the form $0^m 1^n$ for some $m < k$ and some $n \in \mathbb{N}$.

**Theorem 28.** *Let $k \geq 1$.*

(a) $\mathit{minord}(A_k) = \omega^k$, $\mathit{maxord}(A_k) = \omega^{k+1}$.
(b) $\mathit{minord}(C_k^{\{0\}}) = \omega^{k+1}$, $\mathit{maxord}(C_k^{\{0\}}) = \omega^{\omega^{k+1}}$.

**Proof.** (a) We identify each tree $t \in A_k$ with the tuple $(a_0, a_1, \ldots, a_{k-1})$, where $a_i$ is the largest natural number such that $0^i 1^{a_i}$ is a branching node in $t$. This gives us a one-to-one correspondence between $\mathbb{N}^k$ and $A_k$. The lexicographical order on $\mathbb{N}^k$ induces a tree-automatic order on $A_k$ of order type $\omega^k$ and hence $\mathit{minord}(A_k) \leq \omega^k$.

**Claim 29.** *Suppose that $\leq$ is a tree-automatic well-order on $A_k$. Then, there exists a constant $c$ such that for all $b_0, b_1, \ldots, b_{k-1} < c$, the order type of*

$$H_{b_0,b_1,\ldots,b_{k-1}} = \{(ca_0 + b_0, ca_1 + b_1, \ldots, ca_{k-1} + b_{k-1}) \mid a_i > 0, \text{ for all } i < k\},$$

*with respect to $\leq$ is $\omega^k$.*

For ease of notation, for $\tilde{b} = (b_0, b_1, \ldots, b_{k-1})$, let $c \cdot (a_0, a_1, \ldots, a_{k-1}) + \tilde{b}$ denote $(ca_0 + b_0, ca_1 + b_1, \ldots, ca_{k-1} + b_{k-1})$.

To prove the claim, note that for $a_0, a_1, \ldots, a_{k-1} \geq 1$, $a'_0, a'_1, \ldots, a'_{k-1} \geq 1$ the following hold for all $i \leq k-1$ (where the first two items follow using Lemma 5, by taking $c$ below as the factorial of the constant $c$ in the lemma).

- if $a_i > a'_i$ and $c \cdot (a_0, a_1, \ldots, a_i, \ldots, a_{k-1}) + \tilde{b} \leq c \cdot (a'_0, a'_1, \ldots, a'_i, \ldots, a'_{k-1}) + \tilde{b}$ then $c \cdot (a_0, a_1, \ldots, a_i + 1, \ldots, a_{k-1}) + \tilde{b} \leq c \cdot (a'_0, a'_1, \ldots, a'_i, \ldots, , a'_{k-1}) + \tilde{b}$.
- if $c \cdot (a_0, a_1, \ldots, a_i, \ldots, a_{k-1}) + \tilde{b} \leq c \cdot (a'_0, a'_1, \ldots, a'_i, \ldots, a'_{k-1}) + \tilde{b}$ then $c \cdot (a_0, a_1, \ldots, a_i + 1, \ldots, a_{k-1}) + \tilde{b} \leq c \cdot (a'_0, a'_1, \ldots, a'_i + 1, \ldots, a'_{k-1}) + \tilde{b}$.

Since $\leq$ is a well-order, this implies

- $c \cdot (a_0, a_1, \ldots, a_i, \ldots, a_{k-1}) + \tilde{b} \leq c \cdot (a_0, a_1, \ldots, a_i + 1, \ldots, a_{k-1}) + \tilde{b}$.

We can assume without loss of generality, by possibly changing the order of the coordinates, that for all $i < k-1$

- if
  - $a'_i = 2$, $a_{i+1} = 2$,
  - $a_j = 1$ if $j \leq i$ or $i+1 < j \leq k-1$, and
  - $a'_j = 1$ if $j < i$ or $i < j \leq k-1$,

  then $c \cdot (a_0, a_1, \ldots, a_{k-1}) + \tilde{b} \leq c \cdot (a'_0, a'_1, \ldots, a'_{k-1}) + \tilde{b}$.

The previous four statements imply by reverse induction on $i \leq k$ that

- if
  - $(a_j = a'_j$ for $j < i)$,
  - $a_i < a'_i$, and
  - $(a_j, a'_j \geq 1$, for $j \leq k-1)$,

  then $c \cdot (a_0, a_1, \ldots, a_i, \ldots, a_{k-1}) + \tilde{b} \leq c \cdot (a'_0, a'_1, \ldots, a'_i, \ldots, a'_{k-1}) + \tilde{b}$.

Thus the set $\{(ca_0, ca_1, \ldots, ca_{k-1}) \mid a_i > 0$, for $0 \leq i \leq k-1\}$ has order type $\omega^k$ in the ordering $\leq$, and hence $minord(A_k) \geq \omega^k$. Along with the already shown part that $minord(A_k) \leq \omega^k$, we have $minord(A_k) = \omega^k$.

We now show that $maxord(A_k) = \omega^{k+1}$. Let $a_0$ mod $m$ denote the remainder of the division of $a_0$ by $m$. Note that $maxord(A_k) \geq \omega^k \times \{0, 1, \ldots, m-1\}$ by using the order $\leq_m$: $(a_0, a_1, \ldots, a_{k-1}) \leq_m (a'_0, a'_1, \ldots, a'_{k-1})$ iff $a_0$ mod $m < a'_0$ mod $m$ or $a_0$ mod $m = a'_0$ mod $m$ and $(a_0, a_1, \ldots, a_{k-1})$ is lexicographically at most $(a'_0, a'_1, \ldots, a'_{k-1})$. Thus, $maxord(A_k) \geq \omega^{k+1}$.

We now prove that $maxord(A_k) \leq \omega^{k+1}$ by induction on $k$. This holds for $k = 1$ by [34]. So assume that it holds for $k$, and consider an ordering $\leq$ on $A_{k+1}$.

- By Claim 29 there exists a constant $c$ such that for $b_0, b_1, \ldots, b_k < c$, $H_{b_0,b_1,\ldots,b_k} = \{(ca_0 + b_0, ca_1 + b_1, \ldots, ca_k + b_k) \mid a_i > 0$, for all $i \leq k\}$ has order type $\omega^{k+1}$.
- For all $b < c$, let $Y^i_b = \{(a_0, a_1, \ldots, a_k) \mid a_i = b\}$. Then, by induction the order type of $Y^i_b$ is at most $\omega^{k+1}$.

Since, for any ordering of $A_{k+1}$ there exists a $c$ such that the ordering of $A_{k+1}$ can be divided into finitely many parts of the form given by the above two cases, we immediately have that the order type of $A_{k+1}$ is at most $\omega^{k+2}$. This proves part (a).

(b) It follows from Corollary 17 in [35] and Theorem 26 that $maxord(C^{\{0\}}_k) = \omega^{\omega^{k+1}}$ and that $minord(C^{\{0\}}_k) \leq \omega^{k+1}$. Note that $A_{k+1} \subseteq C^{\{0\}}_k$. It follows that $minord(C^{\{0\}}_k) \geq minord(A_{k+1}) \geq \omega^{k+1}$. $\square$

**Theorem 30.** *Let $A$ be an infinite regular tree language. Then $minord(A)$ and $maxord(A)$ fulfill the following conditions:*

(a) *$minord(A)$ is of the form $\omega^\alpha n$, where $n \in \mathbb{N}$ and $n \geq 1$;*

(b) *$maxord(A)$ is of the form $\omega^\alpha n$, where $n \in \mathbb{N}$, $n \geq 1$; furthermore, $\omega^\alpha n$ is not realized and if $maxord(A) = \omega^\alpha n$ with $\alpha \geq \omega$ then $n = 1$.*

**Proof.** (a) Suppose that $\beta = minord(A)$ as witnessed by an ordering $\leq$ on $A$. Suppose that $\beta = \omega^{\alpha_0} n_0 + ... + \omega^{\alpha_k} n_k$ with $\alpha_0 > ... > \alpha_k$ such that $n_i > 0$ for all $i \leq k$. 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 on $A$ of type $\omega^{\alpha_0} n_0$ as follows. Let $a$ be such that $B = \{x \in A \mid x < a\}$ is of order type $\omega^{\alpha_0} n_0$. Let $x \leq' y$ if $[(x, y \in B$ or $x, y \in A \setminus B)$ and $x \leq y]$ or $(y \in B$ and $x \in A \setminus B)$ for $x, y \in A$. It is easy to verify that $\leq'$ has order-type $\omega^{\alpha_0} n_0$. This proves (a).

(b) Suppose that $\beta = maxord(A)$ as witnessed by an ordering $\leq$ on $A$. We first argue that $\beta$ is not realized (and thus $\beta$ is a limit ordinal). Suppose by way of contradiction that $\leq'$ is a tree-automatic well-order on $A$ of order type $\beta$. If $a \in A$ is the least element in $\leq$, then $x \leq' y$ iff $(x \leq y \wedge x \neq a \wedge y \neq a) \vee (y = a)$ defines a well-order of order type $\beta + 1$ on $A$. Thus, $\beta$ is a limit-ordinal which is not realized.

From above, it follows that $maxord(A)$ is of the form $\beta = \omega^{\alpha_0} n_0 + ... + \omega^{\alpha_k} n_k$, where each $n_i > 0$ and $\alpha_0 > \alpha_1 > \ldots > \alpha_k > 0$. Suppose an order $\leq$ on $A$ is of order type $\delta \geq \omega^{\alpha_0} n_0 + ... + \omega^{\alpha_k}(n_k - 1)$. Let $B = \{x \mid x$ is a successor in $\leq$ ordering of $A\}$. Consider the ordering $\leq'$ defined as follows. $x \leq' y$ iff $[(x, y \in B$ or $x, y \in A \setminus B)$ and $x \leq y]$ or $x \in B$ and $y \in A \setminus B$. Then there are two cases:

(i) If $k > 0$, then order type of $\leq'$ is at least $m = \omega^{\alpha_0} n_0 + ... + \omega^{\alpha_k} n_k$ (since, the order type of $B$ is the same as order type of $A$ and the order type of $A \setminus B$ is at least $\omega^\gamma$, for any $\gamma < \alpha_0$), contradicting the above result that $m$ is not attained;

(ii) If $k = 0$, $\alpha_0 \geq \omega$ and $n_0 > 1$, then also the order type of $\leq'$ is at least $\beta = \omega^{\alpha_0} n_0$, contradicting the above result that $\beta$ is not attained.

This completes the proof. $\square$

**Acknowledgments.** We would like to thank the referees of Theoretical Computer Science and of the places of prior submissions for detailed comments which helped improving the paper.

## References

1. Nicolas Bedon. Automata and ordinals. *Theoretical Computer Science*, 156(1–2), 119–144, 1996.
2. Achim Blumensath. *Automatic structures.* Diploma thesis, RWTH Aachen, 1999.
3. Achim Blumensath and Erich Grädel. Automatic structures. *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science, Santa Barbara* (LICS 2000), pages 51-62. IEEE Computer Society, 2000.
4. 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.
5. J. Richard Büchi. On a decision method in the restricted second order arithmetic. *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science*, Stanford University Press, 1960, pages 1–11.

6. J. Richard Büchi. Decision methods in the theory of ordinals. *Bulletin of the American Mathematical Society*, 71, 767–770, 1965.
7. Arnaud Carayol and Christof Löding. MSO on the Infinite Binary Tree: Choice and Order. *Computer Science Logic*, Twentyfirst International Workshop, CSL 2007, Sixteenth Annual Conference of the EACSL 2007, Proceedings. LNCS 4646, pages 161–176. Springer, Berlin 2007.
8. Thomas Colcombet and Christof Löding. Transforming structures by set interpretations. *Logical Methods in Computer Science*, 3(2), 1-36, 2007.
9. Christian Delhommé. *Automaticité des ordinaux et des graphes homogènes. Comptes Rendus Mathematique*, Academy of Sciences in Paris, 339(1), 5–10, 2004.
10. Olivier Finkel and Stevo Todorcevic. A hierarchy of tree-automatic structures. *The Journal of Symbolic Logic*, 77(1), 350-368, 2012.
11. Yuri Gurevich and Sharon Shelah. Rabin's uniformization problem. *The Journal of Symbolic Logic*, 48(4), 1105-1119, 1983.
12. Wilfrid Hodges. *Model Theory.* Cambridge University Press, 1993.
13. Bernard R. Hodgson. On direct products of automaton decidable theories. *Theoretical Computer Science*, 19, 331-335, 1982.
14. Bernard R. Hodgson. Décidabilité par automate fini. *Annales des sciences mathématiques du Québec*, 7(1), 39-57, 1983.
15. Martin Huschenbett. The Rank of Tree-Automatic Linear Orderings. *Symposium on Theoretical Aspects of Computer Science* (STACS 2013), LIPICS 20, pages 586-597, 2013.
16. Martin Huschenbett. Word Automaticity of Tree Automatic Scattered Linear Orderings Is Decidable. *Computability in Europe* (CiE 2012), LNCS 7318, pages 314-322. Springer, Berlin, 2012.
17. Martin Huschenbett and Martin Kartzow and Philipp Schlicht. Pumping for ordinal-automatic structures. to appear in *Computability*, 2016.
18. Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. *Logic and computational complexity* (LCC 1994), LNCS 960, 367-392. Springer, Berlin, 1995.
19. Bakhadyr Khoussainov and Mia Minnes. Model-theoretic complexity of automatic structures. *Annals of Pure and Applied Logic*, 161(3), 416–426, 2009.
20. Bakhadyr Khoussainov and Mia Minnes. Three lectures on automatic structures. *Proceedings of Logic Colloquium (2007)*, Lecture Notes in Logic 35, pages 132-176, 2010.
21. Bakhadyr Khoussainov, André Nies, Sasha Rubin and Frank Stephan. Automatic structures: richness and limitations. *Logical Methods in Computer Science*, 3(2), 2007.
22. Bakhadyr Khoussainov, André Nies, Sasha Rubin and Frank Stephan. Automatic structures: richness and limitations. *Proceedings of the 19th IEEE Symposium on Logic in Computer Science* (LICS 2004), pages 44-53. IEEE Computer Society, 2004.
23. Bakhadyr Khoussainov, Sasha Rubin and Frank Stephan. Automatic linear orders and trees. *ACM Transactions on Computational Logic*, 6(4), 675-700, 2005.
24. Dietrich Kuske. Isomorphisms of scattered automatic linear orders. *Theoretical computer science*, 533, 46-63, 2014.

25. Dietrich Kuske, Jiamon Liu, Markus Lohrey. The isomorphism problem for $\omega$-automatic trees. *Annals of Pure and Applied Logic*, 164(1), 30-48, 2013.
26. Dietrich Kuske, Jiamon Liu, Markus Lohrey. The isomorphism problem for classes of automatic structures. *Proceedings of the 25th IEEE Symposium on Logic in Computer Science* (LICS 2010), pages 160-169. IEEE Computer Society, 2010.
27. Dietrich Kuske, Jiamon Liu, Markus Lohrey. The isomorphism problem on classes of automatic structures with transitive relations. *Transactions of the American Mathematical Society*, 365, 5103–5151, 2013.
28. Dietrich Kuske and Thomas Weidner. Size and Computation of Injective Tree Automatic Presentations. *Mathematical Foundations of Computer Science* (MFCS 2011), LNCS 6907, pages 424-435. Springer, Berlin, 2011.
29. Itay Neeman. Monadic theories of wellorders. *Logic, Methodology and Philosophy of Science*, Proceedings of the Thirteenth International Congress, pages 108-121. College Publications, 2009.
30. André Nies and Pavel Semukhin. Finite automata presentable Abelian groups. *Annals of Pure and Applied Logic*, 161, 458–467, 2009.
31. Michael O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *Transcations of the American Mathematical Society*, 141, 1–35, 1969.
32. Joseph G. Rosenstein. *Linear Orderings.* Academic Press, New York, 1982.
33. Sasha Rubin. Automata presenting structures: a survey of the finite string case. *The Bulletin of Symbolic Logic*, 14(2), 169–209, 2008.
34. Sasha Rubin. Automatic Structures. PhD thesis, The University of Auckland, 2004.
35. Philipp Schlicht and Frank Stephan. Automata on ordinals and automaticity of linear orders. *Annals of Pure and Applied Logic*, 5, 523-527, 2013.
36. Todor Tsankov. The additive group of the rationals does not have an automatic presentation. *The Journal of Symbolic Logic*, 76, 1341-1351, 2011.