Theoretical Computer Science 626 (2016) 83–96

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Tree-automatic scattered linear orders

Sanjay Jain [a,*,1], Bakhadyr Khoussainov [b,*,2], Philipp Schlicht [c,*], Frank Stephan [d,*,3]

[a] Department of Computer Science, National University of Singapore, Singapore 117417, Republic of Singapore
[b] Department of Computer Science, The University of Auckland, Auckland, New Zealand
[c] Mathematical Institute, University of Bonn, Endenicher Allee 60, 53115 Bonn, Germany
[d] Department of Mathematics, National University of Singapore, Singapore 119076, Republic of Singapore

ARTICLE INFO

Article history:
Received 5 August 2014
Received in revised form 4 February 2016
Accepted 9 February 2016
Available online 12 February 2016
Communicated by Z. Esik

Keywords:
Tree automata
Linear orders
Automatic structures

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.

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 [4,9,15,19–24,30,36]. 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 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 ω^{ω^ω}. Our approach can be understood as a refinement of the work of Delhommé leading to an alternative proof of his result in Theorem 26.

* Corresponding authors.
E-mail addresses: sanjay@comp.nus.edu.sg (S. Jain), bmk@cs.auckland.ac.nz (B. Khoussainov), schlicht@math.uni-bonn.de (P. Schlicht), fstephan@comp.nus.edu.sg (F. Stephan).

1 Supported in part by NUS grants R252-000-420-112, R146-000-181-112 and C252-000-087-001.
2 Supported in part by the Marsden Fund of New Zealand.
3 Supported in part by NUS grants R146-000-114-112, R146-000-181-112 and R252-000-420-112.

http://dx.doi.org/10.1016/j.tcs.2016.02.008
0304-3975/

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 Σ. 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 Σ-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 ω^{ω^ω} 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 β, 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 α-th derivative of $\mathcal{L}$.

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

The least ordinal α 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 Σ and $\Sigma^\star$ denotes the set of all finite strings (finite words) over Σ. Let $|\sigma|$ denote the length of a string σ. Let λ denote the empty string. Let $\sigma \preceq \tau$ denote that string σ is a prefix of string τ.

A *finite automaton* over the alphabet Σ 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 $w = a_1 a_2 \ldots a_n$ (where $a_1, a_2 \ldots, a_n$ are members of Σ) is a sequence of states $q_0, q_1, \ldots, q_n$ such that $q_0 = \iota$ and $(q_i, a_{i+1}, q_{i+1}) \in \Delta$ 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 $L(\mathcal{M}) = \{w \mid w$ is accepted by $\mathcal{M}\}$. These languages are called *regular*, *word-automatic*, or *finite automaton recognisable*.

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 σ is a *leaf* of a tree T if σ belongs to T but no proper extension of σ belongs to T. Similarly, σ is an *internal node* of T if σ as well as some proper extension of σ belongs to T. If σ and σa both belong to T (where $a \in \{0, 1\}$), then σa is called a *child* of σ and σ is called the *parent* of σa. A node σ is a *branching node* of T if σ 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 Σ. We say that a (labeled or unlabeled) tree S *embeds* into a tree T if there is an injective map $h: S \to T$ 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 Σ-*tree* is a mapping $t : dom(t) \to \Sigma$ with domain $dom(t)$ being a finite tree such that for every non-leaf node $v \in dom(t)$ we have $v0, v1 \in dom(t)$.[4] The *boundary* of $dom(t)$ is the set $\partial dom(t) = \{xb \mid x$ is a leaf of $dom(t)$ and $b \in \{0, 1\}\}$. The set of all Σ-trees is denoted by $T(\Sigma)$. A *slim* Σ-*tree* is a Σ-tree T such that the branching nodes in T are all pairwise

[4] There are various alternative definitions of Σ-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.