THE ISOMORPHISM PROBLEM FOR TREE-AUTOMATIC ORDINALS WITH ADDITION

SANJAY JAIN, BAKHADYR KHOUSSAINOV,
PHILIPP SCHLICHT AND FRANK STEPHAN

Abstract. This paper studies tree-automatic ordinals (or equivalently, well-founded linearly ordered sets) together with the ordinal addition operation +. Informally, these are ordinals such that their elements are coded by finite trees for which the linear order relation of the ordinal and the ordinal addition operation can be determined by tree automata. We describe an algorithm that, given two tree-automatic ordinals with the ordinal addition operation, decides if the ordinals are isomorphic.

1. Motivation of the problem

Delhommé proved that any ordinal presented by finite automata is strictly less than ω^ω [6]. Ordinals presented by automata are called word-automatic ordinals; precise definitions will be provided in the next section and can be found in the relevant literature [1, 3, 6, 7, 8, 24, 25]. Later Khoussainov, Rubin and Stephan proved that the Cantor-Bendixson rank of any word-automatic ordinal is finite [17]. In particular, this implies that any word-automatic ordinal is strictly less than ω^ω. Khoussainov and Minnes generalized these results by proving that the height of any word-automatic well-founded partially ordered set is below ω^ω [14]. The proofs of these results show that there is an algorithm that, given a word-automatic linear order, decides if the linear order is an ordinal [17]. Moreover, there exists an algorithm that, given two word-automatic ordinals, decides if the ordinals are isomorphic. The decidability of the isomorphism problem for word-automatic ordinals is obtained by extracting the Cantor Normal Form from the given word-automatic ordinals. In contrast, Kuske, Liu and Lohrey proved that the isomorphism problem for word-automatic linearly ordered sets is undecidable [19, 20] and also obtained similar results for ω-automatic trees and other such structures [21].

The results above naturally lead to the study of ordinals presented by tree automata (such ordinals we call tree-automatic ordinals). Delhommé proved that any tree-automatic ordinal is strictly less than ω^{ω^ω} [6]. Jain, Khoussainov, Schlicht and Stephan [12] connected tree-automatic ordinals with automata working on ordinal words [26] and provided an alternative proof of Delhommé's result. However, in contrast to word-automatic ordinals, it is unknown if the isomorphism problem for tree-automatic ordinals is decidable. We address this isomorphism problem in the current paper. We prove two main results. First, we show an ordinal α together with the ordinal addition operation + is tree-automatic if and only if $\alpha < \omega^{\omega^\omega}$. For the proof, by the above mentioned result by Delhommé, it suffices to show that all ordinals $\alpha < \omega^{\omega^\omega}$ with the ordinal addition operation are tree-automatic. We show that

Date: March 15, 2019.

the natural tree-representations of such ordinals preserve tree-automaticity of the addition operation. We further provide an algorithm that, given two tree-automatic ordinals with the addition operation, decides if the ordinals are isomorphic. Just like in the case of word-automatic ordinals, the proof is based on extracting Cantor Normal Forms from tree automata that represent ordinals with addition. The Cantor Normal Form is based on Cantor's result that every ordinal is the sum of a finite descending list of ω-powers [4, 5].

2. Basic definitions

This section gives necessary definitions and background to tree automata and tree-automatic structures. Let $\{0,1\}$ be the binary alphabet and $\leq_p$ be the prefix order on finite binary strings. If $x \leq_p y$ and $x \neq y$ then we say that y properly extends x. By a tree we mean a finite subset X of $\{0,1\}^\star$ such that (1) X is closed under the prefix relation, (2) for every $x \in X$ either no $y \in X$ properly extends x or both $x0$ and $x1$ belong to X.

Let X be a tree and $x \in X$. If no string in X properly extends x then we say that x is a leaf of the tree. In case $x \in X$ and x is not a leaf then x is called an internal node of X; in this case, the node $x0$ is the left-child of x and $x1$ is the right-child of x. For the internal node $x \in X$, both children of x belong to X. The null string λ belongs to any non-empty tree. The empty string is called the root of the tree. Finally, a maximal linearly ordered set of internal nodes of X is called a branch.

For a finite alphabet Σ, a Σ-tree is a function $t : X \to \Sigma$ where X is a tree. In this case the domain of t, denoted by $dom(t)$, coincides with the set X. We denote the set of all Σ-trees by T_Σ. A Σ-tree language (or simply a language) is any set of Σ-trees. Thus, Σ-tree languages are simply subsets of T_Σ.

Definition 2.1. A tree automaton $\mathcal{M}$ is a tuple (S, δ, I, F), where S is a finite set of states, $\delta : S \times \Sigma \to P(S \times S)$ is the transition table, $I \subseteq S$ is the set of initial states and $F \subseteq S$ that of accepting states. These tree automata are sometimes called top-down automata.

Given a tree automaton $\mathcal{M}$ and a Σ-tree t, one naturally defines the notion of a run of $\mathcal{M}$ on the tree t. Namely, a run of $\mathcal{M}$ on t is a function $r : dom(t) \to S$ such that the following conditions are satisfied:

(1) The run starts at the root with an initial state. Namely, $r(\lambda) \in I$.
(2) The run is consistent with the transition table. Namely, for all internal nodes $x \in dom(t)$, if $r(x) = s$ and $t(x) = \sigma$ then $(r(x0), r(x1)) \in \delta(s, \sigma)$.

Note that there could be several runs of $\mathcal{M}$ on any given Σ-tree t. For such a run r, if $r(x) \in F$ for all leaves of the tree $dom(t)$ then we say r is an accepting run of $\mathcal{M}$ on t. We say that $\mathcal{M}$ accepts the tree t if $\mathcal{M}$ has an accepting run on t. Define

$$L(\mathcal{M}) = \{t \mid \text{the automaton } \mathcal{M} \text{ accepts } t\}.$$

We call $L(\mathcal{M})$ the language of the automaton $\mathcal{M}$ or the language recognised by the automaton $\mathcal{M}$.

Note that our restrictive definition of trees above simplifies the transition table in the previous definition, since one need not consider the case of nodes with only one direct successor.

Definition 2.2. A Σ-tree language L is called regular if there exists an automaton $\mathcal{M}$ such that L is the language of the automaton $\mathcal{M}$, that is, $L = L(\mathcal{M})$.

It is well-known that the class of regular Σ-tree languages forms a Boolean algebra under the set-theoretic boolean operations. Tree automata also satisfy an important decidability property that solves the emptiness problem. Namely, there exists an algorithm (that runs in linear time on the sizes of input automata) that, given an automaton $\mathcal{M}$, decides if $\mathcal{M}$ accepts at least one Σ-tree.

Note that a unary relation on T_Σ is recogised by tree automata if and only if it is regular. In order to define n-ary relations that are recognised by tree automata, we need to define the convolution operation on n-tuples of Σ-trees. Here we describe the convolution operation for pairs (t_1, t_2) of Σ-trees; the convolution of n-tuples of Σ-trees can easily be defined in a similar way.

Definition 2.3. Let $\diamond$ be a symbol that does not belong to Σ. Given Σ-trees t_1 and t_2, the convolution $conv(t_1, t_2)$ is a function with domain $dom(t_1) \cup dom(t_2)$ such that for all $x \in dom(t_1) \cup dom(t_2)$ we have the following properties:

(1) If $x \in domt(t_1) \cap dom(t_2)$ then $conv(t_1, t_2)(x) = (t_1(x), t_2(x))$.
(2) If $x \in dom(t_1) \smallsetminus dom(t_2)$ then $conv(t_1, t_2)(x) = (t_1(x), \diamond)$.
(3) If $x \in dom(t_2) \smallsetminus dom(t_1)$ then $conv(t_1, t_2)(x) = (\diamond, t_2(x))$.

This definition allows us to define tree automata recognisable n-ary relations on the set of all Σ-trees:

Definition 2.4. An n-ary relation R on the set T_Σ is tree automata recognisable (or tree-automatic) if there exists a tree automaton $\mathcal{M}$ that recognises the convolution $conv(R)$ of the relation R, that is:

$$L(\mathcal{M}) = \{conv(t_1, t_2, \ldots, t_n) \mid (t_1, t_2, \ldots, t_n) \in R\}.$$

Now we connect the notion of tree automata recognisability with algebraic structures. Recall that a relational algebraic structure is a first order structure of the form

$$\mathcal{A} = (A; R_1^{n_1}, \ldots, R_k^{n_k}),$$

where A is a non-empty set (called the domain of $\mathcal{A}$) and for each $i = 1, \ldots, k$, $R_i^{n_i}$ is a relation of arity n_i on the domain A. These relations are called atomic relations of the structure. Note that any structure with operation symbols can be transformed into a relational structure by replacing each atomic operation with its graph. So, we identify such structures with their relational counterparts as just described.

In this paper, our interest is in ordinals with addition. We adopt the convention to identify each ordinal with the set of its predecessors; ordinals are further (implicitly) identified with isomorphism types of well-founded linearly ordered sets. The ordinal ω^ω with the addition operation can be viewed as the algebraic structure

$$(\omega^\omega;\ \leq, +),$$

where ω^ω equals the set of all ordinals strictly less than ω^ω, $\leq$ is the natural linear order relation $\in$ on ordinals (that is, $\alpha < \beta$ iff $\alpha \in \beta$ for ordinals α and β) and $+$ is the addition operation of ordinals. Here the addition operation on an ordinal δ is identified with its graph $\{(\alpha, \beta, \gamma) \mid \alpha, \beta, \gamma < \delta \ \&\ \alpha + \beta = \gamma\}$. We note that not every ordinal is closed under addition. So, + is a binary partial operation.

The following definition now connects algebraic structures with tree automata:

Definition 2.5. An algebraic structure $\mathcal{A} = (A; R_1^{n_1}, \ldots, R_k^{n_k})$ is tree-automatic if the domain $A \subseteq T_\Sigma$ and the atomic relations $R_1^{n_1}, \ldots, R_k^{n_k}$ on A are all recognised by tree automata. Any tuple of tree-automata that recognise A, $R_1^{n_1}$, …, $R_k^{n_k}$ is called a (tree-automatic) representation of the structure.

If an algebraic structure is isomorphic to a tree-automatic structure, then we say that it is tree-automata presentable. Since we are interested in structures up to isomorphism, we abuse our definition and often call tree-automata presentable structures tree-automatic structures. This will be clear from the context. For this paper we use the following foundational theorem in the theory of automatic structures [2, 9, 10, 15]. The proof of the theorem follows from the closure properties of regular tree-languages and the decidability of the emptiness problem for tree automata:

Theorem 2.6. *There exists an algorithm that given a tree-automatic structure $\mathcal{A}$ and a first order formula $\phi(x_1, \ldots, x_n)$ with n variables x_1, …, x_n, constructs a tree automaton $\mathcal{M}_\phi$ such that $\mathcal{M}_\phi$ accepts a tuple $(t_1, \ldots, t_n)$ if and only if $\mathcal{A} \models \phi(t_1, \ldots, t_n)$. In particular, the first order theory of any tree-automatic structure is decidable.* □

Finally, we refer the reader to the articles of Blumensath and Grädel [2], Delhommé [6], Khoussainov and Minnes [14], Khoussainov and Nerode [16], Kuske [18] and Rubin [24, 25] for the background and open questions in the area of automatic structures. For simple and non-trivial examples of word-automatic structures we refer to the articles of Ishihara, Khoussainov and Rubin [11], Nies [22], Nies and Thomas [23] and Stephan [28].

3. Tree-automatic ordinals with addition

Our main result in this section is the following theorem.

Theorem 3.1. *Let α be an ordinal such that $\alpha < \omega^{\omega^\omega}$. Then the structure $(\alpha; \leq, +)$ of the ordinal α together with ordinal addition is a tree-automatic structure.*

Proof. For the reader and ease of understanding, we provide some background and intuition for ordinals $\alpha \leq \omega^{\omega^\omega}$. These will also be needed for the next theorem. The ordinal ω^ω can be viewed as the sum:

$$\omega^\omega = \omega + \omega^2 + \ldots + \omega^n + \ldots.$$

The ordinals ω^{ω^n}, where $n > 1$, are defined by induction through the supremum of the sequence of ordinals

$$\omega^{\omega^{n-1}},\ \omega^{\omega^{n-1}} \cdot \omega^{\omega^{n-1}},\ \omega^{\omega^{n-1}} \cdot \omega^{\omega^{n-1}} \cdot \omega^{\omega^{n-1}},\quad \omega^{\omega^{n-1}} \cdot \omega^{\omega^{n-1}} \cdot \omega^{\omega^{n-1}} \cdot \omega^{\omega^{n-1}},\ \ldots.$$

So, the ordinal ω^{ω^ω} can be viewed as the sum:

$$\omega^{\omega^\omega} = \omega^\omega + \omega^{\omega^2} + \omega^{\omega^3} + \ldots + \omega^{\omega^n} + \ldots.$$

Let $p(X)$ be a notation for polynomials with non-negative integer coefficients. We represent these polynomials as

$$p(X) = X^n a_n + X^{n-1} a_{n-1} + \ldots + X a_1 + a_0,$$

where $a_n > 0$ whenever $p(X) \neq 0$. Here we explicitly wrote the coefficients on the right side of the variables, since the multiplication and addition operations on ordinals are not commutative. For any ordinal $\alpha < \omega^{\omega^\omega}$, there are unique polynomials $p_0(X)$, …, $p_k(X)$ and integer coefficients c_0, …, c_k with $c_k > 0$ such that

- $\alpha = \omega^{p_0(\omega)}c_0 + \omega^{p_1(\omega)}c_1 + \ldots + \omega^{p_{k-1}(\omega)}c_{k-1} + \omega^{p_k(\omega)}c_k$ and
- $p_0(\omega) > p_1(\omega) > \ldots > p_k(\omega)$.

When adding ordinals described in the form above, one takes into account the following equations from ordinal arithmetic:

$$\omega^\alpha m + \omega^\alpha n = \omega^\alpha(m+n), \text{ and } \omega^\alpha + \omega^\beta = \omega^\beta,$$

where m and n are natural numbers and $\alpha < \beta$ are ordinals. For instance,

$$(\omega^{\omega^3}4 + \omega^{\omega^2}7 + \omega^6 3 + \omega^2 + 1) + (\omega^{\omega^2}2 + \omega^6 3 + \omega^5 + 5) = \omega^{\omega^3}4 + \omega^{\omega^2}9 + \omega^6 3 + \omega^5 + 5.$$

To prove the desired result, it suffices to show that each ordinal ω^{ω^n} with the ordinal addition operation + is a tree-automatic structure. This yields a representation of $(\alpha;\ \leq, +)$ for any ordinal $\alpha < \omega^{\omega^\omega}$, since the class of tree-automatic structures is closed under products. The proof is by induction on n.

We first explain a tree-automatic presentation of ω^ω. Note that every non-null ordinal $\alpha < \omega^\omega$ can be uniquely written as

$$\omega^m b_m + \ldots + \omega^1 b_1 + \omega^0 b_0,$$

where $b_m > 0$ and the coefficients $b_m, \ldots, b_0$ are natural numbers. The integer m is called the degree of α. We represent this α as the following $\{0,1\}$-tree t_α:

1. The leftmost branch of t_α has length $m+1$ and nodes $v_i = 0^i$ for $i \leq m$.
2. For each i, the rightmost branch containing v_i has length k_i above v_i and nodes $w_j = v_i^\frown 1^j$ for $j \leq k_i$ and codes the coefficient $b_i \in [2^{k_i}, 2^{k_i+1})$ in binary format, beginning with the least-significant bit.
3. $dom(t_\alpha)$ does not contain any more nodes than implied by (1) and (2), and $t_\alpha(s) = 0$ for all $s \in dom(t_\alpha)$ not mentioned there.

Let L_1 be the set of all $\{0,1\}$-trees t_α that represent ordinals $\alpha < \omega^\omega$. It is clear that L_1 is a regular tree language. It is not too hard to see that there exist tree automata $\mathcal{M}_\leq^{(1)}$ and $\mathcal{M}_+^{(1)}$ such that, given trees t_α, t_β and t_γ representing ordinals $\alpha, \beta, \gamma < \omega^\omega$, the automata $\mathcal{M}_\leq^{(1)}$ and $\mathcal{M}_+^{(1)}$ verify that $\alpha \leq \beta$ and $\alpha + \beta = \gamma$. For instance, $\mathcal{M}_\leq^{(1)}$ by reading $conv(t_\alpha, t_\beta)$ non-deterministically makes one of the following choices:

1. The automaton guesses and verifies that the degree of β is greater than that of α. In this case $\mathcal{M}_\leq^{(1)}$ accept the convolution $conv(t_\alpha, t_\beta)$.
2. The automaton guesses and verifies that both α and β have the same degree, and then compares the coefficients of α and β lexicographically.

Similarly, using the properties of the addition operation stated above, one can describe the desired automaton $\mathcal{M}_+^{(1)}$.

1. The automaton guesses at each internal node v_i of $dom(t_\alpha)$ whether the Cantor Normal Form of β contains a non-vanishing multiple of ω^k for some $k > i$.
2. For any $i < k$ such that the guess in (1) is positive, we take as coefficient of ω^i in the sum only the one appearing in the representation of β; when the guess is negative, we take the sum of the coefficients of ω^i appearing in α and β.

So, ω^ω with the addition operation is a tree-automatic structure.

Suppose now that we have a regular tree language L_n, that represents all ordinals $\alpha < \omega^{\omega^n}$, and tree automata $\mathcal{M}_\leq^{(n)}$ and $\mathcal{M}_+^{(n)}$ such that, given trees t_α, t_β and t_γ

representing ordinals $\alpha, \beta, \gamma < \omega^{\omega^n}$, the automata $\mathcal{M}_{\le}^{(n)}$ and $\mathcal{M}_{+}^{(n)}$ verify that $\alpha \le \beta$ and $\alpha + \beta = \gamma$.

Elements α of $\omega^{\omega^{n+1}}$ can be identified with finite tuples $(\alpha_0, \dots, \alpha_k)$, where each α_i, $i = 1, \dots, k$, is an element of ω^{ω^n}. Intuitively, the tuple $(\alpha_0, \dots, \alpha_k)$ represents the ordinal $(\omega^{\omega^n})^k \times (\alpha_k) + (\omega^{\omega^n})^{k-1} \times \alpha_{k-1} + \dots + (\omega^{\omega^n}) \times \alpha_1 + (\omega^{\omega^n})^0 \times \alpha_0$. The order between such tuples is given by length-lexicographic order of $(\alpha_k, \dots, \alpha_0)$, when α_k is non-zero or $k = 0$. We represent these tuples α as the following binary tree:

(1) The leftmost branch of t_α has length $k+1$ and nodes $v_i = 0^i$ for $i \le k$.
(2) For each $i \le k$, the right subtree of v_i is a copy of t_{α_i} with its root copied to the right child of v_i.
(3) $dom(t_\alpha)$ does not contain any more nodes than implied by (1) and (2) and $t_\alpha(s) = 0$ for all $s \in dom(t_\alpha)$ not mentioned there.

Let L_{n+1} be the tree language consisting of trees as we described above. The tree language L_{n+1} is a regular tree language. So, the trees from L_{n+1} represent elements of the ordinal $\omega^{\omega^{n+1}}$. Now, using the tree automata $\mathcal{M}_{\le}^{(n)}$ and $\mathcal{M}_{+}^{(n)}$ as subroutines, it is not too hard to construct tree automata $\mathcal{M}_{\le}^{(n+1)}$ and $\mathcal{M}_{+}^{(n+1)}$ that recognise the order relation and the addition operation of the ordinal $\omega^{\omega^{n+1}}$. The proof is similar to the case of $\mathcal{M}_{\le}^{(1)}$ and $\mathcal{M}_{+}^{(1)}$ above. □

Remark 3.2. The anonymous referees asked whether there exist ordinals α for which $(\alpha; \le, +)$ is not automatic while $(\alpha; \le)$ is automatic. Of course, if there are $\beta, \gamma < \alpha$ with $\beta + \gamma \ge \alpha$ then the ordinal α can treat the addition only as an automatic relation $\{(\beta, \gamma, \delta) : \beta, \gamma, \delta < \alpha \wedge \beta + \gamma = \delta\}$ and not as a function, as the latter would have undefined places in the case that the sum exceeds α. For these ordinals, one has that if $(\alpha; \le)$ has an automatic presentation then $(\alpha; \le, +)$ has also an automatic presentation, which might be different from the previous one, as seen above. However, if $\omega \le \alpha < \omega^{\omega^\omega}$, there is a further presentation where $(\alpha; \le)$ is automatic but the addition not. The main idea would be to code the coefficients of the Cantor normal form not in binary as done above, but in unary; then there are trees where the height of the sum is approximately twice that of the original ordinals and an easy application of the pumping lemma shows that such a function cannot be tree-automatic. The natural numbers coded in unary are a well-known example of a word-automatic structure where order and remainders are automatic while the addition is not.

4. Deciding the isomorphism problem

The goal of this section is to prove the following decision theorem that solves the isomorphism problem for tree-automatic ordinals with the addition operation. For this, we need to handle addition on sets which are not closed under addition. Therefore we recall that $(\alpha; +, <)$ is a tree-automatic ordinal with addition iff $(\alpha; <)$ is a tree-automatic ordinal and + is a partial tree-automatic function with tree-automatic domain such that for $\beta, \gamma \in \alpha$, $\beta + \gamma$ is defined and takes as value the ordinal sum of β and γ iff that sum is a member of α, that is, iff it is strictly less than α.

Theorem 4.1. *There exists an algorithm that, given two tree automatic ordinals with the addition operation, decides if the ordinals are isomorphic.*

Proof. Recall that, by Delhommé's result mentioned in the introduction, if $(\alpha; \le, +)$ is a tree-automatic ordinal, then $\alpha < \omega^{\omega^\omega}$. We will be using this fact implicitly. The following well-known claim (see e.g. [13, 27]) shows that we can use the addition operation + to identify powers of the ordinal ω. For the present work, we only consider such powers of the ordinal ω which are of the form $\omega^{p(\omega)}$, where $p(X)$ is a polynomial; larger powers of ω like ω^{ω^ω} do not need to be considered, as they are not tree-automatic.

Claim 4.2. *An ordinal $\beta < \omega^{\omega^\omega}$ is closed under the addition operation + if and only if β is a power of the ordinal ω.*

Indeed, if β is of the form $\omega^{p(\omega)}$, where $p(X)$ is a polynomial, then it is easy to check that β is closed under the addition operation +. Otherwise, β is of the form

$$\beta = \omega^{p(\omega)}c + \beta',$$

where $0 < \beta' < \omega^{p(\omega)}$ and $c > 0$, or $\beta' = 0$ and $c > 1$. So, $\omega^{p(\omega)} < \beta$ yet $\omega^{p(\omega)}(c+1) \geq \beta$. Therefore, β is not closed under the addition operation.

So, let $(\alpha;\ \leq, +)$ be a tree-automatic ordinal with the addition operation +. Consider the following set:

$$P_\alpha = \{\beta \leq \alpha \mid \beta \neq 0 \ \&\ \forall\gamma\forall\gamma'(\gamma < \beta \ \&\ \gamma' < \beta \to \gamma + \gamma' \text{ is defined and } \gamma + \gamma' < \beta)\}.$$

The set P_α is a regular tree language by Theorem 2.6 (since it is first order definable) and the order $\leq$ restricted to P_α is an ordinal; we will identify P_α with this ordinal. Our second claim is the following:

Claim 4.3. *The ordinal P_α is a tree-automatic ordinal strictly less than ω^ω.*

To see this, use that the set P_α consists of the powers of ω less or equal to α by Claim 4.2. If the ordinal P_α equals γ, then $\omega^\gamma \leq \alpha$. Since $\alpha < \omega^{\omega^\omega}$, this implies $\gamma < \omega^\omega$.

Claim 4.4. *Given a tree-automatic ordinal α, we can effectively compute the Cantor Normal Form of the ordinal P_α.*

Here our argument is the same as in the word-automatic case for ordinals [17].

By Theorem 2.6, there is an algorithm that, when given as input any tree-automatic presentation of $(\alpha; +, \leq)$, computes a tree-automatic presentation of the ordinal $(P_\alpha; \leq)$. Let γ_0 denote this ordinal. It is strictly less than ω^ω by Claim 4.3.

We will iterate the following step, beginning with $k = 0$, until the resulting ordinal is the empty set. First we compute the number b_k of steps which one needs to remove the largest element of γ_k until the resulting ordinal is a limit ordinal, that is, has no largest element. Next we let γ_{k+1} be the subset of γ_k of all limit ordinals in γ_k, that is, of all ordinals δ which satisfy that there are smaller ordinals than δ, but among those there is no largest one. Now we let $k = k + 1$ and iterate this process.

For each k it holds that $\gamma_k = \omega \cdot \gamma_{k+1} + b_k$. Thus, in the Cantor Normal Form, the degree of γ_{k+1} is one below that of γ_k and b_k is the last constant part of the Cantor Normal Form of γ_k. Iterating this gives that the Cantor Normal Form of P_α is $\omega^h \cdot b_h + \ldots + \omega \cdot b_1 + b_0$ and h is the least value where γ_{h+1} is the empty set.

Since the procedure is effective by Theorem 2.6, this proves Claim 4.4.

Claim 4.5. *Given a tree-automatic ordinal with addition $(\alpha; \leq, +)$, we can effectively compute a tree-automatic ordinal with addition $(\alpha'; \leq, +)$ such that $\alpha = \omega^{P_\alpha} + \alpha'$ and this procedure allows to compute the Cantor Normal Form of α.*

The ordinal ω^{P_α} is equal to α in the case that α is closed under + and in this case, $\alpha' = 0$. Otherwise ω^{P_α} is the smallest ordinal $\beta \in \alpha$ such that there is no γ with $\beta < \gamma < \alpha$ which is closed under +; this β exists by the well-orderedness of α. Now we define $\alpha' = \{\gamma \in \alpha : \beta + \gamma \text{ is defined and in } \alpha\}$ and inherit the addition from α to α' as follows: for $\gamma, \delta \in \alpha'$, we check whether $\gamma + \delta$ and $\beta + (\gamma + \delta)$ are both defined in $(\alpha; \leq, +)$; if so, then we let $\gamma + \delta$ have in $(\alpha'; \leq, +)$ the same value as in $(\alpha; \leq, +)$, else we let $\gamma + \delta$ be undefined in $(\alpha'; \leq, +)$. We note that $\alpha' < \alpha$, as P_α is the exponent of the highest ω-power with nonzero coefficient in the Cantor Normal Form of α and this coefficient is then in α' at least by one smaller.

Now we let $\alpha_0 = \alpha$ and $\ell = 0$. While α_ℓ is not an ω-power, we update $\alpha_{\ell+1} = \alpha'_\ell$ and $\ell = \ell + 1$. Now let ℓ be the final value of this process and $\alpha_0, \alpha_1, \ldots, \alpha_\ell$ as defined on the way. It is easy to see by induction that the ordinals in this sequence form a strictly descending chain and that

$$\alpha = \sum_{k=0}^{\ell} \omega^{P_{\alpha_k}} = \omega^{P_{\alpha_0}} + \omega^{P_{\alpha_1}} + \ldots + \omega^{P_{\alpha_\ell}}.$$

This is just due to the fact that for the Cantor Normal Form $\sum_{k=0}^{h} \omega^{\beta_k}$ of α, by induction for every k the Cantor Normal Form of α_k equals $\sum_{k'=k}^{h} \omega^{\beta_{k'}}$ and thus $P_{\alpha_k} = \beta_k$. It follows that $\ell = h$, since h is least such that ω^{β_h} is an ω-power. We can effectively determine ℓ and the Cantor Normal Form of all P_{α_k} by Claim 4.4. We can therefore also effectively determine the Cantor Normal Form of α by counting equal ω-powers in the last equation.

Claim 4.6. *The isomorphism problem of tree-automatic ordinals with addition is decidable.*

This follows from Claim 4.5. Suppose that two tree-automatic ordinals with addition $(\alpha; \leq, +)$ and $(\beta; \leq, +)$ are given. Using the process above, we produce the Cantor Normal Forms for both of these ordinals. The ordinals are isomorphic if and only if the two Cantor Normal Forms produced coincide. ☐

Acknowledgments. The authors would like to thank Dietrich Kuske for very helpful correspondence and Peter Koepke for discussions. Part of this work was done while F. Stephan visited the University of Bonn during his sabbatical leave and while P. Schlicht was working there. S. Jain was supported in part by NUS grant C252-000-087-001; B. Khoussainov was supported in part by the Marsden Fund grant of the Royal Society of New Zealand; S. Jain, B. Khoussainov and F. Stephan have been supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2016-T2-1-019 / R146-000-234-112; furthermore, P. Schlicht was supported by a Marie Skłodowska-Curie Individual Fellowship with number 794020.

References

[1] Vince Bárány, Erich Grädel and Sasha Rubin. Automata-based presentations of infinite structures. In Javier Esparza, Christian Michaux and Charles Steinhorn (eds), *Finite and Algorithmic Model Theory*, London Mathematical Society Lecture Note Series, no. 379, pp. 1–76, 2011.

[2] Achim Blumensath and Erich Grädel, Automatic structures, Proceedings *Logic in Computer Science*, LICS 2000, pp. 51–62, 2000.

[3] Achim Blumensath and Erich Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory of Computing Systems*, 37 (6), pp. 641–674, 2004.
[4] Georg Cantor. Beiträge zur Begründung der transfiniten Mengenlehre [Contributions to the foundations of transfinite set theory]. *Mathematische Annalen*, 46, pp. 481–512, 1895.
[5] Georg Cantor. Beiträge zur Begründung der transfiniten Mengenlehre [Contributions to the foundations of transfinite set theory]. *Mathematische Annalen*, 49, pp. 207–246, 1897.
[6] Christian Delhommé. Automaticité des ordinaux et des graphes homogènes [Automaticity of ordinals and of homogeneous graphs]. *Comptes Rendus Mathematique*, 339 (1), pp. 5–10, 2004.
[7] Olivier Finkel and Stevo Todorčević. A hierarchy of tree-automatic structures. *The Journal of Symbolic Logic*, 77 (1), pp. 350–368, 2012.
[8] Olivier Finkel and Stevo Todorčević. Automatic ordinals. *International Journal of Unconventional Computing*, 9 (1–2), pp. 61–70, 2013.
[9] Bernard R. Hodgson. *Théories décidables par automate fini* [Decidable theories via finite automata]. Ph.D. thesis, Département de Mathématiques et de Statistique, Université de Montréal, 1976.
[10] Bernard R. Hodgson. Décidabilité par automate fini [Decidability via finite automata]. *Annales des sciences mathématiques du Québec*, 7 (1), pp. 39–57, 1983.
[11] Hajime Ishihara, Bakhadyr Khoussainov and Sasha Rubin. On isomorphism invariants of some automatic structures. *Proceedings of Logic in Computer Science*, LICS 2002, pp. 235–244, 2002.
[12] Sanjay Jain, Bakhadyr Khoussainov, Philipp Schlicht and Frank Stephan. Tree-automatic scattered linear orders. *Theoretical Computer Science*, 626, pp. 83–96, 2016.
[13] Thomas Jech. *Set Theory: Third Millennium Edition.* Springer Monographs in Mathematics, 2003.
[14] Bakhadyr Khoussainov and Mia Minnes. Three Lectures on Automatic Structures. *Proceedings of the ASL Logic Colloquium 2007, Lecture Notes in Logic* 35, F. Delon et al (eds), pp. 132–176, 2010.
[15] Bakhadyr Khoussainov and Anil Nerode. Automatic Presentations of Structures. D. Lievant (editor) *Proceedings, conference on Logic and Computational Complexity*, pp. 367–393, 1994.
[16] Bakhadyr Khoussainov and Anil Nerode. Open Questions in the Theory of Automatic Structures. *Bulletin of the European Association of Theoretical Computer Science* (EATCS), no. 94, pp. 181–204, 2008.
[17] Bakhadyr Khoussainov, Sasha Rubin and Frank Stephan. Automatic linear orders and trees. *ACM Transactions on Computational Logic* 6 (4), pp. 675–700, 2005.
[18] Dietrich Kuske. Is Cantor's theorem automatic? *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, 22 September 2003, pp. 332–345. Springer, Berlin, Heidelberg, 2003.
[19] Dietrich Kuske. Isomorphisms of scattered automatic linear orders. *Theoretical Computer Science* 533, pp. 46–63, 2014.
[20] Dietrich Kuske, Jiamou Liu and Markus Lohrey. The Isomorphism Problem on Classes of Automatic Structures. *Proceedings of Logic in Computer Science*, LICS 2010, pp. 160–169, 2010.
[21] Dietrich Kuske, Jiamou Liu and Markus Lohrey. The isomorphism problem for ω-automatic trees. *Annals of Pure and Applied Logic*, 164 (1), pp. 30–48, 2013.
[22] André Nies. Describing groups. *Bulletin of Symbolic Logic*, 13 (3), pp. 305–339, 2007.
[23] André Nies and Richard M. Thomas. FA-presentable groups and rings. *Journal of Algebra*, 320 (2), pp. 569–585, 2008.
[24] Sasha Rubin. Automata presenting structures: A survey of the finite string case. *Bulletin of Symbolic Logic*, 14 (2), pp. 169–209, 2008.
[25] Sasha Rubin. Automatic structures. In Jean-Éric Pin (ed), *Handbook of Automata Theory.* To appear.
[26] Philipp Schlicht and Frank Stephan. Automata on ordinals and automaticy of linear orders. *Annals of Pure and Applied Logic*, 164 (5), pp. 523–527, 2013.
[27] Wacław Sierpiński. Cardinal and ordinal numbers. *Polska Akademia Nauk, Monografie Matematyczne.* Tom 34 Państwowe Wydawnictwo Naukowe, Warsaw 1958.

[28] Frank Stephan. Automatic structures – recent results and open questions. *Third International Conference on Science and Engineering in Mathematics, Chemistry and Physics*, ScieTech 2015 (pages 121–130 in proceedings booklet), *Journal of Physics: Conference Series*, 622/1 (Paper 012013) 2015.

Sanjay Jain, School of Computing, National University of Singapore, Singapore 117417, Republic of Singapore
Email address: sanjay@comp.nus.edu.sg

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

Philipp Schlicht, School of Mathematics, University of Bristol, University Walk, Bristol, BS8 1TW, United Kingdom
Email address: philipp.schlicht@bristol.ac.uk

Frank Stephan, Department of Mathematics, National University of Singapore, Block S17, 10 Lower Kent Ridge Road, Singapore 119076, Republic of Singapore
Email address: fstephan@comp.nus.edu.sg