The Isomorphism Problem On Classes of Automatic Structures

Dietrich Kuske
CNRS, LaBRI, Bordeaux, France
kuske@labri.fr

Jiamou Liu, Markus Lohrey*
Universität Leipzig, Institut für Informatik, Germany
{jiamou,lohrey}@informatik.uni-leipzig.de

Abstract

Several undecidability results on isomorphism problems for automatic structures are shown: (i) The isomorphism problem for automatic equivalence relations is Π^0_1-complete. (ii) The isomorphism problem for automatic trees of height $n \geq 2$ is Π^0_{2n-3}-complete. (iii) The isomorphism problem for automatic linear orders is not arithmetical.

1 Introduction

The idea of an automatic structure goes back to Büchi and Elgot who used finite automata to decide, e.g., Presburger arithmetic [4]. Automaton decidable theories [7] and automatic groups [5] are similar concepts. A systematic study was initiated by Khoussainov and Nerode [10] who also coined the name "*automatic structure*". In essence, a structure is automatic if the elements of the universe can be represented as strings from a regular language and every relation of the structure can be recognized by a finite state automaton with several heads that proceed synchronously. Automatic structures received increasing interest over the last years [1, 2, 12, 13, 14, 19]. One of the main motivations for investigating automatic structures is that their first-order theories can be decided uniformly (i.e., the input is an automatic presentation and a first-order sentence).

Automatic structures form a subclass of computable structures. A structure is computable, if its domain as well as all relations are recursive sets of finite words (or naturals). A well-studied problem for computable structures is the isomorphism problem, where it is asked whether two given computable structures over the same signature (encoded by Turing-machines for the domain and all relations) are isomorphic. It is well known that the isomorphism problem for computable structures is complete for the first level of the analytical hierarchy Σ^1_1. In fact, Σ^1_1-completeness holds for many subclasses of computable structures, e.g., for linear orders, trees, undirected graphs, Boolean algebras, Abelian p-groups, see [3, 6]. Σ^1_1-completeness of the isomorphism problem for a class of computable structures implies non-existence of a good classification (in the sense of [3]) for that class.

In [12], it was shown that also for automatic structures the isomorphism problem is Σ^1_1-complete. By a direct interpretation, it follows that for the following classes the isomorphism problem is still Σ^1_1-complete [17]: automatic successor trees, automatic undirected graphs, automatic commutative monoids, automatic partial orders, automatic lattices of height 4, and automatic 1-ary functions. On the other hand, the isomorphism problem is decidable for automatic ordinals [13] and automatic Boolean algebras [12]. An intermediate class is the class of all locally-finite automatic graphs, for which the isomorphism problem is complete for Π^0_3 (third level of the arithmetical hierarchy[1]) [18].

For many interesting classes of automatic structures, the exact status of the isomorphism problem is open. In the recent papers [19, 11] it was asked for instance, whether the isomorphism problem is decidable for automatic equivalence relations and automatic linear orders. For the latter class, this question was already asked in [13]. In this paper, we answer these questions. Our main results are:

- The isomorphism problem for automatic equivalence relations is Π^0_1-complete.
- The isomorphism problem for automatic successor trees of finite height $k \geq 2$ (where the height of a tree is the maximal number of edges along a path from the root to a leaf) is Π^0_{2k-3}-complete.
- The isomorphism problem for automatic linear orders is hard for every level of the arithmetical hierarchy.

Most hardness proofs for automatic structures, in particular the Σ^1_1-hardness proof for the isomorphism problem of automatic structures from [12], use transition graphs of Turing-machines (these graphs are easily seen to be automatic). This technique seems to fail for inherent reasons,

*The second and third author are supported by the DFG research project GELO.

[1] For background on the arithmetical hierarchy see, e.g., [20].

when trying to prove our new results. The reason is most obvious for equivalence relations and linear orders. These structures are transitive but the transitive closure of the transition graph of a Turing-machine cannot be automatic in general (it's first-order theory is undecidable in general). Hence, we have to use a new strategy that is based on Hilbert's 10^{th} problem. Recall that Matiyasevich proved that every recursively enumerable set of natural numbers is Diophantine [16]. This fact was used by Honkala to show that it is undecidable whether the range of a rational power series is $\mathbb{N}$ [8]. Based on a similar technique, we show that the isomorphism problem for automatic successor trees of height 2 is Π^0_1-complete. An inductive argument then allows us to prove that the isomorphism problem for automatic successor trees of height $n \geq 2$ is Π^0_{2n-3}-complete. From the case $n = 2$ we can easily deduce that the isomorphism problem for automatic equivalence relations is Π^0_1-complete. Finally, using a similar but technically more involved reduction, we show that the isomorphism problem for automatic linear orders is hard for every level of the arithmetical hierarchy. In fact, since our proof is uniform on the levels in the arithmetical hierarchy, it follows that the isomorphism problem for automatic linear orders is at least as hard as true arithmetic, i.e., the first-order theory of $(\mathbb{N}; +, \times)$. At the moment it remains open whether the isomorphism problem for automatic linear orders is Σ^1_1-complete. A long version of this extended abstract can be found in [15].

2 Preliminaries

Let $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$. Let $p(x_1, \dots, x_n) \in \mathbb{N}[x_1, \dots, x_n]$ be a polynomial with coefficients in $\mathbb{N}$. Define $\mathrm{Img}_+(p) = \{p(\overline{c}) \mid \overline{c} \in \mathbb{N}^n_+\}$. If $p \neq 0$, then $\mathrm{Img}_+(p) \subseteq \mathbb{N}_+$.

Details on the arithmetical hierarchy can be found for instance in [20]. With Σ^0_n we denote the n^{th} (existential) level of the arithmetical hierarchy; it is the class of all $A \subseteq \mathbb{N}$ such that there exists a recursive predicate $P \subseteq \mathbb{N}^{n+1}$ with $A = \{a \in \mathbb{N} \mid \exists x_1 \forall x_2 \cdots Q x_n : (a, x_1, \dots, x_n) \in P\}$, where $Q = \exists$ ($Q = \forall$) for n odd (even). The set of complements of Σ^0_n-sets is denoted by Π^0_n. By fixing some effective encoding of strings by natural numbers, we can talk about Σ^0_n-sets and Π^0_n-sets of strings over an arbitrary alphabet. A typical example of a set, which does not belong to the arithmetical hierarchy is the first-order theory of $(\mathbb{N}; +, \times)$, which we denote by $\mathsf{FOTh}(\mathbb{N}; +, \times)$.

We assume basic terminologies and notations from automata theory. For a fixed alphabet Σ, a *non-deterministic finite automaton* (NFA) is a tuple $\mathcal{A} = (S, \Delta, I, F)$ where S is the set of states, $\Delta \subseteq S \times \Sigma \times S$ is the transition relation, $I \subseteq S$ is a set of initial states, and $F \subseteq S$ is the set of accepting states. A *run* of $\mathcal{A}$ on a word $u = a_1 a_2 \cdots a_n$ ($a_1, a_2 \dots, a_n \in \Sigma$) is a word over Δ of the form $r = (q_0, a_1, q_1)(q_1, a_2, q_2) \cdots (q_{n-1}, a_n, q_n)$, where $q_0 \in I$. If moreover $q_n \in F$, then r is an *accepting run* of $\mathcal{A}$ on u. We will only apply these definitions in case $n > 0$, i.e., we will only speak of (accepting) runs on non-empty words.

We use *synchronous n-tape automata* to recognize n-ary relations. Such automata have n input tapes, each of which contains one of the input words. The n tapes are read in parallel until all input words are processed. Formally, let $\Sigma_\diamond = \Sigma \cup \{\diamond\}$ where $\diamond \notin \Sigma$. For words $w_1, w_2, \dots, w_n \in \Sigma^*$, their *convolution* is a word $w_1 \otimes \cdots \otimes w_n \in (\Sigma^n_\diamond)^*$ with length $\max\{|w_1|, \dots, |w_n|\}$, and the k^{th} symbol of $w_1 \otimes \cdots \otimes w_n$ is $(\sigma_1, \dots, \sigma_n)$ where σ_i is the k^{th} symbol of w_i if $k \leq |w_i|$, and $\sigma_i = \diamond$ otherwise. An n-ary relation R is *FA recognizable* if the set of all convolutions of tuples $(w_1, \dots, w_n) \in R$ is a regular language.

A *relational structure* $\mathcal{S}$ consists of a *domain* D and atomic relations on the set D. We will only consider structures with countable domain. For a set $\{\mathcal{S}_i \mid i \in I\}$ of relational structures over the same signature, we denote with $\uplus\{\mathcal{S}_i \mid i \in I\}$ the disjoint union of these structures. With $\mathcal{S}_1 \uplus \mathcal{S}_2$ we denote the disjoint union of two structures $\mathcal{S}_1$, $\mathcal{S}_2$. A structure $\mathcal{S}$ is called *automatic* over Σ if its domain is a regular subset of Σ^* and each of its atomic relations is FA recognizable; any tuple $\mathbb{P}$ of automata that accept the domain and the relations of $\mathcal{S}$ is called an *automatic presentation of* $\mathcal{S}$; in this case, we write $\mathcal{S}(\mathbb{P})$ for $\mathcal{S}$. If an automatic structure $\mathcal{S}$ is isomorphic to a structure $\mathcal{S}'$, then $\mathcal{S}$ is called an *automatic copy* of $\mathcal{S}'$ and $\mathcal{S}'$ is *automatically presentable*. In this paper we sometimes abuse the terminology referring to $\mathcal{S}'$ as simply automatic and calling an automatic presentation of $\mathcal{S}$ also automatic presentation of $\mathcal{S}'$. We also simplify our statements by saying "given/compute an automatic structure $\mathcal{S}$" for "given/compute an automatic presentation $\mathbb{P}$ of a structure $\mathcal{S}(\mathbb{P})$". The structures $(\mathbb{N}; \leq, +)$ and $(\mathbb{Q}; \leq)$ are both automatic. On the other hand, $(\mathbb{N}; \times)$ and $(\mathbb{Q}; +)$ have no automatic copies (see [9, 19] and [21]).

Let $\mathsf{FO} + \exists^\infty$ be first-order logic extended by the quantifier $\exists^\infty$ (there exist infinitely many). The following theorem (see [19] for references and generalizations) lays out the main motivation for investigating automatic structures.

Theorem 2.1 *From an automatic presentation $\mathbb{P}$ and a formula $\varphi(\bar{x}) \in \mathsf{FO} + \exists^\infty$ in the signature of $\mathcal{S}(\mathbb{P})$, one can compute an NFA whose language consists of those tuples $\bar{a}$ from $\mathcal{S}(\mathbb{P})$ that make φ true. In particular, the $\mathsf{FO} + \exists^\infty$ theory of any automatic structure $\mathcal{S}$ is (uniformly) decidable.*

Let $\mathcal{K}$ be a class of automatic structures closed under isomorphism. The *isomorphism problem* for $\mathcal{K}$ is the set of pairs $(\mathbb{P}_1, \mathbb{P}_2)$ of automatic presentations with $\mathcal{S}(\mathbb{P}_1) \cong \mathcal{S}(\mathbb{P}_2) \in \mathcal{K}$. The isomorphism problem for the class of all automatic structures is complete for Σ^1_1 — the first level of the analytical hierarchy [12] (this holds already for automatic successor trees). However, if one restricts to special subclasses of automatic structures, this complexity bound can be reduced. For example, for the class of automatic ordinals and also the class of automatic Boolean algebras, the isomorphism problem is decidable. Another interesting result is that the isomorphism problem for locally finite automatic graphs is Π^0_3-complete [18]. All these classes of automatic structures have the nice property that one can decide whether a given automatic presentation describes a structure from this class. Thm. 2.1 implies that this property also holds for the classes of equivalence relations, trees of height at most k, and linear orders, i.e., the classes considered in this paper.

3 Automatic Trees

A *tree* is a structure $T = (V; \leq)$, where $\leq$ is a partial order with a least element, called the *root*, and such that for every $x \in V$, the order $\leq$ restricted to the set $\{y \mid y \leq x\}$ of ancestors of x is a finite linear order. The *level* of a node $x \in V$ is $|\{y \mid y < x\}| \in \mathbb{N}$. The *height* of T is the supremum of the levels of all nodes in V; it may be infinite, but this paper deals with trees of finite height only. One may also view a tree as a directed graph (V, E), where there is an edge $(u, v) \in E$ if and only if u is the largest element in $\{x \mid x < v\}$. The edge relation E is FO-definable in $(V; \leq)$. In this paper, we assume the partial order definition for trees, but will quite often refer to them as graphs for convenience. We use $\mathcal{T}_n$ to denote the class of automatic trees with height at most n. Let n be fixed. Then the tree order $\leq$ is FO-definable in $T = (V, E)$ and this holds even uniformly for all trees from $\mathcal{T}_n$. Moreover, it is decidable whether a given automatic graph belongs to $\mathcal{T}_n$ (since the class of trees of height n can be axiomatized in first-order logic).

In this section, we prove that the isomorphism problem for $\mathcal{T}_n$ is Π^0_{2n-3}-complete. We start with the upper bound:

Proposition 3.1 *The isomorphism problem for the class $\mathcal{T}_n$ of automatic trees of height at most n is (i) decidable for $n = 1$ and (ii) in Π^0_{2n-3} for all $n \geq 2$.*

Proof. We first show that $T_1 \cong T_2$ is decidable for automatic trees $T_1, T_2 \in \mathcal{T}_1$ of height at most 1: It suffices to compute the cardinality of T_i ($i \in \{1, 2\}$) which is possible since the universes of T_1 and T_2 are regular languages.

Now let $n \geq 2$ and consider $T_1, T_2 \in \mathcal{T}_n$. Let $T_i = (V_i, E_i)$, w.l.o.g. $V_1 \cap V_2 = \emptyset$, and $V = V_1 \cup V_2$, $E = E_1 \cup E_2$. For any node u in V, let $T(u)$ denote the subtree (of either T_1 or T_2) rooted at u and let $E(u)$ be the set of children of u. For $k = n-2, n-3, \ldots, 0$, we will define inductively a $\Pi^0_{2n-2k-3}$-predicate $\text{iso}_k(u_1, u_2)$ for $u_1, u_2 \in V$. This predicate expresses that $T(u_1) \cong T(u_2)$ provided u_1 and u_2 belong to level at least k. The result will follow since $T_1 \cong T_2$ if and only if $\text{iso}_0(r_1, r_2)$ holds, where r_σ is the root of T_σ.

For $k = n-2$, the trees $T(u_1)$ and $T(u_2)$ have height at most 2. The statement $\text{iso}_{n-2}(u_1, u_2)$ can be defined as follows: *For all $\kappa \in \mathbb{N} \cup \{\aleph_0\}$ and all $\ell \geq 1$ we have*

$$\exists x_1, \ldots, x_\ell \in E(u_1) : \bigwedge_{1 \leq i < j \leq \ell} x_i \neq x_j \wedge \bigwedge_{i=1}^{\ell} |E(x_i)| = \kappa$$

if and only if

$$\exists y_1, \ldots, y_\ell \in E(u_2) : \bigwedge_{1 \leq i < j \leq \ell} y_i \neq y_j \wedge \bigwedge_{i=1}^{\ell} |E(y_i)| = \kappa.$$

In other words: for every $\kappa \in \mathbb{N} \cup \{\aleph_0\}$, u_1 and u_2 have the same number of children with exactly κ children. Since $\mathsf{FO} + \exists^\infty$ is uniformly decidable for automatic structures, this is indeed a Π^0_1-sentence (note that $2n - 2k - 3 = 1$ for $k = n - 2$). For $0 \leq k < n - 2$, we define $\text{iso}_k(u_1, u_2)$ inductively as follows: *For all $v \in E(u_1) \cup E(u_2)$ and all $\ell \geq 1$ we have*

$$\exists x_1, \ldots, x_\ell \in E(u_1) : \bigwedge_{1 \leq i < j \leq \ell} x_i \neq x_j \wedge \bigwedge_{i=1}^{\ell} \text{iso}_{k+1}(v, x_i)$$

if and only if

$$\exists y_1, \ldots, y_\ell \in E(u_2) : \bigwedge_{1 \leq i < j \leq \ell} y_i \neq y_j \wedge \bigwedge_{i=1}^{\ell} \text{iso}_{k+1}(v, y_i).$$

By quantifying over all $v \in E(u_1) \cup E(u_2)$, we quantify over all isomorphism types of trees that occur as a subtree rooted at a child of u_1 or u_2. For each of these isomorphism types τ, we express that u_1 and u_2 have the same number of children x with $T(x)$ of type τ. Since by induction, $\text{iso}_{k+1}(v, x_i)$ and $\text{iso}_{k+1}(v, y_i)$ are $\Pi^0_{2n-2k-5}$-statements, $\text{iso}_k(u_1, u_2)$ is a $\Pi^0_{2n-2k-3}$-statement. ☐

The rest of this section is devoted to proving that the isomorphism problem for the class $\mathcal{T}_n$ of automatic trees of height at most $n \geq 2$ is also Π^0_{2n-3}-hard (and therefore complete). So let $P_n(x_0)$ be a Π^0_{2n-3}-predicate. In the following lemma and its proof, all quantifiers with unspecified range run over $\mathbb{N}_+$.

Lemma 3.2 *For any Π^0_{2n-3}-predicate $P_n(x_0)$, there exist Π^0_{2i-3}-predicates $P_i(x_0, x_1, y_1, x_2, y_2, \ldots, x_{n-i}, y_{n-i})$ for $2 \leq i < n$ such that*

(a) for all $2 \leq i < n$, $P_{i+1}(\overline{v})$ is logically equivalent to $\forall x_{n-i} \exists y_{n-i} : P_i(\overline{v}, x_{n-i}, y_{n-i})$, and

(b) if $\forall y_{n-i} : \neg P_i(\overline{v}, x_{n-i}, y_{n-i})$ holds, then also $\forall x'_{n-i} \geq x_{n-i}\ \forall y_{n-i} : \neg P_i(\overline{v}, x'_{n-i}, y_{n-i})$,

where $\overline{v} = (x_0, x_1, y_1, \ldots, x_{n-i-1}, y_{n-i-1})$.

Proof. The predicates P_i are constructed by induction, starting with $i = n-1$ down to $i = 2$ where the construction of P_i does not assume that (a) or (b) hold true for P_{i+1}. So let $2 \leq i < n$ such that $P_{i+1}(\overline{v})$ is a $\Pi^0_{2(i+1)-3}$-predicate. Then there exists a Π^0_{2i-3}-predicate $P(\overline{v}, x_{n-i}, y_{n-i})$ such that $P_{i+1}(\overline{v})$ is logically equivalent to $\forall x_{n-i} \exists y_{n-i} : P(\overline{v}, x_{n-i}, y_{n-i})$. But this is logically equivalent to

$$\forall x_{n-i}\, \forall x'_{n-i} \leq x_{n-i}\, \exists y_{n-i} : P(\overline{v}, x'_{n-i}, y_{n-i})\,. \qquad (1)$$

Let $\varphi(\overline{v}, x_{n-i})$ be the formula $\forall x'_{n-i} \leq x_{n-i}\, \exists y_{n-i} : P(\overline{v}, x'_{n-i}, y_{n-i})$. Then for any $x_{n-i} \in \mathbb{N}$,

$$\neg\varphi(\overline{v}, x_{n-i}) \implies \forall x \geq x_{n-i} : \neg\varphi(\overline{v}, x)\,. \qquad (2)$$

Since $\forall x'_{n-i} \leq x_{n-i}$ is a bounded quantifier, the formula $\varphi(\overline{v}, x_{n-i})$ belongs to Σ^0_{2i-2} (see for example [20, p. 61]). Thus there is a Π^0_{2i-3}-predicate $P_i(\overline{v}, x_{n-i}, y_{n-i})$ such that

$$\varphi(\overline{v}, x_{n-i}) \iff \exists y_{n-i} : P_i(\overline{v}, x_{n-i}, y_{n-i})\,. \qquad (3)$$

Therefore (1) (and therefore $P_{i+1}(\overline{v})$) is logically equivalent to $\forall x_{n-i}\, \exists y_{n-i} : P_i(\overline{v}, x_{n-i}, y_{n-i})$, which shows statement (a). For (b) note that $\forall y_{n-i} : \neg P_i(\overline{v}, x_{n-i}, y_{n-i})$ if and only if (by (3)) $\neg\varphi(\overline{v}, x_{n-i})$, which by (2) implies $\forall x \geq x_{n-i} : \neg\varphi(\overline{v}, x)$. By (3) again, this is equivalent to $\forall x \geq x_{n-i}\, \forall y_{n-i} : \neg P_i(\overline{v}, x, y_{n-i})$. □

Let us fix the predicates P_i for the rest of Sec. 3. By induction on $2 \leq i \leq n$, we construct the following trees:

- test trees $T^i_{\overline{c}} \in \mathcal{T}_i$ for $\overline{c} \in \mathbb{N}_+^{1+2(n-i)}$ (which depend on P_i) and
- trees $U^i_\kappa \in \mathcal{T}_i$ for $\kappa \in \mathbb{N}_+ \cup \{\omega\}$ (we assume the standard order on $\mathbb{N}_+ \cup \{\omega\}$).

The idea is that $T^i_{\overline{c}} \cong U^i_\kappa$ if and only if $\kappa = 1 + \inf(\{x_{n-i} \mid \forall y_{n-i} \in \mathbb{N}_+ : \neg P_i(\overline{c}, x_{n-i}, y_{n-i})\} \cup \{\omega\})$. We will not prove this equivalence, but the following simpler consequences for any $\overline{c} \in \mathbb{N}_+^{1+2(n-i)}$:

(P1) $P_i(\overline{c})$ if and only if $T^i_{\overline{c}} \cong U^i_\omega$.

(P2) $\neg P_i(\overline{c})$ if and only if $T^i_{\overline{c}} \cong U^i_m$ for some $m \in \mathbb{N}_+$.

The first property is certainly sufficient for proving Π^0_{2n-3}-hardness (with $i = n$), the second property and therefore the trees U^i_m for $m < \omega$ are used in the inductive step. We also need the following property for the construction.

(P3) No leaf of any of the trees $T^i_{\overline{c}}$ or U^i_κ is a child of the root.

In Section 3.1, we will describe the trees $T^i_{\overline{c}}$ and U^i_κ of height at most i and prove (P1) and (P2). Condition (P3) will be obvious from the construction. Section 3.2 is then devoted to proving the effective automaticity of these trees.

3.1 Construction of trees

We start with a few definitions: A forest is a disjoint union of trees. Let H and J be two forests. The forest H^ω is the disjoint union of countably many copies of H. Formally, if $H = (V, E)$, then $H^\omega = (V \times \mathbb{N}, E')$ with $((v, i), (w, j)) \in E'$ if and only if $(v, w) \in E$ and $i = j$. We write $H \sim J$ for $H^\omega \cong J^\omega$. Then $H \sim J$ if they are formed, up to isomorphism, by the same set of trees (i.e., any tree is isomorphic to some connected component of H if and only if it is isomorphic to some connected component of J). If r does not belong to the domain of H, then we denote with $r \circ H$ the tree that results from adding r to H as new least element.

3.1.1 Induction base: construction of $T^2_{\overline{c}}$ and U^2_κ

For notational simplicity, we write k for $1 + 2(n-2)$. Hence, P_2 is a k-ary predicate. By Matiyasevich's theorem, we find two non-zero polynomials $p_1(x_1, \ldots, x_\ell)$, $p_2(x_1, \ldots, x_\ell) \in \mathbb{N}[\overline{x}]$, $\ell > k$, such that for any $\overline{c} \in \mathbb{N}^k_+$:

$$P_2(\overline{c}) \iff \forall \overline{x} \in \mathbb{N}_+^{\ell-k} : p_1(\overline{c}, \overline{x}) \neq p_2(\overline{c}, \overline{x})\,.$$

It is well known that the function $C : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ with

$$C(x, y) = (x+y)^2 + 3x + y \qquad (4)$$

is injective ($C(x, y)/2$ defines a pairing function, see e.g. [8]). For two numbers $m, n \in \mathbb{N}_+$, let $T[m, n]$ denote the tree of height 1 with exactly $C(m, n)$ leaves. Then define the following forests, where $\kappa \in \mathbb{N}_+ \cup \{\omega\}$:

$$H^2 = \biguplus\{T[m, n] \mid m, n \in \mathbb{N}_+, m \neq n\}$$

$$H^2_{\overline{c}} = H^2 \uplus \biguplus\{T[p_1(\overline{c}, \overline{x}) + x_{\ell+1}, p_2(\overline{c}, \overline{x}) + x_{\ell+1}] \mid \overline{x} \in \mathbb{N}_+^{\ell-k}, x_{\ell+1} \in \mathbb{N}_+\}$$

$$J^2_\kappa = H^2 \uplus \biguplus\{T[x, x] \mid x \in \mathbb{N}_+, x > \kappa\}$$

Note that $J^2_\omega = H^2$. Moreover, the forests J^2_κ ($\kappa \in \mathbb{N}_+ \cup \{\omega\}$) are pairwise non-isomorphic, since C is injective.

The tree $T^2_{\overline{c}}$ (resp. U^2_κ) is obtained from $H^2_{\overline{c}}$ (resp. J^2_κ) by taking countably many copies and adding a root:

$$T^2_{\overline{c}} = r \circ (H^2_{\overline{c}})^\omega \quad \text{and} \quad U^2_\kappa = r \circ (J^2_\kappa)^\omega, \qquad (5)$$

see Fig.1 and 2. The following lemma states (P1) for the Π^0_1-predicate P_2 , i.e., for $i = 2$.

Lemma 3.3 *For all* $\overline{c} \in \mathbb{N}^k_+$*:* $P_2(\overline{c}) \iff T^2_{\overline{c}} \cong U^2_\omega$.

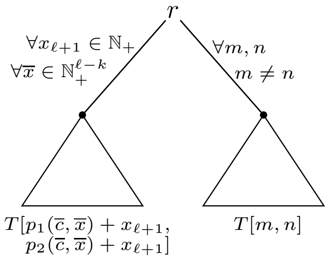

Figure 1. The tree $T^2_{\overline{c}}$

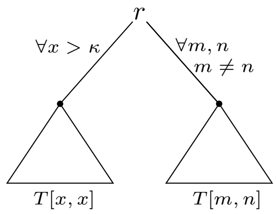

Figure 2. The tree U^2_κ

Proof. By (5), it suffices to show that $P_2(\overline{c})$ holds if and only if $H^2_{\overline{c}} \sim J^2_\omega$. So first assume $P_2(\overline{c})$ holds. We have to prove that the forests $H^2_{\overline{c}}$ and $J^2_\omega = H^2$ contain the same trees (up to isomorphism). Clearly, every tree from H^2 is contained in $H^2_{\overline{c}}$. For the other direction, let $\overline{x} \in \mathbb{N}_+^{\ell-k}$ and $x_{\ell+1} \in \mathbb{N}_+$. Then the tree $T[p_1(\overline{c}, \overline{x}) + x_{\ell+1}, p_2(\overline{c}, \overline{x}) + x_{\ell+1}]$ occurs in $H^2_{\overline{c}}$. Since $P_2(\overline{c})$ holds, we have $p_1(\overline{c}, \overline{x}) \neq p_2(\overline{c}, \overline{x})$ and therefore $p_1(\overline{c}, \overline{x}) + x_{\ell+1} \neq p_2(\overline{c}, \overline{x}) + x_{\ell+1}$. Hence this tree also occurs in H^2.

Conversely suppose $H^2_{\overline{c}} \sim H^2$ and let $\overline{x} \in \mathbb{N}_+^{\ell-k}$. Then the tree $T[p_1(\overline{c}, \overline{x})+1, p_2(\overline{c}, \overline{x})+1]$ occurs in $H^2_{\overline{c}}$ and therefore in H^2. Hence $p_1(\overline{c}, \overline{x}) \neq p_2(\overline{c}, \overline{x})$. Since $\overline{x}$ was chosen arbitrarily, this implies $P_2(\overline{c})$. ☐

Now consider the forest $H^2_{\overline{c}}$ once more. If it contains a tree of the form $T[m, m]$ for some m (necessarily $m \geq 2$), then it contains all trees $T[x, x]$ for $x \geq m$. Hence, $H^2_{\overline{c}} \sim J^2_\kappa$ for some $\kappa \in \mathbb{N}_+ \cup \{\omega\}$, which implies $T^2_{\overline{c}} \cong U^2_\kappa$ for some $\kappa \in \mathbb{N}_+ \cup \{\omega\}$. Thus, with Lemma 3.3 we get:

$$\begin{aligned} \neg P_2(\overline{c}) \quad &\iff \quad T^2_{\overline{c}} \ncong U^2_\omega \\ &\iff \quad \exists m \in \mathbb{N}_+ : T^2_{\overline{c}} \cong U^2_m \end{aligned}$$

Thus, we proved (P2) for the Π^0_1-predicate P_2. This finishes the construction of the trees $T^2_{\overline{c}}$ and U^2_κ for $\kappa \in \mathbb{N}_+ \cup \{\omega\}$, and the verification of properties (P1) and (P2). Clearly, also (P3) holds for $T^2_{\overline{c}}$ and U^2_κ (all maximal paths have length 2).

3.1.2 Induction step: construction of $T^{i+1}_{\overline{c}}$ and U^{i+1}_κ

Again, we write k for $1 + 2(n - i - 1)$. Thus, P_{i+1} is a k-ary predicate and P_i a $(k + 2)$-ary one. We now apply the induction hypothesis. For any $\overline{c} \in \mathbb{N}_+^k$, $x, y \in \mathbb{N}_+$,

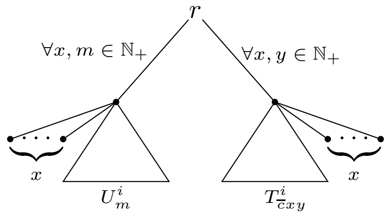

Figure 3. The tree $T^{i+1}_{\overline{c}}$

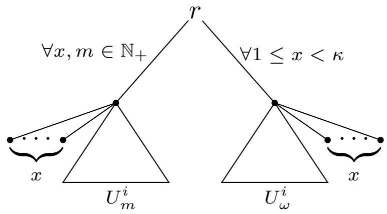

Figure 4. The tree U^{i+1}_κ

$\kappa \in \mathbb{N}_+ \cup \{\omega\}$ let $T^i_{\overline{c}xy}$ and U^i_κ be trees of height at most i such that:

$$\begin{aligned} P_i(\overline{c}, x, y) \quad &\iff \quad T^i_{\overline{c}xy} \cong U^i_\omega \\ \neg P_i(\overline{c}, x, y) \quad &\iff \quad \exists m \in \mathbb{N}_+ : T^i_{\overline{c}xy} \cong U^i_m. \end{aligned}$$

In a first step, we build trees $T'_{\overline{c}xy}$ and $U'_{\kappa,x}$ ($x \in \mathbb{N}_+$) from $T^i_{\overline{c}xy}$ and U^i_κ, resp., by adding x leaves as children of the root. This ensures:

$$T'_{\overline{c}xy} \cong U'_{\kappa,x'} \quad \iff \quad x = x' \wedge T^i_{\overline{c}xy} \cong U^i_\kappa \ , \qquad (6)$$

since, by property (P3), no leaf of any of the trees $T^i_{\overline{c}xy}$ or U^i_κ is a child of the root. Next, we collect these trees into forests as follows:

$$\begin{aligned} H^{i+1} &= \biguplus \{U'_{m,x} \mid x, m \in \mathbb{N}_+\} \ , \\ H^{i+1}_{\overline{c}} &= H^{i+1} \uplus \biguplus \{T'_{\overline{c}xy} \mid x, y \in \mathbb{N}_+\} \ , \text{ and} \\ J^{i+1}_\kappa &= H^{i+1} \uplus \biguplus \{U'_{\omega,x} \mid 1 \leq x < \kappa\} \text{ for } \kappa \in \mathbb{N}_+ \cup \{\omega\} \end{aligned}$$

The tree $T^{i+1}_{\overline{c}}$ (resp. U^{i+1}_κ) is obtained from the forest $H^{i+1}_{\overline{c}}$ (resp. J^{i+1}_κ) by taking countably many copies and adding a root:

$$T^{i+1}_{\overline{c}} = r \circ (H^{i+1}_{\overline{c}})^\omega \quad \text{and} \quad U^{i+1}_\kappa = r \circ (J^{i+1}_\kappa)^\omega, \qquad (7)$$

see Fig.3 and 4. Note that the height of any of these trees is one more than the height of the forests defining them and therefore at most $i + 1$. Since none of the connected components of the forests $H^{i+1}_{\overline{c}}$ and J^{i+1}_κ is a singleton, none of the trees in (7) has a leaf that is a child of the root and therefore (P3) holds. The next lemma states (P1) for $i + 1$:

Lemma 3.4 *For all $\overline{c} \in \mathbb{N}_+^k$: $P_{i+1}(\overline{c}) \iff T^{i+1}_{\overline{c}} \cong U^{i+1}_\omega$.*

Proof. By (7), it suffices to show that $P_{i+1}(\overline{c})$ if and only if $H^{i+1}_{\overline{c}} \sim J^{i+1}_{\omega}$. First assume $H^{i+1}_{\overline{c}} \sim J^{i+1}_{\omega}$ and let $x \geq 1$ be arbitrary. We have to find some $y \geq 1$ with $P_i(\overline{c}, x, y)$. Note that $U'_{\omega,x}$ belongs to J^{i+1}_{ω} and therefore to $H^{i+1}_{\overline{c}}$. Since $U'_{\omega,x} \not\cong U'_{m,x'}$ for any $m, x, x' \in \mathbb{N}_+$, this implies the existence of $x', y' \geq 1$ with $T'_{\overline{c}x'y'} \cong U'_{\omega,x}$. By (6), this is equivalent to $x = x'$ and $T^i_{\overline{c}xy'} \cong U^i_{\omega}$. Now the induction hypothesis implies that $P_i(\overline{c}, x, y')$ holds. Since $x \geq 1$ was chosen arbitrarily, we get $P_{i+1}(\overline{c})$.

Conversely suppose $P_{i+1}(\overline{c})$. Let T belong to $H^{i+1}_{\overline{c}}$. By the induction hypothesis, it is one of the trees $U'_{\kappa,x}$ for some $x \in \mathbb{N}_+$, $\kappa \in \mathbb{N}_+ \cup \{\omega\}$. In any case, it also belongs to J^{i+1}_{ω}. Hence it remains to show that any tree of the form $U'_{\omega,x}$ belongs to $H^{i+1}_{\overline{c}}$. So let $x \in \mathbb{N}_+$. Then, by $P_{i+1}(\overline{c})$, there exists $y \in \mathbb{N}_+$ with $P_i(\overline{c}, x, y)$. By the induction hypothesis, we have $T^i_{\overline{c}xy} \cong U^i_{\omega}$ and therefore $T'_{\overline{c}xy} \cong U'_{\omega,x}$ (which belongs to $H^{i+1}_{\overline{c}}$ by the very definition). ◻

Lemma 3.5 *For all $\overline{c} \in \mathbb{N}^k_+$ there exists $\kappa \in \mathbb{N}_+ \cup \{\omega\}$ such that $T^{i+1}_{\overline{c}} \cong U^{i+1}_{\kappa}$.*

Proof. It suffices to prove that $H^{i+1}_{\overline{c}} \sim J^{i+1}_{\kappa}$ for some $\kappa \in \mathbb{N}_+ \cup \{\omega\}$. Let κ be the smallest value in $\mathbb{N}_+ \cup \{\omega\}$ with $\forall x \geq \kappa \, \forall y : \neg P_i(\overline{c}, x, y)$. By property (b) from Lemma 3.2 for P_i, we get $\forall 1 \leq x < \kappa \, \exists y : P_i(\overline{c}, x, y)$. By the induction hypothesis, we get $\forall x \geq \kappa \, \forall y : T'_{\overline{c}xy} \not\cong U'_{\omega,x}$ and $\forall 1 \leq x < \kappa \, \exists y : T'_{\overline{c}xy} \cong U'_{\omega,x}$. Thus, $H^{i+1}_{\overline{c}}$ contains, apart from the trees in $H^{i+1} = \biguplus \{U'_{m,x} \mid x, m \in \mathbb{N}_+\}$, exactly the trees from $\{U'_{\omega,x} \mid 1 \leq x < \kappa\}$, i.e., $H^{i+1}_{\overline{c}} \sim J^{i+1}_{\kappa}$. ◻

Lemma 3.4 and 3.5 immediately imply also (P2) for $i+1$. Finally, (P1) for $i = n$ gives:

Proposition 3.6 *For the Π^0_{2n-3}-predicate $P(x)$ we have for all $c \in \mathbb{N}_+$: $P(c)$ if and only if $T^n_c \cong U^n_{\omega}$.*

It remains to show that the trees T^n_c and U^n_{ω} are effectively automatic – this is the topic of the next section.

3.2 Automaticity

For constructing automatic presentations for the trees from Section 3.1, it is actually easier to work with *dags* (*directed acyclic graphs*). The *height* of a dag D is the length (number of edges) of a longest directed path in D. We only consider dags of finite height. A *root* of a dag is a node without incoming edges. A dag $D = (V, E)$ can be unfolded into a forest $\mathrm{unfold}(D)$ in the usual way: Nodes of $\mathrm{unfold}(D)$ are directed paths in D that cannot be extended to the left (i.e., the initial node of the path is a root) and there is an edge between two paths p, p' if and only if p' extends p by one more node. For a node $v \in V$ of D, we define the tree $\mathrm{unfold}(D, v)$ as follows: First we restrict D to those nodes that are reachable from v and then we unfold the resulting dag. We need the following lemma.

Lemma 3.7 *From given $k \in \mathbb{N}$ and an automatic dag $D = (V, E)$ of height at most k, one can construct effectively an automatic presentation $\mathbb{P}$ with $\mathcal{S}(\mathbb{P}) \cong \mathrm{unfold}(D)$.*

Proof. The universe for our automatic copy of $\mathrm{unfold}(D)$ is the set P of all convolutions $v_1 \otimes v_2 \otimes \cdots \otimes v_m$, where v_1 is a root and $(v_i, v_{i+1}) \in E$ for all $1 \leq i < m$. Since D has height at most k, we have $m \leq k$. Since the edge relation of D is automatic and since the set of all roots in D is first-order definable and hence regular, P is indeed a regular set. Moreover, the edge relation of $\mathrm{unfold}(D)$ becomes clearly FA recognizable on P. ◻

For $2 \leq i \leq n$, let F^i be the forest

$$\biguplus \{T^i_{\overline{c}} \mid \overline{c} \in \mathbb{N}^{1+2(n-i)}_+\} \uplus \biguplus \{U^i_{\kappa} \mid \kappa \in \mathbb{N}_+ \cup \{\omega\}\} \ .$$

By induction over i, we will prove:

Proposition 3.8 *There is an an automatic copy $\mathcal{F}^i$ of F^i and an isomorphism $f^i : F^i \to \mathcal{F}^i$ that maps (i) the root of the tree $T^i_{\overline{c}}$ to $a^{\overline{c}}$ (for all $\overline{c} \in \mathbb{N}^{1+2(n-i)}_+$), (ii) the root of the tree U^i_{ω} to ε, and (iii) the root of the tree U^i_m to b^m (for all $m \in \mathbb{N}_+$).*

This will give the desired result since T^n_c is then isomorphic to the connected component of $\mathcal{F}^n$ that contains the word a^c (and similarly for U^n_{κ}). Note that this connected component is again (effectively) automatic by Thm. 2.1, since the forest $\mathcal{F}^n$ has bounded height.

By Lemma 3.7, it suffices to construct an automatic dag $\mathcal{D}^i$ such that there is an isomorphism $h : \mathrm{unfold}(\mathcal{D}^i) \to \mathcal{F}^i$ that is the identity on the set of roots of $\mathcal{D}^i$.

3.2.1 Induction base: the automatic dag $\mathcal{D}^2$

Recall that, for $i = 2$, we used two polynomials p_1 and p_2 from Matiyasevitch's theorem and constructed the trees $T^i_{\overline{c}}$ and U^i_{κ} that then formed the forest F^2. To show automaticity of this forest (more precisely: of a suitable dag $\mathcal{D}^2$), we therefore have to represent polynomials by automata. The basis for this representation, that is inspired by Honkala's work [8], is provided by the following construction.

For a symbol a, let Σ^a_k denote the alphabet $\Sigma^a_k = \{a, \diamond\}^k \setminus \{(\diamond, \ldots, \diamond)\}$ and let σ_i denote the i^{th} component of $\sigma \in \Sigma^a_k$. For $\overline{e} = (e_1, \ldots, e_k) \in \mathbb{N}^k_+$, define

$$a^{\overline{e}} = a^{e_1} \otimes a^{e_2} \otimes \cdots \otimes a^{e_k} \in (\Sigma^a_k)^* \ .$$

For a language L, we write $\otimes_k(L)$ for the language

$$\{u_1 \otimes u_2 \otimes \cdots \otimes u_k \mid u_1, \ldots, u_k \in L\}.$$

Lemma 3.9 *There exists an algorithm that, given a nonzero polynomial $p(\overline{x}) \in \mathbb{N}[\overline{x}]$ in k variables, constructs an NFA $\mathcal{A}[p(\overline{x})]$ on the alphabet Σ_k^a with $L(\mathcal{A}[p(\overline{x})]) = \otimes_k(a^+)$ such that for all $\overline{c} \in \mathbb{N}_+^k$: $\mathcal{A}[p(\overline{x})]$ has exactly $p(\overline{c})$ accepting runs on input $a^{\overline{c}}$.*

Proof. The NFA $\mathcal{A}[p(\overline{x})]$ is build by induction on the construction of the polynomial p, the base case is provided by the polynomials 1 and x_i.

Let $\mathcal{A}[1]$ be a deterministic automaton with $L(\mathcal{A}[1]) = \otimes_k(a^+)$. Next, suppose $p(x_1, \ldots, x_k) = x_i$ for some $1 \leq i \leq k$. Let $\mathcal{A}[p(\overline{x})] = (\{q_1, q_2\}, \{q_1\}, \Delta, \{q_2\})$ with $\Delta = \{(q_1, \sigma, q_j) \mid j \in \{1,2\}, \sigma \in \Sigma_k^a, \sigma_i = a\} \cup \{(q_2, \sigma, q_2) \mid \sigma \in \Sigma_k^a\}$. When the NFA $\mathcal{A}[p(\overline{x})]$ runs on an input word $a^{\overline{c}}$, it has exactly c_i many times the chance to move from state q_1 to the final state q_2. Therefore there are exactly $c_i = p(\overline{c})$ many accepting runs on $a^{\overline{c}}$.

Let $p_1(\overline{x})$ and $p_2(\overline{x})$ be polynomials in $\mathbb{N}[\overline{x}]$. Assume as inductive hypothesis that there are two NFA $\mathcal{A}[p_i(\overline{x})] = (S_i, \Delta_i, I_i, F_i)$ such that the number of accepting runs of $\mathcal{A}[p_i(\overline{x})]$ on $a^{\overline{c}}$ equals $p_i(\overline{c})$ for $i \in \{1,2\}$

For $p(\overline{x}) = p_1(\overline{x}) + p_2(\overline{x})$, let $\mathcal{A}[p(\overline{x})]$ denote the disjoint union of $\mathcal{A}[p_1(\overline{x})]$ and $\mathcal{A}[p_2(\overline{x})]$. For any word $a^{\overline{c}}$, the number of accepting runs of $\mathcal{A}[p(\overline{x})]$ on u is equal to the sum of the numbers of accepting runs of $\mathcal{A}[p_1(\overline{x})]$ and $\mathcal{A}[p_2(\overline{x})]$ on $a^{\overline{c}}$, which is $p(\overline{c})$.

For $p(\overline{x}) = p_1(\overline{x}) \cdot p_2(\overline{x})$, let $\mathcal{A}[p(\overline{x})] = (S_1 \times S_2, \Delta, I_1 \times I_2, F_1 \times F_2)$, where $\Delta = \{((p_1, p_2), \sigma, (q_1, q_2)) \mid (p_1, \sigma, q_1) \in \Delta_1, (p_2, \sigma, q_2) \in \Delta_2\}$. Then the number of accepting runs of $\mathcal{A}[p(\overline{x})]$ on a word $a^{\overline{c}}$ is the product of the numbers of accepting runs of $\mathcal{A}[p_1(\overline{x})]$ and $\mathcal{A}[p_2(\overline{x})]$ on $a^{\overline{c}}$, which is $p(\overline{c})$. ☐

Lemma 3.10 *Let $q_1, q_2 \in \mathbb{N}[x_1, \ldots, x_\ell]$ and let a be some symbol. There is an automatic forest of height 1 over an alphabet $\Sigma_\ell^a \uplus \Gamma$ such that: (i) the roots are the words from $\otimes_\ell(a^+)$, (ii) the leaves are words from Γ^+, and (iii) the tree rooted at $a^{\overline{e}}$ is isomorphic to $T[q_1(\overline{e}), q_2(\overline{e})]$.*

Proof. Set $p(\overline{x}) = C(q_1(\overline{x}), q_2(\overline{x}))$ (C is defined in (4)) and let $\mathcal{A}[p] = (S, I, \Delta, F)$ be the NFA over the alphabet Σ_ℓ^a from Lemma 3.9. Define the NFA $\mathcal{B}[p] = (S, I, \Delta', F)$ with alphabet Δ and $\Delta' = \{(p, (p, \sigma, q), q) \mid (p, \sigma, q) \in \Delta\}$; it accepts the set of accepting runs of $\mathcal{A}[p]$. Let $\pi : \Delta^* \to (\Sigma_\ell^a)^*$ be the projection morphism with $\pi(p, a, q) = a$. Then, for all $\overline{e} \in \mathbb{N}_+^\ell$, the size of $\pi^{-1}(a^{\overline{e}}) \cap L(\mathcal{B}[p])$ equals the number of accepting runs of $\mathcal{A}[p]$ on $a^{\overline{e}}$, which is $p(\overline{e})$. Let

$$L = \otimes_\ell(a^+) \cup (\pi^{-1}(\otimes_\ell(a^+)) \cap L(\mathcal{B}[p])) \text{ and}$$
$$E = \{(u, v) \mid u \in \otimes_\ell(a^+), v \in \pi^{-1}(u) \cap L(\mathcal{B}[p])\}.$$

Then L is regular and E is FA recognizable, i.e., $(L; E)$ is an automatic graph. It is actually a forest of height 1, the words from $\otimes_\ell(a^+)$ form the roots, and the tree rooted at $a^{\overline{e}}$ has precisely $p(\overline{e})$ leaves, i.e., it is isomorphic to $T[q_1(\overline{e}), q_2(\overline{e})]$. ☐

From now on, we use the notations from Sec. 3.1.1. By Lemma 3.10, we can compute automatic forests $\mathcal{F}_1$ and $\mathcal{F}_2$ over alphabets $\Sigma_{\ell+1}^a \uplus \Gamma_1$ and $\Sigma_2^b \uplus \Gamma_2$, resp., such that

(a) the roots of $\mathcal{F}_1$ (resp. $\mathcal{F}_2$) are the words from $\otimes_{\ell+1}(a^+)$ (resp. $\otimes_2(b^+)$),

(b) the leaves of $\mathcal{F}_i$ are words from Γ_i^+ ($i \in \{1,2\}$),

(c) the tree rooted at $a^{\overline{e}e_{\ell+1}}$ is isomorphic to the tree $T[p_1(\overline{e}) + e_{\ell+1}, p_2(\overline{e}) + e_{\ell+1}]$ for $\overline{e} \in \mathbb{N}_+^\ell$, $e_{\ell+1} \in \mathbb{N}_+$,

(d) the tree rooted at $b^{e_1 e_2}$ is isomorphic to $T[e_1, e_2]$ for $e_1, e_2 \in \mathbb{N}_+$.

We can assume that the alphabets Γ_1, Γ_2, $\Sigma_{\ell+1}^a$, and Σ_2^b are mutually disjoint. Let $\mathcal{F} = (V_\mathcal{F}, E_\mathcal{F})$ be the disjoint union of $\mathcal{F}_1$ and $\mathcal{F}_2$; it is effectively automatic. The universe of the automatic dag $\mathcal{D}^2$ is the regular language

$$\otimes_k(a^+) \cup b^* \cup (\$^* \otimes V_\mathcal{F}),$$

where $ is a new symbol. We have the following edges:

- For $u, v \in V_\mathcal{F}$, $\$^m \otimes u$ is connected to $\$^n \otimes v$ if and only if $m = n$ and $(u, v) \in E_\mathcal{F}$. This produces $\aleph_0$ many copies of $\mathcal{F}$.
- $a^{\overline{c}}$ is connected to all words from $\$^* \otimes (\{a^{\overline{c}\,\overline{x}} \mid \overline{x} \in \mathbb{N}_+^{\ell-k+1}\} \cup \{b^{e_1 e_2} \mid e_1 \neq e_2\})$. By point (c) and (d) above, this means that the tree $\mathrm{unfold}(\mathcal{D}^2, a^{\overline{c}})$ has $\aleph_0$ many subtrees isomorphic to $T[p_1(\overline{c}\,\overline{x}) + x_{\ell+1}, p_2(\overline{c}\,\overline{x}) + x_{\ell+1}]$ for $\overline{x} \in \mathbb{N}_+^{\ell-k}$, $x_{\ell+1} \in \mathbb{N}_+$ and $T[e_1, e_2]$ for $e_1, e_2 \in \mathbb{N}_+$, $e_1 \neq e_2$. Hence, $\mathrm{unfold}(\mathcal{D}^2, a^{\overline{c}}) \cong T_{\overline{c}}^2$.
- ε is connected to all words from $\$^* \otimes \{b^{e_1 e_2} \mid e_1 \neq e_2\}$. By (d) above, this means that the tree $\mathrm{unfold}(\mathcal{D}^2, \varepsilon)$ has $\aleph_0$ many subtrees isomorphic to $T[e_1, e_2]$ for $e_1, e_2 \in \mathbb{N}_+$, $e_1 \neq e_2$. Hence, $\mathrm{unfold}(\mathcal{D}^2, \varepsilon) \cong U_\omega^2$.
- b^m ($m \in \mathbb{N}_+$) is connected to all words from $\$^* \otimes \{b^{e_1 e_2} \mid e_1 \neq e_2 \text{ or } e_1 = e_2 > m\}$. By (d), this means that the tree $\mathrm{unfold}(\mathcal{D}^2, b^m)$ has $\aleph_0$ many subtrees isomorphic to $T[e_1, e_2]$ for all $e_1, e_2 \in \mathbb{N}_+$ with $e_1 \neq e_2$ or $e_1 = e_2 > m$. Thus, $\mathrm{unfold}(\mathcal{D}^2, b^m) \cong U_m^2$.

Hence, $\mathrm{unfold}(\mathcal{D}_2) \cong F^2$ and the roots are as required in Prop. 3.8. Moreover, it is clear that $\mathcal{D}_2$ is automatic.

3.2.2 Induction step: the automatic dag $\mathcal{D}^{i+1}$

Suppose $\mathcal{D}^i = (V, E)$ is such that $\mathcal{F}^i = \mathrm{unfold}(\mathcal{D}^i)$ is as described in Prop. 3.8. We use the notations from Sec. 3.1.2. We first build another automatic dag $\mathcal{D}'$, whose unfolding contains (copies of) all trees $U'_{\kappa,x}$ ($\kappa \in \mathbb{N}_+ \cup \{\omega\}$, $x \in \mathbb{N}_+$) and $T'_{\overline{c}xy}$ ($\overline{c} \in \mathbb{N}_+^k$, $x, y \in \mathbb{N}_+$). Recall that the set of roots of $\mathcal{D}^i$ is $\otimes_{k+2}(a^+) \cup b^* \subseteq V$. The universe of $\mathcal{D}'$ consists of the following regular set, where $\sharp$, $\sharp_1$, and $\sharp_2$ are new symbols:

$$(V \setminus b^*) \cup (\sharp^+ \otimes b^*) \cup \sharp_1^+ \sharp_2^* .$$

We have the following edges in $\mathcal{D}'$:

- All edges from E except those with an initial node in b^* are present in $\mathcal{D}'$.
- $a^{\overline{c}xy} \in V$ is connected to all words of the form $\sharp_1^i \sharp_2^{x-i}$ for $\overline{c} \in \mathbb{N}_+^k, x, y \in \mathbb{N}_+$, and $1 \le i \le x$. This ensures that the subtree rooted at $a^{\overline{c}xy}$ gets x new leaves, which are children of the root. Thus $\mathrm{unfold}(\mathcal{D}', a^{\overline{c}xy}) \cong T'_{\overline{c}xy}$.
- $\sharp^x \otimes b^m$ for $x \in \mathbb{N}_+$ and $m \in \mathbb{N}$ is connected to (i) all nodes to which b^m is connected in $\mathcal{D}^i$ and to (ii) all nodes from $\sharp_1^i \sharp_2^{x-i}$ for $1 \le i \le x$. This ensures that $\mathrm{unfold}(\mathcal{D}', \sharp^x \otimes b^m) \cong U'_{m,x}$ in case $m \in \mathbb{N}_+$ and $\mathrm{unfold}(\mathcal{D}', \sharp^x \otimes \varepsilon) \cong U'_{\omega,x}$.

In summary, $\mathcal{D}'$ is a dag, whose unfolding consists of (a copy of) $U'_{\omega,x}$ rooted at $\sharp^x \otimes \varepsilon$, $U'_{m,x}$ ($m \in \mathbb{N}_+$) rooted at $\sharp^x \otimes b^m$, and $T'_{\overline{c}xy}$ rooted at $a^{\overline{c}xy}$.

From the automatic dag $\mathcal{D}'$, we now build in a final step the automatic dag $\mathcal{D}^{i+1}$. This is very similar to the constructions of $\mathcal{D}^2$ and $\mathcal{D}'$ above. Let V' be the universe of $\mathcal{D}'$. The universe of $\mathcal{D}^{i+1}$ is the regular language

$$\otimes_k(a^+) \cup b^* \cup (\$^* \otimes V') .$$

The edges are as follows:

- For $u, v \in V'$, $\$^m \otimes u$ is connected to $\$^n \otimes v$ if and only if $m = n$ and (u, v) is an edge of $\mathcal{D}'$. This generates $\aleph_0$ many copies of $\mathcal{D}'$.
- $a^{\overline{c}}$ is connected to every word from $\$^* \otimes (\{a^{\overline{c}xy} \mid x, y \in \mathbb{N}_+\} \cup (\sharp^+ \otimes b^+))$. Hence, the tree $\mathrm{unfold}(\mathcal{D}^{i+1}, a^{\overline{c}})$ has $\aleph_0$ many subtrees isomorphic to $T'_{\overline{c}xy}$ for $x, y \in \mathbb{N}_+$ and $U'_{m,x}$ for $x, m \in \mathbb{N}_+$. Thus, $\mathrm{unfold}(\mathcal{D}^{i+1}, a^{\overline{c}}) \cong T_{\overline{c}}^{i+1}$.
- ε is connected to all words from $\$^* \otimes (\sharp^+ \otimes b^*)$. Hence, the tree $\mathrm{unfold}(\mathcal{D}^{i+1}, \varepsilon)$ has $\aleph_0$ many subtrees isomorphic to $U'_{\kappa,x}$ for all $x \in \mathbb{N}_+$ and $\kappa \in \mathbb{N}_+ \cup \{\omega\}$. Thus, $\mathrm{unfold}(\mathcal{D}^{i+1}, \varepsilon) \cong U_{\omega}^{i+1}$.
- b^m ($m \in \mathbb{N}_+$) is connected to all words from $\$^* \otimes ((\sharp^+ \otimes b^+) \cup \{\sharp^x \otimes \varepsilon \mid 1 \le x < m\})$. This means that the tree $\mathrm{unfold}(\mathcal{D}^{i+1}, b^m)$ has $\aleph_0$ many subtrees isomorphic to $U'_{m,x}$ for all $m, x \in \mathbb{N}_+$ and $U'_{\omega,x}$ for all $1 \le x < m$. Hence, $\mathrm{unfold}(\mathcal{D}^{i+1}, b^m) \cong U_m^{i+1}$.

This finishes the proof of Prop. 3.8.

Theorem 3.11 *For any $n \ge 2$, the isomorphism problem for automatic trees of height at most n is Π^0_{2n-3}-complete.*

The isomorphism problem for the class of automatic trees of finite height is recursively equivalent to $\mathsf{FOTh}(\mathbb{N}; +, \times)$.

Proof. We first prove the first statement. Containment in Π^0_{2n-3} was shown in Prop. 3.1. For the hardness, let $P_n \subseteq \mathbb{N}_+$ be any Π^0_{2n-3}-predicate and let $c \in \mathbb{N}_+$. Then, above, we constructed the automatic forest $\mathcal{F}^n$ of height n. The trees T_c^n and U_ω^n are first-order definable in $\mathcal{F}^n$ since they are (isomorphic to) the trees rooted at $a^{\overline{c}}$ and ε, resp. Hence these two trees are effectively automatic. By Prop. 3.6, they are isomorphic if and only if $P_n(c)$ holds.

We now come to the second statement. Since the proof of Prop. 3.1 is uniform in the level n, we can compute from two automatic trees T_1, T_2 of finite height an arithmetical formula, which is true if and only if $T_1 \cong T_2$. The other direction follows from the first statement because of the uniformity in constructing the trees T_c^n and U_ω^n. ☐

From Thm. 3.11 we can easily deduce a corollary on automatic equivalence structures. An equivalence structure is of the form $\mathcal{E} = (D; E)$ where E is an equivalence relation on D.

Corollary 3.12 *The isomorphism problem for automatic equivalence structures is Π^0_1-complete.*

Proof. By Thm. 3.11 for $k = 2$ it suffices to show that the isomorphism problem for $\mathcal{T}_2$ is recursively equivalent to the isomorphism problem for automatic equivalence structures. First, let $\mathcal{E} = (V; \equiv)$ be an automatic equivalence structure and let $\le_{\mathrm{lex}}$ be the length-lexicographic order on V. Now build the tree $T(\mathcal{E})$ of height at most 2 as follows: Let r be a new letter that serves as root. Its children are the $\le_{\mathrm{lex}}$-minimal elements u of the equivalence classes of $\equiv$, and the children of u are the remaining elements of the equivalence class $[u]$. It is clear that $T(\mathcal{E})$ is a tree of height at most 2. Moreover, if $\mathcal{E}$ is automatic, then also $T(\mathcal{E})$ is automatic and an automatic presentation for $T(\mathcal{E})$ can be computed from an automatic presentation for $\mathcal{E}$. Finally, $\mathcal{E}_1 \cong \mathcal{E}_2$ if and only if $T(\mathcal{E}_1) \cong T(\mathcal{E}_2)$. This gives us a reduction from the isomorphism problem for automatic equivalence structures to the isomorphism problem for $\mathcal{T}_2$.

For the reverse reduction, let T be a tree of height 2. We construct an equivalence structure $\mathcal{E}(T)$ as follows: W.l.o.g. assume that T is not a single node. Then we first add to each child of the root of T a further child. This ensures that every maximal path in T has length 2. Let T' be the resulting tree. Then the elements of $\mathcal{E}(T)$ are the leaves of T' and two leaves u and v are equivalent if and only if the have the same parent node. Again it is easy to see that: (i) If T is automatic then also $\mathcal{E}(T)$ is automatic and an automatic presentation for $\mathcal{E}(T)$ can be computed from an automatic presentation for T. (ii) $\mathcal{T}_1 \cong \mathcal{T}_2$ if and only if $\mathcal{E}(T_1) \cong \mathcal{E}(T_2)$. ☐

Let us close this section, with a brief discussion on the isomorphism problem for computable trees of finite height.

Theorem 3.13 *For every $n \geq 1$, the isomorphism problem for computable trees of height at most n is Π^0_{2n}-complete.*

Proof. For the upper bound, let us first assume that $n = 1$. Two computable trees T_1 and T_2 of height 1 are isomorphic if and only if: for every $k \geq 0$, there exist at least k nodes in T_1 if and only if there exist at least k nodes in T_2. This is a Π^0_2-statement. For the inductive step, we can reuse the arguments from the proof of Prop. 3.1.

For the lower bound, we first deal with the case $n = 1$. It is known that the problem whether a given recursively enumerable set is infinite is Π^0_2-complete [20]. For a given deterministic Turing-machine M, we construct a computable tree T_M of height 1 as follows: the set of leaves of T_M is the set of all accepting computations of M. We add a root to the tree and connect the root to all leaves. If $L(M)$ is infinite, then T_M is isomorphic to the height-1 tree with infinitely many leaves. If $L(M)$ is finite, then there exists $m \in \mathbb{N}$ such that T_M is isomorphic to the height-1 tree with m leaves. We can use this construction as the base case for our construction in Sec. 3.1.2. This yields the lower bound for all $n \geq 1$. ☐

4 Automatic Linear Orders

Our main result for automatic linear orders is:

Theorem 4.1 *The isomorphism problem for the class of automatic linear orders is at least as hard as $\mathsf{FOTh}(\mathbb{N}; +, \times)$.*

The proof of this result follows our arguments for trees of finite height but is technically more involved. Looking back to the proof of Thm. 3.11, we see that trees are used in order to encode sets of sets of ... sets of natural numbers. For linear orders, we replace the basic tree operation of gluing together a set of trees into a single tree by adding a new root by the *shuffle sum*. The shuffle sum of a countable set of linear order types $\mathcal{L}$ is constructed as follows: First, we densely color $\mathbb{Q}$ with the order types in $\mathcal{L}$, i.e., for all rationals $x < y$ and all $L \in \mathcal{L}$ there exists $x < z < y$ such that z is colored with the order type L (it doesn't matter which dense coloring we choose). The shuffle sum of $\mathcal{L}$ is the linear order that results from $(\mathbb{Q}, <)$ by replacing each L-colored rational ($L \in \mathcal{L}$) with the order L. Assuming that every order type in $\mathcal{L}$ starts with some ordinal $\omega \cdot i$ ($i \in \mathbb{N}$) and does not contain $\omega \cdot i$ as an interval elsewhere, the shuffle sum of $\mathcal{L}$ encodes the set $\mathcal{L}$ as a linear order. In our proof of Thm. 4.1 we use iterated shuffle sums. In order to stay within automatic linear orders, we have to realize shuffle sums in an automatic way, details can be found in the complete version [15] of this paper.

In [13], it is shown that every linear order has finite FC-rank. We do not define the FC-rank of a linear order in general, see e.g. [13]. A linear order L has FC-rank 1, if after identifying all $x, y \in L$ such that the interval $[x, y]$ is finite, one obtains a dense ordering or the singleton linear order. The result of [13] mentioned above suggests that the isomorphism problem might be simpler for linear orders of low FC-rank. We now prove that this is not the case:

Corollary 4.2 *The isomorphism problem for automatic linear orders of FC-rank 1 is at least as hard as $\mathsf{FOTh}(\mathbb{N}; +, \times)$.*

Proof. We provide a reduction from the isomorphism problem for automatic linear orders of arbitrary rank. If L is an automatic linear order, then so is $\widetilde{L} = ((-1, 0] + [1, 2)) \cdot L$. This linear order is obtained from L by replacing each point with a copy of the rational numbers in $(-1, 0] \cup [1, 2)$. Then $\widetilde{L}$ has FC-rank 1: Only the copies of 0 and 1 will be identified, and the resulting order is isomorphic to $(\mathbb{Q}, \leq)$. Moreover, L is isomorphic to the set of all $x \in \widetilde{L}$ satisfying $\exists z > x \, \forall y : (x < y \leq z \to y = z)$. Hence $L_1 \cong L_2$ if and only if $\widetilde{L}_1 \cong \widetilde{L}_2$, which completes the reduction. ☐

5 Arithmetical isomorphisms

We conclude this paper with an application of Thms. 3.11 and 4.1. The following corollary shows that although automatic structures look simple (especially for automatic trees), there may be no "simple" isomorphism between two automatic copies of the same structure. An isomorphism f between two automatic structures with domains L_1 and L_2, resp., is a Σ^0_k-isomorphism, if the set $\{(x, f(x)) \mid x \in L_1\}$ belongs to Σ^0_k.

Corollary 5.1 *For any $k \in \mathbb{N}$, there exist two isomorphic automatic trees of finite height (and two automatic linear orders) without any Σ^0_k-isomorphism.*

Proof. Assume that between any two isomorphic automatic trees of finite height, there always exists a Σ^0_k-isomorphism. Then the isomorphism problem for automatic trees of finite height would belong to Σ^0_{k+2} (which contradicts Thm. 3.11): two automatic trees $T_1 = (D_1; E_1)$ and $T_2 = (D_2; E_2)$ of finite height are isomorphic if *there exists a Σ^0_k-predicate $P(x, y)$ such that for all $x_1, x_2 \in D_1$ there exist $y_1, y_2 \in D_2$ (and vice versa) such that:* $P(x_1, y_1)$, $P(x_2, y_2)$, $(x_1 = x_2 \leftrightarrow y_1 = y_2)$, and $((x_1, x_2) \in E_1 \leftrightarrow (y_1, y_2) \in E_2)$. Since P is a Σ^0_k-predicate, this is a Σ^0_{k+2}-statement, which expresses the existence of a Σ^0_k-isomorphism from T_1 to T_2. For linear orders we can argue in the same way. ☐

6 Open problems

The main open problem, which remains, is the precise complexity of the isomorphism problem for automatic linear orders. Is this problem Σ^1_1-complete or does it belong to the hyperarithmetical hierarchy (which makes up $\Sigma^1_1 \cap \Pi^1_1$)? Another interesting problem is the isomorphism problem for automatic well-founded trees (trees without an infinite path). In the proof of [12] (Σ^1_1-completeness of the isomorphism problem for automatic successor trees), trees with infinite paths arise. Finally, it seems to be open, whether the isomorphism problem for automatic groups (in the sense of [10] and not [5]) is decidable.

References

[1] V. Bárány, L. Kaiser, and S. Rubin. Cardinality and counting quantifiers on omega-automatic structures. In *STACS 2008*, pages 385–396. IFIB Schloss Dagstuhl, 2008.

[2] A. Blumensath and E. Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory Comput. Syst.*, 37(6):641–674, 2004.

[3] W. Calvert and J. F. Knight. Classification from a computable viewpoint. *Bull. Symbolic Logic*, 12(2):191–218, 2006.

[4] C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Am. Math. Soc.*, 98:21–51, 1961.

[5] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, and W. P. Thurston. *Word processing in groups*. Jones and Bartlett, Boston, 1992.

[6] S. S. Goncharov and J. F. Knight. Computable structure and antistructure theorems. *Algebra i Logika*, 41(6):639–681, 2002.

[7] B. R. Hodgson. On direct products of automaton decidable theories. *Theoret. Comput. Sci.*, 19:331–335, 1982.

[8] J. Honkala. On the problem whether the image of an N-rational series equals N. *Fund. Inform.*, 73(1-2):127–132, 2006.

[9] B. Khoussainov and M. Minnes. Model theoretic complexity of automatic structures. In *TAMC 08*, LNCS 4978, pp 514–525, Springer, 2008.

[10] B. Khoussainov and A. Nerode. Automatic presentations of structures. In *LCC: International Workshop on Logic and Computational Complexity*, LNCS 960, pages 367–392, Springer 1995.

[11] B. Khoussainov and A. Nerode. Open questions in the theory of automatic structures. *Bulletin of the EATCS*, 94, pages 184–204, 2008.

[12] B. Khoussainov, A. Nies, S. Rubin, and F. Stephan. Automatic structures: richness and limitations. *Log. Methods Comput. Sci.*, 3(2):2:2, 18 pp. (electronic), 2007.

[13] B. Khoussainov, S. Rubin, and F. Stephan. Automatic linear orders and trees. *ACM Trans. Comput. Log.*, 6(4):675–700, 2005.

[14] D. Kuske and M. Lohrey. Some natural decision problems in automatic graphs. *J. Symbolic Logic*, 2009. To appear.

[15] D. Kuske, J. Liu, and M. Lohrey. The isomorphism problem on classes of automatic structures. `http://arxiv.org/abs/1001.2086`

[16] Y. V. Matiyasevich. *Hilbert's Tenth Problem*. MIT Press, Cambridge, Massachusetts, 1993.

[17] A. Nies. Describing groups. *Bull. Symbolic Logic*, 13(3):305–339, 2007.

[18] S. Rubin. *Automatic Structures*. PhD thesis, University of Auckland, 2004.

[19] S. Rubin. Automata presenting structures: A survey of the finite string case. *Bull. Symbolic Logic*, 14:169–209, 2008.

[20] R. I. Soare. *Recursively enumerable sets and degrees*. Perspectives in Mathematical Logic. Springer, 1987.

[21] T. Tsankov. The additive group of the rationals does not have an automatic presentation. `http://arxiv.org/abs/0905.1505`