Climbing up the Elementary Complexity Classes with Theories of Automatic Structures

Faried Abu Zaid
TU Ilmenau, Germany

Dietrich Kuske
TU Ilmenau, Germany

Peter Lindner
RWTH Aachen University, Germany

Abstract

Automatic structures are structures that admit a finite presentation via automata. Their most prominent feature is that their theories are decidable. In the literature, one finds automatic structures with non-elementary theory (e.g., the complete binary tree with equal-level predicate) and automatic structures whose theories are at most 3-fold exponential (e.g., Presburger arithmetic or infinite automatic graphs of bounded degree). This observation led Durand-Gasselin to the question whether there are automatic structures of arbitrary high elementary complexity.

We give a positive answer to this question. Namely, we show that for every $h \geq 0$ the forest of (infinitely many copies of) all finite trees of height at most $h+2$ is automatic and it's theory is complete for $\mathrm{STA}(*, \exp_h(n, \mathrm{poly}(n)), \mathrm{poly}(n))$, an alternating complexity class between h-fold exponential time and space. This exact determination of the complexity of the theory of these forests might be of independent interest.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic

Keywords and phrases Automatic Structures, Complexity Theory, Model Theory

Digital Object Identifier 10.4230/LIPIcs.CSL.2018.3

1 Introduction

The idea of an automatic structure goes back to Büchi and Elgot who used finite automata to decide, e.g., Presburger arithmetic [6]. In essence, a structure is automatic if the elements of the universe are strings form a regular language and every relation of the structure is synchronously-rational [11]. The notion was introduced in [13] and a systematic study was initiated by Khoussainov and Nerode [15] and started to attract quite some interest with the work by Blumensath and Grädel [3, 4], see the surveys [23, 1, 24, 14]. One of the main motivations for investigating automatic structures is that their first-order theories are decidable. This decidability holds even if one extends first-order logic by quantifiers "there exist infinitely many" [3], "the number of elements satisfying φ is a finite multiple of p" [16], and "there exists an infinite relation satisfying φ" (provided φ mentions the infinite relation only negatively) [19].

Already in [3, 4], the authors observe that the first-order theory of an automatic structure is, in general, non-elementary (i.e., does not belong to n-EXPSPACE for any $n \in \mathbb{N}$). The simplest example is provided by the set of binary words with the prefix relation, the two successor relations, and the equal-length predicate. An inspection of the decidability proof for arbitrary automatic structures shows that validity of a formula in Σ_{n+1} can be decided in n-EXPSPACE. Note that this problem has two inputs: a formula from Σ_{n+1} and an

27th EACSL Annual Conference on Computer Science Logic (CSL 2018).
Editors: Dan Ghica and Achim Jung; Article No. 3; pp. 3:1–3:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

automatic structure (given by a tuple of automata). In [18], it is shown that fixing one of the two inputs does not make the problem simpler. In other words: both the expression and the data complexity are complete for n-EXPSPACE.

On the positive side, there are also automatic structures whose theories are much simpler. One example is Presburger's arithmetic, i.e., the structure $(\mathbb{N}, +)$ that is automatic [6] and has a theory in 2-EXPSPACE [22, 8]. Another example are automatic structures of bounded degree [20] whose theories are in 2-EXPSPACE. Finally, let us mention structures, which have an automatic presentation over a unary alphabet, e.g. the natural Numbers with successor $(\mathbb{N}, S)$. The first-order theory of every such structure is decidable in polynomial time [17].

To the authors' knowledge, no automatic structure is known whose theory is elementary but not in 2-EXPSPACE. In this article, we provide such examples. More precisely, for any $h \in \mathbb{N}$, we provide an automatic structure whose theory is complete for the class of problems that can be decided in h-fold exponential time with polynomially many alternations, i.e., for Berman's complexity class $\mathrm{STA}(*, \exp_h(2, \mathrm{poly}(n)), \mathrm{poly}(n))$ [2].

This structure is the forest F_{h+2} consisting of countably many copies of all trees of height at most $h + 2$. Containment in $\mathrm{STA}(*, \exp_h(2, \mathrm{poly}(n)), \mathrm{poly}(n))$ is shown as follows: Let φ be a first-order sentence of quantifier rank r. In a first step, we show that any tree of height $\leq h + 2$ is indistinguishable from some tree of size h-fold exponential in r by any formula of quantifier rank r. Consequently, to determine the truth of the sentence φ in the forest F_{h+2}, it suffices to determine it in a forest whose trees have size h-fold exponential in r. Since the elements of this forest can be described by words of h-fold exponential size, its model checking can be done in the said complexity class.

For the lower bound, we first reduce any problem in the said complexity class to the theory of the free monoid where quantification is restricted to words of h-fold exponential length. This theory is then reduced to the theory of the forest F_{h+2}. This second step is based on an encoding of h-fold exponential numbers and their addition in the forest.

Thus, technically, the main result of this paper is the complete characterisation of the complexity of the theory of the forest F_{h+2}. Since this forest is automatic, we get an affirmative answer to the open question from the theory of automatic structures. Besides this, the forest F_{h+2} is a natural structure, so that our result can have consequences in other contexts as well.

The results presented in this paper close the gap that was left open in the third author's master thesis [21].

2 Preliminaries

The set of natural numbers is denoted $\mathbb{N} = \{0, 1, 2, \ldots\}$; $\mathbb{N}_{>0} = \{1, 2, 3, \ldots\}$ denotes the positive natural numbers. For $m, n, r \in \mathbb{N}$ we write $m =_r n$ if $m = n$ or $m, n \geq r$. Inductively, we define the class of functions $\exp_m : \mathbb{N}^2 \to \mathbb{N}$ for $m, c, n \in \mathbb{N}$:

$$\exp_m(c, n) = \begin{cases} n & \text{if } m = 0 \\ c^{\exp_{m-1}(c,n)} & \text{if } m > 0 \end{cases}$$

Intuitively, $\exp_m(c, n)$ is a stack of cs of height m with the number n on top of this stack. By $\mathrm{poly}(n)$ we denote the class of all polynomial functions $\mathbb{N} \to \mathbb{N}$.

We assume that the reader is familiar with the basics of automata theory and formal logic, especially first-order logic. We use this section to recall some of the key notions in order to fix our notation.

A *(directed) graph* is a tuple $G = (V, E)$, where V is a set and $E \subseteq V \times V \setminus \{(v, v) \mid v \in V\}$ is a binary irreflexive relation. A *tree* is a finite graph $T = (V, E)$ such that, for some node $r \in V$, any node $v \in V$ has precisely one path from r to v. The node r, being unique, is called the *root* of T. Now let $T = (V, E)$ be a tree and $v \in V$. The *depth of* v is the length of the path from r to v (i.e., the number of edges such that the depth of the root is 0). The *height of* v is the maximal length of a path starting in v. A node v is a *leaf* if its height is 0. The *height of* T is the height of the root r or, equivalently, the maximal depth of a node in T. A subtree is an induced subgraph of a tree $T = (V, E)$ whose vertex set is of the form $\{w \in V \mid w \text{ is reachable from } v\}$ for some node $v \in V$. Note that v is the root of this subtree and every subtree is uniquely dertermined by its root. Therefore we denote the subtree with root v by T_v.

An *automatic graph* is a graph $G = (V, E)$ such that $V \subseteq \Sigma^*$ is a regular language over some alphabet Σ and the edge relation E is synchronously rational [11].

First-order formulas (over the language of graphs) are build up from variables $\{x_i \mid i \in \mathbb{N}\}$, the Boolean connectives $\{\neg, \vee, \wedge, \rightarrow\}$, the edge relation symbol E, quantifiers $\{\forall, \exists\}$, and the bracket symbols $\{(,)\}$. The *quantifier rank* $\mathrm{qr}(\varphi)$ of a formula φ is the maximal nesting depth of quantifiers within φ. Two graphs G and H are r*-equivalent* (denoted $G \equiv_r H$) if they cannot be distinguished by any formula of quantifier rank $\leq r$. For a tuple $\overline{a} = (a_1, \ldots, a_k) \in A^k$ and $B \subseteq A$ let $\overline{a}\restriction_B$ denote the restriction of $\overline{a}$ to the components in B, i.e. the tuple $(a_{i_1}, \ldots, a_{i_\ell})$ with $\{i_1, \ldots, i_k\} = \{i \mid a_i \in B\}$ and $i_1 < i_2 < \cdots < i_\ell$.

The *Ehrenfeucht-Fraïssé-game* is a game-theoretic characterisation of elementary equivalence. It is played on two graphs G and H, where the two players, Spoiler and Duplicator, choose alternately elements of these two structures for a prescribed number of rounds. More precisely the i-th round of an r-round Ehrenfeucht-Fraisse-game on $G = (V^G, E^G)$ and $H = (V^H, E^H)$ ($G_r(G, H)$) has the following form: First Spoiler picks an element a_i from G or an element b_i from H. Duplicator answers by choosing an element b_i from H or an element a_i from G, respectively. Therefore the two players iteratively construct two tuples $(a_1, \ldots, a_r) \in (V^G)^r$ and $(b_1, \ldots, b_r) \in (V^H)^r$. Duplicator wins if the mapping $a_i \mapsto b_i$ is a partial isomorphism, that is if $a_i = a_j \Leftrightarrow b_i = b_j$ and $(a_i, a_j) \in E^G \Leftrightarrow (b_i, b_j) \in E^H$ for all $1 \leq i, j \leq r$. Otherwise Spoiler wins.

▶ **Theorem 1** ([5]). *Let G and H be two graphs. Then Duplicator has a winning strategy in the game $G_r(G, H)$ if, and only if, $G \equiv_r H$.*

The main object of study in this paper is the following forest:

▶ **Definition 2.** For $H \in \mathbb{N}$, let F_H denote the disjoint union of $\aleph_0$ many copies of all trees of height at most H.

Thus, F_H is the forest of all trees of height at most H, containing countably many copies of every such tree.

▶ **Remark 3.** *Natural variants of this forest are, among others, the following:*

- *The disjoint union F_H^∞ of $\aleph_0$ many copies of all countably infinite (or at most countably infinite) trees of height at most H.*
- *The disjoint union F_H^1 of all finite (or at most countably infinite) trees of height at most H up to isomorphism (i.e., one tree per isomorphism class).*

We will show that F_H is automatic which is not the case for F_H^∞ (it is ω-automatic) and we conjecture that also F_H^1 is not automatic.

Nevertheless, the proofs of the complexity results can easily be transformed to show that also the theories of these forests are complete for $\mathrm{STA}(*, \exp_{H-2}(2, \mathrm{poly}(n)), \mathrm{poly}(n))$.

3 Trees of Bounded Height

The goal of this section is to provide an automatic copy of the forrest F_H for every $H \in \mathbb{N}$. The idea is to use XML-like notation to describe a tree and to encode an element by marking its position in the tree that it belongs to. Because the nesting-depth of parentheses will be bounded for every H, the resulting languages remain regular. Let $\Sigma = \{\langle\rangle, \langle\backslash\rangle, \langle x\rangle, \langle\backslash x\rangle\}$. We define regular languages J_H and K_H for every $H \in \mathbb{N}$:

$$J_0 = \{\langle\rangle\langle\backslash\rangle\}$$
$$K_0 = \{\langle x\rangle\langle\backslash x\rangle\}$$

and

$$J_{H+1} = \langle\rangle J_H^* \langle\backslash\rangle$$
$$K_{H+1} = \langle x\rangle J_H^* \langle\backslash x\rangle \cup \langle\rangle J_H^* K_H J_H^* \langle\backslash\rangle.$$

Every word in $w \in K_H$ contains the tag $\langle x\rangle \ldots \langle\backslash x\rangle$ exactly once. This tag marks the selected node in the tree that is presented by w.

Next we show that the edge relation on K_H is synchronously-rational [11]. Two nodes u and v from F_H are connected by a directed edge if, and only if, they belong to the same tree and u is the parent of v. To describe the edge relation E_H of our automatic copy, write $L^{\odot 2} = \left\{ \binom{w}{w} : w \in L \right\}$ for any language L. Then we have

$$E_0 = \emptyset$$
$$E_1 = \binom{\langle x\rangle}{\langle\rangle} \begin{pmatrix} \langle\rangle & \langle\backslash\rangle \\ \langle\rangle & \langle\backslash\rangle \end{pmatrix}^* \begin{pmatrix} \langle\rangle & \langle\backslash\rangle \\ \langle x\rangle & \langle\backslash x\rangle \end{pmatrix} \begin{pmatrix} \langle\rangle & \langle\backslash\rangle \\ \langle\rangle & \langle\backslash\rangle \end{pmatrix}^* \binom{\langle\backslash x\rangle}{\langle\backslash\rangle}$$
$$E_{H+2} = \binom{\langle x\rangle}{\langle\rangle} (J_{H+1}^{\odot 2})^* \binom{\langle\rangle}{\langle x\rangle} (J_H^{\odot 2})^* \binom{\langle\backslash\rangle}{\langle\backslash x\rangle} (J_{H+1}^{\odot 2})^* \binom{\langle\backslash x\rangle}{\langle\backslash\rangle}$$
$$\cup \binom{\langle\rangle}{\langle\rangle} (J_{H+1}^{\odot 2})^* E_{H+1} (J_{H+1}^{\odot 2})^* \binom{\langle\backslash\rangle}{\langle\backslash\rangle}.$$

Note that the languages that we defined so far do not induce an isomorphic copy of F_H. We need to modify the languages such that every tree of height at most H will appear infinitely often. Therefore let $L_H = \$^* K_H$ and $E'_H = \binom{\$}{\$}^* E_H$. Then $(L_H, E'_H) \cong F_H$ is an automatic copy of F_H.

4 Upper Bound

We provide a simple decision procedure for the theory of F_{H+2} that runs in alternating H-fold exponential time while making only polynomially many alternations. We found it more convenient to first prove this result in the realm of order trees: An *order tree* is a finite partial order $(V, \leq)$ with a minimal element such that, for any $v \in V$, the set $\{w \in V \mid w \leq v\}$ is finite and linearly ordered by $\leq$. An *order forest* is a disjoint union of order trees. The *length* of an order forest is the maximal size of a linearly ordered subset, its *height* is the predecessor of its length.

Let oF_h denote the order version of the forest F_h, i.e., the disjoint union of infinitely many copies of any order tree of height $\leq H$. The theory of this order forest can be decided as follows: We determine from a sentence φ of quantifier rank r a finite order forest satisfying

φ iff $\mathrm{oF}_{H+2} \models \varphi$. The size of this order forest can be bounded since, as we show below, every finite order tree of height $\leq H+2$ is r-equivalent to an order tree of size at most $\exp_{H+1}(r+1, \mathrm{poly}(n+1))$. The elements of this finite order forest have encodings by words of length $\leq \exp_{H+1}(r+1, \mathrm{poly}(n+1))$. Then, the standard alternating model checking algorithm is applied to this forest (without computing it explicitely). The result on the forest F_{H+2} follows because of a polynomial-time reduction of the theory of the forest F_{H+2} to that of the order forest oF_{H+2}.

The following lemma on order forests prepares the construction of a "small" equivalent order tree.

▶ **Lemma 4.** *Let $(S_i)_{i\in I}$ and $(T_j)_{j\in J}$ be nonempty (possibly infinite) families of order trees such that*

$$|\{i \in I \mid S_i \in \tau\}| =_r |\{j \in J \mid T_j \in \tau\}| \tag{1}$$

holds for any $\equiv_r$-equivalence class τ. Then

$$\biguplus_{i\in I} S_i \equiv_r \biguplus_{j\in J} T_j \,. \tag{2}$$

Proof. We show that Duplicator has a winning strategy in the r-round Ehrenfeucht-Fraïssé-game on the forests $S = \biguplus_{i\in I} S_i$ and $T = \biguplus_{j\in J} T_j$. More precisely we show that Duplicator can maintain the following invariant after $\ell \in \{0,1,\ldots,r\}$ rounds (when the current position is $(\overline{a}, \overline{b})$):

> For all $i \in I$, there exists $j \in J$ such that for all $k \in \{1,2,\ldots,\ell\}$, we have $a_k \in S_i \iff b_k \in T_j$ and $(S_i, \overline{a}{\restriction_{S_i}}) \equiv_{r-\ell} (T_j, \overline{b}{\restriction_{T_j}})$.

Since no edge connects distinct trees in a forest, every position $(a_1,\ldots,a_r,b_1,\ldots,b_r)$ satisfying this invariant describes a partial isomorphism $a_i \mapsto b_i$. Therefore it remains to be shown that Duplicator can maintain this invariant.

So let $0 \leq \ell < r$, $a_1,\ldots,a_\ell \in S$, and $b_1,\ldots,b_\ell \in T$ such that the invariant holds. Note that the invariant is equivalent to its dual:

> For all $j \in J$, there exists $i \in I$ such that for all $k \in \{1,2,\ldots,\ell\}$, we have $b_k \in T_i \iff a_k \in S_j$ and $(T_j, \overline{a}{\restriction_{T_J}}) \equiv_{r-\ell} (S_i, \overline{a}{\restriction_{S_i}})$.

Hence, by symmetry, we can assume that Spoiler chooses an element $a_{\ell+1}$ of S in round $\ell+1 \leq r$. Then there is $i \in I$ such that $a_{\ell+1}$ is a node from S_i. We distinguish two cases: either there is $k \in \{1,2,\ldots,\ell\}$ with $a_k \in S_i$ or there is no such k.

First, assume $a_k \in S_i$ for some $1 \leq k \leq \ell$. By the induction hypothesis, there exists $j \in J$ with $b_k \in T_j$ and $(S_i, \overline{a}{\restriction_{S_i}}) \equiv_{r-\ell} (T_j, \overline{b}{\restriction_{T_j}})$. Hence, there is $b_{\ell+1} \in T_j$ with $(S_i, \overline{a}a_{\ell+1}{\restriction_{S_i}}) \equiv_{r-\ell-1} (T_j, \overline{b}b_{\ell+1}{\restriction_{T_j}})$. Chosing this element $b_{\ell+1}$, Duplicator can move the play into a position that satisfies the invariant.

Now consider the second case, $a_k \notin S_i$ for all $1 \leq k \leq \ell$. Let $I' = \{i' \in I \mid S_i \equiv_r S_{i'}\}$ and, similarly, $J' = \{j' \in J \mid S_i \equiv_r T_{j'}\}$. If $|I'| = |J'|$, the invariant implies the existence of $j \in J'$ such that no element b_k belongs to T_j. Otherwise, we have $|J'| \geq r$ by (1). Since only $\ell < r$ many nodes b_k have been chosen so far, also in this case there exists $j \in J'$ such that no element b_k belongs to T_j. Because of $S_i \equiv_r T_j$, the tree T_j has some element $b_{\ell+1}$ with $(S_i, a_{\ell+1}) \equiv_{r-1} (T_j, b_{\ell+1})$ (and therefore also $(S_i, a_{\ell+1}) \equiv_{r-\ell-1} (T_j, b_{\ell+1})$). Thus, also in this case, Duplicator can move the play into a position that satisfies the invariant. ◀

▶ **Lemma 5.** *Let $r, h \in \mathbb{N}$. There exists a polynomial function $p_h\colon \mathbb{N} \to \mathbb{N}$ such that the following holds: For any order tree S of height $\leq h$, there exists an $\equiv_r$-equivalent order tree T of height $\leq h$ and size*

$$\leq \begin{cases} p_h(r+1) & \text{if } h \leq 2 \\ \exp_{h-2}(r+1, p_h(r+1)) & \text{if } h > 2\,. \end{cases}$$

Proof. For each $h, r \in \mathbb{N}$, we let $\equiv_r^h$ denote the restriction of the relation $\equiv_r$ to order trees of height $\leq h$.

By induction on h, we prove in addition

$$\operatorname{index}(\equiv_r^h) \leq \begin{cases} 1 & \text{if } h = 0 \\ \exp_{h-1}(r+1, r+1) & \text{if } h \geq 1\,. \end{cases}$$

For $h = 0$, there is only one order tree of height h and this tree has size 1, hence we set $p_0(x) = 1$. Furthermore, $\operatorname{index}(\equiv_r^0) = 1$ is obvious.

Now let $h > 0$ and let S be some order tree of height h. Let I denote the set of nodes of depth 1 and, for $i \in I$, let S_i denote the subtree of S rooted at i. By the induction hypothesis, any $\equiv_r^{h-1}$-equivalence class τ contains some order tree T_τ of size $\leq p_{h-1}(r+1)$ (if $h \leq 3$) and $\leq \exp_{h-3}(r+1, p_{h-1}(r+1))$ otherwise. For $i \in I$, let $T_i = T_{[S_i]}$ be the representative of the $\equiv_r^{h-1}$-class of S_i. Let $J \subseteq I$ such that

$$\min\bigl(r, |\{i \in I \mid S_i \in \tau\}|\bigr) = |\{j \in J \mid T_j \in \tau\}|$$

for any $\equiv_r$-equivalence class τ. Then (1) from Lemma 4 holds, implying $\biguplus_{i\in I} S_i \equiv_r \biguplus_{j \in J} T_j$ by Lemma 4. Let the order tree T arise from the order forest $\biguplus_{j\in J} T_j$ by the addition of a root that is smaller than any other node. Note that T is quantifier free definable in the disjoint sum of $\biguplus_{j \in J} T_j$ and a single node.[1] Since S arises in the same way from the order forest $\biguplus_{i\in I} S_i$, we get $S \equiv_r T$ [7].

Next, we prove the upper bound for the size of the order tree T. Note that this size is at most $|J|$ multiplied with the maximal size of an order tree T_j. Since J contains at most r elements per $\equiv_r^{h-1}$-equivalence class, we obtain

$$\begin{aligned} |J| &\leq r \cdot \operatorname{index}(\equiv_r^{h-1}) \\ &\leq r \cdot \begin{cases} 1 & \text{if } h = 1 \\ r+1 & \text{if } h = 2 \\ \exp_{h-2}(r+1, r+1) & \text{if } h \geq 3\,. \end{cases} \end{aligned}$$

Since the size of the order trees T_j is bounded as described above, the size of the order tree T is

$$\begin{aligned} &\leq r \cdot \begin{cases} 1 \cdot p_0(r+1) & \text{if } h = 1 \\ (r+1) \cdot p_1(r+1) & \text{if } h = 2 \\ \exp_1(r+1, r+1) \cdot p_2(r+1) & \text{if } h = 3 \\ \exp_{h-2}(r+1, r+1) \cdot \exp_{h-3}(r+1, p_{h-1}(r+1)) & \text{if } h > 3 \end{cases} \\ &\leq \begin{cases} p_h(r+1) & \text{if } h \leq 2 \\ \exp_{h-2}(r+1, p_h(r+1)) & \text{if } h \geq 3 \end{cases} \end{aligned}$$

for a suitably chosen polynomial function p_h. This proves the claim from the lemma.

---

[1] Here we need order trees since this does not hold for successor trees (V, E).

It remains to prove the additional inductive invariant on the number of equivalence classes of $\equiv_r^h$. Note that the order tree T constructed above is completely given by a mapping from the $\equiv_r^{h-1}$-equivalence classes into the set of numbers $\{0, 1, \ldots, r\}$. Hence, the number of distinct order trees T that can arise in the above way, is

$$\begin{aligned}
&\leq (r+1)^{\mathrm{index}(\equiv_r^{h-1})} \\
&\leq \begin{cases} (r+1) & \text{if } h = 1 \\ (r+1)^{\exp_{h-2}(r+1, r+1)} & \text{if } h > 1 \end{cases} \\
&= \exp_{h-1}(r+1, r+1)\,.
\end{aligned}$$ ◀

For $r, k \in \mathbb{N}$, we let $\mathrm{oF}_h^{r,k}$ denote the disjoint union of r copies of every order tree of height $\leq h$ and size $\leq k$.

▶ **Proposition 6.** *Let $r, h \in \mathbb{N}$. There exists a polynomial function $p_h\colon \mathbb{N} \to \mathbb{N}$ such that $\mathrm{oF}_h \equiv_r \mathrm{oF}_h^{r,k}$ with*

$$k = \begin{cases} p_h(r+1) & \text{if } h \leq 2 \\ \exp_{h-2}(r+1, p_h(r+1)) & \text{if } h > 2\,. \end{cases}$$

Proof. Let τ be some $\equiv_r$-equivalence class containing some order tree S of height $\leq h$. The order forest oF_h contains infinitely many copies of S. By Lemma 5, there exists an order tree T in $\mathrm{oF}_h^{r,k}$ with $T \in \tau$. More precisely, there are $\geq r$ such order trees (possibly isomorphic). From Lemma 4, we obtain $F_h \equiv_r F_{r,k}^h$. ◀

▶ **Corollary 7.** *For $H \in \mathbb{N}$, the theory of oF_{H+2} belongs to $\mathrm{STA}(*, \exp_H(2, \mathrm{poly}(n)), \mathrm{poly}(n))$.*

Proof. Let φ be a sentence of size n. Without loss of generality, we assume φ to be in prenex normal form. Let furthermore p be the polynomial p_{H+2} from Proposition 6.

The quantifier rank of φ is $\leq n$. Hence, by Proposition 6, it suffices to decide whether φ holds in the finite order forest $\mathrm{oF}_{H+2}^{n,k}$ with $k = \exp_H(r+1, p(r+1))$. Using the encoding of F_{H+2} as automatic structure, the elements of $\mathrm{oF}_{H+2}^{n,k}$ can be encoded as strings of length $\mathcal{O}(n+k)$. Hence the standard alternating model-checking algorithm for first-order logic uses time $\mathcal{O}(\mathrm{poly}(n+k))$ and $\leq n$ alternations. Note that this algorithm does not calculate the order forest $\mathrm{oF}_{H+2}^{n,k}$ explicitely, but only handles words of length $\mathcal{O}(n+k)$. ◀

As a consequence, we get the following result about the forest F_{H+2}.

▶ **Theorem 8.** *For $H \in \mathbb{N}$, the theory of F_{H+2} belongs to $\mathrm{STA}(*, \exp_H(2, \mathrm{poly}(n)), \mathrm{poly}(n))$.*

Proof. We reduce this theory to the theory of the ordered forest oF_{H+2}: Let φ be a sentence in the signature of trees. In φ, replace every occurrence of the atomic formula $E(x, y)$ by

$$x < y \land \neg\exists z\colon x < z < y$$

and call the resulting sentence φ'. Then $F_{H+2} \models \varphi \iff \mathrm{oF}_{H+2} \models \varphi'$. Since φ' can be computed from φ in polynomial time, the claim follows from Corollary 7. ◀

5 Lower Bound

Let $H \geq 1$ be fixed throughout this section. We want to show that the theory of the forest F_{H+2} is hard for the class $\mathrm{STA}(*, \exp_H(2, \mathrm{poly}(n)), \mathrm{poly}(n))$.

We will reduce an arbitrary language $L \subseteq \Sigma^*$ from the said complexity class to the theory of the forest F_{H+2} in two steps: First, we reduce L to the theory of the free monoid Δ^*. In this reduction, we can restrict quantification to words of length $\leq \exp_H(2, \mathrm{poly}(|x|))$. In a second step, we reduce this bounded theory of the free monoid to the theory of the forest F_{H+2}.

Let φ be a formula and $k \geq 1$. Then $\exists^{\geq k} y \colon \varphi$ abbreviates the formula

$$\exists y_1, y_2, \ldots, y_k \colon \bigwedge_{1 \leq i < j \leq k} y_i \neq y_j \wedge \forall y \colon \left(\left(\bigvee_{1 \leq i \leq k} y = y_i \right) \to \varphi \right)$$

and $\exists^{=k} y \colon \varphi$ stands for $\exists^{\geq k} y\, \varphi \wedge \neg \exists^{\geq k+1} y\, \varphi$. Note that the size of these formulas is $O(k^2 + |\varphi|)$.

5.1 Reduction to the theory of the bounded free monoid

Let $N \geq 0$ and let Δ be an alphabet. The *N-bounded free monoid* is the structure

$$(\Delta^{\leq \exp_H(N,N)}, \cdot, (a)_{a \in \Delta})$$

where $\Delta^{\leq \exp_H(N,N)}$ is the set of words over Δ of length $\leq \exp_H(N, N)$, $\cdot$ is the concatenation of such words (considered as a ternary relation such that the product of two "long" words is not defined), and any letter $a \in \Delta$ serves as a constant.

An alternating Turing machine is a tuple $M = (Q, \Sigma, \Gamma, \delta, \iota, \Box, \mathrm{tp}, F)$ where Q is the finite set of states, $\Sigma \subseteq \Gamma$ are the input- and tape-alphabets, $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{-1, 0, 1\}$ is the transition relation, $\iota \in Q$ is the initial state, $\Box \in \Gamma \setminus \Sigma$ is the blank symbol, $\mathrm{tp} \colon Q \to \{\forall, \exists\}$ is the type function with $\mathrm{tp}(\iota) = \exists$, and $F \subseteq Q$ is the set of final states. We assume the tape of M to be infinite on the right, only. We write Δ for the set $\Gamma \cup Q \cup \{\triangleleft, \triangleright\}$ (assuming these three sets to be mutually disjoint).

A *configuration* is a word from $\triangleright \Gamma^* Q \Gamma^* \triangleleft$. We write $c \vdash c'$ for configurations c and c' if the machine can move from c to c' in one step. The *type* of a configuration is the type of its state. A *computation* is a finite sequence of configurations $(c_i)_{0 \leq i \leq n}$ for some $n \in \mathbb{N}$ with $c_i \vdash c_{i+1}$ for all $0 \leq i < n$. We say that it is a computation from c_0 to c_n. It is *existential* if all configurations are existential; it is *homogeneous* if

- the types of c_0, c_1, $\ldots$, c_{n-1} are the same and
- the types of c_0 and c_n are different.

For configurations c and c', we write

$$c \vdash_\exists c' \text{ and } c \vdash_{hom} c'$$

if there exists an existential and a homogeneous computation, respectively, from c to c'. Note that the latter implies that c and c' have distinct types.

Let $f \colon \mathbb{N} \to \mathbb{N}$ be a function. The alternating Turing machine is *$f(n)$-time bounded* if any computation $(c_i)_{0 \leq i \leq N}$ with first configuration in $\triangleright \iota w \Box^* \triangleleft$ and $w \in \Sigma^*$ makes $\leq f(|w|)$ steps, i.e., satisfies $N + 1 \leq f(|w|)$.

Now let $a \in \mathbb{N}$ be odd and $w \in \Sigma^*$. Then x is accepted by M with a alternations if there exists a configuration $c_0 \in \triangleright \iota w \Box^* \triangleleft$ such that the following holds:

$$\begin{aligned}
&\exists \text{ configuration } c_1 \text{ with } c_0 \vdash_{hom} c_1\\
&\quad\forall \text{ configurations } c_2 \text{ with } c_1 \vdash_{hom} c_2\\
&\qquad\exists \text{ configuration } c_3 \text{ with } c_2 \vdash_{hom} c_3\\
&\qquad\quad\forall \text{ configurations } c_4 \text{ with } c_3 \vdash_{hom} c_4\\
&\qquad\qquad\ddots\\
&\qquad\qquad\quad\exists \text{ configuration } c_{a-2} \text{ with } c_{a-3} \vdash_{hom} c_{a-2}\\
&\qquad\qquad\qquad\forall \text{ configuration } c_{a-1} \text{ with } c_{a-2} \vdash_{hom} c_{a-1}\\
&\qquad\qquad\qquad\quad\exists \text{ accepting configuration } c_a\colon c_{a-1} \vdash_{\exists} c_a
\end{aligned} \tag{3}$$

For our reduction, fix a language $L \in \mathrm{STA}(*, \exp_H(2, \mathrm{poly}(n)), \mathrm{poly}(n))$. Then there exist an alternating Turing machine M and polynomial functions $p, q\colon \mathbb{N} \to \mathbb{N}$ such that M is $\exp_H(2, p(n))$-time bounded and L is the set of words w that are accepted by M with $q(|w|)$ alternations. For notational simplicity, we assume $q(n)$ to be odd for all $n \in \mathbb{N}$.

Let $w \in \Sigma^*$. Furthermore, let $N = p(|w|)^2$. We want to express the acceptance of w by M by a formula of polynomial size over

$$\mathcal{M}_{p(n)^2} = (\Delta^{\leq \exp_H(p(n)^2, p(n)^2)}, \cdot, (a)_{a\in\Delta})\,.$$

To achieve this, first note the following:

- A word c is an existential configuration if it satisfies

$$\begin{aligned}\mathrm{conf}_{\exists}(c) = \exists x, y \forall z_1, z_2\colon & \bigwedge_{a \in Q \cup \{\triangleleft, \triangleright\}} (x \neq z_1 a z_2 \wedge y \neq z_1 a z_2)\\ & \wedge \bigvee_{q\in Q, \mathrm{tp}(q) = \exists} c = \triangleright x q y \triangleleft\,.\end{aligned}$$

  Universal and accepting configurations are described similarly by formulas $\mathrm{conf}_{\forall}(c)$ and $\mathrm{conf}_{\mathrm{acc}}(c)$, respectively. Let $\mathrm{conf} = \mathrm{conf}_{\exists} \vee \mathrm{conf}_{\forall}$.
- A word c is an initial configuration with input w, i.e., $c \in \triangleright \iota w \Box^* \triangleleft$, iff it satisfies

$$\mathrm{init}_w(c) = \exists y \left(c = \triangleright \iota w y \triangleleft \wedge \forall z_1, z_2\colon \bigwedge_{a\in \Delta \setminus \{\Box\}} y \neq z_1 a z_2 \right).$$

- $c \vdash_M c'$ iff they satisfy

$$\mathrm{step}(c, c') = \mathrm{conf}(c) \wedge \mathrm{conf}(c') \wedge \exists x, y\colon \bigvee_{(\ell, r)\in R} (c = x\ell y \wedge c' = xry)$$

  where R is some finite subset of $\Delta^3 \times \Delta^3$.

▶ **Lemma 9.** *There is a formula* $\mathrm{comp}_{\mathrm{hom}}(\overline{x}, \overline{y})$ *such that for any configurations* c *and* c', *we have* $\mathcal{M}_{p(n)^2} \models \mathrm{comp}_{\mathrm{hom}}(c, c')$ *if, and only if, there exists a homogeneous computation*

$$c = c_0 \vdash c_1 \vdash c_2 \vdash \cdots \vdash c_K = c'$$

with

$$\sum_{0 \leq i \leq K} |c_i| \leq \exp_H(p(n)^2, p(n)^2)\,. \tag{4}$$

Similarly, there is a formula $\mathrm{comp}_{\exists}$ *expressing the existence of an existential computation with the same length bound.*

Proof. We will express the existence of a word $W = c_0\, c_1\, c_2\, \dots\, c_K$ such that

- $c = c_0$,
- $c_i \vdash_M c_{i+1}$ for all $0 \le i < K$,
- $c_K = c'$,
- and all configurations c_i for $i < K$ have the type of c_0.

Note that this is the case iff there exists a word W such that

- c is a prefix of W,
- c' is a suffix of W,
- any factor x of W that is a configuration is either a suffix of W or followed by a factor y which is a configuration satisfying $x \vdash_M y$. In the latter case, its type is that of c.

If we consider this formula in the free monoid Δ^*, then it expresses the existence of a homogeneous computation from c to c' *of arbitrary length.* In the structure $\mathcal{M}_{p(n)^2}$, the length of the word W is bounded by $\exp_H(p(n)^2, p(n)^2)$. Hence we get (4). ◀

▶ **Proposition 10.** *From $w \in \Sigma^*$ with $|w| = n$, we can compute in polynomial time a sentence φ_w such that $w \in L$ if, and only if, $\mathcal{M}_{p(n)^2} \models \varphi_w$.*

Proof. Let φ_w be the following sentence:

$$\begin{aligned}
&\exists c_0 \colon \mathit{init}_w(c_0)\\
&\quad \wedge \exists c_1 \colon \mathrm{conf}(c_1) \wedge \mathrm{comp}_{\mathrm{hom}}(c_0, c_1)\\
&\qquad \wedge \forall c_2 \colon \mathrm{conf}(c_2) \wedge \mathrm{comp}_{\mathrm{hom}}(c_1, c_2)\\
&\qquad\quad \to \exists c_3 \colon \mathrm{conf}(c_3) \wedge \mathrm{comp}_{\mathrm{hom}}(c_2, c_3)\\
&\qquad\qquad \wedge \forall c_4 \colon \mathrm{conf}(c_4) \wedge \mathrm{comp}_{\mathrm{hom}}(c_3, c_4)\\
&\qquad\qquad\quad \ddots\\
&\qquad\qquad\qquad \exists c_{q(n)-2} \colon \mathrm{conf}(c_{q(n)-2}) \wedge \mathrm{comp}_{\mathrm{hom}}(c_{q(n)-3}, c_{q(n)-2})\\
&\qquad\qquad\qquad\quad \wedge \forall c_{q(n)-1} \colon \mathrm{conf}(c_{q(n)-1}) \wedge \mathrm{comp}_{\mathrm{hom}}(c_{q(n)-2}, c_{q(n)-1})\\
&\qquad\qquad\qquad\qquad \to \exists c_{q(n)} \colon \quad \mathrm{conf}_{\mathrm{acc}}(c_{q(n)})\\
&\qquad\qquad\qquad\qquad\qquad \wedge \quad \mathrm{comp}_{\exists}(c_{q(n)-1}, c_{q(n)})
\end{aligned}$$

Since this is the direct translation of the acceptance condition by alternating Turing machines (3), we obtain that $\mathcal{M}_{p(n)^2} \models \varphi_w$ implies $w \in L$.

Conversely, suppose $w \in L$, i.e., (3) holds. Since M is $\exp(2, p(n))$-time bounded, any computation starting from a configuration $c_0 \in \triangleright\iota w\Box^*\triangleleft$ has length $\le \exp_H(2, p(n))$; in particular, the machine's head can only move $\exp_H(2, p(n))$ cells to the right. Since (3) quantifies over reachable configurations, only, we can restrict quantification in (3) to configurations of length $\le \exp_H(2, p(n))$. Furthermore, (3) quantifies over computations (hidden in the statements $c_i \vdash_{hom} c_{i+1}$ and $c_{a-1} \vdash_\exists c_a$). Since these computations start in reachable configurations, their length is at most $\exp_H(2, p(n))$ and all intermediate configurations are reachable and therefore of length $\le \exp_H(2, p(n))$. Note that

$$(\exp_H(2, p(n)) + 1) \cdot \exp_H(2, p(n)) \le \exp_H(p(n)^2, p(n)^2)\,.$$

Hence, statements of the form $c_i \vdash_{hom} c_{i+1}$ can be replaced by statements of the form $\mathcal{M}_{p(n)^2} \models \mathrm{comp}_{\mathrm{hom}}(c_i, c_{i+1})$ (and similarly for $c_{q(n)-1} \vdash_\exists c_{q(n)}$). Thus, in summary, we get $\mathcal{M}_{p(n)^2} \models \varphi_w$. ◀

5.2 Interpretation of the bounded free monoid in F_{H+2}

To complete the reduction of L to the theory of the forest F_{H+2}, it remains to provide an interpretation of the theory of $\mathcal{M}_{p(n)^2}$ in F_{H+2}. This interpretation has to be computable in time polynomial in $N = p(n)^2$. This reduction requires to express certain numerical properties. Therefore, we first show how to encode numbers by nodes from F_{H+2} and how to do some restricted form of arithmetic.

5.2.1 Nodes as numbers

Let $N \geq 3$. We define the number $[\![v]\!]_N$ for any node v of the forest F_{H+2}. Let $v_1, \ldots, v_\ell$ be the children of v (if v is of height 0, then there is no such child, i.e., $\ell = 0$). For $k \in \mathbb{N}$, let t_k denote the number of children v_i with $[\![v_i]\!]_N = k$, i.e.,

$$t_k = |\{i \mid 1 \leq i \leq \ell, [\![v_i]\!]_N = k\}| .$$

Note that $t_k = 0$ for almost all k since any node of F_{H+2} has only finitely many children. We want to consider the number t_k as k-th digit in a base-N-representation of some natural number. Therefore, we normalize this number to

$$d_k = \min(t_k, N-1)$$

such that $d_k \in \{0, 1, \ldots, N-1\}$. Let $\chi_N(v) = (d_k)_{k \in \mathbb{N}}$ denote the *characteristic* of v and define

$$[\![v]\!]_N = \sum_{k \in \mathbb{N}} d_k \cdot b^k .$$

Note that the sequence $\chi_N(v)$ is the base-N-representation of the number $[\![v]\!]_N$.[2]

▶ **Example 11.** The number 0 is represented by all nodes of height 0, i.e., all leaves in F_{H+2}. A number $i \in \{1, 2, \ldots, N-2\}$ is represented by all nodes of height 1 with precisely i children. Any height-1-node with $\geq N-1$ children represents the number $N-1$. If $a_m \in \{0, 1, \ldots, N-1\}$ for $0 \leq m < n$, then $a = \sum_{0 \leq m < N} a_m b^m$ is represented, e.g., by a height-2-node v such that a_m children v have m children, i.e., represent the number m (for all $0 \leq m < N$). If $a_m = N-1$, then we can even add further children representing m without changing $[\![v]\!]_N$.

By induction, one obtains for any node v of height h:

$$\begin{aligned} [\![v]\!]_N &= 0 && \text{if } h = 0 \\ \exp_{h-2}(N, N) \leq [\![v]\!]_N &< \exp_{h-1}(N, N) && \text{if } h \geq 1 \end{aligned}$$

Conversely (for $h \leq H+2$), any $a < \exp_{h-1}(N, N)$ is represented by some node of height $\leq h$.

We next show that the relations $[\![v_1]\!]_N < [\![v_2]\!]_N$ and $[\![v_1]\!]_N = [\![v_2]\!]_N$ can be defined by first-order formulas.

[2] For $N = 2$, this is a simple variation of the encoding from [9]. For this case, Flum and Grohe also prove Lemma 12i, but neither Lemma 12ii nor Lemma 13. In contrast to them, we measure the size of our formulas in terms of N while H is considered a constant.

▶ **Lemma 12.** *From $N \in \mathbb{N}$, one can compute formulas $\mathrm{eq}_N(x_1, x_2)$ and $\mathrm{less}_N(x_1, x_2)$ in time polynomial in N such that for any two nodes v_1 and v_2 in F_{H+2} the following hold:*

(i) $(F_{H+2}, v_1, v_2) \models \mathrm{eq}_N$ *if, and only if,* $[\![v_1]\!]_N = [\![v_2]\!]_N$ *and*

(ii) $(F_{H+2}, v_1, v_2) \models \mathrm{less}_N$ *if, and only if,* $[\![v_1]\!]_N < [\![v_2]\!]_N$.

Proof. For $0 \le h \le H+2$, we can construct a formula in time $O(h)$ expressing that the height of a node is at most h: $\neg\exists x_0, x_1, \ldots, x_{h+1} \colon x = x_0 \wedge \bigwedge_{0\le i\le h} E(x_i, x_{i+1})$. We abbreviate this formula by $\mathrm{hgt}_{\le h}(x)$.

Let v_1 and v_2 be nodes of F_{H+2}. Then $[\![v_1]\!]_N = [\![v_2]\!]_N$ if, and only if, $\chi_N(v_1) = \chi_N(v_2)$. But this is the case if, and only if, for all children v of v_1 or v_2, the number of children v_1' of v_1 with $[\![v]\!]_N = [\![v_1']\!]_N$ equals the number of children v_2' of v_2 with $[\![v]\!]_N = [\![v_2']\!]_N$ or both numbers are $\ge N-1$. Thus, to build the formula eq_N, we have to apply the same formula to nodes of smaller height. Therefore, we first construct formulas eq_N^h that satisfy i at least for all nodes v_1 and v_2 of height at most h (for $0 \le h \le H+2$). The first claim then follows with $\mathrm{eq}_N = \mathrm{eq}_N^{H+2}$.

The formula $\mathrm{eq}_N^0 = (x_1 = x_1)$ satisfies i for nodes of height ≤ 0 since, whenever v_1 and v_2 are nodes of height 0, they both represent 0. We define eq_N^{h+1} as follows:

$$\mathrm{eq}_N^{h+1} = \forall y\colon \left(\big(E(x_1,y) \vee E(x_2,y)\big) \rightarrow \bigwedge_{1\le i<N} \begin{pmatrix} & \exists^{\ge i} y_1\colon E(x_1,y_1) \wedge \mathrm{eq}_N^h(y,y_1) \\ \leftrightarrow & \exists^{\ge i} y_2\colon E(x_2,y_2) \wedge \mathrm{eq}_N^h(y,y_2) \end{pmatrix} \right)$$

By the above explanation and by induction, this formula satisfies *i* for all nodes of height $\le h+1$. This completes the definition of the formula $\mathrm{eq}_N = \mathrm{eq}_N^{H+2}$.

By induction, there are constants $c_1, c_2, \ldots, c_{H+2}$ such that, for sufficiently large n, we have $|\mathrm{eq}_N^{h+1}| \le c_{h+1}|(n^3 + |\mathrm{eq}_N^h|)$. Consequently,

$$|\mathrm{eq}_N^{H+1}| \in O(N^{3\cdot(H+2)})\,.$$

Since H was fixed from the beginning, the formula $\mathrm{eq}_N^{H+2} = \mathrm{eq}_N$ can be constructed from N in time polynomial in N.

Similarly, we construct formulas less_N^h that satisfy *ii* at least for all nodes v_1 and v_2 of height at most h (for $0 \le h \le H+2$). The second claim then follows with $\mathrm{less}_N = \mathrm{less}_N^{H+2}$.

Let $\chi_N(v_i) = (d_k^i)_{k\in\mathbb{N}}$ for $i \in \{1,2\}$ be the characteristic of v_i. Then $[\![v_1]\!]_N < [\![v_2]\!]_N$ if, and only if, $\chi_N(v_1)$ is lexicographically properly smaller than $\chi_N(v_2)$. This means that there is some $k \in \mathbb{N}$ with $d_k^1 < d_k^2$ and $d_i^1 \le d_i^2$ for all $i < k$. Since, in particular, $d_k^2 > 0$, there is a child v' of v_2 with $[\![v']\!]_N = k$.

The formula $\mathrm{less}_N^0 = (x_1 = x_1)$ satisfies the required property. Let less_N^{h+1} denote the following formula:

$$\begin{aligned} \exists y\colon \quad & E(x_2,y) \wedge \bigvee\nolimits_{1\le i<N} \begin{pmatrix} & \neg\exists^{\ge i} y_1\colon E(x_1,y_1) \wedge \mathrm{eq}_N^h(y,y_1) \\ \wedge & \exists^{\ge i} y_2\colon E(x_2,y_2) \wedge \mathrm{eq}_N^h(y,y_2) \end{pmatrix} \\ \wedge \quad & \bigwedge_{1\le i<N} \forall z\colon \begin{pmatrix} & \big(E(x_1,z) \wedge \mathrm{less}_N^h(z,y) \wedge \exists^{\ge i} z_1\colon E(x_1,z_1) \wedge \mathrm{eq}_N^h(z,z_1)\big) \\ \rightarrow & \exists^{\ge i} z_2\colon E(x_2,z_2) \wedge \mathrm{eq}_N^h(z,z_2) \end{pmatrix} \end{aligned}$$

By induction, there are constants $c_1, c_2, \ldots, c_{H+2}$ such that, for sufficiently large N, we have $|\mathrm{less}_N^{h+1}| \le c_{h+1}(N^3 + |\mathrm{eq}_N^h| + |\mathrm{less}_N^h|)$. Consequently,

$$|\mathrm{less}_N^{H+1}| \in O(N^{3\cdot(H+2)})\,.$$

Since H was fixed from the beginning, the formula $\mathrm{less}_N^{H+2} = \mathrm{less}_N$ can be constructed from N in time polynomial in N. ◀

Using the two formulas from above, we are now able to also define addition:

▶ **Lemma 13.** *From $N \in \mathbb{N}$, one can compute a formula* $\mathrm{add}_N(x_1, x_2, x_3)$ *in time polynomial in N such that for any three nodes v_1, v_2, and v_3 in F_{H+2}, the following holds:*
$(F_{H+2}, v_1, v_2, v_3) \models \mathrm{add}_N$ *if, and only if,* $[\![v_1]\!]_N + [\![v_2]\!]_N = [\![v_3]\!]_N$.

Proof. In the following explanations, let $t = \exp_H(N, N)$.

Let v_1, v_2, and v_3 be nodes from F_{H+2}, and let $\chi_N(v_i) = (d^i_k)_{k\in\mathbb{N}}$ for all $1 \le i \le 3$. Then $d^i_k = 0$ for all $k \ge t$ since the height of v_i is $\le H+2$, i.e, its children (being of height $\le H+1$) represent numbers $< t$. Since $(d^i_k)_{0\le i<t}$ is the base-N-representation of $[\![v_i]\!]_N$, the following are equivalent:

- $[\![v_1]\!]_N + [\![v_2]\!]_N = [\![v_3]\!]_N$
- There exist $e_k \in \{0, 1\}$ (the carry bits) for $0 \le k < t$ such that
  (a) $e_0 = 0$,
  (b) $d^3_k + N \cdot e_{k+1} = d^1_k + d^2_k + e_k$ for $0 \le k < t-1$, and
  (c) $d^3_{t-1} = d^1_{t-1} + d^2_{t-1} + e_{t-1}$.

We will translate this description into the formula add_N. Note that nodes of height $H+2$ have characteristics of length t (more precisely: from the entry number t on, they are constantly zero). Hence any sequence $(e_0, e_1, \dots, e_{t-1}, 0, 0, \dots)$ of bits is the characteristics of some node y. Furthermore note that we have to quantify over numbers k with $0 \le k < t$ – but these are precisely the values of nodes of height $\le H+1$. Therefore, the following formulas succ_N and $\max_N$ will become useful.

The formula

$$\mathrm{succ}_N(z, z') = \mathrm{hgt}_{\le H+1}(z) \wedge \mathrm{hgt}_{\le H+1}(z') \wedge \mathrm{less}_N(z, z') \wedge \neg\exists z''\colon \mathrm{less}_N(z, z'') \wedge \mathrm{less}_N(z'', z')$$

expresses that z and z' are two nodes of height $\le H+1$ satisfying $[\![z]\!]_N + 1 = [\![z']\!]_N$. Furthermore, the formula

$$\max_N(z) = \mathrm{hgt}_{\le H+1}(z) \wedge \neg\exists z'\colon \mathrm{hgt}_{\le H+1}(z') \wedge \mathrm{less}_N(z, z')$$

expresses that z is a node of height at most $H+1$ that represents the maximal possible value for such a node, i.e., $[\![z]\!]_N = t-1$.

Let I denote the set of quintuples $(a_1, a_2, b_1, a_3, b_2)$ of natural numbers from $\{0, 1, \dots, n-1\}$ with $a_1 + a_2 + b_1 = a_3 + N \cdot b_2$. Finally, for $i \in \{0, 1, \dots, N-1\}$ set

$$Q^i x\,\varphi = \begin{cases} \exists^{=i} x\,\varphi & \text{if } i < N-1 \\ \exists^{\ge N-1} x\,\varphi & \text{if } i = N-1\,. \end{cases}$$

Now consider the following formula $\mathrm{add}_N(x_1, x_2, x_3)$:

$$\begin{aligned}
\exists y\,\forall z, z'\colon \quad & \big(E(y,z) \wedge E(y,z') \wedge \mathrm{eq}_N(z,z')\big) \to \big(z = z' \wedge \exists y'\colon E(z,y')\big) \\
\wedge \quad & \mathrm{succ}(z,z') \to \bigvee_{(a_1,a_2,b_1,a_3,b_2)\in I} \left(\begin{array}{ll} & Q^{a_1} x'_1\colon E(x_1,x'_1) \wedge \mathrm{eq}_N(x'_1,z) \\ \wedge & Q^{a_2} x'_2\colon E(x_2,x'_2) \wedge \mathrm{eq}_N(x'_2,z) \\ \wedge & Q^{b_1} y'\colon E(y,y') \wedge \mathrm{eq}_N(y',z) \\ \wedge & Q^{a_3} x'_3\colon E(x_3,x'_3) \wedge \mathrm{eq}_N(x'_3,z) \\ \wedge & Q^{b_2} y'\colon E(y,y') \wedge \mathrm{eq}_N(y',z') \end{array}\right) \\
\wedge \quad & \max(z) \to \bigvee_{(a_1,a_2,b_1,a_3,0)\in I} \left(\begin{array}{ll} & Q^{a_1} x'_1\colon E(x_1,x'_1) \wedge \mathrm{eq}_N(x'_1,z) \\ \wedge & Q^{a_2} x'_2\colon E(x_2,x'_2) \wedge \mathrm{eq}_N(x'_2,z) \\ \wedge & Q^{b_1} y'\colon E(y,y') \wedge \mathrm{eq}_N(y',z) \\ \wedge & Q^{a_3} x'_3\colon E(x_3,x'_3) \wedge \mathrm{eq}_N(x'_3,z) \end{array}\right)
\end{aligned}$$

Let y be some node of F_{H+2} such that the formula starting with $\forall z$ holds. Let furthermore $(e_k)_{k\in\mathbb{N}}$ be the characteristic of the node y. Since the height of y is $\leq H+2$, we get $e_k = 0$ for all $k \geq t$. The first conjunct expresses $e_k \in \{0,1\}$ (since no two distinct children of y represent the same number) and $e_0 = 0$ (since no child of y has height 0, i.e., represents 0). Having said this, it is clear that the second and third conjunct ensure properties (b) and (c) from above. Thus, indeed, the formula add_N expresses the relation $[\![v_1]\!]_N + [\![v_2]\!]_N = [\![v_3]\!]_N$.

Furthermore note that $|I| \leq N^5$. Hence, using Lemma 12, the formula add_N can be constructed in polynomial time from N. ◀

5.2.2 Tuples of nodes as words

In the previous section, we agreed how to consider a node v of depth ≥ 1 (and therefore of height $\leq H+1$) as a number $[\![v]\!]_N$ between 0 and $\exp_H(N,N)-1$. Now, we want to consider a tuple $\overline{v} = (v_a)_{a\in\Delta}$ of nodes as word $\mathrm{word}_N(\overline{v})$ over the alphabet Δ. To this aim, let

$$P_a = \{[\![v'_a]\!]_N \mid (v_a, v'_a) \in E\}$$

denote the set of numbers represented by children of the node v_a (for $a \in \Delta$). The word $\mathrm{word}_N(\overline{v})$ is defined only in case these sets of numbers are mutually disjoint and the union of these sets is an initial segment of the natural numbers. Let $\ell = \sup\big(\bigcup_{a\in\Delta} P_a\big)$. Then $\mathrm{word}_N(\overline{v})$ is the word

$$a_0 a_1 a_2 \ldots a_\ell$$

with $a_k = a \iff k \in P_a \iff k = [\![v'_a]\!]_N$ for some child v'_a of v_a. Thus, the children of the node v_a represent the positions of the letter a in $\mathrm{word}_N(\overline{v})$. Since children of nodes have height $\leq H+1$, the word $\mathrm{word}_N(\overline{v})$ has length $\leq \exp_H(N,N)$. Conversely, any word of this length can be represented by a tuple of nodes $word_N(\overline{v})$.

▶ **Lemma 14.** *From $N \in \mathbb{N}$, one can compute in polynomial time formulas $\mathrm{is\_word}_N(\overline{x})$ and $\mathrm{prod}(\overline{x},\overline{y},\overline{z})$, such that, for any Δ-tuples $\overline{u}$, $\overline{v}$, and $\overline{w}$ of nodes, the following hold:*
- $(F_{H+2},\overline{v}) \models \mathrm{is\_word}_N$ *if, and only if, the tuple* $\mathrm{word}_N(\overline{v})$ *is defined.*
- $(F_{H+2},\overline{u},\overline{v},\overline{w}) \models \mathrm{prod}$ *if, and only if,* $\mathrm{word}_N(\overline{u})$, $\mathrm{word}_N(\overline{v})$, *and* $\mathrm{word}_N(\overline{w})$ *are defined and* $\mathrm{word}_N(\overline{u})\ \mathrm{word}_N(\overline{v}) = \mathrm{word}_N(\overline{w})$.

Proof. The formula $\mathrm{is\_word}_N$ looks as follows:

$$\forall x,y \left(\bigvee_{a\in\Delta} E(x_a,y) \wedge \mathrm{less}_N(x,y)\right) \to \exists x' \left(\bigvee_{b\in\Delta} E(x_b,x') \wedge \mathrm{eq}_N(x,x')\right)$$
$$\wedge \bigwedge_{a,b\in\Delta,a\neq b} \Big(\big(E(x_a,x) \wedge E(x_b,y)\big) \to \neg\mathrm{eq}_N(x,y)\Big)$$

The first line expresses that $\bigcup_{a\in\Delta} P_a$ is an initial segment of $(\mathbb{N},\leq)$, the second one ensures that the sets P_a are mutually disjoint.

Note that the length of the word $\mathrm{word}_N(\overline{v})$ is the successor of the maximal number represented by any of the children of nodes v_a from the tuple $\overline{v}$. Therefore, the following formula ensures that the length of $\mathrm{word}_N(\overline{x})$ equals $[\![\ell]\!]_N$:

$$\forall x\colon \bigwedge_{a\in\Delta} \big(E(x_a,x) \to \mathrm{less}_N(x,\ell)\big)$$
$$\wedge \exists x\colon \bigvee_{a\in\Delta} E(x_a,x) \wedge \neg\exists y\colon \mathrm{less}_N(x,y) \wedge \mathrm{less}_N(y,\ell)$$

We just remark that representable words have length $\leq \exp_H(N,N)$. Hence, their length is always represented by some node of height $\leq H+2$.

We denote the above formula by $|\operatorname{word}_N(\overline{x})| = \ell$. Now the formula prod looks as follows:

$$\begin{aligned}\exists \ell_x, \ell_y, \ell_z\colon\quad & \operatorname{is\_word}_N(\overline{x}) \wedge \operatorname{is\_word}_N(\overline{y}) \wedge \operatorname{is\_word}_N(\overline{z})\\ & \wedge |\operatorname{word}_N(\overline{x})| = \ell_x \wedge |\operatorname{word}_N(\overline{y})| = \ell_y \wedge |\operatorname{word}_N(\overline{z})| = \ell_z\\ & \wedge \operatorname{add}_N(\ell_x, \ell_y, \ell_z)\\ & \wedge \bigwedge_{a\in\Delta} \forall x \exists z\colon E(x_a, x) \to E(z_a, z) \wedge \operatorname{eq}_N(x, z)\\ & \wedge \bigwedge_{a\in\Delta} \forall y \exists z\colon E(y_a, y) \to E(z_a, z) \wedge \operatorname{add}_N(y, \ell_x, z)\end{aligned}$$ ◀

▶ **Observation 15.** *From $N \in \mathbb{N}$ and $a \in \Delta$, one can construct in polynomial time a formula* $\operatorname{is\_letter}_{N,a}(\overline{x})$ *such that, for any Δ-tuple $\overline{u}$ of nodes, we have*

$$(F_{H+2}, \overline{u}) \models \operatorname{is\_letter}_{N,a}(\overline{x}) \iff \operatorname{word}_N(\overline{u}) \textit{ is defined and equals } a\,.$$

This is obtained by the formula

$$\operatorname{is\_word}_N(\overline{u}) \wedge \bigwedge_{b\neq a} \forall y\, \neg E(x_b, y) \wedge \exists^{=1} y\, E(x_a, y)\,.$$

This finishes the construction of an interpretation of the bounded free monoid $\mathcal{M}_N$ in the forest F_{H+2}. Since all the formulas $\operatorname{is\_word}_N$, prod_N, and $\operatorname{is\_letter}_{N,a}$ can be computed in polynomial time, we can reduce the theory of the bounded free monoid $\mathcal{M}_N$ in polynomial time to the theory of F_{H+2}. Together with Proposition 10, this finishes the proof of the following theorem:

▶ **Theorem 16.** *The theory of the forest F_{H+2} is hard for the class*

$$\mathrm{STA}(*, \exp_H(2, \mathrm{poly}(n)), \mathrm{poly}(n)).$$

6 Conclusion

We have shown that for every h there is an automatic structure, whose theory is complete for the Berman complexity class $\mathrm{STA}(*, \exp_h(2, \mathrm{poly}(n)), \mathrm{poly}(n))$. Therefore theories of automatic structures are distributed across all stages of elementary complexity. The variants F_H^1 and F_H^∞ of our structure F_H that we mentioned in the beginning might be interesting in their own right. A careful analysis of our proof reveals without much effort that the theories of these two structures have the same complexity as the theory of F_H.

▶ **Theorem 17.** *The theories of F_H^1 and F_H^∞ are complete for*

$$\mathrm{STA}(*, \exp_H(2, \mathrm{poly}(n)), \mathrm{poly}(n)).$$

Finally let us mention a related problem from parameterized complexity theory.

▶ **Conjecture 18.** *There is no algorithm that determines correctly for every tree T of height at most H and every first-order sentence φ whether $T \models \varphi$ in time $\exp_{H-3}(2, \mathrm{poly}(|\varphi|)) \cdot \mathrm{poly}(|T|)$.*

An upper bound this problem is given in [12]. It might be possible to prove Conjecture 18 (under suitable complexity theoretic assumptions) with a similar strategy as it was used in [10] for the class of all finite trees. The formulas that we defined for our lower bound might be useful in this case.

References

1 V. Bárány, E. Grädel, and S. Rubin. Automata-based presentations of infinite structures. In *Finite and Algorithmic Model Theory*, pages 1–76. Cambridge University Press, 2011.

2 L. Berman. The complexity of logical theories. *Theoretical Computer Science*, 11:71–77, 1980.

3 A. Blumensath. Automatic structures. Technical report, RWTH Aachen, 1999.

4 A. Blumensath and E. Grädel. Automatic Structures. In *LICS'00*, pages 51–62. IEEE Computer Society Press, 2000.

5 Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49(2):129–141, 1961. URL: `http://eudml.org/doc/213582`.

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

7 S. Feferman and R.L. Vaught. The first order properties of algebraic systems. *Fund. Math.*, 47:57–103, 1959.

8 J. Ferrante and Ch. Rackoff. *The Computational Complexity of Logical Theories*. Lecture Notes in Mathematics vol. 718. Springer, 1979.

9 J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, Heidelberg, 2006.

10 Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Ann. Pure Appl. Logic*, 130(1-3):3–31, 2004.

11 Ch. Frougny and J. Sakarovitch. Synchronized rational relations of finite and infinite words. *Theor. Comput. Sci.*, 108:45–82, 1993.

12 Jakub Gajarský and Petr Hlinený. Faster deciding MSO properties of trees of fixed height, and some consequences. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, pages 112–123, 2012. `doi:10.4230/LIPIcs.FSTTCS.2012.112`.

13 B.R. Hodgson. On direct products of automaton decidable theories. *Theoretical Computer Science*, 19:331–335, 1982.

14 B. Khoussainov and M. Minnes. Three lectures on automatic structures. In *Proceedings of Logic Colloquium*, pages 132—-176, 2007.

15 B. Khoussainov and A. Nerode. Automatic presentations of structures. In *Logic and Computational Complexity*, Lecture Notes in Comp. Science vol. 960, pages 367–392. Springer, 1995.

16 B. Khoussainov, S. Rubin, and F. Stephan. Definability and regularity in automatic structures. In *STACS'04*, Lecture Notes in Comp. Science vol. 2996, pages 440–451. Springer, 2004.

17 Bakhadyr Khoussainov, Jiamou Liu, and Mia Minnes. Unary automatic graphs: an algorithmic perspective. *Mathematical Structures in Computer Science*, 19(1):133–152, 2009.

18 D. Kuske. Theories of automatic structures and their complexity. In *CAI 2009*, Lecture Notes in Comp. Science vol. 5725, pages 81–98. Springer, 2009.

19 D. Kuske and M. Lohrey. Some natural decision problems in automatic graphs. *Journal of Symbolic Logic*, 75(2):678–710, 2010.

20 D. Kuske and M. Lohrey. Automatic structures of bounded degree revisited. *Journal of Symbolic Logic*, 76(4):1352–1380, 2011.

21 P. Lindner. Theorien automatischer Strukturen in der Exponentialzeithierarchie. Master's thesis, TU Ilmenau, 2017.

22 D.C. Oppen. A $2^{2^{2^{cn}}}$ upper bound on the complexity of Presburger arithmetic. *Journal of Computer and System Sciences*, 16:323–332, 1978.

23 S. Rubin. Automata presenting structures: A survey of the finite string case. *Bulletin of Symbolic Logic*, 14:169–209, 2008.

24 F. Stephan. Automatic structures – recent results and open questions. *Journal of Physics: Conference Series*, 632:012013, 2015.