Foundations of Software Technology and Theoretical Computer Science (2010) Submission

The First-Order Theory of Ground Tree Rewrite Graphs

Stefan Göller[1], **Markus Lohrey**[2*]

[1] Department of Computer Science, University of Bremen

[2] Department of Computer Science, University of Leipzig

ABSTRACT. We prove that the complexity of the uniform first-order theory of ground tree rewrite graphs is in $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, O(n))$. Providing a matching lower bound, we show that there is a fixed ground tree rewrite graph whose first-order theory is hard for $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, \mathsf{poly}(n))$ with respect to logspace reductions. Finally, we prove that there is a fixed ground tree rewrite graph together with a single unary predicate in form of a regular tree language such that the resulting structure has a non-elementary first-order theory. For a long version of this paper with complete proofs see [11].

1 Introduction

Pushdown systems (PDS) are natural abstractions of sequential recursive programs. Moreover, the positive algorithmic properties of their transition graphs (pushdown graphs) make them attractive for the verification of sequential recursive programs. In particular, the model-checking problem for MSO (monadic second-order logic) and hence for most temporal logics (e.g. LTL, CTL) is decidable over pushdown graphs and precise complexity results are known (cf. [3, 14, 19, 20]). Ground tree rewrite systems [4, 8, 13] (GTRS), which are also known as ground term rewrite systems, generalize PDS from strings to trees. While the rules of a PDS rewrite a prefix of a given word, the rules of a GTRS rewrite a *subtree* of a given tree. GTRS can model (on an abstract level) concurrent programs with the ability to spawn new subthreads that are hierarchically structured, which in turn may terminate and return some values to their parents.

The transition graphs of GTRS (ground tree rewrite graphs) do not share all the nice algorithmic properties of pushdown graphs. For instance, the infinite grid is easily seen to be a ground tree rewrite graph, which implies that MSO is undecidable over GTRS. This holds even for most linear-time and branching-time temporal logics such as LTL and CTL (cf. [13, 17]). On the positive side, reachability, recurrent reachability, fair termination and certain fragments of LTL are decidable (cf. [4, 7, 13, 16, 17]). Moreover, first-order logic (FO) and first-order logic with reachability predicates are decidable [8]. This implies that model-checking of the CTL-fragment EF is decidable for GTRS; the precise complexity was recently clarified in [10].

While model-checking FO with reachability predicates is clearly non-elementary over GTRS (this holds already for the infinite binary tree, which is a pushdown graph [15]), the precise complexity of model-checking FO over GTRS is open, although the problem

*The second author is supported by the DFG project GELO.

is known to be decidable since more than 20 years [8]. The algorithm provided in [8] has non-elementary complexity due to an exponential blowup when dealing with negation.

The main contribution of this paper solves this problem. We prove that (i) the first-order theory of every ground tree rewrite graph belongs to $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, O(n))$ (doubly exponential alternating time, where the number of alternations is bounded linearly) and (ii) that there exists a fixed ground tree rewrite graph with an $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, \mathsf{poly}(n))$-complete first-order theory. The upper bound of $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, O(n))$ even holds uniformly, which means that the GTRS may be part of the input. The complexity class $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, \mathsf{poly}(n))$ appears also in other contexts. For instance, Presburger Arithmetic (the first-order theory of $(\mathbb{N}, +)$) is known to be complete for $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, \mathsf{poly}(n))$ [1].

The upper bound of $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, \mathsf{poly}(n))$ is shown by the method of Ferrante and Rackoff [9]. Basically, the idea is to show the existence of a winning strategy of the duplicator in an Ehrenfeucht-Fraïssé game, where the duplicator chooses "small" elements. This method is one of the main tools for proving upper bounds for FO-theories. We divide the upper bound proof into two steps. In a first step, we reduce the FO-theory for a ground tree rewrite graph to the FO-theory for a very simple word rewrite graph, where all word rewrite rules replace one symbol by another symbol. The alphabet consists of all trees, whose size is bounded by a singly exponential function in the input size (hence, the alphabet size is doubly exponential in the input size; this is the reason for the doubly exponential time bound). Basically, we obtain a word over this alphabet from a tree t by cutting off some prefix-closed set C in the tree and taking the resulting sequence of trees. Intuitively, the set C consists of all nodes u of t such that the subtree rooted in u is "large". Here, "large" has to be replaced by a concrete value $m \in \mathbb{N}$ such that a sequence of n rewrite steps applied to a tree t cannot touch a node from the upward-closed set C. Clearly, m depends on n. In our context, n will be exponential in the input size and so will m. In a second step (see the long version [11]), we provide an upper bound for the FO-theory of a word rewrite graph of the above form.

For the lower bound, we show in this extended abstract only hardness for $\mathsf{2NEXP}$ (doubly exponential nondeterministic time) using a $(2^{2^n} \times 2^{2^n})$ tiling problem (completeness for $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, \mathsf{poly}(n))$ is shown in the long version [10] using an alternating version of this tiling problem). In this problem, we are given a word w of length n over some fixed set of tiles, and it is asked, whether this word can be completed to a tiling of an array of size $(2^{2^n} \times 2^{2^n})$, where the word w is an initial part of the first row. There exists a fixed set of tiles, for which this problem is $\mathsf{2NEXP}$-complete. From this fixed set of tiles, we construct a fixed GTRS such that the following holds: From a given word w of length n over the tiles, one can construct (in logspace) a first-order formula that evaluates to true in our fixed ground tree rewrite graph if and only if the word w is a positive instance of the $(2^{2^n} \times 2^{2^n})$ tiling problem. Our construction is inspired by [10], where it is shown that the model-checking problem for a fragment of the logic EF (consisting of those EF-formulas, where on every path of the syntax tree at most one EF-operator occurs) over GTRS is $\mathsf{P}^{\mathsf{NEXP}}$-complete.

We finally state that there exists a fixed ground tree rewrite graph together with a single unary predicate in form of a regular tree language such that the resulting structure has a non-elementary first-order theory. This result is shown by a reduction from first-order satisfiability of finite binary words, which is non-elementary [15]. It should be noted that the first-order theory of a pushdown graph extended by regular unary predicates still has an elementary first-order theory: it is an automatic structure of bounded degree, hence its first-order theory belongs to 2EXPSPACE [12]. However, we remark that ground tree rewrite graphs are not of bounded degree, hence the result from [12] for tree-automatic structures of bounded degree (stating that their first-order theories belong to 3EXPTIME) cannot be applied to obtain an elementary upper bound in our setting.

2 Preliminaries

Let $\mathbb{N} = \{0, 1, \ldots\}$ be the set of *non-negative integers*. For $i, j \in \mathbb{N}$ we define the interval $[i, j] = \{i, i+1, \ldots, j\}$ and $[j] = [0, j]$. For an alphabet A (possibly infinite), we denote with $A^+ = A^* \setminus \{\varepsilon\}$ the set of all non-empty words over A. The length of the word $w \in A^*$ is denoted by $|w|$. For $B \subseteq A$, we denote with $|w|_B$ the number of occurrences of symbols from B in the word w. Let $f : A \to B$ be a mapping. For $A' \subseteq A$, we denote with $f\restriction A' : A' \to B$ the restriction of f to A'. For sets A, B, C (where A and B may have a non-empty intersection) and two mappings $f : A \to C$ and $g : B \to C$, we say that f and g are *compatible* if $f\restriction(A \cap B) = g\restriction(A \cap B)$. Finally, for mappings $f : A \to C$ and $g : B \to C$ with $A \cap B = \emptyset$, we define $f \uplus g : A \cup B \to C$ as the mapping with $(f \uplus g)(a) = f(a)$ for $a \in A$ and $(f \uplus g)(b) = g(b)$ for $b \in B$.

We will deal with alternating complexity classes, see [5] for more details. For functions $t(n)$ and $a(n)$ with $a(n) \leq t(n)$ for all $n \geq 0$ let $\mathsf{ATIME}(t(n), a(n))$ denote the class of all problems solvable on an alternating Turing-machine in time $t(n)$ with at most $a(n)$ alternations. We note that $\mathsf{ATIME}(t(n), t(n))$ is contained in $\mathsf{DSPACE}(t(n))$ if $t(n) \geq n$ [5].

2.1 Labelled graphs and first-order logic

A (directed) *graph* is a pair $(V, \to)$, where V is a set of *nodes* and $\to \subseteq V \times V$ is a binary relation. A *labelled graph* is a tuple $\mathfrak{G} = (V, \Sigma, \{\xrightarrow{a} \mid a \in \Sigma\})$, where V is a set of *nodes*, Σ is a finite set of *actions*, and $\xrightarrow{a}$ is a binary relation on V for all $a \in \Sigma$. We note that (labelled) graphs may have infinitely many nodes. We also write $v \in \mathfrak{G}$ for $v \in V$. For $u, v \in V$, we define $d_{\mathfrak{G}}(u, v)$ as the length of a shortest undirected path between u and v in the graph $(V, \bigcup_{a \in \Sigma} \xrightarrow{a})$. For $n \in \mathbb{N}$ and $u \in V$ let $S_n(\mathfrak{G}, u) = \{v \in V \mid d_{\mathfrak{G}}(u, v) \leq n\}$ be the *sphere* of radius n around u. Moreover, for $u_1, \ldots, u_k \in V$ let $S_n(\mathfrak{G}, u_1, \ldots, u_k) = \bigcup_{1 \leq i \leq k} S_n(\mathfrak{G}, u_i)$. We identify $S_n(\mathfrak{G}, u_1, \ldots, u_k)$ with the subgraph of $\mathfrak{G}$ induced by the set $S_n(\mathfrak{G}, u_1, \ldots, u_k)$, where in addition every u_i ($1 \leq i \leq k$) is added as a constant. For two labelled graphs $\mathfrak{G}_1$ and $\mathfrak{G}_2$ and nodes $u_1, \ldots, u_k \in \mathfrak{G}_1$, $v_1, \ldots, v_k \in \mathfrak{G}_2$, we will consider isomorphisms $f : S_n(\mathfrak{G}_1, u_1, \ldots, u_k) \to S_n(\mathfrak{G}_2, v_1, \ldots, v_k)$. Such an isomorphism has to map u_i to v_i. We write $S_n(\mathfrak{G}_1, u_1, \ldots, u_k) \cong S_n(\mathfrak{G}_2, v_1, \ldots, v_k)$ if there is an isomorphism $f : S_n(\mathfrak{G}_1, u_1, \ldots, u_k) \to S_n(\mathfrak{G}_2, v_1, \ldots, v_k)$. The following lemma is straightforward.

LEMMA 1. *Let $\mathfrak{G}_1, \mathfrak{G}_2$ be labelled graphs with the same set of actions. Let $\overline{u} \in \mathfrak{G}_1^k$, $\overline{v} \in \mathfrak{G}_2^k$, $u \in \mathfrak{G}_1$, and $v \in \mathfrak{G}_2$ such that $u \notin S_{2n+1}(\mathfrak{G}_1, \overline{u})$ and $v \notin S_{2n+1}(\mathfrak{G}_1, \overline{v})$. Finally, let $f : S_n(\mathfrak{G}_1, \overline{u}) \to S_n(\mathfrak{G}_2, \overline{v})$ and $f' : S_n(\mathfrak{G}_1, u) \to S_n(\mathfrak{G}_2, v)$ be isomorphisms. Then $f \uplus f' : S_n(\mathfrak{G}_1, u, \overline{u}) \to S_n(\mathfrak{G}_2, v, \overline{v})$ is an isomorphism as well.*

Later, we have to lift a relation $\to$ from a set A to a larger set. We will denote this new relation again by $\to$. Two constructions will be needed. Assume that $\to$ is a binary relation on a set A and let $A \subseteq B$. We lift $\to$ to the set B^+ of non-empty words over B as follows: For all $u, v \in B^+$, we have $u \to v$ if and only if there are $x, y \in B^*$ and $a, b \in A$ such that $a \to b$ and $u = xay$, $v = xby$. Note that this implies $|u| = |v|$. The second construction lifts $\to \subseteq A \times A$ from A to $\mathbb{N} \times A$ as follows: For $a, b \in A$ and $m, n \in \mathbb{N}$ let $(m, a) \to (n, b)$ if and only if $m = n$ and $a \to b$. Note that $(\mathbb{N} \times A, \to)$ consists of $\aleph_0$ many disjoint copies of $(A, \to)$. Finally, for a labelled graph $\mathfrak{G} = (V, \Sigma, \{\xrightarrow{a} \mid a \in \Sigma\})$, we define the labelled graph $\mathfrak{G}^+ = (V^+, \Sigma, \{\xrightarrow{a} \mid a \in \Sigma\})$. By the above definition, $\xrightarrow{a}$ is lifted to a relation on V^+.

We will consider first-order logic (FO) with equality over labelled graphs. Thus, for a set Σ of actions, we have for each $a \in \Sigma$ a binary relation symbol $a(x, y)$ in our language. The meaning of $a(x, y)$ is of course $x \xrightarrow{a} y$. If $\varphi(x_1, \ldots, x_n)$ is a first-order formula with free variables $x_1, \ldots, x_n$, $\mathfrak{G} = (V, \Sigma, \{\xrightarrow{a} \mid a \in \Sigma\})$ is a labelled graph, and $v_1, \ldots, v_n \in V$, then we write $\mathfrak{G} \models \varphi(v_1, \ldots, v_n)$ if φ evaluates to true in $\mathfrak{G}$, when variable x_i is instantiated by v_i $(1 \leq i \leq n)$. The *first-order theory* of a labelled transition graph $\mathfrak{G}$ is the set of all first-order sentences (i.e., first-order formulas without free variables) φ with $\mathfrak{G} \models \varphi$. The *quantifier rank* of a first-order formula is the maximal number of nested quantifiers in φ.

2.2 Trees and ground tree rewrite systems

Let $\preceq$ denote the prefix order on $\mathbb{N}^*$, i.e., $x \preceq y$ for $x, y \in \mathbb{N}^*$ if there exists $z \in \mathbb{N}^*$ with $y = xz$. A set $D \subseteq \mathbb{N}^*$ is called *prefix-closed* if for all $x, y \in \mathbb{N}^*$, $x \preceq y \in D$ implies $x \in D$. A *ranked alphabet* is a collection of finite and pairwise disjoint alphabets $A = (A_i)_{i \in [k]}$ for some $k \geq 0$ such that $A_0 \neq \emptyset$. For simplicity we identify A with $\bigcup_{i \in [k]} A_i$. A *ranked tree* over the ranked alphabet A is a mapping $t : D_t \to A$, where (i) D_t is non-empty, finite, and prefix-closed, and (ii) for each $x \in D_t$ with $t(x) \in A_i$ we have $x1, \ldots, xi \in D_t$ and $xj \notin D_t$ for each $j > i$. By Tr_A we denote the set of all ranked trees over the ranked alphabet A. Let $t \in \mathsf{Tr}_A$. Elements of D_t are called *nodes*. A *leaf* of t is a node x with $t(x) \in A_0$. An *internal node* of t is a node, which is not a leaf. We also refer to $\varepsilon \in D_t$ as the *root* of t. Let $\mathsf{size}(t) = |D_t|$ be the size of the tree t. It is easy to show that $|\{t \in \mathsf{Tr}_A \mid \mathsf{size}(t) \leq n\}| \leq |A|^n$. For $x \in [1, k]^*$ we define $xD_t = \{xy \in [1, k]^* \mid y \in D_t\}$ and $x^{-1}D_t = \{y \in [1, k]^* \mid xy \in D_t\}$. By $t^{\downarrow x}$ we denote the *subtree of* t with root x; it is defined by $D_{t^{\downarrow x}} = x^{-1}D_t$ and $t^{\downarrow x}(y) = t(xy)$. For a second tree $s \in \mathsf{Tr}_A$ and $x \in D_t$ we denote by $t[x/s]$ the tree that is obtained by replacing $t^{\downarrow x}$ in t by s. More formally, $D_{t[x/s]} = (D_t \setminus xD_{t^{\downarrow x}}) \cup xD_s$, $t[x/s](y) = t(y)$ for $y \in D_t \setminus xD_{t^{\downarrow x}}$, and $t[x/s](xz) = s(z)$ for $z \in D_s$.

Let C be a prefix-closed subset of D_t. We define the string of subtrees $t \setminus C$ as follows: If $C = \emptyset$, then $t \setminus C = t$. If $C \neq \emptyset$, then $t \setminus C = t^{\downarrow v_1} \cdots t^{\downarrow v_m}$, where $v_1, \ldots, v_m$ is a list of all nodes from $((C \cdot \mathbb{N}) \cap D_t) \setminus C$ in lexicographic order. Intuitively, we remove from the tree t the prefix-closed subset C and list all remaining maximal subtrees. For $n \in \mathbb{N}$ and a tree t we define the prefix-closed subset $\mathsf{up}(t, n) \subseteq D_t$ as $\mathsf{up}(t, n) = \{v \in D_t \mid \mathsf{size}(t^{\downarrow v}) > n\}$. Note that $t \setminus \mathsf{up}(t, n)$ is a list of all maximal subtrees of size at most n in t.

A *ground tree rewrite system (GTRS)* is tuple $\mathcal{R} = (A, \Sigma, R)$, where A is a ranked alphabet, Σ is finite set of actions, and $R \subseteq \mathsf{Tr}_A \times \Sigma \times \mathsf{Tr}_A$ is a finite set of rewrite rules. A rule (s, a, t) is also written as $s \overset{a}{\longmapsto} t$. The corresponding *ground tree rewrite graph* is $\mathfrak{G}(\mathcal{R}) =$

$(\mathsf{Tr}_A, \Sigma, \{\xrightarrow{a} \mid a \in \Sigma\})$, where for each $a \in \Sigma$, we have $t \xrightarrow{a} t'$ if and only if there exists a rule $(s \overset{a}{\longmapsto} s') \in R$ and $x \in D_t$ such that $t^{\downarrow x} = s$ and $t' = t[x/s']$. The following two lemmas are not very hard to prove.

LEMMA 2. *Let $\mathcal{R} = (A, \Sigma, R)$ be a GTRS and let r be the maximal size of a tree that appears in R. Let s and t be ranked trees such that $d_{\mathfrak{G}(\mathcal{R})}(s, t) \leq n$. Then $\mathsf{size}(t) \leq \mathsf{size}(s) + r \cdot n$.*

LEMMA 3. *Let $\mathcal{R} = (A, \Sigma, R)$ be a GTRS and let r be the maximal size of a tree that appears in R. Let t be a ranked tree, $k \in \mathbb{N}$, and let $C \subseteq \mathsf{up}(t, r \cdot k)$ be prefix-closed. Then we have $S_k(\mathfrak{G}(\mathcal{R}), t) \cong S_k(\mathfrak{G}(\mathcal{R})^+, t \setminus C)$ (where $\mathfrak{G}(\mathcal{R})^+$ is defined in Section 2.1).*

3 An $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, O(n))$ upper bound

In this section we will prove the following result:

THEOREM 4. *The problem of checking $\mathfrak{G}(\mathcal{R}) \models \varphi$ for a given GTRS $\mathcal{R} = (A, \Sigma, R)$ and an FO-sentence φ over the signature of $\mathfrak{G}(\mathcal{R})$ belongs to the class $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, O(n))$.*

It suffices to prove Thm. 4 for the case that the ranked alphabet A contains a symbol of rank at least two. A ground tree rewrite graph, where all symbols have rank at most 1 is in fact a suffix rewrite graph, i.e., pushdown graph. But every pushdown graph is first-order interpretable in a full $|\Gamma|$-ary tree Γ^* (with Γ finite), where the defining first-order formulas can be easily computed from the pushdown automaton. Finally, the first-order theory of a full tree Γ^* (with $|\Gamma| \geq 2$) is complete for the class $\mathsf{ATIME}(2^{O(n)}, O(n))$ [6, 18].

The proof of Thm. 4 will be divided into two steps. In a first step, we reduce the FO-theory for a given ground tree rewrite graph to the FO-theory for a very simple word rewrite graph of the form $\mathfrak{G}^+$, where $\mathfrak{G}$ is a finite labelled graph. Note that if V is the set of nodes of $\mathfrak{G}$, then V^+ is the set of nodes of $\mathfrak{G}^+$. Moreover, every edge in $\mathfrak{G}^+$ replaces a single symbol in a word by another symbol. In our reduction, the size of the set V is doubly exponential in the input size (which is the size of the input formula plus the size of the input GTRS). In a second step, we solve the FO-theory of a simple word structure $\mathfrak{G}^+$ on an alternating Turing machine. More precisely, in the long version [11] of this paper, we prove:

THEOREM 5. *There exists an alternating Turing-machine M, which accepts all pairs $(\mathfrak{G}, \varphi)$, where $\mathfrak{G}$ is a finite labelled graph and φ is an FO-sentence over the signature of $\mathfrak{G}$ with $\mathfrak{G}^+ \models \varphi$. Moreover, M runs in time $O(n^\ell \cdot |\varphi|)$, where n is the number of nodes of $\mathfrak{G}$ and ℓ is the quantifier rank of φ. Moreover, the number of alternations is bounded by $O(\ell)$.*

The proof of Thm. 5 uses an application of the method of Ferrante and Rackoff [9], which is one of most successful techniques for proving upper bounds for the complexity of first-order theories. In the rest of this section, we will derive Thm. 4 from Thm. 5.

Fix a GTRS $\mathcal{R} = (A, \Sigma, R)$. We restrict to the case $A_1 \neq \emptyset \neq A_2$; the general case (see [11]) is technically more complicated, but the main idea is the same. Let $\mathfrak{G} = \mathfrak{G}(\mathcal{R})$ and let

$$\varphi = Q_\ell x_\ell \cdots Q_1 x_1 \; Q_0 x_0 : \psi$$

be a FO-sentence of quantifier rank $\ell + 1$, where $Q_0, \ldots, Q_\ell \in \{\forall, \exists\}$ and ψ is quantifier-free. We want to check, whether $\mathfrak{G} \models \varphi$. Let r be the maximal size of a tree that appears in R, and let $p \geq 2$ the maximal rank of a symbol from A. Let us define the following subsets of Tr_A (where $0 \leq i \leq \ell$):

$$\begin{aligned} U &= \{t \in \mathsf{Tr}_A \mid \mathsf{size}(t) \leq r \cdot (p+1) \cdot 4^{\ell} + 1\}, \\ V_i &= \{t \in \mathsf{Tr}_A \mid \mathsf{size}(t) \leq r \cdot 4^{i}\} \subseteq U \text{ and} \\ W_i &= \{\alpha(u_1, \dots, u_q) \mid q \geq 1, \alpha \in A_q, u_1, \dots, u_q \in V_i\} \setminus V_i \subseteq U. \end{aligned}$$

Intuitions on these sets will be given below. Note that $\mathsf{size}(t) \leq r \cdot p \cdot 4^i + 1$ for all $t \in W_i$. We consider the set U as a finite alphabet and the sets V_i and W_i as subalphabets. Note that $|U| \leq |A|^{r\cdot(p+1)\cdot 4^{\ell}+1}$. On the set $(\mathbb{N} \times U^+) \cup U$ we define a labelled graph with the set of actions Σ. Take an action $\sigma \in \Sigma$. By our lifting constructions from Section 2.1, the binary relation $\xrightarrow{\sigma}$ on Tr_A is implicitly lifted to a binary relation on Tr_A^+ and $\mathbb{N} \times \mathsf{Tr}_A^+$. Since $(\mathbb{N} \times U^+) \cap U = \emptyset$, $\xrightarrow{\sigma}$ can be viewed as a binary relation on $(\mathbb{N} \times U^+) \cup U$; simply take the disjoint union of the relations on $(\mathbb{N} \times U^+)$ and U. We define the labelled graph

$$\mathfrak{S}_1 = ((\mathbb{N} \times U^+) \cup U,\ \Sigma,\ \{\xrightarrow{\sigma} \mid \sigma \in \Sigma\}).$$

For $0 \leq i \leq \ell$ and nodes $s_{i+1}, \dots, s_\ell \in (\mathbb{N} \times U^+) \cup U$ define the following nodes sets in $\mathfrak{S}_1$:

$$L_i = (\mathbb{N} \times V_i^* W_i V_i^*) \cup V_i, \quad L_i(s_{i+1}, \dots, s_\ell) = L_i \cup S_{3\cdot 4^i}(\mathfrak{S}_1, s_{i+1}, \dots, s_\ell).$$

Finally, define the FO-sentence (with relativized quantifiers)

$$\varphi_1 \;=\; Q_\ell x_\ell \in L_\ell \; Q_{\ell-1} x_{\ell-1} \in L_{\ell-1}(x_\ell) \;\cdots\; Q_0 x_0 \in L_0(x_1, \dots, x_\ell) : \psi \tag{1}$$

over the signature of $\mathfrak{S}_1$ Note that in φ_1 the constraint set for a variable x_i depends on the values for the already quantified variables $x_{i+1}, \dots, x_\ell$. Based on the following lemma, we show that $\mathfrak{G} \models \varphi$ if and only if $\mathfrak{S}_1 \models \varphi_1$.

LEMMA 6. *Assume that* $0 \leq i \leq \ell$,

- $\bar{s} = (s_{i+1}, \dots, s_\ell) \in ((\mathbb{N} \times U^+) \cup U)^{\ell-i}$ *with* $s_j \in L_j \cup S_{3\cdot 4^j}(\mathfrak{S}_1, s_{j+1}, \dots, s_\ell)$ *for all* $j \in [i+1, \ell]$,
- $\bar{t} = (t_{i+1}, \dots, t_\ell) \in \mathsf{Tr}_A^{\ell-i}$, *and*
- $f : S_{4^{i+1}}(\mathfrak{S}_1, \bar{s}) \to S_{4^{i+1}}(\mathfrak{G}, \bar{t})$ *is an isomorphism such that* $f\restriction S_{4^{i+1}}(\mathfrak{S}_1, s_j)$ *is the identity for all* $j \in [i+1, \ell]$ *with* $t_j \in V_{i+1}$ *or* $s_j \in V_{i+1}$.†

Then, the following holds:

(a) For all $t_i \in \mathsf{Tr}_A$ *there exists* $s_i \in L_i \cup S_{3\cdot 4^i}(\mathfrak{S}_1, \bar{s})$ *and an isomorphism* $g : S_{4^i}(\mathfrak{S}_1, s_i, \bar{s}) \to S_{4^i}(\mathfrak{G}, t_i, \bar{t})$ *such that* f *and* g *are compatible and* $g\restriction S_{4^i}(\mathfrak{S}_1, s_j)$ *is the identity for all* $j \in [i, \ell]$ *with* $t_j \in V_i$ *or* $s_j \in V_i$.

(b) For all $s_i \in L_i \cup S_{3\cdot 4^i}(\mathfrak{S}_1, \bar{s})$ *there exists* $t_i \in \mathsf{Tr}_A$ *and an isomorphism* $g : S_{4^i}(\mathfrak{S}_1, s_i, \bar{s}) \to S_{4^i}(\mathfrak{G}, t_i, \bar{t})$ *such that* f *and* g *are compatible and* $g\restriction S_{4^i}(\mathfrak{S}_1, s_j)$ *is the identity for all* $j \in [i, \ell]$ *with* $t_j \in V_i$ *or* $s_j \in V_i$.

Before we prove the lemma, let us provide some intuition. For case (a) we will basically distinguish two cases: In case t_i is "close" to some tree in the tuple $\bar{t}$, then the simulating

†For $i = \ell$, f is the isomorphism between empty sets.

s_i can safely be chosen as t_i itself. In case t_i is "far" to all trees in $\bar{t}$, we distinguish two cases: Either the size of t_i exceeds $r \cdot 4^i$ or not. If it does, then s_i will be chosen as a pair from $\{n\} \times V_i^* W_i V_i^*$ for some fresh number n that does not appear as a first component of any element in $\bar{t}$, and where the second component of s_i consists basically of $t_i \setminus C$ for some prefix-closed subset C of t_i's nodes. Intuitively, this means that s_i does not have to be "too big" in order to simulate t_i: only "small" subtrees of t_i have to be accounted for. Lemma 3 will be crucial. In case $|t_i| \leq r \cdot 4^i$, we can prove that we can set $s_i = t_i \in V_i$. For case (b) we can proceed similarly, but the main crux is that for each element $s_i \in \mathbb{N} \times V_i^* W_i V_i^*$ we can build a tree $t_i \in \mathsf{Tr}_A$ such that the spheres of radius 4^i around s_i and t_i are isomorphic.

PROOF (of Lemma 6). Let $f : S_{4^{i+1}}(\mathfrak{S}_1, \bar{s}) \to S_{4^{i+1}}(\mathfrak{G}, \bar{t})$ be an isomorphism such that $f \restriction S_{4^{i+1}}(\mathfrak{S}_1, s_j)$ is the identity for all $i+1 \leq j \leq \ell$ with $t_j \in V_{i+1}$ or $s_j \in V_{i+1}$. Let us first prove statement (a). Let $t_i \in \mathsf{Tr}_A$. We distinguish two cases:

Case 1. $t_i \in S_{3 \cdot 4^i}(\mathfrak{G}, \bar{t})$. Thus, t_i belongs to the range of the isomorphism f. Moreover, $S_{4^i}(\mathfrak{G}, t_i, \bar{t}) \subseteq S_{4^{i+1}}(\mathfrak{G}, \bar{t})$. Then, we set $s_i = f^{-1}(t_i) \in S_{3 \cdot 4^i}(\mathfrak{S}_1, \bar{s})$. We define g as the restriction of f to the set $S_{4^i}(\mathfrak{S}_1, s_i, \bar{s}) \subseteq S_{4^{i+1}}(\mathfrak{S}_1, \bar{s})$. Now, assume that $t_i \in V_i$, i.e., $\mathsf{size}(t_i) \leq r \cdot 4^i$. We have to show that $g \restriction S_{4^i}(\mathfrak{S}_1, s_i)$ is the identity. Let t_j $(j > i)$ be such that $d_{\mathfrak{G}}(t_i, t_j) \leq 3 \cdot 4^i$. Lemma 2 implies $\mathsf{size}(t_j) \leq \mathsf{size}(t_i) + r \cdot 3 \cdot 4^i \leq r \cdot 4^i + r \cdot 3 \cdot 4^i = r \cdot 4^{i+1}$. Hence, $t_j \in V_{i+1}$ and $f \restriction S_{4^{i+1}}(\mathfrak{S}_1, s_j)$ is the identity by assumption. Since $S_{4^i}(\mathfrak{S}_1, s_i) \subseteq S_{4^{i+1}}(\mathfrak{S}_1, s_j)$, it follows that $g \restriction S_{4^i}(\mathfrak{S}_1, s_i)$ is the identity too. If $s_i \in V_i$, then we can argue analogously.

Case 2. $t_i \notin S_{3 \cdot 4^i}(\mathfrak{G}, \bar{t})$. Thus, $t_i \notin S_{2 \cdot 4^i + 1}(\mathfrak{G}, \bar{t})$. We will find $s_i \in L_i$ and an isomorphism $f' : S_{4^i}(\mathfrak{S}_1, s_i) \to S_{4^i}(\mathfrak{G}, t_i)$ such that $s_i \notin S_{3 \cdot 4^i}(\mathfrak{S}_1, \bar{s})$. Then, Lemma 1 implies that $g = (f \restriction S_{4^i}(\mathfrak{S}_1, \bar{s})) \uplus f'$ is an isomorphism from $S_{4^i}(\mathfrak{S}_1, s_i, \bar{s})$ to $S_{4^i}(\mathfrak{G}, t_i, \bar{t})$, which is compatible with f. Moreover, we will show that if $t_i \in V_i$ or $s_i \in V_i$, then f' is the identity. In order to define s_i, let $t_i \setminus \mathsf{up}(t_i, r \cdot 4^i) = u_1 \cdots u_m$. Thus $u_1, \ldots, u_m \in V_i$. Choose a number $n \in \mathbb{N}$ such that n does not appear as a first component of a pair from $\{s_{i+1}, \ldots, s_\ell\} \cap (\mathbb{N} \times U^+)$.

Case 2.1. $\mathsf{size}(t_i) > r \cdot 4^i$. Then there exists a symbol $\alpha \in A$ of rank $q \geq 1$, an index $1 \leq j \leq m - q + 1$, and a prefix-closed subset $C \subseteq \mathsf{up}(t_i, r \cdot 4^i)$ such that $\alpha(u_j, \ldots, u_{j+q-1}) \in W_i$ and $t_i \setminus C = u_1 u_2 \cdots u_{j-1} \alpha(u_j, \ldots, u_{j+q-1}) u_{j+q} \cdots u_m$. We have $t_i \setminus C \in V_i^* W_i V_i^*$. Let $s_i = (n, t_i \setminus C) \in L_i$. Due to the choice of n, we have $s_i \notin S_\rho(\mathfrak{S}_1, \bar{s})$ for all ρ. Moreover, Lemma 2 and the definition of the set U implies that the sphere of radius 4^i around every tree from $u_1, u_2, \ldots, u_{j-1}, \alpha(u_j, \ldots, u_{j+q-1}), u_{j+q}, \ldots, u_m$ is completely contained in U. With Lemma 3 (setting $k = 4^i$), we get $S_{4^i}(\mathfrak{S}_1, s_i) \cong S_{4^i}(\mathfrak{G}, t_i)$ via an isomorphism f'. Finally, $\mathsf{size}(t_i) > r \cdot 4^i$ and $s_i \notin U$. Hence, neither $s_i \in V_i$ nor $t_i \in V_i$.

Case 2.2. $\mathsf{size}(t_i) \leq r \cdot 4^i$. Hence, $m = 1$ and $t_i = u_1$. We set $s_i = t_i = u_1 \in V_i \subseteq L_i$. Thus $\mathsf{size}(s_i) = \mathsf{size}(t_i) \leq r \cdot 4^i$. We have $S_{4^i}(\mathfrak{G}, t_i) \subseteq U$, which implies $S_{4^i}(\mathfrak{S}_1, s_i) = S_{4^i}(\mathfrak{G}, t_i)$. Assume that $s_i \in S_{3 \cdot 4^i}(\mathfrak{S}_1, \bar{s})$. We will deduce a contradiction. Let $j > i$ such that $d_{\mathfrak{S}_1}(s_i, s_j) \leq 3 \cdot 4^i$. Since $s_i \in U$, we must have $s_j \in U$ as well (there is no path in $\mathfrak{S}_1$ between the sets U and $\mathbb{N} \times U^+$). Moreover, Lemma 2 implies $\mathsf{size}(s_j) \leq \mathsf{size}(s_i) + r \cdot 3 \cdot 4^i \leq r \cdot 4^{i+1}$, i.e., $s_j \in V_{i+1}$. Hence, $f \restriction S_{4^{i+1}}(\mathfrak{S}_1, s_j)$ is the identity and $t_i \in S_{3 \cdot 4^i}(\mathfrak{G}, t_j)$, a contradiction. We can finally choose for f' the identity isomorphism on $S_{4^i}(\mathfrak{S}_1, s_i) = S_{4^i}(\mathfrak{G}, t_i)$. This proves (a).

Let us now prove (b). Let $s_i \in L_i \cup S_{3 \cdot 4^i}(\mathfrak{S}_1, \bar{s})$. Again, we distinguish two cases.

Case 1. $s_i \in S_{3\cdot 4^i}(\mathfrak{S}_1, \bar{s})$. This implies $S_{4^i}(\mathfrak{S}_1, s_i, \bar{s}) \subseteq S_{4^{i+1}}(\mathfrak{S}_1, \bar{s})$. We set $t_i = f(s_i) \in S_{3\cdot 4^i}(\mathfrak{G}, \bar{t})$. We can conclude as in Case 1 for point (a) above.

Case 2. $s_i \notin S_{3\cdot 4^i}(\mathfrak{S}_1, \bar{s})$. Hence, $s_i \in L_i$. We will find $t_i \in \mathsf{Tr}_A$ and an isomorphism $f' : S_{4^i}(\mathfrak{S}_1, s_i) \to S_{4^i}(\mathfrak{G}, t_i)$ such that $t_i \notin S_{3\cdot 4^i}(\mathfrak{G}, \bar{t})$. Then, Lemma 1 implies that the mapping $g = (f \restriction S_{4^i}(\mathfrak{S}_1, \bar{s})) \uplus f'$ is an isomorphism from $S_{4^i}(\mathfrak{S}_1, s_i, \bar{s})$ to $S_{4^i}(\mathfrak{G}, t_i, \bar{t})$, which is compatible with f. Moreover, we will show that if $t_i \in V_i$ or $s_i \in V_i$, then f' is the identity.

Case 2.1. $s_i \in V_i \subseteq \mathsf{Tr}_A$. We set $t_i = s_i$. Hence, $\mathsf{size}(t_i) \le r \cdot 4^i$ and one can argue analogously to Case 2.2 in the proof for statement (a).

Case 2.2. $s_i \in \mathbb{N} \times V_i^* W_i V_i^*$. Let $s_i = (n, w)$ where $w \in V_i^* W_i V_i^*$. Hence, $w = u_1 \cdots u_m$ with $u_1, \ldots, u_m \in V_i \cup W_i$. Moreover, there is a unique index j such that $u_j \in W_i$. Assume that $u_j = \alpha(u'_1, \ldots, u'_q)$ with $\alpha \in A$ and $q \ge 1$. By the definition of the set W_i we have $u'_1, \ldots, u'_q \in V_i$. Since we assume that $A_1 \neq \emptyset \neq A_2$, we can choose for t_i a tree with the following properties:

- $t_i \setminus \mathsf{up}(t_i, r \cdot 4^i) = u_1 \cdots u_{j-1} u'_1 \cdots u'_q u_{j+1} \cdots u_m$. For this, we connect all trees $u_1, \ldots, u_m$ to one tree using a chain of binary symbols, starting from $u_j \in W_i$.
- $t_i \notin S_{3\cdot 4^i}(\mathfrak{G}, \bar{t})$. This can be enforced by adding a long enough chain of unary symbols to the root.

With Lemma 3, the first point implies $S_{4^i}(\mathfrak{S}_1, s_i) \cong S_{4^i}(\mathfrak{G}, t_i)$. Moreover, since t_i contains a subtree from W_i, we have $t_i \notin V_i$. ∎

Using the classical back-and-forth argument from the proof of the Ehrenfeucht-Fraïssé-Theorem, we can deduce the following lemma from Lemma 6.

LEMMA 7. *Assume that* $-1 \le i \le \ell$,

- $\bar{s} = (s_{i+1}, \ldots, s_\ell) \in ((\mathbb{N} \times U^+) \cup U)^{\ell - i}$ *with* $s_j \in L_j \cup S_{3\cdot 4^j}(s_{j+1}, \ldots, s_\ell)$ *for* $j \in [i+1, \ell]$,
- $\bar{t} = (t_{i+1}, \ldots, t_\ell) \in \mathsf{Tr}_A^{\ell - i}$, *and*
- $f : S_{4^{i+1}}(\mathfrak{S}_1, \bar{s}) \to S_{4^{i+1}}(\mathfrak{G}, \bar{t})$ *is an isomorphism such that* $f \restriction S_{4^{i+1}}(\mathfrak{S}_1, s_j)$ *is the identity for all* $j \in [i+1, \ell]$ *with* $t_j \in V_{i+1}$ *or* $s_j \in V_{i+1}$.

Then, for every quantifier-free FO-formula ψ *over the signature of* $\mathfrak{G}$ *and all* $Q_0, \ldots, Q_i \in \{\forall, \exists\}$ *we have:* $\mathfrak{S}_1 \models Q_i x_i \in L_i(\bar{s}) \cdots Q_0 x_0 \in L_0(x_1, \ldots, x_i, \bar{s}) : \psi(x_0, \ldots, x_i, \bar{s})$ *if and only if* $\mathfrak{G} \models Q_i x_i \cdots Q_0 x_0 : \psi(x_0, \ldots, x_i, \bar{t})$.

Setting $i = \ell$ in Lemma 7, it follows $\mathfrak{G} \models \varphi$ if and only if $\mathfrak{S}_1 \models \varphi_1$, where φ_1 is from (1). It now requires rather easy but technical arguments to compute a finite labelled graph $\mathfrak{G}'$ with doubly exponentially (in our input size) many nodes and a first-order sentence φ' (of polynomial size and quantifier rank $O(\ell)$) such that $\mathfrak{S}_1 \models \varphi_1$ if and only if $(\mathfrak{G}')^+ \models \varphi'$. To get rid of the direct product with $\mathbb{N}$ in the node set of $\mathfrak{S}_1$, we use the simple fact that for an arbitrary graph $(V, \to)$, $((V \cup \{\$\})^+ \setminus \{\$\}^+, \to)$ (where $\$ \notin V$ is a new symbol) is isomorphic to $(\mathbb{N} \times V^+, \to)$; here we use the lifting constructions from Section 2.1. Then, Thm. 4 can be easily derived from Thm. 5. We refer to [11] for details.

4 A lower bound

In the long version of this paper [11], we prove the following lower bound:

THEOREM 8. *There is a fixed GTRS $\mathcal{R}$ such that the first-order theory of $\mathfrak{G}(\mathcal{R})$ is hard for* $\mathsf{ATIME}(2^{2^{\mathsf{poly}(n)}}, \mathsf{poly}(n))$ *under logspace reductions.*

In this section, we will sketch a proof for the slightly weaker lower bound of 2NEXP. This will be achieved using a tiling problem. Tiling problems turned out to be an important tool for proving lower bounds in logic, see e.g. [2]. So, let us start with a few definitions concerning tiling systems. A *tiling system* is a tuple $S = (\Theta, \mathbb{H}, \mathbb{V})$, where Θ is a finite set of *tile types*, $\mathbb{H} \subseteq \Theta \times \Theta$ is a *horizontal matching relation*, and $\mathbb{V} \subseteq \Theta \times \Theta$ is a *vertical matching relation*. A mapping $\sigma : [0, k-1] \times [0, k-1] \to \Theta$ (where $k \geq 0$) is a *k-solution for S* if for all $x, y \in [0, k-1]$ the following holds: (i) if $x < k-1$, $\sigma(x, y) = \theta$, and $\sigma(x+1, y) = \theta'$, then $(\theta, \theta') \in \mathbb{H}$, and (ii) if $y < k-1$, $\sigma(x, y) = \theta$, and $\sigma(x, y+1) = \theta'$, then $(\theta, \theta') \in \mathbb{V}$. Let $\mathsf{Sol}_k(S)$ denote the set of all k-solutions for S. Let $w = \theta_0 \cdots \theta_{n-1} \in \Theta^n$ be a word and let $k \geq n$. With $\mathsf{Sol}_k(S, w)$ we denote the set of all $\sigma \in \mathsf{Sol}_k(S)$ such that $\sigma(x, 0) = \theta_x$ for all $x \in [0, n-1]$. For a fixed tiling system S, its $(2^{2^n} \times 2^{2^n})$ *tiling problem* asks for a given word $w \in \Theta^n$, whether $\mathsf{Sol}_{2^{2^n}}(S, w) \neq \emptyset$ holds. Using the standard encoding of Turing machine computations by tilings, it follows easily that there exists a fixed tiling system S_0 whose $(2^{2^n} \times 2^{2^n})$ tiling problem is 2NEXP-hard under logspace reductions; see also [2]. Let us fix such a tiling system $S_0 = (\Theta_0, \mathbb{H}_0, \mathbb{V}_0)$ for the rest of the section.

We now define a fixed GTRS $\mathcal{R}_0 = (A, \Sigma, R)$ and prove that the first-order theory of $\mathfrak{G}(\mathcal{R}_0)$ is 2NEXP-hard under logspace reductions. We define $A_0 = \{\heartsuit, \mathbb{1}, \mathbb{1}_\dagger, \mathbb{1}_\ddagger, \mathbb{O}, \mathbb{O}_\dagger, \mathbb{O}_\ddagger\}$, $A_1 = \Theta_0$, $A_2 = \{\bullet\}$, and $\Sigma = \{\ell, r, h, u, m_\dagger, m_\ddagger, \} \cup \Theta_0 \cup A_0$. The set of rewrite rules R is given as follows:

$$
\begin{array}{ll}
X \xmapsto{X} X \text{ for all } X \in A_0 & \theta(X_\ddagger) \xmapsto{\theta} \theta(X_\ddagger) \text{ for all } \theta \in \Theta_0, X \in \{\mathbb{1}, \mathbb{O}\} \\
X \xmapsto{m_\dagger} X_\dagger \text{ for all } X \in \{\mathbb{1}, \mathbb{O}\} & \bullet(\heartsuit, \heartsuit) \xmapsto{u} \heartsuit \\
X_\dagger \xmapsto{m_\ddagger} X_\ddagger \text{ for all } X \in \{\mathbb{1}, \mathbb{O}\} & \bullet(\heartsuit, X_\ddagger) \xmapsto{r} X_\ddagger \text{ for all } X \in \{\mathbb{1}, \mathbb{O}\} \\
X_\dagger \xmapsto{h} \heartsuit \text{ for all } X \in \{\mathbb{1}, \mathbb{O}\} & \bullet(X_\ddagger, \heartsuit) \xmapsto{\ell} X_\ddagger \text{ for all } X \in \{\mathbb{1}, \mathbb{O}\}
\end{array}
$$

For the rest of this section we fix $\mathfrak{G}_0 = \mathfrak{G}(\mathcal{R}_0)$ and an input $w = \theta_0 \cdots \theta_{n-1} \in \Theta_0^n$ of the $(2^{2^n} \times 2^{2^n})$ tiling problem for S_0. Our goal is to compute in logspace from w a first-order sentence φ over Σ such that $\mathsf{Sol}_{2^{2^n}}(S_0, w) \neq \emptyset$ if and only if $\mathfrak{G}_0 \models \varphi$. We will need the following lemma, which goes back to the work of Fischer and Rabin.

LEMMA 9. *Given a subset of actions $\Gamma \subseteq \Sigma$ and the binary representation of $j \in [0, 2^{n+1}]$, one can compute in logspace a first-order formula $\Gamma^j(x, y)$ such that for all $t, t' \in \mathsf{Tr}_A$ we have $\mathfrak{G}_0 \models \Gamma^j(t, t')$ if and only if there is a path of length j from t to t' in the graph $(\mathsf{Tr}_A, \bigcup_{\gamma \in \Gamma} \xrightarrow{\gamma})$.*

If $\Gamma = \{\gamma\}$, we write $\gamma^j(x, y)$ for the formula $\Gamma^j(x, y)$. For $\Gamma_1, \ldots, \Gamma_k \subseteq \Sigma$ and $j_1, \ldots, j_k \in \mathbb{N}$, we write $[\Gamma_1^{j_1} \cdots \Gamma_k^{j_k}](x, y)$ for $\exists x_0, \ldots, x_k : \big(x_0 = x \wedge x_k = y \wedge \bigwedge_{i=1}^k \Gamma_i^{j_i}(x_{i-1}, x_i)\big)$.

A tree $t \in \mathsf{Tr}_A$ is a *tile tree* if $t = \theta(t')$ for some $\theta \in \Theta_0$ and $t' \in \mathsf{Tr}_{\{\mathbb{O}, \mathbb{1}, \bullet\}}$ such that $\{1, 2\}^{n+1}$ is the set of leaves of t'. Fix a tile tree $t = \theta(t')$. Then t has precisely $2^{n+1} = 2 \cdot 2^n$ leaves. For a leaf λ of t let $\mathsf{lex}(\lambda) \in [0, 2^{n+1} - 1]$ be the position of λ among all leaves w.r.t. the lexicographic order (starting with 0). The intention is that t represents the θ-labeled grid element $(M, N) \in [0, 2^{2^n} - 1]^2$, where each leaf λ that is a left (resp. right) child represents the $\lfloor \frac{\mathsf{lex}(\lambda)}{2} \rfloor^{\text{th}}$ least significant bit of the 2^n-bit binary presentation of M (resp. N): In case λ is a left child, then $t(\lambda) = \mathbb{O}$ (resp. $t(\lambda) = \mathbb{1}$) if and only if the $\lfloor \frac{\mathsf{lex}(\lambda)}{2} \rfloor^{\text{th}}$ least significant bit of M equals 0 (resp. 1) and analogously if λ is a right child this corresponds to N. We say a leaf λ of a tree t is *marked* (resp. *selected*) if $t(\lambda) = X_\dagger$ (resp. $t(\lambda) = X_\ddagger$) for some $X \in \{\mathbb{O}, \mathbb{1}\}$. A *marked tile tree* is a tree that can be obtained from a tile tree t by marking *every* leaf of t. For the rest of this section, let $D = 2^{n+1} - (n+2)$.

LEMMA 10. *One can compute in logspace a first-order formula* $\mathsf{marked}(x)$ *such that for every tree* $t \in \mathsf{Tr}_{A\setminus\{\mathbb{O}_\ddagger,\mathbb{1}_\ddagger,\heartsuit\}}$ *with precisely* 2^{n+1} *marked leaves we have:* $\mathfrak{G}_0 \models \mathsf{marked}(t)$ *if and only if the marked leaves of* t *are the leaves of some (unique) marked tile subtree of* t*.*

PROOF. The formula $\mathsf{marked}(x)$ states that once we select any of the 2^{n+1} marked leaves, we can execute from the resulting tree some sequence in the language $h^{2^{n+1}-1}u^D\{\ell,r\}^{n+1}\Theta_0$. Formally, we define $\mathsf{marked}(x) = \forall y (m_\ddagger(x,y) \to \exists z : [h^{2^{n+1}-1}u^D\{\ell,r\}^{n+1}\Theta_0](y,z))$. Let us explain the intuition behind this. Assume that we select exactly one of the 2^{n+1} marked leaves of t, and let t' be the resulting tree. First, note that by executing the sequence $h^{2^{n+1}-1}$ from t', we replace each of the marked leaves of t' with the symbol $\heartsuit$, reaching a tree t''. Second, by executing u^D from t'' we reach (in case t contains a marked tile subtree) a tree t''', which has the form of a chain where the lowest leaf is labeled with $\mathbb{O}_\ddagger$ or $\mathbb{1}_\ddagger$ and all other leaves are labeled with $\heartsuit$. Next, we can "shrink" the chain t''' to the tree $\theta(X_\ddagger)$ by executing some sequence from $\{\ell,r\}^{n+1}$. To $\theta(X_\ddagger)$ we can finally apply the action $\theta \in \Theta_0 \subseteq \Sigma$. ∎

A *grid tree* is a tree t for which every leaf is inside a subtree of t that is a tile tree. The following lemma can be proven similarly as Lemma **??**.

LEMMA 11. *One can compute in logspace a first-order formula* $\mathsf{grid}(x)$ *such that for all* $t \in \mathsf{Tr}_A$ *we have* $\mathfrak{G}_0 \models \mathsf{grid}(t)$ *if and only if* t *is a grid tree.*

A *marked grid tree* is a tree that can be obtained from a grid tree t by replacing exactly one tile subtree of t by some marked tile tree. A *selected grid tree* is a tree that can be obtained from a marked grid tree t by selecting *precisely one* marked leaf of t. For each $i \in [1, n+1]$ we define the logspace computable FO-formula

$$\mathrm{bit}_i(x) = \exists y : [h^{2^{n+1}-1}u^D\{\ell,r\}^{i-1}r](x,y).$$

It is not hard to see that for every selected grid tree t with selected leaf λ we have that the i^{th} least significant bit of $\mathsf{lex}(\lambda)$ is 1 if and only if $\mathfrak{G}_0 \models \mathrm{bit}_i(t)$. Next, compute for each $\circ \in \{<,=\}$ in logspace a first-order formula $\varphi_\circ(x,y)$ such that for every two selected grid trees t_1 and t_2 with selected leaves λ_1 and λ_2 we have $\mathfrak{G}_0 \models \varphi_\circ(t_1,t_2)$ if and only if $\mathsf{lex}(\lambda_1) \circ \mathsf{lex}(\lambda_2)$. We define

$$\varphi_<(x,y) = \bigvee_{j\in[1,n+1]} \left((\neg\mathrm{bit}_j(x) \wedge \mathrm{bit}_j(y)) \wedge \bigwedge_{1\le i<j} (\mathrm{bit}_i(x) \leftrightarrow \mathrm{bit}_i(y)) \right)$$

and $\varphi_=(x,y) = \neg\varphi_<(x,y) \wedge \neg\varphi_<(y,x)$. Recall that the marked tile subtree of a marked grid tree t represents a θ-labeled grid element $(M,N) \in [0, 2^{2^n}-1]^2$ for some $\theta \in \Theta_0$. Let us define $M(t) = M$, $N(t) = N$, and $\Theta_0(t) = \theta$.

LEMMA 12. *One can compute in logspace first-order formulas $\varphi_\theta(x)$, $\varphi^i_M(x,x')$, $\varphi^i_N(x,x')$, where $\theta \in \Theta_0$ and $i \in \{0,1\}$ such that for all marked grid trees t and t' the following holds:*
(1) $\mathfrak{G}_0 \models \varphi_\theta(t)$ if and only if $\Theta_0(t) = \theta$ and
(2) $\mathfrak{G}_0 \models \varphi^i_Y(t,t')$ (where $Y \in \{M,N\}$) if and only if $Y(t)+i = Y(t')$

PROOF. For point (1), let $\varphi_\theta(x) = \exists y : [m_\ddagger h^{2^{n+1}-1} u^D \{\ell, r\}^{n+1}\theta](x,y)$. For point (2), we only construct the formula $\varphi^1_M(x,x')$. For a selected grid tree z, the formula $l(z) = \exists u, v(h(z,u) \wedge \ell(u,v))$ expresses that the selected leaf is a left child. Then define

$$\varphi^1_M(x,y) = \exists x', y'(m_\ddagger(x,x') \wedge m_\ddagger(y,y') \wedge \varphi_=(x',y') \wedge \mathbb{O}_\ddagger(x',x') \wedge \mathbb{1}_\ddagger(y',y') \wedge l(x') \wedge \psi_1 \wedge \psi_2).$$

Thus, we select a position $p \in [0, 2^n-1]$ that is set to 0 (resp. 1) in the binary representation of $M(t)$ (resp. $M(t')$). The formula $\psi_1(x,y,x',y')$ is the conjunction

$$\forall z((m_\ddagger(x,z) \wedge \varphi_<(z,x') \wedge l(z)) \to \mathbb{1}_\ddagger(z,z)) \quad \wedge \quad \forall z((m_\ddagger(y,z) \wedge \varphi_<(z,y') \wedge l(z)) \to \mathbb{O}_\ddagger(z,z)).$$

It expresses that each bit at some position that is smaller than p is set to 1 (resp. 0) in $M(t)$ (resp. $M(t')$). Finally, the formula $\psi_2(x,y,x',y')$ is

$$\forall u, v\, ((m_\ddagger(x,u) \wedge m_\ddagger(y,v) \wedge \varphi_=(u,v) \wedge \varphi_<(x',u) \wedge l(u)) \to (\mathbb{1}_\ddagger(u,u) \leftrightarrow \mathbb{1}_\ddagger(v,v))).$$

It expresses that the binary representations of $M(t)$ and $M(t')$ agree on each position $> p$. ∎

Recall that $w = \theta_0 \cdots \theta_{n-1}$. We define the formula $\mathsf{sol}(x)$ as the conjunction of $\mathsf{grid}(x)$ and the following formulas, where $\mathsf{mark}(z_1,z_2)$ abbreviates $m_\ddagger^{2^{n+1}}(z_1,z_2) \wedge \mathsf{marked}(z_2)$:

- Grid element $(j,0)$ is labeled by θ_j for all $j \in [0,n-1]$:

$$\exists y_0,\ldots,y_{n-1}\Big(\bigwedge_{j\in[0,n-1]} (\mathsf{mark}(x,y_j) \wedge \varphi_{\theta_j}(y_j)) \;\wedge\; \forall z(m_\ddagger(y_0,z) \to \mathbb{O}_\ddagger(z,z)) \;\wedge$$
$$\bigwedge_{j\in[1,n-1]} (\varphi^1_M(y_{j-1},y_j) \wedge \varphi^0_N(y_{j-1},y_j))\Big)$$

- If we mark a tile subtree of x that corresponds to the grid element (M,N) and $M < 2^{2^n}-1$, a tile subtree of x that corresponds to $(M+1,N)$ satisfies the horizontal matching relation:

$$\forall y((\mathsf{mark}(x,y) \wedge \exists z(m_\ddagger(y,z) \wedge \mathbb{O}_\ddagger(z,z) \wedge l(z))) \to$$
$$\exists y'(\mathsf{mark}(x,y') \wedge \varphi^1_M(y,y') \wedge \varphi^0_N(y,y') \wedge \bigvee_{(\theta,\theta')\in\mathbb{H}_0} (\varphi_\theta(y) \wedge \varphi_{\theta'}(y'))))$$

- Analogously, we express that the vertical matching relation is respected and for each grid element there is at most one tile type.

It follows by construction that $\mathsf{Sol}_{2^{2^n}}(S_0,w) \neq \emptyset$ if and only if $\mathfrak{G}_0 \models \exists x : \mathsf{sol}(x)$. Thus, the first-order theory of $\mathfrak{G}_0$ is indeed 2NEXP-hard under logspace reductions.

We conclude with a non-elementary lower bound for ground tree rewrite graphs with an additional unary predicate. For a GTRS $\mathcal{R} = (A, \Sigma, R)$ and a set of trees $L \subseteq \mathsf{Tr}_A$, we denote with $(\mathfrak{G}(\mathcal{R}), L)$ the structure that results from the labelled graph $\mathfrak{G}(\mathcal{R})$ by adding the set L as an additional unary predicate. Note that if L is a regular set of trees, then $(\mathfrak{G}(\mathcal{R}), L)$ is a tree-automatic structure, and hence has a decidable first-order theory. On the other hand, a reduction from satisfiability for first-order logic over binary words (see [11]) shows:

THEOREM 13. *There exists a fixed GTRS $\mathcal{R}_1 = (A, \Sigma, R)$ and a fixed regular tree language $L \subseteq \mathsf{Tr}_A$ such that the first-order theory of $(\mathfrak{G}(\mathcal{R}_1), L)$ is non-elementary.*

References

[1] L. Berman. The complexity of logical theories. *Theoret. Comput. Sci.*, 11:71–77, 1980.

[2] E. Börger, E. Grädel, and Y. Gurevich. *The classical decision problem*. Springer, 2001.

[3] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model-checking. In *Proc. CONCUR'97*, LNCS 1243, pages 135–150. Springer, 1997.

[4] W. S. Brainerd. Tree generating regular systems. *Inform. and Control*, 14(2):217–231, 1969.

[5] A. K. Chandra, D. C. Kozen, and L. J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981.

[6] K. J. Compton and C. W. Henson. A uniform method for proving lower bounds on the computational complexity of logical theories. *Ann. Pure Appl. Logic*, 48(1):1–79, 1990.

[7] J.-L. Coquidé, M. Dauchet, R. Gilleron, and S. Vágvölgyi. Bottom-up tree pushdown automata: Classification and connection with rewrite systems. *Theor. Comput. Sci.*, 127(1):69–98, 1994.

[8] M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *Proc. LICS'90*, pages 242–248. IEEE Computer Society, 1990.

[9] J. Ferrante and C. Rackoff. *The Computational Complexity of Logical Theories*. Number 718 in Lecture Notes in Mathematics. Springer, 1979.

[10] S. Göller and A. W. Lin. The complexity of verifying ground tree rewrite systems. In *Proc. LICS 2011*, pages 279-288. IEEE Computer Society, 2011.

[11] S. Göller and M. Lohrey. The first-order theory of ground tree rewrite graphs. arXiv.org, 2011. `http://arxiv.org/abs/1107.0919`.

[12] D. Kuske and M. Lohrey. Automatic structures of bounded degree revisited. In *Proc. CSL 2009*, LNCS 5771, pages 364–378. Springer, 2009.

[13] C. Löding. *Infinite Graphs Generated by Tree Rewriting*. PhD thesis, RWTH Aachen, 2003.

[14] D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theor. Comput. Sci.*, 37:51–75, 1985.

[15] L. J. Stockmeyer. *The complexity of decision problems in automata theory and logic*. PhD thesis, Department of Electrical Engineering, MIT, 1974.

[16] S. Tison. Fair termination is decidable for ground systems. In *Proc. RTA'89*, LNCS 355, pages 462–476. Springer, 1989.

[17] A. W. To. *Model Checking Infinite-State Systems: Generic and Specific Approaches*. PhD thesis, LFCS, School of Informatics, University of Edinburgh, 2010.

[18] H. Vogel. Turing machines with linear alternation, theories of bounded concatenation and the decision problem of first-order theories,. *Theoret. Comput. Sci.*, 23:333–337, 1983.

[19] I. Walukiewicz. Model checking CTL properties of pushdown systems. In *Proc. FSTTCS 2000*, LNCS 1974, pages 127–138. Springer, 2000.

[20] I. Walukiewicz. Pushdown processes: Games and model-checking. *Inf. Comput.*, 164(2):234–263, 2001.