Decidability of MSO Theories of Tree Structures

Angelo Montanari and Gabriele Puppis

Dipartimento di Matematica e Informatica, Università di Udine
via delle Scienze 206, 33100 Udine, Italy
{montana,puppis}@dimi.uniud.it

Abstract. In this paper we provide an automaton-based solution to the decision problem for a large set of monadic second-order theories of tree structures. We achieve it in two steps: first, we reduce the considered problem to the problem of determining, for any Rabin tree automaton, whether it accepts a given tree; then, we exploit a suitable notion of tree equivalence to reduce (a number of instances of) the latter problem to the decidable case of regular trees. We prove that such a reduction works for a large class of trees, that we call residually regular trees. We conclude the paper with a short discussion of related work.

1 Introduction

The automatic verification of properties of infinite state systems is a crucial problem in computer science, which turns out to be undecidable in many cases. A natural approach to this problem is to model a transition system as a directed graph, whose vertices (resp. edges) represent system configurations (resp. transitions). The expected behavior of the system is then expressed by a logical formula, which can be satisfied or not by the corresponding graph. The verification problem consists in deciding the satisfiability (resp. truth) of a given formula (resp. sentence) over a fixed graph structure. In this paper, we address the verification problem for systems of monadic second-order (MSO) logic interpreted over tree structures.

A fundamental result in the case of finite state systems is Büchi's theorem [2], that shows the decidability of the MSO theory of the linear order $(\mathbb{N}, <)$. Such a result takes advantage of closure properties of language acceptors (Büchi automata) with respect to union, intersection, complementation, and projection. Later, Rabin extended this result to the theory of the infinite (complete) binary tree by exploiting a new class of automata, called Rabin tree automata [24]. Büchi's theorem has also been used to deal with expansions of $(\mathbb{N}, <)$ with suitable unary predicates. Given a unary predicate $P \subseteq \mathbb{N}$, the decision problem for the theory of the expanded structure $(\mathbb{N}, <, P)$ is the problem of determining, for any Büchi automaton M, whether M accepts (the infinite word that characterizes) P. Elgot and Rabin gave a positive answer to this problem for some relevant predicates, such as the factorial one [15]. Recently, Carton and Thomas generalized such a result to the class of the so-called residually ultimately periodic words (which includes the class of morphic infinite words) [7]. In [23], Muller and Schupp brought the interest to MSO theories of graphs by identifying a large family of decidable graphs. Several approaches to the problem of deciding graph theories have been proposed in the literature. The transformational approach solves the problem for those graphs that are obtained by applying decidability-preserving transformations to structures which are known to be decidable, e.g., unfoldings [14], tree-graph operation [27], first-order interpretations and inverse rational mappings [10], MSO definable transductions [13]. Other approaches capture decidable graph structures through rewriting systems [8], transducers [4], or equational systems [11, 12]. It is worth pointing out that different characterizations of the same family of graphs have been obtained by following different approaches. As an example, prefix-recognizable graphs [3] can be equivalently described by means of rational restrictions of inverse rational mappings of the infinite complete binary tree [10], MSO interpretations of infinite regular trees [26], and vertex-replacement equational graphs [1].

In this paper we extend Carton and Thomas' automaton-based approach [7] to cope with the decision problem for a large set of MSO theories of tree structures. First, we reduce the considered problem to the problem of determining, for any Rabin tree automaton, whether it accepts a given tree. Then, we exploit a suitable notion of tree equivalence to reduce (a number of instances of) the latter problem to the decidable case of regular trees. We prove that such a reduction works for a large class of trees, that we call residually regular trees. We show the effectiveness of the proposed technique by exploiting it to decide the theories of some meaningful relational structures, including several trees in the Caucal hierarchy [9] and trees outside it [6]. We conclude the paper with a short discussion of related work.

2 Basic Notions

In the following, we provide some basic definitions about structures, logics, and automata. In the following, we briefly introduce the transformational approach of Caucal [9], which, to our knowledge, is up to now the most powerful tool for establishing the decidability of MSO theories of graph structures.

2.1 MSO Logics

MSO logics over graph structures are defined as follows. Given a finite alphabet Λ, a Λ-labeled graph structure is a tuple $\mathcal{G} = (S, (E_l)_{l \in \Lambda})$, where S (also denoted $\mathcal{D}om(\mathcal{G})$) is a countable set of vertices (states) and $(E_l)_{l \in \Lambda}$ are binary relations defining the edge labels. A graph is said to be *deterministic* if, for each relation E_l, $(u, v) \in E_l$ and $(u, w) \in E_l$ imply $v = w$.

MSO formulas are built up from atoms of the forms $x_i = x_j$, $X_k(x_j)$, and $E_l(x_i, x_j)$ by means of the Boolean connectives $\vee$ and $\neg$ and the existential quantification over first-order variables $x_i, x_j, \ldots$, interpreted as single vertices, and second-order ones $X_k, \ldots$, interpreted as sets of vertices. The semantics of an MSO formula is defined in the standard way [25]. For a given MSO formula

$\varphi(x_1, \ldots, x_n, X_1, \ldots, X_m)$, with free variables $x_1, \ldots, x_n, X_1, \ldots, X_m$, we write $\mathcal{G} \vDash \varphi[v_1, \ldots, v_n, V_1, \ldots, V_m]$ whenever φ holds in the structure $\mathcal{G}$ with the interpretation v_i for x_i, for $1 \leq i \leq n$, and V_j for X_j, for $1 \leq j \leq m$. In the following, we shall adopt a simplified, but expressively equivalent, set-up where all variables are second-order and atomic formulas are of the forms $X_i \subseteq X_j$ and $E_l(X_i, X_j)$.

The decision problem for a given structure $\mathcal{G}$ is the problem of establishing, for any MSO sentence φ, whether $\mathcal{G} \vDash \varphi$. We shall focus our attention on expanded graph structures $(\mathcal{G}, \bar{V})$, where $\bar{V} = (V_1, \ldots, V_m)$, with $V_j \subseteq \mathcal{D}om(\mathcal{G})$ for $1 \leq j \leq m$. The decision problem for expanded structures $(\mathcal{G}, \bar{V})$ is the problem of establishing, for any MSO formula $\varphi(X_1, \ldots, X_m)$, whether $\mathcal{G} \vDash \varphi[\bar{V}]$. The set of all sentences (resp. formulas) that hold in a structure $\mathcal{G}$ (resp. $(\mathcal{G}, \bar{V})$) is called the *MSO theory* of $\mathcal{G}$ (resp. $(\mathcal{G}, \bar{V})$), denoted by $MTh(\mathcal{G})$ (resp. $MTh(\mathcal{G}, \bar{V})$). $MTh(\mathcal{G})$ (resp. $MTh(\mathcal{G}, \bar{V})$) is said to be decidable iff there is an effective way to test whether any MSO sentence (resp. formula) ϕ belongs to $MTh(\mathcal{G})$ (resp. $MTh(\mathcal{G}, \bar{V})$).

As a matter of fact, any expanded structure $(\mathcal{G}, \bar{V})$ can be encoded into a *Σ-colored* graph $\mathcal{G}_{\bar{V}}$, with $|\Sigma| = 2^m$, called the *canonical representation* of $\bar{V}$. Each color $c \in \Sigma$ can be viewed as a subset of $\{1, \ldots, m\}$: for any vertex $v \in \mathcal{D}om(\mathcal{G})$, the color of v is the set of all and only the indexes i such that $v \in V_i$. Furthermore, we say that an n-ary relation R is *MSO definable in $\mathcal{G}$* if there is a formula $\varphi(x_1, \ldots, x_n)$ such that $(v_1, \ldots, v_n) \in R$ iff $\mathcal{G} \vDash \varphi[v_1, \ldots, v_n]$.

2.2 Trees

For any $k > 0$, let $[k]$ be the set $\{1, \ldots, k\}$. A *k-ary (Σ-colored) tree* is a $[k]$-labeled Σ-colored graph whose domain is a *prefix-closed* language over $[k]$, and whose edge relations are such that $(u, v) \in E_l$ iff $v = ul$, for every $l \in [k]$. Given a tree $\mathcal{T}$, we denote by $\mathcal{T}(v)$ the color of the vertex v. For convenience, we shall use terms to denote trees, for instance $b\langle w, w, w\rangle$ denotes the full ternary $\{w, b\}$-colored tree which consists of a b-colored root and 3 w-colored leaves. The *frontier* $\mathcal{F}r(\mathcal{T})$ of $\mathcal{T}$ is the set $\{u \in \mathcal{D}om(\mathcal{T}) : \forall\, l \in [k].\ ul \notin \mathcal{D}om(\mathcal{T})\}$. Note that $\mathcal{F}r(\mathcal{T})$ is a *prefix-free* language, namely, for every $u \in \mathcal{F}r(\mathcal{T})$ and for every prefix v of u, $v \in \mathcal{F}r\mathcal{T}$ iff $v = u$. In this paper, we mainly deal with *full* trees, namely, trees such that if $(u, ul) \in E_l$ for some $l \in [k]$, then $(u, ui) \in E_i$ for every $i \in [k]$. Though the standard notion of full tree includes both empty trees and singletons, it is convenient to exclude them.

A *path* of $\mathcal{T}$ is a (finite or infinite) word u such that every finite prefix of u belongs to $\mathcal{D}om(\mathcal{T})$. Given a path u of $\mathcal{T}$, we denote by $\mathcal{T}|u$ the sequence of colors associated with the vertices of u (formally, the finite or infinite sequence $\mathcal{T}(u_0)\mathcal{T}(u_1)\mathcal{T}(u_2)\ldots$, where u_i denotes the i-character prefix $u[1..i]$ of u). A *branch* is a maximal path, namely, a path which is not a proper prefix of any word in $\mathcal{D}om(\mathcal{T})$. We denote the set of all (finite or infinite) branches by $\mathcal{B}ch(\mathcal{T})$.

2.3 Tree Automata

A *k-ary Rabin tree automaton* over the alphabet Σ is a quadruple $M = (S, I, E, AP)$, where S is a finite set of states, $I \subseteq S$ is a set of initial states, $E \subseteq S \times \Sigma \times S^k$ is a transition relation, and AP is a finite set of accepting pairs (L_i, U_i), with $L_i, U_i \subseteq S$ [24].

Given an infinite complete k-ary (Σ-colored) tree $\mathcal{T}$, a *run* of the automaton M on $\mathcal{T}$ is any infinite complete k-ary (S-colored) tree $\mathcal{R}$ such that

$$(\mathcal{R}(u), \mathcal{T}(u), \mathcal{R}(u1), \ldots, \mathcal{R}(uk)) \in E \qquad \forall\, u \in \mathcal{D}om(\mathcal{R})..$$

We say that $\mathcal{R}$ is *successful*, and thus $\mathcal{T}$ is *accepted* by M, if $\mathcal{R}(\varepsilon) \in I$ and, for every branch u, there exists (L_i, U_i) such that $\mathcal{I}nf(\mathcal{R}|u) \cap L_i = \emptyset$ and $\mathcal{I}nf(\mathcal{R}|u) \cap U_i \neq \emptyset$, where $\mathcal{I}nf(\alpha)$ is the set of elements that occur infinitely often in α (we further denote by $\mathcal{I}mg(\alpha)$ the set of elements that occur in α). The language $\mathscr{L}(M)$ is the set of all trees accepted by M.

2.4 The Caucal Hierarchy

We first recall the basic notions of unfolding and MSO interpretation.

Let $\mathcal{G} = (S, (E_l)_{l \in \Lambda})$ be a graph structure and let $v_0 \in S$ be a designated vertex of $\mathcal{G}$. The *unfolding of* $\mathcal{G}$ *from* v_0, denoted by $Unf(\mathcal{G}, v_0)$, is a tree structure $(S', (E'_l)_{l \in \Lambda})$, where S' is the set of all the finite paths of the form $v_0 l_0 v_1 \ldots l_{n-1} v_n$, and E'_l is the set of all the pairs of paths of the form $(v_0 l_0 v_1 \ldots l_{n-1} v_n, v_0 l_0 v_1 \ldots l_{n-1} v_n l v_{n+1})$.

Notice that the unfolding of any finite graph is (isomorphic to) a regular tree. It is well known [14] that unfoldings from MSO definable vertices preserve the decidability of the MSO theories of graph structures.

Another transformation which preserves decidability is the MSO interpretation, which is defined as follows. Given a graph structure $\mathcal{G} = (S, (E_l)_{l \in \Lambda})$ and a finite set of labels Γ, an *MSO interpretation of* $\mathcal{G}$ *in* Γ is a family $(\varphi_l)_{l \in \Gamma}$ of MSO formulas over $\mathcal{G}$. It gives raise to a graph $\mathcal{G}' = (S', (E'_l)_{l \in \Gamma})$, where, for each $l \in \Sigma$, $E'_l = \{(v, w) \in S \times S : \mathcal{G} \vDash \varphi_l[v, w]\}$, and $S' \subseteq S$ is the set of all vertices occurring in the edge relations E'_l.

The level 0 of the Caucal hierarchy consists of all the finite graphs. At level $n+1$, we find all the graphs which are obtained from level n graphs by applying an unfolding followed by an MSO interpretation. Since both unfoldings and MSO interpretations preserve decidability, the resulting hierarchy contains only graphs with a decidable MSO theory.

3 An Automaton-based Approach to Decidability

In this section, we develop an automaton-based method to decide MSO theories of infinite complete trees. It can be viewed as a generalization of Carton and Thomas' method, which exploits noticeable properties of *residually ultimately periodic words* to decide MSO theories of labeled linear orderings [7].

As a first step, we show how to reduce the decision problem for the considered MSO theories to the acceptance problem for Rabin tree automata. Rabin's Theorem [24] establishes a strong correspondence between MSO formulas satisfied by an expanded tree structure $(\mathcal{T}, \bar{V})$ and Rabin tree automata accepting its canonical representation $\mathcal{T}_{\bar{V}}$: for every formula $\varphi(\bar{X})$, we can compute a Rabin tree automaton M (and, conversely, for every Rabin tree automaton M, we can compute a formula $\varphi(\bar{X})$) such that $\mathcal{T} \vDash \varphi[\bar{V}]$ iff $\mathcal{T}_{\bar{V}} \in \mathscr{L}(M)$. Let us denote by $Acc(\mathcal{T}_{\bar{V}})$ the problem of deciding whether a given Rabin tree automaton recognizes $\mathcal{T}_{\bar{V}}$. We have that

$$MTh(\mathcal{T}, \bar{V}) \text{ is decidable} \quad \text{iff} \quad Acc(\mathcal{T}_{\bar{V}}) \text{ is decidable.}$$

By exploiting the closure under intersection and the decidability of the emptiness problem for Rabin tree automata, one can easily show that the problem $Acc(\mathcal{T}_{\bar{V}})$ is decidable for any *regular* tree $\mathcal{T}_{\bar{V}}$ (a regular tree is a tree containing only finitely many distinct subtrees). In the following, we shall extend the class of trees for which this acceptance problem turns out to be decidable. We introduce the class of *residually regular trees* and we solve their acceptance problem by reducing them to equivalent regular trees (according to a suitable notion of tree equivalence).

Let us preliminarily define some tools for tree manipulation (for the sake of simplicity, hereafter we shall omit the subscript $\bar{V}$, thus writing $\mathcal{T}$ for $\mathcal{T}_{\bar{V}}$).

Definition 1. *Let $\mathcal{T}$ be a k-ary tree, $U \subseteq \mathcal{D}om(\mathcal{S})$ be a prefix-free language, and $(\mathcal{R}_u)_{u \in U}$ be a family of k-ary trees. We denote by $\mathcal{T}[u/\mathcal{R}_u]_{u \in U}$ the tree resulting from the simultaneous substitution in $\mathcal{T}$ of each sub-tree rooted at $u \in U$ by $\mathcal{R}_u$, formally, the tree $\mathcal{T}'$ defined by*

$$\mathcal{D}om(\mathcal{T}') = \mathcal{D}om(\mathcal{T}) \setminus (U \cdot [k]^*) \; \cup \; \bigcup_{u \in U} \{u\} \cdot \mathcal{D}om(\mathcal{R}_u),$$

$$\mathcal{T}'(v) = \begin{cases} \mathcal{T}(v) & \textit{no prefix of } v \textit{ belongs to } U, \\ \mathcal{R}_u(w) & v{=}u\ w \textit{ with } u \in U \textit{ and } w \in \mathcal{D}om(\mathcal{R}_u). \end{cases}$$

Definition 2. *For every pair of full k-ary (Σ-colored) trees $\mathcal{T}_1$ and $\mathcal{T}_2$ and every color $c \in \Sigma$, the concatenation $\mathcal{T}_1 \cdot_c \mathcal{T}_2$ is the tree resulting from the simultaneous substitution of all the c-colored leaves of $\mathcal{T}_1$ by $\mathcal{T}_2$, namely, the full k-ary (Σ-colored) tree $\mathcal{T}_1[u/\mathcal{T}_2]_{u \in U}$, where $U = \{u \in \mathcal{F}r(\mathcal{T}_1) \,:\, \mathcal{T}_1(u) = c\}$.*

Note that tree concatenation is not associative. As an example take the two $\{w, b, c\}$-colored trees $\mathcal{T}_1 = b\langle c, w\rangle$ and $\mathcal{T}_2 = w\langle w, w\rangle$; clearly, $(\mathcal{T}_1 \cdot_c \mathcal{T}_1) \cdot_w \mathcal{T}_2 \neq \mathcal{T}_1 \cdot_c (\mathcal{T}_1 \cdot_w \mathcal{T}_2)$. We assume that tree concatenation associates to the left.

Definition 2 can be generalized to the case of *infinite* concatenations. Given an infinite sequence $(c_n)_{n \in \mathbb{N}}$ of colors in Σ and an infinite sequence $(\mathcal{T}_n)_{n \in \mathbb{N}}$ of full k-ary (Σ-colored) trees, the infinite concatenation $\mathcal{S} = \mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots$ is defined as follows: $\mathcal{D}om(\mathcal{S}) = \bigcup_{n \in \mathbb{N}} \mathcal{D}om(\mathcal{S}_n)$, where $\mathcal{S}_0 = \mathcal{T}_0$, $\mathcal{S}_{n+1} = \mathcal{S}_n \cdot_{c_n} \mathcal{T}_{n+1}$, and $\mathcal{S}(u) = c$ if and only if $\mathcal{S}_n(u) = c$ for all, but finitely many, n. A *factorization* is a finite or infinite concatenation $\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots$ (we denote infinite concatenations by $\prod_{i \in \mathbb{N}} (\mathcal{T}_i)_{c_i}$). A factorization is *regular* if every $\mathcal{T}_n$ is a regular full tree and there are two positive integers p and q (called respectively *prefix* and *period*) such that, for every $n \geq p$, $c_n = c_{n+q}$ (if c_{n+q} exists) and $\mathcal{T}_n = \mathcal{T}_{n+q}$ (if $\mathcal{T}_{n+q}$ exists). The following proposition links regular factorizations to regular trees. From such a result, it immediately follows that $Acc(\mathcal{T})$ is decidable for any infinite (complete) tree $\mathcal{T}$ generated by a regular factorization.

Proposition 1. *A full tree $\mathcal{T}$ is regular iff it has a regular factorization.*

Proof. Note that finite concatenations of regular trees produce regular trees, thus we can restrict our attention to the case of infinite factorizations. Let $\prod_{n \in \mathbb{N}} (\mathcal{U}_n)_{c_n}$ be an infinite regular factorization of a full k-ary Σ-colored tree $\mathcal{T}$ and let p and q be a prefix and a period of such a factorization. For every $v \in \mathcal{D}om(\mathcal{T})$, the subtree $\mathcal{T}/v$ of $\mathcal{T}$ rooted at node v has the following form:

$$\mathcal{T}/v = \begin{cases} \emptyset & \text{if } v \notin \mathcal{D}om(\mathcal{T}), \\ \prod_{n \in \mathbb{N}} (\mathcal{R}'_n)_{c_n} & \text{if } v \in \mathcal{D}om(\mathcal{U}_0) \setminus \mathcal{F}r(\mathcal{U}_0), \\ & \quad \mathcal{U}'_0 = \mathcal{U}_0/v, \\ & \quad \mathcal{U}'_n = \mathcal{U}_n \text{ for every } n > 0, \\ \mathcal{T}_v/w & \text{if } v = uw \text{ with } u \in \mathcal{F}r(\mathcal{U}_0), \\ & \quad \mathcal{T}_v = \prod_{n > n_v} (\mathcal{U}_n)_{c_n}, \\ & \quad n_v \text{ is the first integer such that } \mathcal{U}_0(u) = c_{n_v}. \end{cases}$$

As regards the value of n_v in the above equation, we can claim that $0 \leq n_v \leq p+q$ (since $c_n = c_{n+q}$ for all $n \geq p$). From this it follows that there are only finitely many distinct trees of the form $\mathcal{T}_v = \prod_{n > n_v} (\mathcal{U}_n)_{c_n}$. Furthermore, since $\mathcal{U}_0$ is a full tree, w is a proper suffix of v. This can be used to verify, by induction on the length of v, that there are only finitely many distinct trees of the form $\mathcal{T}/v$.

For the converse implication, we start from the fact that any regular colored tree is isomorphic to the unfolding of a finite deterministic colored graph from a designated vertex [9]. For a k-ary Σ-colored tree $\mathcal{T}$, we can assume that the generating graph is of the form $(S, (E_i)_{i \in [k]}, (V_c)_{c \in \Sigma})$, where S and Σ are disjoints finite sets. For every vertex $v \in S$, we define the finite k-ary $(\Sigma \cup S)$-colored tree $\mathcal{U}_v$ such that

i) $\mathcal{D}om(\mathcal{U}_v) = \{\varepsilon\} \cup [k]$,

ii) $\mathcal{U}_v(\varepsilon) = c$ where c is the color of vertex v ($v \in V_c$),

iii) $\mathcal{U}_v(l) = w$, where w is the (unique) vertex such that $(v, w) \in E_l$.

Now, we fix an order for the vertexes of S, say $v_0, \ldots, v_{q-1}$, and we assume that designated vertex for the unfolding of the graph is v_0. We further define the sequence of colors $c_0 = v_0, \ldots, c_{q-1} = v_{q-1}$. It is easy to verify that $\prod_{n \in \mathbb{N}} (\mathcal{U}_{v_{n \bmod q}})_{c_{n \bmod q}}$ is a regular factorization that generates exactly the unfolding $\mathcal{T}$ of the graph. This proves that any regular full colored tree enjoys a regular factorization. □

The following lemma is almost trivial, but it will be extensively used in the next section.

Lemma 1. *For every pair of full trees $\mathcal{T}_1, \mathcal{T}_2$ and for every color $c \in \Sigma$, v is a branch of tree $\mathcal{T}_1 \cdot_c \mathcal{T}_2$ iff one of the following conditions holds:*
1. *v is a branch of $\mathcal{T}_1$ and $\mathcal{T}_1(v)$ (if defined) differs from c,*
2. *there are a branch u of $\mathcal{T}_1$ and a branch w of $\mathcal{T}_2$ such that $v = uw$ and $\mathcal{T}_1(u) = c$.*

3.1 Residually Regular Factorizations and Trees

We now show how to reduce the acceptance problem for a large class of infinite (complete) trees to the acceptance problem for equivalent (according to a suitable notion of tree equivalence $\equiv_M$) regular trees. As a preliminary step, we introduce the notion of (finite or infinite) computation of a Rabin tree automaton $M = (S, I, E, AP)$: a *computation* of M on a full, finite or infinite, (Σ-colored) tree $\mathcal{T}$ is a full (S-colored) tree $\mathcal{C}$ such that (i) $\mathcal{D}om(\mathcal{C}) = \mathcal{D}om(\mathcal{T})$ and (ii) $(\mathcal{C}(v), \mathcal{T}(v), \mathcal{C}(v1), \ldots, \mathcal{C}(vk)) \in E$, for every $v \in \mathcal{D}om(\mathcal{C}) \setminus \mathcal{F}r(\mathcal{C})$.

Definition 3. *Given a Rabin tree automaton $M = (S, I, E, AP)$ over Σ, and two full (Σ-colored) trees $\mathcal{T}_1$ and $\mathcal{T}_2$, $\mathcal{T}_1 \equiv_M \mathcal{T}_2$ holds iff, for every computation $\mathcal{C}_1$ of M on $\mathcal{T}_1$, there exists a computation $\mathcal{C}_2$ of M on $\mathcal{T}_2$ (and vice versa) such that*
1. *$\mathcal{T}_1(\varepsilon) = \mathcal{T}_2(\varepsilon)$ and $\mathcal{C}_1(\varepsilon) = \mathcal{C}_2(\varepsilon)$;*
2. *for every $v \in \mathcal{F}r(\mathcal{T}_1)$, there exists $u \in \mathcal{F}r(\mathcal{T}_2)$ such that $\mathcal{T}_1(v) = \mathcal{T}_2(u)$, $\mathcal{C}_1(v) = \mathcal{C}_2(u)$, and $\mathcal{I}mg(\mathcal{C}_1|v)) = \mathcal{I}mg(\mathcal{C}_2|u)$, and vice versa;*
3. *for any (infinite branch) $v \in \mathcal{B}ch(\mathcal{T}_1)$, there exists (an infinite branch) $u \in \mathcal{B}ch(\mathcal{T}_2)$ such that $\mathcal{I}nf(\mathcal{C}_1|v) = \mathcal{I}nf(\mathcal{C}_2|u)$, and vice versa.*

The equivalence $\equiv_M$ satisfies the following properties:
1. $\equiv_M$ has *finite index*;
2. for every pair of factorizations $\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots$ and $\mathcal{T}'_0 \cdot_{c_0} \mathcal{T}'_1 \cdot_{c_1} \ldots$ such that $\mathcal{T}_i \equiv_M \mathcal{T}'_i$ for every i, we have $\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots \equiv_M \mathcal{T}'_0 \cdot_{c_0} \mathcal{T}'_1 \cdot_{c_1} \ldots$;
3. for every pair of $\equiv_M$-equivalent infinite (complete) trees $\mathcal{T}_1, \mathcal{T}_2$, we have that $\mathcal{T}_1 \in \mathscr{L}(M)$ iff $\mathcal{T}_2 \in \mathscr{L}(M)$ (in such a case we say that $\mathcal{T}_1$ and $\mathcal{T}_2$ are *indistinguishable* by automaton M).

It is worth pointing out that the proposed notion of equivalence can be easily tailored to different kinds of automata, such as, for instance, Muller tree automata and parity tree automata.

Proposition 2. *For every Rabin tree automaton M, $\equiv_M$ has finite index.*

Proof. Let $M = (S, I, E, AP)$ be a tree automaton over the alphabet Σ and let $T^{ful}_{k,\Sigma}$ be the language of all the full k-ary Σ-colored trees. Further denote by $\mathcal{E} : T^{ful}_{k,\Sigma} \rightarrow \Sigma \times \mathscr{P}(S \times \mathscr{P}(\mathscr{P}(S)) \times \mathscr{P}(\Sigma \times S \times \mathscr{P}(S)))$ the function defined, for every tree $\mathcal{T} \in T^{ful}_{k,\Sigma}$, as

$$\mathcal{E}(\mathcal{T}) = \Big(\mathcal{T}(\varepsilon), \Big\{\Big(\mathcal{C}(\varepsilon), \{\mathcal{I}nf(\mathcal{C}|v) \,:\, v \in \mathcal{B}ch(\mathcal{T})\}, \{\Phi_{\mathcal{T},\mathcal{C}}(v) \,:\, v \in \mathcal{F}r(\mathcal{T})\}\Big) \;: \\ \mathcal{C} \text{ is a computation of } M \text{ on } \mathcal{T}\Big\}\Big)$$

where $\Phi_{\mathcal{T},\mathcal{C}} : \mathcal{F}r(\mathcal{T}) \to \Sigma \times S \times \mathscr{P}(S)$ is the function that maps a finite branch u of $\mathcal{T}$ to the tuple $(\mathcal{T}(u), \mathcal{C}(u), \mathcal{I}mg(\mathcal{C}|u))$. Clearly, whenever $\mathcal{E}(\mathcal{T}_1) = \mathcal{E}(\mathcal{T}_2)$, $\mathcal{T}_1 \equiv_M \mathcal{T}_2$ follows. From the finiteness of the image set of the function $\mathcal{E}$, we conclude that $\equiv_M$ has finite index. $\square$

Proposition 3. *For every Rabin tree automaton M, $\equiv_M$ respects finite and infinite factorizations, namely, for every pair of factorizations $\mathcal{U}_0 \cdot_{c_0} \mathcal{U}_1 \cdot_{c_1} \ldots$ and $\mathcal{U}'_0 \cdot_{c_0} \mathcal{U}'_1 \cdot_{c_1} \ldots$ such that $\mathcal{U}_i \equiv_M \mathcal{U}'_i$ for every i, we have*

$$\mathcal{U}_0 \cdot_{c_0} \mathcal{U}_1 \cdot_{c_1} \ldots \equiv_M \mathcal{U}'_0 \cdot_{c_0} \mathcal{U}'_1 \cdot_{c_1} \ldots$$

Proof. Let $\mathcal{T} = (\mathcal{U}_0 \cdot_{c_0} \mathcal{U}_1 \cdot_{c_1} \ldots)$, $\mathcal{T}' = (\mathcal{U}'_0 \cdot_{c_0} \mathcal{U}'_1 \cdot_{c_1} \ldots)$, and $\mathcal{C}$ be a computation of the automaton $M = (S, I, E, AP)$ on $\mathcal{T}$. Clearly, both $\mathcal{T}$ and $\mathcal{T}'$ are full trees. We want to prove that there is a computation $\mathcal{C}'$ on $\mathcal{T}'$ that satisfies the three conditions of Definition 3. The inverse implication will follow similarly. Consider the sets of nodes of $\mathcal{T}$ and $\mathcal{T}'$ in which substitutions occur. Precisely, let $I_1, I_2, \ldots$ and $J_1, J_2, \ldots$ be two (finite or infinite) sequences of disjoint prefix-free languages such that $\mathcal{T} = \mathcal{U}_0[u/\mathcal{U}_1]_{u \in I_1}[u/\mathcal{U}_2]_{u \in I_2} \ldots$ and $\mathcal{T}' = \mathcal{U}'_0[v/\mathcal{U}'_1]_{v \in J_1}[v/\mathcal{U}'_2]_{v' \in J_2} \ldots$ hold. Further let $I_0 = J_0 = \{\varepsilon\}$. Now, for every word $v \in J_n$, we define, by induction on n,

i) a finite word $u_v \in I_n$ such that $\mathcal{T}(u_v) = \mathcal{T}'(v)$,

ii) a computation $\mathcal{C}_{u_v}$ on $\mathcal{U}_n$, which is obtained by taking the subtree $\mathcal{C}/u_v$ of $\mathcal{C}$ rooted at u_v and by restricting the domain to $\mathcal{D}om(\mathcal{U}_n)$,

iii) a computation $\mathcal{C}'_v$ on $\mathcal{U}'_n$ such that
- $\mathcal{C}_{u_v}(\varepsilon) = \mathcal{C}'_v(\varepsilon)$,
- $\{\mathcal{I}nf(\mathcal{C}_{u_v}|x) \ : \ x \in \mathcal{B}ch(\mathcal{U}_n)\} = \{\mathcal{I}nf(\mathcal{C}'_v|y) \ : \ y \in \mathcal{B}ch(\mathcal{U}'_n)\}$,
- $\{\Phi_{\mathcal{U}_n,\mathcal{C}_{u_v}}(x) \ : \ x \in \mathcal{F}r(\mathcal{U}_n)\} = \{\Phi_{\mathcal{U}'_n,\mathcal{C}'_v}(y) \ : \ y \in \mathcal{F}r(\mathcal{U}'_n)\}$

where $\Phi_{\mathcal{T},\mathcal{C}} : \mathcal{F}r(\mathcal{T}) \to \Sigma \times S \times \mathscr{P}(S)$ is the function that maps a finite branch u of $\mathcal{T}$ to the tuple $(\mathcal{T}(u), \mathcal{C}(u), \mathcal{I}mg(\mathcal{C}|u))$.

For $n = 0$, $v \in J_0$ iff $v = \varepsilon$. Thus, we define $u_\varepsilon = \varepsilon$ and $\mathcal{C}_\varepsilon = \mathcal{C}|\mathcal{D}om(\mathcal{U}_0)$. Since $\mathcal{U}_0 \equiv_M \mathcal{U}'_0$, we know that there is a computation, say $\mathcal{C}'_\varepsilon$, on $\mathcal{U}'_0$ satisfying the above properties. Now, for the induction step, assume that $v \in J_n$. Clearly, v is a branch of the tree $\mathcal{U}'_0 \cdot_{c_0} \mathcal{U}'_1 \cdot_{c_1} \ldots \mathcal{U}'_{n-1}$. By Lemma 1, there exist an index $i \leq n-1$, a word $x \in J_i$, and a finite word $y \in \mathcal{B}ch(\mathcal{U}'_i)$ such that $v = xy$. From the inductive hypothesis, we know that there exist a word $u_x \in I_i$, a computation $\mathcal{C}_{u_x} = (\mathcal{C}/u_x)|\mathcal{D}om(\mathcal{U}_i)$ on $\mathcal{U}_i$, and a computation $\mathcal{C}'_x$ on $\mathcal{U}'_i$ satisfying the above properties. Moreover, since $\mathcal{U}_i \equiv_M \mathcal{U}'_i$, there is a finite word $z \in \mathcal{F}r(\mathcal{U}_i)$ such that $\mathcal{U}_i(z) = \mathcal{U}'_i(y)$, $\mathcal{C}_{u_x}(z) = \mathcal{C}'_x(y)$, and $\mathcal{I}mg(\mathcal{C}_{u_x}|z) = \mathcal{I}mg(\mathcal{C}'_x|y)$. Thus, we can define $u_v = u_x z$ and $\mathcal{C}_{u_v} = (\mathcal{C}/u_v)|\mathcal{D}om(\mathcal{U}_n)$ and we can easily verify that $u_v \in I_n$ (since $\mathcal{U}_i(z) = \mathcal{U}'_i(y) = c_{n-1}$), $\mathcal{T}(u_v) = \mathcal{U}_n(\varepsilon) = \mathcal{U}'_n(\varepsilon) = \mathcal{T}'(v)$ (since $\mathcal{U}_n \equiv_M \mathcal{U}'_n$), and $\mathcal{C}_{u_v}$ is a computation on $\mathcal{U}_n$. Finally, since $\mathcal{U}_n \equiv_M \mathcal{U}'_n$, we know that there is a computation $\mathcal{C}'_v$ on $\mathcal{U}'_n$ satisfying the above properties.

Such a result allows us to define $\mathcal{C}'$ as $\mathcal{C}'_\varepsilon[v/\mathcal{C}'_v]_{v \in J_1}[v/\mathcal{C}'_v]_{v \in J_2} \ldots$. We prove that $\mathcal{C}'$ is a computation of M on $\mathcal{T}'$, namely, we prove that $\mathcal{C}'$ respects the transitions of the automaton both on the internal nodes and on the frontier nodes of its factors. For every $v \in J_n$ and for every $w \in \mathcal{D}om(\mathcal{U}'_n) \setminus \mathcal{F}r(\mathcal{U}'_n)$, $(\mathcal{C}'_v(w), \mathcal{U}'_n(w), \mathcal{C}'_v(w1), \ldots, \mathcal{C}'_v(wk)) \in E$ and hence $(\mathcal{C}'(vw), \mathcal{T}'(vw), \mathcal{C}'(vw1), \ldots,$

$\mathcal{C}'(vwk)) \in E$ (since $\mathcal{C}'_v(y) = \mathcal{C}'_{vy}(\varepsilon)$ for any $y \in \mathcal{F}r(\mathcal{U}'_n)$). Moreover, for every $v \in J_n$ and for every $w \in \mathcal{F}r(\mathcal{U}'_n)$, we have $(\mathcal{C}'_{vw}(\varepsilon), \mathcal{U}'_n(w), \mathcal{C}'_{vw}(1), \ldots, \mathcal{C}'_{vw}(k)) \in E$, and hence $(\mathcal{C}'(vw), \mathcal{T}(vw), \mathcal{C}'(vw1), \ldots, \mathcal{C}'(vwk)) \in E$ (since $\mathcal{U}'_n(w) = \mathcal{U}_{n'}(\varepsilon)$ with n' satisfying $vw \in J_{n'}$). This proves that $\mathcal{C}'$ is a computation on $\mathcal{T}'$.

It remains to show that $\mathcal{C}'$ satisfies the three conditions of Definition 3. Clearly, we have $(\mathcal{T}(\varepsilon), \mathcal{C}(\varepsilon)) = (\mathcal{T}'(\varepsilon), \mathcal{C}'(\varepsilon))$ since $(\mathcal{U}_0(\varepsilon), \mathcal{C}_\varepsilon(\varepsilon)) = (\mathcal{U}'_0(\varepsilon), \mathcal{C}'_\varepsilon(\varepsilon))$. This accounts for the first condition.

Let u be an infinite branch of $\mathcal{T}$. We have to find an infinite branch v of $\mathcal{T}'$ such that $\mathcal{I}nf(\mathcal{C}|u) = \mathcal{I}nf(\mathcal{C}'|v)$. We know that the branch u can be broken into a (finite or infinite) sequence of factors $x_{i_1}, x_{i_2}, \ldots$, where $0 = i_1 < i_2 < \ldots$ and each x_{i_j} is a branch of $\mathcal{U}_{i_j}$. Let us denote by u_j the prefix $x_{i_1} x_{i_2} \ldots x_{i_{j-1}}$ of u. Clearly, $u_j \in I_{i_j}$ and hence we can denote by $\mathcal{C}_{u_j}$ the computation $(\mathcal{C}/u_j)|\mathcal{D}om(\mathcal{U}_{i_j})$ on $\mathcal{U}_{i_j}$. Since $\mathcal{U}_{i_j} \equiv_M \mathcal{U}'_{i_j}$, we can further define two sequences $y_{i_1}, y_{i_2}, \ldots$ and $v_{i_1}, v_{i_2}, \ldots$, where y_{i_j} is a branch of $\mathcal{U}'_{i_j}$, $v_{i_j} = y_{i_1} y_{i_2} \ldots y_{i_{j-1}} \in J_{i_j}$, and $\mathcal{C}'_{v_{i_j}} = (\mathcal{C}'/v_{i_j})|\mathcal{D}om(\mathcal{U}'_{i_j})$ is a computation on $\mathcal{U}'_{i_j}$ such that

i) $\mathcal{T}(u_{i_j}) = \mathcal{U}_{i_j}(\varepsilon) = \mathcal{U}'_{i_j}(\varepsilon) = \mathcal{T}'(v_{i_j})$,

ii) $\mathcal{C}(u_{i_j}) = \mathcal{C}_{u_{i_j}}(\varepsilon) = \mathcal{C}'_{v_{i_j}}(\varepsilon) = \mathcal{C}'(v_{i_j})$,

iii) $\mathcal{I}nf(\mathcal{C}_{u_{i_j}}|x_{i_j}) = \mathcal{I}nf(\mathcal{C}'_{v_{i_j}}|y_{i_j})$,

iv) $\mathcal{U}_{i_j}(x_{i_j}) = \mathcal{U}'_{i_j}(y_{i_j})$,

v) $\mathcal{C}_{u_{i_j}}(x_{i_j}) = \mathcal{C}'_{v_{i_j}}(y_{i_j})$,

vi) $\mathcal{I}mg(\mathcal{C}_{u_{i_j}}|x_{i_j}) = \mathcal{I}mg(\mathcal{C}'_{v_{i_j}}|y_{i_j})$.

We define $v = x_{i_1} x_{i_2} \ldots$. It easy to see that v is a branch of $\mathcal{T}'$. We further know that a state s belongs to $\mathcal{I}nf(\mathcal{C}|u)$ iff either it occurs infinitely often in a sequence $\mathcal{C}_{u_j}|x_{i_j}$ or it occurs in infinitely many sequences $\mathcal{C}_{u_j}|x_{i_j}$. We formalize such a property by writing $s \in \mathcal{I}nf(\mathcal{C}|u)$ iff $\exists\, j.\ s \in \mathcal{I}nf(\mathcal{C}_{u_j}|x_{i_j})$ or $\exists^\omega\, j.\ s \in \mathcal{I}mg(\mathcal{C}_{u_j}|x_{i_j})$. From the above properties, $s \in \mathcal{I}nf(\mathcal{C}|u)$ iff $\exists\, j.\ s \in \mathcal{I}nf(\mathcal{C}'_{v_j}|y_{i_j})$ or $\exists^\omega\, j.\ s \in \mathcal{I}mg(\mathcal{C}'_{v_j}|y_{i_j})$, iff $s \in \mathcal{I}nf(\mathcal{C}'|v)$. This proves that $\mathcal{I}nf(\mathcal{C}|\mathcal{B}ch(\mathcal{T})) \subseteq \mathcal{I}nf(\mathcal{C}'|\mathcal{B}ch(\mathcal{T}'))$. The inverse inclusion can be similarly proved. This accounts for the second condition of Definition 3.

Finally, we prove that, given a finite word $u \in \mathcal{F}r(\mathcal{T})$, there is a finite word $v \in \mathcal{F}r(\mathcal{T}')$, such that $\mathcal{T}(u) = \mathcal{T}'(v)$, $\mathcal{C}(u) = \mathcal{C}'(v)$, and $\mathcal{I}mg(\mathcal{C}|u) = \mathcal{I}mg(\mathcal{C}'|v)$. The converse implication will follow similarly. So, let u be a finite branch of $\mathcal{T}$. As usual, we brake u into a finite sequence of finite words $x_{i_1}, x_{i_2}, \ldots$, and we define the corresponding word $v = y_{i_1} y_{i_2} \ldots \in \mathcal{F}r(\mathcal{T}')$. It easy to verify that $\mathcal{I}mg(\mathcal{C}|u) = \bigcup_j \mathcal{I}mg(\mathcal{C}_{u_{i_j}}|x_{i_j}) = \bigcup_j \mathcal{I}mg(\mathcal{C}'_{v_{i_j}}|y_{i_j}) = \mathcal{I}mg(\mathcal{C}'|v)$. This accounts for the third condition of Definition 3. □

Proposition 4. *For every Rabin tree automaton M, and for every pair of infinite (complete) $\equiv_M$-equivalent trees $\mathcal{T}$ and $\mathcal{T}'$,*

$$\mathcal{T} \in \mathscr{L}(M) \quad \textit{iff} \quad \mathcal{T}' \in \mathscr{L}(M).$$

Proof. Let $M = (S, I, E, AP)$ be a Rabin tree automaton and let $\mathcal{T}, \mathcal{T}'$ be two infinite complete trees such that $\mathcal{T} \equiv_M \mathcal{T}'$. By definition, $\mathcal{T} \in \mathscr{L}(M)$ iff there is a run $\mathcal{R}$ of M on $\mathcal{T}$ such that $\mathcal{R}(\varepsilon) \in I$ and, for every branch u of $\mathcal{T}$, there exists an accepting pair (L_i, U_i) that satisfies both $\mathcal{I}nf(\mathcal{R}|u) \cap L_i = \emptyset$ and $\mathcal{I}nf(\mathcal{R}|u) \cap U_i \neq \emptyset$. From the third condition of Definition 3 (since every run is also a computation), it trivially follows that $\mathcal{T} \in \mathscr{L}(M)$ iff $\mathcal{T}' \in \mathscr{L}(M)$. $\square$

3.2 Exploiting Indistinguishability

Taking advantage of previous results, we identify a large class of trees, that we call residually regular trees, whose acceptance problem is decidable. Intuitively, we say that an infinite sequence $\mathcal{S}$ of finite full trees is 1-residually regular if, for every Rabin tree automaton M, $\mathcal{S}$ is $\equiv_M$-equivalent to an ultimately periodic sequence of finite trees (which can be effectively constructed). We call 1-*residually regular* trees those trees that are obtained by concatenating the trees in a 1-residually regular sequence. The notion of 1-residually regular can be extended to level n, with n being any ordinal, by no longer considering finite trees but level $n' < n$ residually regular trees. For every ordinal n, n-residually regular trees can be defined as follows, where we denote by $[i]_{p,q}$ either i or $p + ((i - p) \bmod q)$, depending on whether $i < p$ or not.

Definition 4. *Given an ordinal n, a factorization $\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \dots$ is an n-*residually regular factorization *iff the following two conditions hold:*

1. *for every i, either $\mathcal{T}_i$ is a finite full tree or we can provide an n'-residually regular factorization of $\mathcal{T}_i$, with $n' < n$;*
2. *for any Rabin tree automaton M, there exist two positive integers p and q (called* prefix *and* period *of the factorization with respect to $\equiv_M$) such that $c_i = c_{[i]_{p,q}}$ and $\mathcal{T}_i \equiv_M \mathcal{T}_{[i]_{p,q}}$, for every i.*

*An n-*residually regular tree *is a tree enjoying an n-residually regular factorization. A* residually regular factorization *is an n-residually regular factorization, for some ordinal n. A* residually regular tree *is a tree enjoying a residually regular factorization.*

It is worth noticing that the above definition allows residually regular factorizations to encompass residually regular factors of any arbitrary level. For instance, we can start with some factors $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \dots$ which respectively are level $1, 2, 3, \dots$ residually regular, and concatenate them to build an ω-residually regular tree; then, we can concatenate ω-residually regular trees to obtain an $(\omega + 1)$-residually regular tree, and so on.

In order to reduce the decision problem for (n-)residually regular trees to regular trees, we introduce the notion of $\equiv_M$-regular form. Such a notion is defined by transfinite induction on n, given a Rabin tree automaton M and an n-residually regular factorization. Precisely, an $\equiv_M$-regular form of a 1-residually regular factorization $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$ is a tree $\prod_{i\in\mathbb{N}}(\mathcal{T}'_i)_{c_i}$, where $\mathcal{T}'_i = \mathcal{T}_{[i]_{p,q}}$ and p and q are respectively a prefix and a period of the factorization with respect to $\equiv_M$. For any ordinal $n > 1$, an $\equiv_M$-regular form of an n-residually regular factorization $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$ is a tree $\prod_{i\in\mathbb{N}}(\mathcal{T}'_i)_{c_i}$, where, depending on whether $\mathcal{T}_i$ is finite or not, $\mathcal{T}'_i$ is either $\mathcal{T}_{[i]_{p,q}}$ or an $\equiv_M$-regular form of an n'-residually regular factorization of $\mathcal{T}_{[i]_{p,q}}$, with $n' < n$ and p and q being respectively a prefix and a period of the factorization with respect to $\equiv_M$. It is easy to verify that an $\equiv_M$-regular form of a residually regular factorization $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$ is a *regular* tree which is $\equiv_M$-equivalent to the tree generated by $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$, which leads to the following theorem [21].

Theorem 1. *Let $\mathcal{T}$ be the infinite (complete) tree resulting from an n-residually regular factorization $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$, M be a Rabin tree automaton, and $\mathcal{T}'$ be an $\equiv_M$-regular form of $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$. We have that*

$$\mathcal{T} \in \mathscr{L}(M) \quad \text{iff} \quad \mathcal{T}' \in \mathscr{L}(M) \ \text{(which is decidable).}$$

The upshot of such a result is that residually regular trees enjoy a decidable acceptance problem and hence a decidable MSO theory.

3.3 Structural Properties of Residually Regular Factorizations

We now identify some structural properties that allow us to easily build residually regular factorizations. The resulting framework somehow generalizes previous results by Zhang [28] and Carton and Thomas [7]. Let $T^{full}_{k,\Sigma}$ denote the language of all full k-ary Σ-colored trees. For any Rabin tree automaton M, the equivalence $\equiv_M$ induces an homomorphism from the infinite groupoid (i.e., a set endowed with a binary operation) $(T^{full}_{k,\Sigma}, \cdot_c)$ to the finite groupoid $([T^{full}_{k,\Sigma}]_{/\equiv_M}, \cdot_c)$. Given a groupoid $(G, \cdot)$, an element $g \in G$, and a number $n \in \mathbb{N}$, we denote by g^{n+1} the n-fold iteration of $\cdot g$ applied to g, namely, $g^{n+1} = g \cdot g \cdot \ldots g$. We define ultimately periodic functions with respect to finite groupoids.

Definition 5. *A function $f : \mathbb{N} \to \mathbb{N}$ is* ultimately periodic with respect to finite groupoids *(*residually periodic *for short) if, for every finite groupoid $(G, \cdot)$ and every $g \in G$, there exist $p \geq 0$ and $q > 0$ such that $g^{f(n)+1} = g^{f([n]_{p,q})+1}$, that is, $(g^{f(n)+1})_{n\in\mathbb{N}}$ is an ultimately periodic sequence.*

We say that a function is *effectively* residually periodic iff, for every groupoid $(G, \cdot)$ and every $g \in G$, it is possible to compute a prefix p and a period q of the ultimately periodic sequence $(g^{f(n)+1})_{n\in\mathbb{N}}$. As a simple example, the identity function is effectively residually periodic. From now on, we restrict our attention to effectively residually periodic functions. The following proposition provides a characterization of (effectively) residually periodic functions. [1]

Proposition 5. *A function $f : \mathbb{N} \to \mathbb{N}$ is (effectively) residually periodic iff for all $l \geq 0$ and $r > 0$, one can compute $p \geq 0$ and $q > 0$ such that*

$$[f(n)]_{l,r} = [f([n]_{p,q})]_{l,r} \,.$$

[1] it is worth mentioning that such a result implies that our notion of ultimately periodic functions with respect to finite groupoids is equivalent to the notion of residually ultimately periodic sequence, which has been introduced in [7]; indeed, an equivalent characterization in terms of congruences with ultimately periodic sequences holds for all the sequences of the form $(f(n))_{n\in\mathbb{N}}$, with f being a residually periodic function.

Proof. Suppose that $f : \mathbb{N} \to \mathbb{N}$ is a periodic function. Let $l \geq 0$ and $r > 0$. We can chose a finite group $(G, \cdot)$ and an element $g \in G$ such that r turns out to be the least integer such that g^r is the identity. Note that, by the definition of g, for every pair of integers y and z, $g^y = g^z$ implies $y = z \pmod r$, and hence $[y]_{l,r} = [z]_{l,r}$. Since f is periodic and G is also a groupoid, there are $p \geq 0$ and $q > 0$ such that $g^{f(n)+1} = g^{f([n]_{p,q})+1}$ holds for every $n \in \mathbb{N}$. Hence, it follows that, for every $n \in \mathbb{N}$, $[f(n)]_{l,r} = [f([n]_{p,q})]_{l,r}$.

For the converse implication, suppose that, given $l \geq 0$ and $r > 0$, there are $p \geq 0$ and $q > 0$ such that, for every $n \in \mathbb{N}$, $[f(n)]_{l,r} = [f([n]_{p,q})]_{l,r}$ holds. Given a finite groupoid $(G, \cdot)$ and an element $g \in G$, we know, from the Pigeonhole Principle, that there exist two positive integers $l \geq 0$ and $r > 0$ such that, for every $x \in \mathbb{N}$, $g^{x+1} = g^{[x]_{l,r}+1}$. Thus, from the hypothesis on f, we have $g^{f(n)+1} = g^{[f(n)]_{l,r}+1} = g^{[f([n]_{p,q})]_{l,r}+1} = g^{f([n]_{p,q})+1}$. □

Definition 6. *A function $f : \mathbb{N} \to \mathbb{N}$* has unbounded infima *if it holds that lim $inf_{n \to \infty} f(n) = \infty$. In this case, we assume that, for any k, we can compute n_0 such that $f(n) \geq k$ for all $n \geq n_0$.*

The following theorem provides a number of ways to build residually periodic functions [21]. Examples of generated functions are n^2, 2^n, $2^n - n^2$, n^n, $n!$, and the exponential tower $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$.

Theorem 2. *Let f and g be residually periodic functions. The following functions are residually periodic as well:*

1. **(Sum)** $f + g$;
2. **(Difference)** $f - g$, *provided that it has unbounded infima;*
3. **(Product)** $f * g$;
4. **(Quotient)** h *defined by* $h(n) = \lfloor \frac{f(n)}{d} \rfloor$, *with* $d > 0$;
5. **(Exponentiation)** f^g, *provided that it has unbounded infima;*
6. **(Exponential tower)** h *defined by* $h(0) = 1$ *and* $h(n+1) = b^{h(n)}$, *with* $b > 0$;
7. **(Fibonacci numbers)** h *defined by* $h(0) = h(1) = 1$ *and* $h(n+2) = h(n) + h(n+1)$;
8. **(Generalized sum)** h *defined by* $h(n) = \sum_{i=0}^{n-1} f(i)$;
9. **(Generalized product)** h *defined by* $h(n) = \prod_{i=0}^{n-1} f(i)$;
10. **(Substitution)** $g \circ f$.

Proof. For cases *1.* and *3.* it suffices to note that $[\ ]_{l,r}$ respects sums and products, in the sense that $[x+y]_{l,r} = [[x]_{l,r} + [y]_{l,r}]_{l,r}$ and $[x*y]_{l,r} = [[x]_{l,r} * [y]_{l,r}]_{l,r}$ hold for every pair of integers x and y. Similarly, one can show that $[\ ]_{l,r}$ respects differences, namely we have $[x-y]_{l,r} = [[x]_{l,r} - [y]_{l,r}]_{l,r}$, provided that $x - y \geq l$. Thus case *2.* follows by assuming that $f - g$ has unbounded infima.

Case *4.* is proved by noticing that, for every $l \geq 0$ and $r > 0$, $[h(n)]_{l,r}$ is either 0 or $[[h(n-d)]_{l,r} + 1]_{l,r}$ depending on whether $n < d$ or $n \geq d$. By defining

$(A_1(n), \ldots, A_d(n)) = ([h(n)]_{l,r}, \ldots, [h(n+d-1)]_{l,r})$, we obtain

$$(A_1(n), \ldots, A_d(n)) = \begin{cases} (0, \ldots, 0) & \text{if } n = 0, \\ (A_2(n-1), \ldots, A_d(n-1), [A_1(n-1)+1]_{l,r}) & \text{if } n > 0. \end{cases}$$

Since each $A_i(n)$ ranges over the finite domain $\{0, \ldots, l+r-1\}$, we can apply the Pigeonhole Principle and claim that there are two (computable) integers $p \geq 0$ and $q > 0$ such that $A_i(n) = A_i(n+q)$ for every $n \geq p$ and for every $i \in \{1, \ldots, d\}$. This proves that $[h(n)]_{l,r} = [h([n]_{p,q})]_{l,r}$.

As regards case *5.*, we preliminarily recall the definition of the Euler Totient Function

$$\phi(n) = n \prod_{p \text{ prime dividing } n} \left(1 - \frac{1}{p}\right).$$

and the following two properties:

$$x^y = x^y + \sum_{i=1}^{y} \left(\binom{y}{i} x^{y-i} * 0\right) = x^y + \sum_{i=1}^{y} \left(\binom{y}{i} x^{y-i} * m^i\right) = (x+m)^y \pmod{m},$$
$$x^y = x^y * 1 = x^y * x^{\phi(m)} = x^{y+\phi(m)} \pmod{m}.$$

Since f and g are periodic, there are $p, p' \geq 0$ and $q, q' \geq 0$ such that $[f(n)]_{0,m} = [f([n]_{p,q})]_{0,m}$ and $[g(n)]_{0,\phi(m)} = [g([n]_{p',q'})]_{0,\phi(m)}$. Now, by letting $r = max(p, p')$ and $s = lcm(q, q')$, we can write

$$\begin{aligned} \left[f(n)^{g(n)}\right]_{0,m} &= \left[\left([f(n)]_{0,m}\right)^{[g(n)]_{0,\phi(m)}}\right]_{0,m} \\ &= \left[\left([f([n]_{r,s})]_{0,m}\right)^{[g([n]_{r,s})]_{0,\phi(m)}}\right]_{0,m} \\ &= \left[f\left([n]_{r,s}\right)^{g([n]_{r,s})}\right]_{0,m}. \end{aligned}$$

We can generalize the above result. Let $\sigma(l)$ be the first $n \geq r$ such that $f(n)^{g(n)} \geq l$ (such a value exists and it is computable by hypothesis). Thus, for every $n \geq \sigma(l)$, we have

$$\begin{aligned} \left[f(n)^{g(n)}\right]_{l,m} &= \left[f(n)^{g(n)} - l\right]_{0,m} + l \\ &= \left[\left[f(n)^{g(n)}\right]_{0,m} - l\right]_{0,m} + l \\ &= \left[\left[f\left([n]_{\sigma(l),s}\right)^{g([n]_{\sigma(l),s})}\right]_{0,m} - l\right]_{0,m} + l \\ &= \left[f\left([n]_{\sigma(l),s}\right)^{g([n]_{\sigma(l),s})}\right]_{l,m}. \end{aligned}$$

This proves that f^g is periodic.

In succession, we prove case *6.*. We have $h(0) = 1$ and $h(n) = b^{h(n-1)}$ for every positive n. Let $r > 0$ and, for every $l \geq 0$, let $\sigma(l) = \lceil log_b(l) \rceil$. We prove by induction on i that

$$\forall\, i \in [0, r].\ \forall\, l \geq 0.\ \forall\, n \geq \sigma(l).\ [h(n+i)]_{l,\phi^{r-i}(r)} = [h(n+i+1)]_{l,\phi^{r-i}(r)}.$$

The case $i = 0$ is almost trivial. Since $\phi(n)$ is strictly decreasing for $n > 1$, we have $\phi^r(r) = 1$. This implies that $[h(n)]_{l,\phi^r(r)} = l = [h(n+1)]_{l,\phi^r(r)}$ for every $n \geq \sigma(l)$. Now, let $i > 0$. For every $l \geq 0$ and for every $n \geq \sigma(l)$, we have

$$
\begin{aligned}
\Big[h(n+i)\Big]_{l,\phi^{r-i}(r)} &= \Big[b^{h(n+(i-1))}\Big]_{l,\phi^{r-i}(r)} \\
&= \Big[b^{[h(n+(i-1))]_{\sigma(l),\phi^{r-(i-1)}(r)}}\Big]_{l,\phi^{r-i}(r)} \\
&= \Big[b^{[h(n+(i-1)+1)]_{\sigma(l),\phi^{r-(i-1)}(r)}}\Big]_{l,\phi^{r-i}(r)} \\
&= \Big[b^{h(n+(i-1)+1)}\Big]_{l,\phi^{r-i}(r)} \\
&= \Big[h(n+i+1)\Big]_{l,\phi^{r-i}(r)}.
\end{aligned}
$$

In particular, by setting $i = r$, we obtain

$$\forall\, l \geq 0.\ \forall\, n \geq \sigma(l).\ [h(n+r)]_{l,r} = [h(n+r+1)]_{l,r}.$$

Therefore, we conclude that, for every $l \geq 0$, $r > 0$, and $n \geq 0$, $[h(n)]_{l,r} = [h([n]_{\sigma(l)+r,1}]_{l,r}$ holds.

For case *7.*, we can write $[h(n)]_{l,r} = [[h(n-2)]_{l,r} + [h(n-1)]_{l,r}]_{l,r}$ whenever $n \geq 2$. By defining $(A_1(n), A_2(n)) = ([h(n)]_{l,r}, [h(n+1)]_{l,r})$, we have

$$(A_1(n), A_2(n)) = \begin{cases} ([1]_{l,r}, [1]_{l,r}) & \text{if } n = 0, \\ (A_2(n-1), [A_1(n-1) + A_2(n-1)]_{l,r}) & \text{if } n > 0. \end{cases}$$

Since $A_1(n)$ and $A_2(n)$ range over the finite domain $\{0, \ldots, l+r-1\}$, we can apply the Pigeonhole Principle and claim that there are two (computable) integers $p \geq 0$ and $q > 0$ such that $(A_1(n), A_2(n)) = (A_1(n+q), A_2(n+q))$ for every $n \geq p$. This implies that $[h(n)]_{l,r} = [h([n]_{p,q})]_{l,r}$.

We now prove case *8.*; case *9.* will follow similarly. Given a periodic function f, we fix two integers l, r and we denote by p, q the integers such that $[f(n)]_{l,r} = [f([n]_{p,q})]_{l,r}$. We further define $S = \sum_{i=p+1}^{p+q} f(i)$ and we note that $\Big[\sum_{i=p+1}^{p+kq} f(i)\Big]_{l,r} = [kS]_{l,r}$ holds, for every integer $k \geq 0$. From the Pigeonhole Principle, there are suitable integers p', q' such that $[p'kS]_{l,r} = [(p'+q')kS]_{l,r}$. Moreover, for any $n \geq p+qq'$, there is a k_n such that $n = [n]_{p,qq'} + k_n qq'$. Finally, for any $n \geq p + qq'$, we obtain

$$\begin{aligned}
\Big[h(n)\Big]_{l,r} &= \Big[\sum_{i=0}^{n} f(i)\Big]_{l,r} \\
&= \Big[\sum_{i=0}^{[n]_{p,qq'}} f(i) + \sum_{i=[n]_{p,qq'}}^{n} f(i)\Big]_{l,r} \\
&= \Big[\sum_{i=0}^{[n]_{p,qq'}} f(i) + q'k_n S\Big]_{l,r} \\
&= \Big[\sum_{i=0}^{[n]_{p,qq'}} f(i) + q'[k_n]_{p',q'} S\Big]_{l,r} \\
&= \Big[h([n]_{p,qq'})\Big]_{l,r}.
\end{aligned}$$

It remains to show case *10.*. This is easily proved by noticing that, for every $l \geq 0$ and $r > 0$, there are $p, p' \geq 0$ and $q, q' > 0$ such that $[g(f(n))]_{l,r} = [g([f(n)]_{p,q})]_{l,r} = [g([f([n]_{p',q'})]_{p,q})]_{l,r} = [g(f([n]_{p',q'}))]_{l,r}$. $\square$

The next theorem shows how to combine (colored) trees to obtain residually regular factorizations [21]. In particular, case *1.* links residually periodic functions to residually regular trees; case *2.* states that by interleaving the factors of residually regular factorizations we obtain again a residually regular factorization; case *3.* gives the possibility of periodically grouping the factors of a given factorization; case *4.* is useful to recursively define the factors of a residually regular factorization.

Theorem 3. *Given an ultimately periodic sequence of colors $c_1c_2c_3\ldots$, the factorization $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$ is residually regular in each of the following cases:*

1. **(Iteration)** *if $\mathcal{T}_i = \mathcal{U}^{(f(i)+1)_c}$, where $\mathcal{U}^{(f(i)+1)_c}$ denotes the $f(i)$-fold iteration of $\cdot_c\mathcal{U}$ applied to $\mathcal{U}$, with $\mathcal{U}$ being a residually regular tree and f being a residually periodic function;*
2. **(Interleaving)** *if there is $q > 0$ such that, for every $0 \leq i < q$, $\prod_{j\in\mathbb{N}}(\mathcal{T}_{jq+i})_c$ is a residually regular factorization;*
3. **(Grouping)** *if there is $q > 0$ and there is a residually regular factorization $\prod_{j\in\mathbb{N}}(\mathcal{U}_j)_c$ such that, for every $i \in \mathbb{N}$, $\mathcal{T}_i = (\mathcal{U}_{iq} \cdot_c \mathcal{U}_{iq+1} \cdot_c \ldots \cdot_c \mathcal{U}_{iq+q-1})$;*
4. **(Recursion)** *if $\mathcal{T}_0$ is a residually regular tree and there is a residually regular factorization $\prod_{j\in\mathbb{N}}(\mathcal{U}_j)_c$ such that $\mathcal{T}_{i+1} = \mathcal{U}_i \cdot_c (\mathcal{T}_i \cdot_d \mathcal{U}_i) \cdot_c (\mathcal{T}_i \cdot_d \mathcal{U}_i) \cdot_c \ldots$.*

Proof. The proof of case *1.* is almost trivial. Indeed, for any Rabin tree automaton M, the quotient set $[T^{ful}_{k,\Sigma}]_{/\equiv_M}$ endowed with a concatenation operator $\cdot_c$ is a finite groupoid. Thus, for all but finitely many i and for a suitable $q > 0$, we have:

$$\begin{aligned}
\Big[\mathcal{T}_i\Big]_{/\equiv_M} &= \Big[\mathcal{U}^{(f(i)+1)_c}\Big]_{/\equiv_M} \\
&= \Big[\mathcal{U}\Big]^{f(i)+1}_{/\equiv_M} \\
&= \Big[\mathcal{U}\Big]^{f(i+q)+1}_{/\equiv_M} \\
&= \Big[\mathcal{U}^{(f(i+q)+1)_c}\Big]_{/\equiv_M} \\
&= \Big[\mathcal{T}_{i+q}\Big]_{/\equiv_M}.
\end{aligned}$$

Hence, $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$ is a residual regular factorization.

For case *2.*, it suffices to see that, whenever there exist $p_0,\ldots,p_{q-1} \geq 0$ and $q_0,\ldots,q_{q-1} > 0$ such that, for every $0 \leq i < q$ and for every $j \geq p_i$, $\mathcal{T}_{jq+i} \equiv_M \mathcal{T}_{(j+q_i)q+i}$, then, by setting $p' = max\{p_i q : 0 \leq i < q\}$ and $q' = lcm\{q_i q : 0 \leq i < q\}$, we have $\mathcal{T}_i \equiv_M \mathcal{T}_{i+q'}$ for every $i \geq p'$.

We prove case *3.*. Given a Rabin tree automaton M, let $p \geq 0$ and $q > 0$ be integers such that $[\mathcal{U}_j]_{/\equiv_M} = [\mathcal{U}_{[j]_{p,q}}]_{/\equiv}$. By Theorem 2, there exists two (computable) integers $p' \geq 0$ and $q' > 0$ such that $[iq+j]_{p,q} = [[i]_{p',q'}q + j]_{p,q}$ for every $0 \leq j < q$ and for every i. Therefore, for every $i \geq 0$, we have:

$$\begin{aligned}
\Big[\mathcal{T}_i\Big]_{/\equiv_M} &= \Big[\mathcal{U}_{iq}\Big]_{/\equiv_M} \cdot_c \Big[\mathcal{U}_{iq+1}\Big]_{/\equiv_M} \cdot_c \cdots \Big[\mathcal{U}_{iq+q-1}\Big]_{/\equiv_M} \\
&= \Big[\mathcal{U}_{[iq]_{p,q}}\Big]_{/\equiv_M} \cdot_c \Big[\mathcal{U}_{[iq+1]_{p,q}}\Big]_{/\equiv_M} \cdot_c \cdots \Big[\mathcal{U}_{[iq+q-1]_{p,q}}\Big]_{/\equiv_M} \\
&= \Big[\mathcal{U}_{[[i]_{p',q'}q]_{p,q}}\Big]_{/\equiv_M} \cdot_c \Big[\mathcal{U}_{[[i]_{p',q'}q+1]_{p,q}}\Big]_{/\equiv_M} \cdot_c \cdots \Big[\mathcal{U}_{[[i]_{p',q'}q+q-1]_{p,q}}\Big]_{/\equiv_M} \\
&= \Big[\mathcal{U}_{[i]_{p',q'}q}\Big]_{/\equiv_M} \cdot_c \Big[\mathcal{U}_{[i]_{p',q'}q+1}\Big]_{/\equiv_M} \cdot_c \cdots \Big[\mathcal{U}_{[i]_{p',q'}q+q-1}\Big]_{/\equiv_M} \\
&= \Big[\mathcal{T}_{[i]_{p',q'}}\Big]_{/\equiv_M}.
\end{aligned}$$

Consequently, $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$ is a residually regular factorization.

Finally, we prove case *4.*. Given a Rabin tree automaton M, let $l \geq 0$ and $m > 0$ be integers such that $[\mathcal{U}_i]_{/\equiv_M} = [\mathcal{U}_{[i]_{l,m}}]_{/\equiv_M}$. We further define $A_i = [\mathcal{T}_i]_{/\equiv_M}$, for every $i \geq 0$. Consider now the subsequence $(A_{l+im})_{i\in\mathbb{N}}$ of $(A_i)_{i\in\mathbb{N}}$. Since the A_{l+im}'s range over a finite domain, by Pigeonhole Principle, there exist two integers $r \geq 0$ and $s > 0$ such that $A_{l+rm} = A_{l+(r+s)m}$. We prove, by induction on i, that $A_{l+rm+i} = A_{l+(r+s)m+i}$. The base case is trivial; so, for the induction step, suppose that $A_{l+rm+i} = A_{l+(r+s)m+i}$. Since $\mathcal{U}_{l+rm+i} \equiv_M \mathcal{U}_{l+(r+s)m+i}$ and $\mathcal{T}_{l+rm+i} \equiv_M \mathcal{T}_{l+(r+s)m+i}$ hold, by Proposition 3, we have:

$$\begin{aligned}
A_{l+rm+i+1} &= \Big[\mathcal{U}_{l+rm+i} \cdot_c \big(\mathcal{T}_{l+rm+i} \cdot_d \mathcal{U}_{l+rm+i}\big) \cdot_c \cdots\Big]_{/\equiv_M} \\
&= \Big[\mathcal{U}_{l+(r+s)m+i} \cdot_c \big(\mathcal{T}_{l+(r+s)m+i} \cdot_d \mathcal{U}_{l+(r+s)m+i}\big) \cdot_c \cdots\Big]_{/\equiv_M} \\
&= A_{l+(r+s)m+i+1}.
\end{aligned}$$

Therefore, for every i, we have $\mathcal{T}_i \equiv_M \mathcal{T}_{[i]_{l+rm,sm}}$. This proves that $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$ is a residually regular factorization. $\square$

3.4 On the Effectiveness of the Automaton-based Approach

In this section we show the effectiveness of the proposed automaton-based approach. In particular, we exploit the results outlined in the previous section to prove decidability results for two meaningful structures.

As a first example, let us consider the semi-infinite line $\mathcal{L}$ with forward edges, backward edges and loops (cf. Figure 1), which belongs to the Caucal hierarchy, and its unfolding from the leftmost vertex. We provide an alternative proof of the decidability of the MSO theory of the unfolding of $\mathcal{L}$. The unfolded graph can be embedded into the infinite complete $\{w, b\}$-colored tree $\mathcal{T}$ of Figure 2. Black nodes b correspond to nodes of the original structure, while white nodes w are added to complete the tree. For the sake of readability, we adopt $\{1, \bar{1}, \#\}$ instead of $\{1, 2, 3\}$ as the set of edge labels. Using a set of three colors $\{w, b, c\}$, a factorization $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_c$ of $\mathcal{T}$ can be inductively defined as follows (see the dashed regions in Figure 2):

- $\mathcal{T}_0 = \mathcal{U} \cdot_w \mathcal{W}$;
- $\mathcal{T}_{i+1} = \mathcal{U} \cdot_w (\mathcal{T}_i \cdot_c \mathcal{U}) \cdot_w (\mathcal{T}_i \cdot_c \mathcal{U}) \cdot_w \ldots$,

where $\mathcal{U} = \prod_{j\in\mathbb{N}}(b\langle w, b, c\rangle)_b$ (we denote by $b\langle w, b, c\rangle$ the full finite ternary tree with a b-colored root and 3 leaves colored by w, b, and c, respectively) and $\mathcal{W}$ is the infinite complete ternary $\{w\}$-colored tree. From Theorem 3 (case *4.*), it follows that such a factorization is residually regular. This accounts for the decidability of $MTh(\mathcal{T})$.

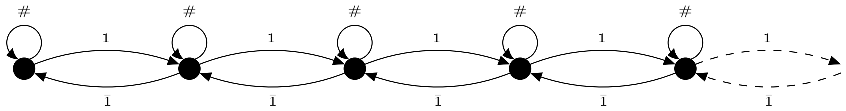

Fig. 1. The semi-infinite line $\mathcal{L}$.

As a second example, consider the infinite binary $\{w, b\}$-colored tree $\mathcal{T}_{tow}$ such that $\mathcal{T}_{tow}(u) = b$ iff $u = 1^n0^m$, with $m < tow(n)$, where tow is the exponential tower defined by $tow(0) = 1$ and $tow(n+1) = 2^{tow(n)}$ (cf. Figure 3). In [6], Carayol and Wöhrle show that such a tree does not belong to the Caucal hierarchy, but it enjoys a decidable MSO theory. We give an alternative proof of the decidability of $MTh(\mathcal{T}_{tow})$ by providing a residually regular factorization of $\mathcal{T}_{tow}$. A factorization $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_c$ of $\mathcal{T}_{tow}$ can be defined as follows (see the dashed regions in Figure 3):

- $\mathcal{T}_0 = b\langle w, c\rangle \cdot_w \mathcal{W}$;
- $\mathcal{T}_i = b\langle b, c\rangle \cdot_b \left(b\langle b, w\rangle \cdot_w \mathcal{W}\right)^{(tow(i)-1)_b} \cdot_b \mathcal{W}$ for $i \geq 1$,

where $\mathcal{W}$ is the infinite complete binary $\{w\}$-colored tree. From Theorem 3 (case *1.*), it follows that such a factorization is residually regular, and thus $MTh(\mathcal{T}_{tow})$ is decidable.

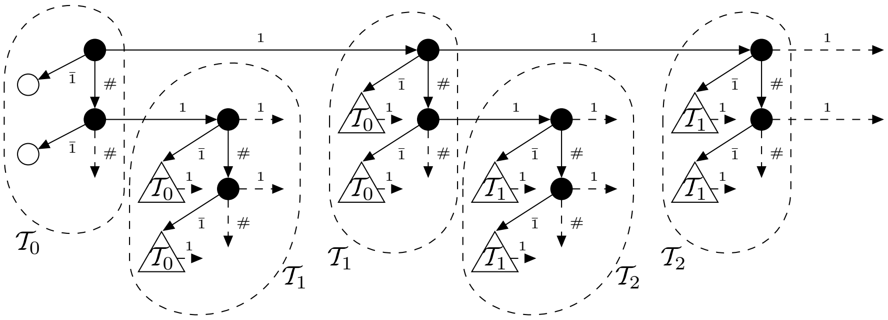

Fig. 2. The tree $\mathcal{T}$ embedding the unfolding of $\mathcal{L}$.

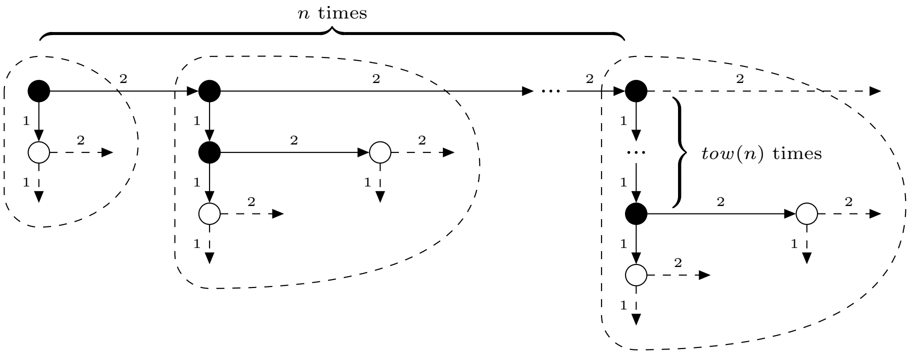

Fig. 3. The tree $\mathcal{T}_{tow}$.

4 Layered Temporal Structures

In this section we define (ω-)layered structures and we explore the relationships among them. In particular, we show that the theories of (k-refinable) downward/upward unbounded ω-layered structures can be embedded into the theory of (k-refinable) totally unbounded ω-layered strucutures. In Section 4.3, we shall exploit the automaton-based framework to solve the decision problem for such a kind of structures.

Layered structures have been originally proposed by Montanari et al. in [17, 18, 20] to model finite and infinite hierarchies of time granularities. They focus their attention on three distinct layered structures: the n-layered k-refinable structure, abbreviated n-LS, which consists of a fixed finite number n of temporal layers such that each time point can be refined into k time points of the immediately finer layer, if any, and the downward (resp. upward) unbounded k-refinable ω-layered structure, abbreviated k-DULS (resp. k-UULS), which consists of an infinite number of arbitrarily fine (resp. coarse) layers. The MSO theories of these structures have been shown to be expressive enough to capture meaningful temporal properties of reactive systems (such as "P holds at all time points k^i, with $i \geq 0$") and decidable. The decidability of the theories of the k-refinable n-LS, DULS, and UULS has been proved by reducing them to the MSO theories of one successor ($S1S$), of k successors (SkS), and of the k-ary systolic tree ($S1S^k$), respectively. Both the DULS and the UULS can naturally be viewed as tree structures. The DULS can be viewed as an infinite sequence of infinite k-ary trees, while the UULS can be seen as a complete k-ary infinite tree generated from the leaves or, equivalently, as an infinite sequence of finite increasing k-ary trees [16]. We further introduce a new structure, namely, the k-refinable totally unbounded ω-layered structure (k-TULS), which can be viewed as the composition of the k-DULS and the k-UULS. We show that this structure embeds both the k-DULS and the k-UULS.

Definition 7. *A* k-refinable layered structure *is a graph structure* $\mathcal{G} = (\bigcup_{i \in I} L_i, <, (\downarrow_l)_{l \in [k]})$, *where* $I \subseteq \mathbb{Z}$, $L_i = \{(i, n) : n \in \mathbb{N}\}$, $<$ *is a suitable total order over* $\bigcup_{i \in I} L_i$, *and, for all* $l \in [k]$, $\downarrow_l$ *is a function that maps the element* (i, n) *to the element* $(i + 1, kn + l - 1)$ *(if there exists such an element).*

For all $i \in I$, L_i is called a *layer* of the structure and, for all $l \in [k]$, $\downarrow_l$ is called the *l-th projection function*, since it maps elements of a given layer to elements of the immediately finer layer (if any).

We further denote by $\prec_0$, $\prec_1$, and $\prec_2$ the order relations for 2-refinable layered structures, which are given, respectively, by the pre-order, in-order, and post-order visits of the vertexes. For generic k-refinable ω-layered structures, one can easily generalize the definitions through $k+1$ ordering relations $\prec_0, \ldots, \prec_k$.

Definition 8. *For every* $i \in \{0, \ldots, k\}$, *we define the (partial) ordering* $\prec_i$ *such that, for every pair of vertices in the k-refinable ω-layered structure* $\mathcal{G}$, $x \prec_i y$ *iff one of the following conditions holds:*

1. *there are* $z \in \mathcal{G}$, $i_0, i_1 \in [k]$ *such that* $i_0 < i_1$, $(z, x) \in \downarrow_{i_0} (\bigcup_{i \in [k]} \downarrow_i)^*$, *and* $(z, y) \in \downarrow_{i_1} (\bigcup_{i \in [k]} \downarrow_i)^*$;
2. *there is* $i_1 \in [k]$ *such that* $i_1 \geq i$ *and* $(z, y) \in \downarrow_{i_1} (\bigcup_{i \in [k]} \downarrow_i)^*$;
3. *there is* $i_0 \in [k]$ *such that* $i_0 < i$ *and* $(z, x) \in \downarrow_{i_0} (\bigcup_{i \in [k]} \downarrow_i)^*$.

In this paper, we restrict our attention to the k-DULS (see Figure 4), the k-UULS (see Figure 5), and the k-TULS. The k-DULS can be seen as an infinite sequence of infinite complete k-ary trees, while the k-UULS can be seen either as an infinite k-branching tree generated from the leaves or as an infinite sequence of finite increasing k-trees. Formally, DULS are obtained by setting $I = \mathbb{N}$ in Definition 7 and defining $<$ as the total order given by the pre-order visit (for elements of the same tree) and by the linear order of the trees (for elements belonging to different trees), while UULS are obtained by setting $I = -\mathbb{N}$ and defining $<$ as the total order given by the in-order visit of the nodes of the tree. TULS are obtained by setting $I = \mathbb{Z}$ and defining $<$ as a suitable total order (e.g., the total order induced by the pre-order visit or the in-order one).

In order to systematically analyze the relationships between TULS and DULS (resp. UULS), we define a couple of auxiliary relations. We denote the layer

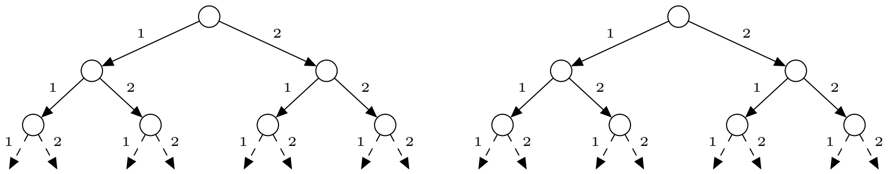

Fig. 4. The 2-DULS.

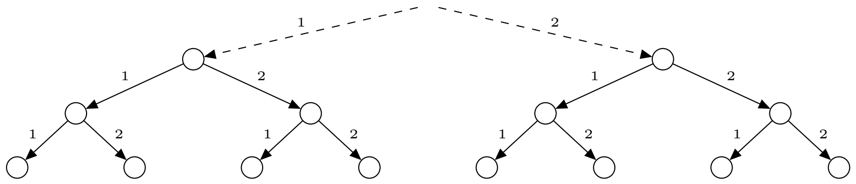

Fig. 5. The 2-UULS.

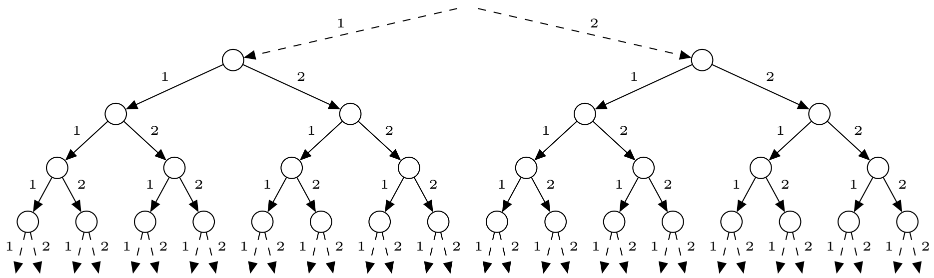

Fig. 6. The 2-TULS.

$\{(0, n) : n \in \mathbb{N}\}$ by L_0 (L_0 is the top layer of DULS, the bottom layer of UULS, and a distinguished intermediate layer of TULS). Furthermore, we denote by $<_0$ (resp. $+_0$) the order relation (resp. the successor function) over L_0, which are defined as follows: for every $n, n' \in \mathbb{N}$, $(0, n) <_0 (0, n')$ iff $n < n'$ and $+_0((0, n)) = (0, n')$ iff $n' = n + 1$ (it is well-know that $<_0$ is MSO definable in terms of $+_0$). In Figure 7, we depict a little part of the 2-TULS, pointing out the elements of L_0 by black vertices and the successor function $+_0$ by bold arrows.

4.1 Definability of Order Relations

We now investigate the expressiveness of extensions of ω-layered structures. In particular we show that (i) the usual order relations of UULS and TULS are definable in terms of reflexive and transitive closures of projection functions, and (ii) the usual order relations of DULS are definable in terms of reflexive and transitive closures of projection functions and in terms of the successor function defined on the elements of the top layer. The following lemma is useful to define

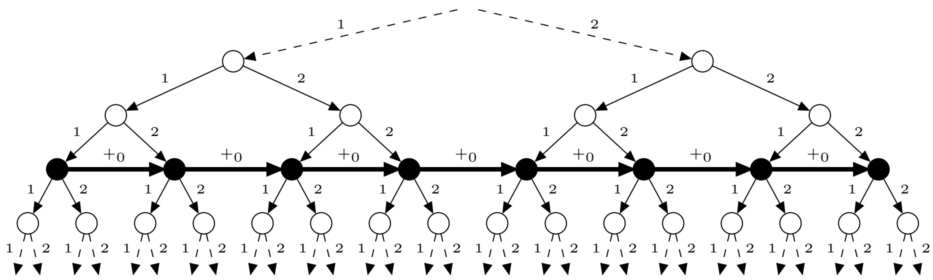

Fig. 7. The 2-TULS extended with L_0 and $+_0$.

reflexive and transitive closures of binary predicates and to prove a series of results, such as Propositions 6, 7, 8, and 9.

Lemma 2. *For any graph structure $\mathcal{G}$ endowed with a binary relation R, the reflexive and transitive closure R^* of R is MSO definable in $\mathcal{S}$.*

Proof. The formula $\varphi(x, y) = \forall\, Z.\ \forall\, z_1.\ \forall\, z_2.\ ((Z(z_1) \wedge R(z_1, z_2)) \rightarrow Z(z_2)) \wedge Z(x)) \rightarrow Z(y)$ defines the relation R^*. $\square$

Proposition 6. *For every $0 \leq i \leq k$, the relation $\prec_i$ is MSO definable in any k-refinable ω-layered structure.*

Proof. Trivial by definition of $\prec_i$ and by Lemma 2. $\square$

Proposition 7. *Given a k-refinable ω-layered structure $\mathcal{G}$, $<_0$ is MSO definable in $(\mathcal{G}, +_0)$ and $+_0$ is MSO definable in $(\mathcal{G}, <_0)$.*

Proof. The formula $\varphi(x, y) = \neg(x = y) \ \wedge \ +_0^*(x, y)$ defines the partial order $<_0$ through function $+_0$, where $+_0^*$ denotes the MSO formula for the reflexive and transitive closure of $+_0$ (such a formula exists by Lemma 2). Similarly, the formula $\varphi'(x, y) = x <_0 y \ \wedge \ \forall\, z.\ (x <_0 z \ \rightarrow \ (y = z \ \vee \ y <_0 z))$ defines the successor function $+_0$ through relation $<_0$. $\square$

It is clear that the unary predicate L_0 is MSO definable in the DULS (since $L_0 = \{x : \nexists\, y.\ (y, x) \in \downarrow_1\}$) and in the UULS (since $L_0 = \{x : \exists\, y.\ (x, y) \in <_0\}$ and $<_0$ is MSO definable in the UULS). Furthermore, L_0 is MSO definable in the TULS extended with $+_0$, because $L_0 = \{x : \exists\, y.\ (x, y) \in +_0\}$. Similar results hold for the relations $<_0$ and $+_0$ with respect to UULS and TULS, but not with respect to DULS.

Proposition 8. *The function $+_0$ (and hence the relation $<_0$) is MSO definable in the UULS and in the TULS extended with L_0, but not in the DULS.*

Proof. The formula $\varphi(x, y) = L_0(x) \wedge L_0(y) \wedge \exists\, z_1, z_2, z_3.\ \bigvee_{i \in [k]} (\downarrow_i (z_1, z_2) \wedge \downarrow_k^* (z_2, x) \ \wedge \ \downarrow_{i+1} (z_1, z_3) \ \wedge \ \downarrow_1^* (z_3, y))$ defines the successor function $+_0$ in the

UULS and in the TULS extended with L_0, provided that $\downarrow_i^*$ is the formula defining the reflexive and transitive closure of $\downarrow_i$ (such a formula exists by Lemma 2). Note that, by above observations, the relation L_0 can be removed from the UULS without affecting the result.

In order to show that $+_0$ is not MSO definable in the DULS, it is sufficient to see that there is no way of comparing elements in the top layer of the k-DULS. Formally, for every formula $\varphi(x, y)$ and for every pair of vertexes $v_1, v_2 \in L_0$, the following property holds:

$$k\text{-DULS} \vDash \varphi[v_1, v_2] \quad \text{iff} \quad k\text{-DULS} \vDash \varphi[v_2, v_1].$$

Such a property implies that $+_0$ is not MSO definable in the DULS. □

The MSO logics of DULS, UULS and TULS endowed only with the usual projection functions are pairwise expressively non-comparable because each of such structure allows one to capture properties that cannot be defined in the other structures. Summing up all the above results, we can claim that the MSO logic over the TULS extended with L_0 is at least as expressive as the MSO logics over the DULS and the UULS extended with $<_0$, $+_0$, and L_0.

4.2 Embedding Layered Structures

In [16], the decidability of the MSO theories of DULS and UULS has been proved by embedding them into MSO theories of suitable (different) tree structures. In the following, we deal with the decision problem for the MSO theory of TULS extended with L_0. In the previous section we showed that the MSO theories of the (extended) DULS and UULS can be embedded in the MSO theory of the extended TULS. Thus it make sense to tackle the decision problem for $MTh(k\text{-TULS}, L_0)$. The method we are going to present also provides an alternative (uniform) decidability proof for the theories of DULS and UULS.

As a preliminary step, we show that the MSO theories of DULS, UULS, and TULS extended with L_0 can be embedded into the MSO theories of suitable infinite (complete) $(k+1)$-ary (colored) trees. Notice that such an embedding allows us to move from the setting of layered structures to the more standard framework of (colored) trees. Precisely, we translate MSO formulas for ω-layered structures into equivalent MSO formulas interpreted over tree structures. We preliminarily define the injective functions $f_{k-\text{DULS}}$, $f_{k-\text{UULS}}$, and $f_{k-\text{TULS}}$, which map the vertexes in the k-refinable layered structures to vertices in the infinite $(k+1)$-ary tree:

$$f_{k\text{-DULS}}((i,n)) = \underbrace{(k+1)\cdot \ldots \cdot (k+1)}_{\lfloor \frac{n}{k^i} \rfloor \text{ times}} \cdot \underbrace{(\tfrac{n}{k^{i-1}} \bmod k+1)\cdot \ldots \cdot (\tfrac{n}{k^0} \bmod k+1)}_{i \text{ times}}$$

$$f_{k\text{-UULS}}((i,n)) = \underbrace{(k+1)\cdot \ldots \cdot (k+1)}_{A \text{ times}} \cdot \underbrace{(\tfrac{n}{k^{A+i-1}} \bmod k+1)\cdot \ldots \cdot (\tfrac{n}{k^0} \bmod k+1)}_{A+i \text{ times}}$$

$$f_{k\text{-TULS}}((i,n)) = \underbrace{(k+1)\cdot \ldots \cdot (k+1)}_{A \text{ times}} \cdot \underbrace{(\tfrac{n}{k^{A+i-1}} \bmod k+1)\cdot \ldots \cdot (\tfrac{n}{k^0} \bmod k+1)}_{A+i \text{ times}}$$

where A is either $\lceil log_k(n+1)\rceil - i$ or 0, depending on whether $n+1 \geq k^i$ or not.

The above functions allow us to embed the ω-layered structures into tree structures. For any ω-layered structure $\mathcal{G}$, we define the unary predicates $D_{\mathcal{G}} = f_{\mathcal{G}}(\mathcal{D}om(\mathcal{G}))$ and $L_{\mathcal{G}} = f_{\mathcal{G}}(L_0)$. Both $D_{\mathcal{G}}$ and $L_{\mathcal{G}}$ are subsets of the domain of the infinite $(k+1)$-ary tree $([k+1]^*, (E_i)_{i\in[k+1]})$, henceforth denoted Δ_{k+1}. Thus, they can be given the status of vertex-coloring relations over Δ_{k+1}. Figures 8, 9, and 10 depict such relations; shaded nodes represent the vertices of the domain of the corresponding ω-layered structure, and black nodes represent the vertices corresponding to the layer L_0. Note the following characterization in terms of regular/context-free languages:

- $D_{k\text{-DULS}} = \{k+1\}^* \cdot [k]^*$,
- $D_{k\text{-UULS}} = \{\varepsilon\} \cup \bigcup_{0\leq m\leq n}\{k+1\}^{n+1} \cdot \{k\} \cdot [k]^m$,
- $D_{k\text{-TULS}} = [k]^* \cup \{k+1\} \cdot \{k+1\}^* \cdot \{k\} \cdot [k]^*$,
- $L_{k\text{-DULS}} = \{k+1\}^*$,
- $L_{k\text{-UULS}} = \{\varepsilon\} \cup \bigcup_{n\geq 0}\{k+1\}^{n+1} \cdot \{k\} \cdot [k]^n$,
- $L_{k\text{-TULS}} = \{\varepsilon\} \cup \bigcup_{n\geq 0}\{k+1\}^{n+1} \cdot \{k\} \cdot [k]^n$.

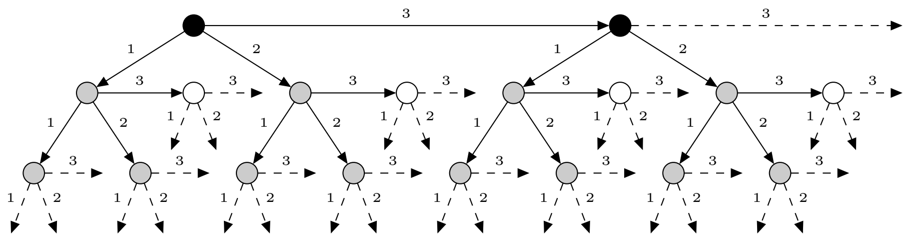

Fig. 8. The predicates $D_{2\text{-DULS}}$ and $L_{2\text{-DULS}}$.

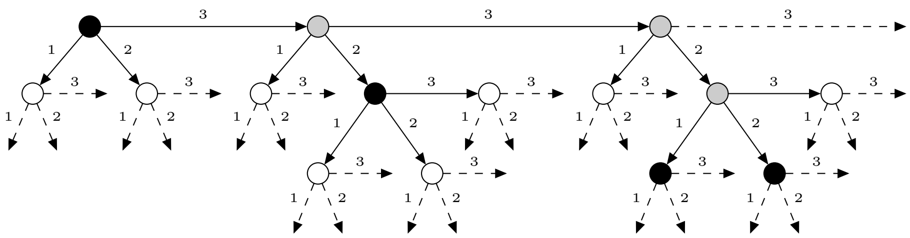

Fig. 9. The predicates $D_{2\text{-UULS}}$ and $L_{2\text{-UULS}}$.

The following theorem states that MSO formulas interpreted over ω-layered structures can be turned into equivalent MSO formulas interpreted over suitable (colored) trees. As an example, we have:

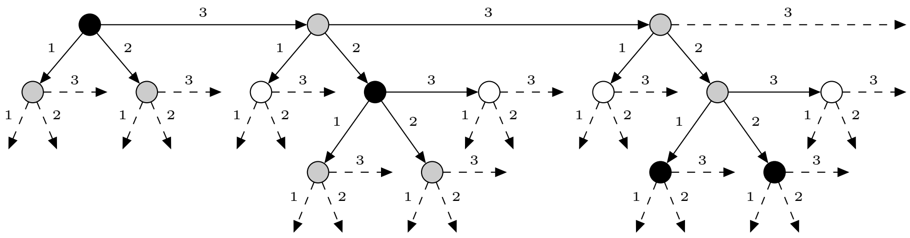

Fig. 10. The predicates $D_{2\text{-TULS}}$ and $L_{2\text{-TULS}}$.

i) $(2\text{-TULS}, L_0) \models L_0[v]$ iff $(\Delta_3, D_{2\text{-TULS}}, L_{2\text{-TULS}}) \models L_{2\text{-TULS}}[f_{2\text{-TULS}}(v)]$,
ii) $(2\text{-TULS}, L_0) \models \downarrow_1 (u, v)$ iff $(\Delta_3, D_{2\text{-TULS}}, L_{2\text{-TULS}}) \models E_1(f_{2\text{-TULS}}(u), f_{2\text{-TULS}}(v)) \vee E_3(f_{2\text{-TULS}}(v), f_{2\text{-TULS}}(u))$.

From this it follows that, whenever the MSO theory of $(\Delta_{k+1}, D_{k\text{-TULS}}, L_{k\text{-TULS}})$ has been proved to be decidable, then the MSO theory of the ω-layered structure $(k\text{-TULS}, L_0)$ would be decidable as well.

Theorem 4. *Let $(\mathcal{G}, L_0)$ be a k-refinable ω-layered structure extended with the predicate L_0. For any MSO formula $\varphi(Z, \bar{x}, \bar{X})$, there is a MSO formula $\varphi'(Y, Z, \bar{x}, \bar{X})$ such that, for every tuple $\bar{v}$ of vertices and for every tuple $\bar{V}$ of subsets of $\mathcal{D}om(\mathcal{G})$,*

$$\mathcal{G} \models \varphi[L_0, \bar{v}, \bar{V}] \quad \textit{iff} \quad \Delta_{k+1} \models \varphi'[D_{\mathcal{G}}, L_{\mathcal{G}}, f_{\mathcal{G}}(\bar{v}), f_{\mathcal{G}}(\bar{V})].$$

Proof. We write $\varphi(Z, \bar{x}, \bar{X}) \rightsquigarrow \varphi'(Y, Z, \bar{x}, \bar{X})$ to shortly denote the property claimed by the theorem. Hereafter, $\downarrow_1, \ldots, \downarrow_k$ denote the k projection functions of the ω-layered structure and $E_1, \ldots, E_{k+1}$ denote the $k+1$ edge relations of the infinite complete tree Δ_{k+1}. Now, in order to prove the theorem, it suffices to see that

- $x_{i_1} = x_{i_2} \rightsquigarrow x_{i_1} = x_{i_2}$;
- $X_{i_1}(x_{i_2}) \rightsquigarrow X_{i_1}(x_{i_2})$;
- $\downarrow_i (x_{i_1}, x_{i_2}) \rightsquigarrow E_i(x_{i_1}, x_{i_2})$,
  provided that $i \neq 1$ or $\mathcal{G} = k$-DULS;
- $\downarrow_1 (x_{i_1}, x_{i_2}) \rightsquigarrow E_1(x_{i_1}, x_{i_2}) \ \vee \ E_{k+1}(x_{i_2}, x_{i_1})$,
  provided that $\mathcal{G}$ is either the k-UULS or the k-TULS;
- $\neg\varphi(Z, \bar{x}, \bar{X}) \rightsquigarrow \neg\varphi'(Y, Z, \bar{x}, \bar{X})$,
  provided that $\varphi(Z, \bar{x}, \bar{X}) \rightsquigarrow \varphi'(Y, Z, \bar{x}, \bar{X})$;
- $\varphi_1(Z, \bar{x}, \bar{X}) \ \vee \ \varphi_2(Y, Z, \bar{x}, \bar{X}) \rightsquigarrow \varphi'_1(Z, \bar{x}, \bar{X}) \ \vee \ \varphi'_2(Y, Z, \bar{x}, \bar{X})$,
  provided that $\varphi_1(Z, \bar{x}, \bar{X}) \rightsquigarrow \varphi'_1(Y, Z, \bar{x}, \bar{X})$ and $\varphi_2(Z, \bar{x}, \bar{X}) \rightsquigarrow \varphi'_2(Y, Z, \bar{x}, \bar{X})$;
- $\exists\, x_{n+1}.\ \varphi(Z, \bar{x}, x_{n+1}, \bar{X}) \rightsquigarrow \exists\, x_{n+1}.\ \varphi'(Y, Z, \bar{x}, x_{n+1}, \bar{X}) \ \wedge \ Y(x_{n+1})$,
  provided that $\varphi(Z, \bar{x}, x_{n+1}, \bar{X}) \rightsquigarrow \varphi'(Y, Z, \bar{x}, x_{n+1}, \bar{X})$;
- $\exists\, X_{m+1}.\ \varphi(Z, \bar{x}, \bar{X}, X_{m+1}) \rightsquigarrow \exists\, X_{m+1}.\ \varphi'(Y, Z, \bar{x}, \bar{X}, X_{m+1}) \ \wedge \ X_{m+1} \subseteq Y$,
  provided that $\varphi(Z, \bar{x}, \bar{X}, X_{m+1}) \rightsquigarrow \varphi'(Y, Z, \bar{x}, \bar{X}, X_{m+1})$.

$\square$

As regards the decidability of trees embedding ω-layered structures, we provide the following results. As a matter of fact, since the predicate $D_{k\text{-TULS}}$ turns out to be MSO definable in the infinite (complete) tree Δ_{k+1}, we can ignore it in coping with the decidability of the extended TULS.

Proposition 9. *The predicates $D_{k\text{-}DULS}$, $D_{k\text{-}TULS}$, $L_{k\text{-}DULS}$ are MSO definable in Δ_{k+1}. Conversely, the predicates $D_{k\text{-}UULS}$, $L_{k\text{-}UULS}$, $L_{k\text{-}TULS}$ are not MSO definable in Δ_{k+1}.*

Proof. Let $root(x) = \neg\exists\, y.\ \bigvee_{i\in[k+1]} E_i(y,x)$. Clearly, $root(x)$ restricts the variable x to be interpreted by the root node in the tree structure Δ_{k+1}. Further let $anc_k(x,y)$ denote the formula for the reflexive and transitive closure of the relation $\bigcup_{i\in[k]} E_i$ (such a formula exists by Lemma 2). Now, the formulas

$$\begin{aligned}
&\varphi_1(x) = \exists\, y,z.\ (root(y)\ \wedge\ E^*_{k+1}(y,z)\ \wedge\ anc_k(z,x)) \\
&\varphi_2(x) = \exists\, y.\ (root(y)\ \wedge\ anc_k(y,x))\ \vee\ \exists\, y,z,v.\ (root(y)\ \wedge\ E^*_{k+1}(y,z) \\
&\qquad\quad \wedge\ y \neq z\ \wedge\ E_k(z,v)\ \wedge\ anc_k(v,x)) \\
&\varphi_3(x) = \exists\, y.\ (root(y)\ \wedge\ E^*_{k+1}(y,x))
\end{aligned}$$

restrict the variable x to be interpreted by elements of $D_{k\text{-DULS}}$, $D_{k\text{-TULS}}$, and $L_{k\text{-DULS}}$, respectively.

Conversely, we prove that $D_{k\text{-UULS}}$ is not MSO definable in Δ_{k+1}; the proofs for $L_{k\text{-UULS}}$ and $L_{k\text{-TULS}}$ are similar. Assume, by way of contradiction, that there exists an MSO formula $\varphi(x)$ defining $D_{k\text{-UULS}}$ in Δ_{k+1}. This would imply that $\psi(X) = \forall\, x.\ X(x)\ \leftrightarrow\ \varphi(x)$ holds in Δ_{k+1} iff X is interpreted as $D_{k\text{-UULS}}$. By Rabin's Theorem [24], there would be a Rabin automaton M such that $\mathscr{L}(M)$ is the singleton consisting of the single tree Δ_{k+1} colored by $D_{k\text{-UULS}}$. Such a tree would be non-regular since it would have infinitely many non-isomorphic subtrees (see Figure 9). Because any non-empty Rabin-recognizable set of trees contains a regular tree, $\mathscr{L}(M)$ could not be a singleton. This is a contradiction and hence $D_{k\text{-UULS}}$ is not MSO definable in Δ_{k+1}. □

4.3 Deciding MSO Theories of Layered Structures

Here we exploit the automaton-based framework to provide a uniform decision method for the MSO theories of k-refinable ω-layered structures. From the results of previous sections, we may choose to restrict ourself to the crucial problem of decidability of $MTh(k\text{-TULS}, L_0)$ (recall that the decidability of such a theory implies the decidability of the MSO theories of k-DULS and k-UULS extended with relations $<_0$, $+_0$, and L_0). We can embed $(k\text{-TULS}, L_0)$ into a suitable infinite (complete) $(k+1)$-ary $\{w,b\}$-colored tree $\mathcal{T}_{k\text{-TULS}}$ (the colors w and b stand for 'white' and 'black', respectively). As an example, Figure 11 depicts the tree $\mathcal{T}_{2\text{-TULS}}$ for the 2-TULS extended with L_0. By Theorem 4, MSO formulas interpreted over the k-TULS can be turned into equivalent MSO formulas interpreted over $\mathcal{T}_{k\text{-TULS}}$. This means that the decidability of the MSO theory of $(k\text{-TULS}, L_0)$ directly follows from the decidability of the MSO theory of the corresponding colored tree $\mathcal{T}_{k\text{-TULS}}$.

$MTh(\mathcal{T}_{k\text{-TULS}})$ can be easily proved to be decidable by providing a residually regular factorization of $\mathcal{T}_{k\text{-TULS}}$. As an example, a possible choice for the components of a factorization of $\mathcal{T}_{2\text{-TULS}}$ is represented by dashed regions in Figure 11. The elements corresponding to L_0 are represented by b-colored notes. By adding an auxiliary third color c to manage concatenations, the factorization can be written as $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_c$, where

- $\mathcal{T}_0 = b\langle w, w, c\rangle \cdot_w \mathcal{W}$,
- $\mathcal{T}_{i+1} = w\langle w, b, c\rangle \cdot_w \mathcal{W} \cdot_b \mathcal{U}_i$,
- $\mathcal{U}_0 = b\langle w, w, w\rangle \cdot_w \mathcal{W}$,
- $\mathcal{U}_{i+1} = w\langle b, b, w\rangle \cdot_w \mathcal{W} \cdot_b \mathcal{U}_i$,
- $\mathcal{W}$ is the infinite complete ternary white ($\{w\}$-colored) tree.

The above factorization can be easily proved to be residually regular by exploiting Theorem 3. Theorem 1 implies the decidability of $MTh(\mathcal{T}_{2\text{-TULS}})$. Such a method can be trivially generalized to the case of a generic k-TULS.

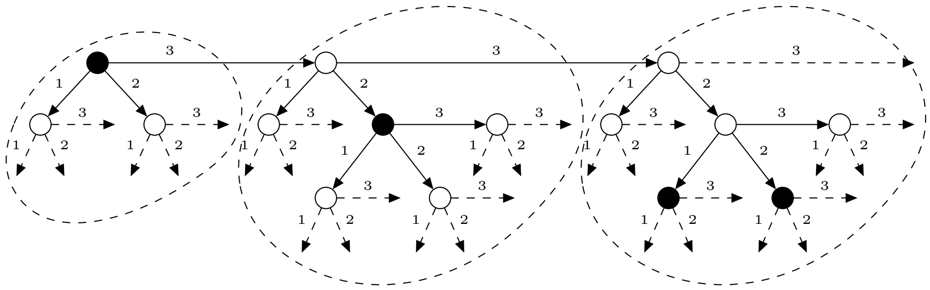

Fig. 11. The tree $\mathcal{T}_{2\text{-TULS}}$ embedding (k-TULS, L_0).

5 On the Relationship with the Caucal Hierarchy

We exploit the automaton-based method to provide an embedding of some representative graphs of the so-called Caucal hierarchy into our framework. Precisely, we show that residually regular trees allow us to capture tree generators for MSO interpretations in the Caucal hierarchy. In [6], Carayol and Wöhrle show that, for each level of the Caucal hierarchy, there exists a representative graph, called generator, from which all the other graphs belonging to that level can be obtained by the application of the basic operations of rational marking and MSO interpretation. These generators are closely related to the 'tree generators' introduced by Cachat to simulate games on higher order pushdown systems [5]. The level 1 tree generator $\mathcal{C}_1$ is the infinite complete binary tree. The level 2 tree generator $\mathcal{C}_2$ is the tree obtained from $\mathcal{C}_1$ by adding reverse edges and a loop for each vertex (labeled by fresh symbols, e.g., $\bar{1}, \bar{2}, \#$), and then by unfolding the resulting graph from its root. The level 3 tree generator $\mathcal{C}_3$ is obtained by applying the same operation (which we shortly denote by $MSOUnf$) to $\mathcal{C}_2$, and so on. Since *MSOUnf* is an MSO interpretation followed by an unfolding, tree generators belong to the Caucal hierarchy.

Proposition 10. *Every level n graph of the Caucal hierarchy can be obtained from $\mathcal{C}_n$ by applying a rational marking followed by an inverse rational mapping (we refer to [6] for formal definitions).*

Proof. The proof is based on an idea of Carayol. We denote by Δ_2^n the graph obtained from Δ_2 by an $(n-1)$-fold application of the treegraph operation (cf. [6]). Δ_2^n has been showed to generate any level n graph of the Caucal hierarchy via rational markings followed by inverse rational mappings. It turns out that the generator Δ_2^n can be obtained from $\mathcal{C}_n$ by applying an MSO interpretation. Precisely, we have $\Delta_2^n = \mathcal{I}_2(\mathcal{I}_1(\mathcal{C}_n))$, where $\mathcal{I}_1$ is the MSO interpretation that restricts the domain of $\mathcal{C}_n$ to the set of vertices corresponding to a word which does not contain $l\bar{l}$ or $\bar{l}l$, for any label l, and $\mathcal{I}_2$ is the MSO interpretation that replaces each $\bar{l}$-labeled edge (u, v) by an l-labeled edge (v, u). Since Δ_2^n is a generator for rational markings and inverse rational mappings (i.e. MSO transductions, cf. [6]), then $\mathcal{C}_n$ is a tree generator for MSO rational markings and inverse rational mappings.

Tree generators can be embedded into residually regular trees as intuitively explained in the following. First, $\mathcal{C}_1$ can be viewed as a uniformly colored tree and thus it obviously is a residually regular tree. Then, by generalizing the construction used in the case of the semi-infinite line, one can prove that the class of residually regular trees is closed under *MSOUnf*. As matter of fact, we need to slightly modify the definition of *MSOUnf* to operate inside the class of *full* trees; however, this is a trivial generalization which preserves all results about decidability and expressiveness of tree generators.

Formally, we identify any *non-full* k-ary Σ-colored tree $\mathcal{T}$ with the *full* k-ary $(\Sigma \cup \{\perp\})$-colored tree $\mathcal{T}'$ obtained by appending to $\mathcal{T}$ the missing leaves and by coloring them with $\perp$. Then, given the full k-ary $(\Sigma \cup \{\perp\})$-colored tree $\mathcal{T}'$, the application of *MSUnf* to $\mathcal{T}'$ is a full $(2k+1)$-ary $(\Sigma \cup \{\perp\})$-colored tree $\bar{\mathcal{T}}$, where each vertex is of the form $v = l_0 l_1 \ldots l_{n-1}$, with $l_i \in \{1, \ldots, k, \bar{1}, \ldots, \bar{k}, \#\}$, and it denotes a path $v_0 l_0 v_1 \ldots l_{n-1} v_n$ in the graph $\mathcal{T}$ extended with backward edges, loops and $\perp$-colored leaves such that $v_0 = \varepsilon$, $\mathcal{T}(v_i) \neq \perp$ for every $i < n$, and either $\bar{\mathcal{T}}(v) = \mathcal{T}(v_n)$ or $\bar{\mathcal{T}}(v) = \perp$ holds, depending on whether $v_n \in \mathcal{D}om(\mathcal{T})$ or not. Finally, given a residually regular factorization of $\mathcal{T}$, one can generalize the construction for the unfolding of the semi-infinite line to provide a residually regular factorization of $\bar{\mathcal{T}}$. Using such a correspondence, we prove that tree generators can be embedded into (infinite complete) residually regular trees. Such a result, along with the fact that residually regular trees capture also some trees outside the Caucal hierarchy, provides a good example of the expressiveness of our framework.

Theorem 5. *The class of residually regular colored trees is closed under MSOUnf.*

Proof. Given a residually regular tree $\mathcal{T}$, we show how to build a residually regular tree $\bar{\mathcal{T}}$ (i.e. the tree generated by applying *MSOUnf*), starting from a residually regular factorization of $\mathcal{T}$, applying *MSOUnf* to its factors, and then combining them to obtain $\bar{\mathcal{T}}$. Such an approach is somehow a generalization of the example of the unfolding of the semi-infinite line. We will extend the set of labels $[k]$ with fresh labels from $\{\bar{l} : l \in [k]\} \cup \{\#\}$. In order to prove the theorem, we shall use transfinite induction to show that, for every residually regular factorization of a k-ary Σ-colored tree $\mathcal{T}$, we can compute:

- a residually regular factorization of the $(2k+1)$-ary $(\Sigma \cup \{\perp\})$-colored tree $\bar{\mathcal{T}}$, with edges labeled over $\{1,\ldots,k,\bar{1},\ldots,\bar{k},\#\}$;
- a residually regular factorization of the $(2k+1)$-ary $(\Sigma \cup \{\perp, \widetilde{1},\ldots,\widetilde{k}\})$-colored tree $\widetilde{\mathcal{T}}$ obtained from $\bar{\mathcal{T}}$ by assigning a new color $\widetilde{l}$ to each $\perp$-colored leaf of the form $\#^n\bar{l}$, with $n \in \mathbb{N}$.

(The $\widetilde{l}$-colored leaves in $\widetilde{\mathcal{T}}$ will act as concatenation endpoints for backwards edges in the final tree.)

As a preliminary step to correctly manage concatenation endpoints, we need to show how one can recolor the factors of a generic tree $\mathcal{U}$ in such a way that the leaves involved in the concatenations have distinct colors whenever the entering edges have distinct labels. New colors will be of the form (c,l), with c being the original color and l being the label of the entering edge. So, let $\prod_{i\in\mathbb{N}}(\mathcal{V}_i)_{c_i}$ be a residually regular factorization of a k-ary Σ-colored tree $\mathcal{U}$. We define the infinite factorization $\mathcal{V}'_0 \cdot_{(c_0,1)} \mathcal{V}'_1 \cdot_{(c_0,2)} \cdots \cdot_{(c_0,k)} \mathcal{V}'_1 \cdot_{(c_1,1)} \mathcal{V}'_2 \cdot_{(c_1,2)} \cdots \cdot_{(c_1,k)} \mathcal{V}'_2 \cdot_{(c_2,1)} \cdots$, where, for every $c \in \Sigma$ and $l \in [k]$, (c,l) is a fresh color, and each $\mathcal{V}'_i$ is obtained from $\mathcal{V}_i$ by recoloring the leaves in such a way that, for every $v \in \mathcal{D}om(\mathcal{V}_i)$, $\mathcal{V}_i(v) = c$ iff either $\mathcal{V}'_i(v) = (c,l)$ or $\mathcal{V}'_i(v) = c$ holds, depending on whether $v = ul$ is a c-colored leaf with $c \in \{c_j : j \geq i\}$ and $l \in [k]$ or not. It is easy to see that the new factorization is residually regular and it generates exactly the tree $\mathcal{U}$.

Now, by exploiting the above observations, let $\prod_{i\in\mathbb{N}}(\mathcal{U}_i)_{(c_i,l_i)}$ be a residually regular factorization of a k-ary Σ-colored tree $\mathcal{T}$, where, for every $i \in \mathbb{N}$, l_i is the last label in a path from the root to a (c_i,l_i)-colored leaf of $\mathcal{U}_0 \cdot_{(c_0,l_0)} \ldots \cdot_{(c_{i-2},l_{i-2})} \mathcal{U}_{i-1}$. Note that the sequence $(l_i)_{i\in\mathbb{N}}$ is ultimately periodic. By transfinite induction on the structure of such a factorization, we can obtain, from each factor $\mathcal{U}_i$, a residually regular factorization of $\bar{\mathcal{U}}_i$ and $\widetilde{\mathcal{U}}_i$. Note that, if $\mathcal{U}_i$ is a finite full tree, $\bar{\mathcal{U}}_i$ is obtained by simply applying *MSOUnf* to $\mathcal{U}_i$. Similarly, $\widetilde{\mathcal{U}}_i$ is obtained from $\mathcal{U}_i$ by (i) introducing backward edges, loops and $\perp$-colored leaves, (ii) recoloring each leaf of the form $\#^n\bar{l}$ with the new color $\widetilde{l}$ (for any $n \in \mathbb{N}$ and $l \in [k]$), and (iii) by unfolding the graph from its root. Thus, since unfoldings of finite graphs generate regular trees, under the hypothesis of $\mathcal{U}_i$ being a finite full tree, both $\bar{\mathcal{U}}_i$ and $\widetilde{\mathcal{U}}_i$ turn out to be regular trees. Otherwise, if $\mathcal{U}_i$ is a residually regular tree, by inductive hypothesis, $\bar{\mathcal{U}}_i$ and $\widetilde{\mathcal{U}}_i$ are also residually regular trees (and we can recursively compute their factorizations). Now, we can build a residually regular factorization of $\bar{\mathcal{T}}$ and $\widetilde{\mathcal{T}}$. Such factorizations are respectively of the form $\prod_{i\in\mathbb{N}}(\bar{\mathcal{V}}_i)_{(c_i,l_i)}$ and $\prod_{i\in\mathbb{N}}(\widetilde{\mathcal{V}}_i)_{(c_i,l_i)}$, where:

- $\bar{\mathcal{V}}_0 = \bar{\mathcal{U}}_0$,
- $\bar{\mathcal{V}}_{i+1} = \widetilde{\mathcal{U}}_{i+1} \cdot_{\widetilde{l}_i} (\bar{\mathcal{V}}_i \cdot_{(c_i,l_i)} \widetilde{\mathcal{U}}_{i+1}) \cdot_{\widetilde{l}_i} (\bar{\mathcal{V}}_i \cdot_{(c_i,l_i)} \widetilde{\mathcal{U}}_{i+1}) \cdot_{\widetilde{l}_i} \ldots$,
- $\widetilde{\mathcal{V}}_0 = \widetilde{\mathcal{U}}_0$,

- $\widetilde{\mathcal{V}}_{i+1} = \widetilde{\mathcal{U}}_{i+1} \cdot_{\widetilde{l_i}} (\widetilde{\mathcal{V}}_i \cdot_{(c_i,l_i)} \widetilde{\mathcal{U}}_{i+1}) \cdot_{\widetilde{l_i}} (\widetilde{\mathcal{V}}_i \cdot_{(c_i,l_i)} \widetilde{\mathcal{U}}_{i+1}) \cdot_{\widetilde{l_i}} \ldots$.

By Theorem 3, the above factorizations are residually regular. We can verify by transfinite induction that $\prod_{i\in\mathbb{N}}(\bar{\mathcal{V}}_i)_{(c_i,l_i)}$ generates exactly $\bar{\mathcal{T}}$ and $\prod_{i\in\mathbb{N}}(\widetilde{\mathcal{V}}_i)_{(c_i,l_i)}$ generates exactly $\widetilde{\mathcal{T}}$. The proof is rather involved and we briefly sketch it in the following.

We first prove that $\widetilde{\mathcal{T}}$ features only (finite) paths in the extended tree $\mathcal{T}$ (a similar proof holds for $\bar{\mathcal{T}}$). So, let v be a c-colored vertex of $\widetilde{\mathcal{T}}$. There exist $k+1$ indexes $0 = i_0 < i_1 < \ldots < i_k$ and a vertex $v_{i_j} \in \mathcal{D}om(\widetilde{\mathcal{V}}_{i_j})$ for each index i_j such that $v = v_{i_0} \cdot v_{i_1} \cdot \ldots \cdot v_{i_k}$. The first factor v_{i_0} is a vertex of $\bar{\mathcal{U}}_0$, and hence by inductive hypothesis, it either denotes a path from the root to a $\widetilde{\mathcal{U}}_0(v_{i_0})$-colored vertex in the graph $\mathcal{U}_0$ extended with backward edges and loops, or it is of the form $x\bar{l}$, with x denoting a path in the extended graph $\mathcal{U}_0$ from the root to a vertex u without l-labeled entering edges. In the latter case, we easily verify that $v = v_{i_0}$ and hence x denotes a path in the extended tree $\mathcal{T}$ from the root to a vertex u such that $u[|u|] \neq l$. In the former case, we know that, for any $j > 0$, the word v_{i_j} can in turn be broken into a finite sequence of words $u_0^{(j)}, w_0^{(j)}, u_1^{(j)} \ldots$, with $u_h^{(j)} \in \mathcal{D}om(\widetilde{\mathcal{U}}_{i_j})$ and $w_h^{(j)} \in \mathcal{D}om(\widetilde{\mathcal{V}}_{i_j-1})$. By inductive hypothesis, we can assume that each word $u_h^{(j)}$ either denotes a path in the extended graph $\mathcal{U}_{i_j}$ from the root to a $\widetilde{\mathcal{U}}_{i_j}(u_h^{(j)})$-colored vertex or it is of the form $x \cdot \bar{l}$, with x denoting a path in the extended graph $\mathcal{U}_{i_j}$ from the root to a vertex u without l-labeled entering edges. In the latter case, $u_h^{(j)}$ would be the last factor of v. In the former case, if $u_h^{(j)}$ is not the last factor of v, then we have $u_h^{(j)}[|u_h^{(j)}|] = l_{i_j}$ and $\widetilde{\mathcal{U}}_{i_j}(u_h^{(j)}) = (c_{i_j}, l_{i_j})$. Similar properties hold for the factors $w_h^{(j)}$ (in this case, the paths spread over the extended graph $\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-2},l_{i_j-2})} \mathcal{U}_{i_j-1}$ starting from some vertices in $\mathcal{F}r(\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-3},l_{i_j-3})} \mathcal{U}_{i_j-2})$). Therefore, it turns out that each word $v_{i_j} = u_0^{(j)} w_0^{(j)} u_1^{(j)} \ldots$ either denotes a path in the extended graph $\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-2},l_{i_j-2})} \mathcal{U}_{i_j-1}$ starting from some vertex in $\mathcal{F}r(\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-3},l_{i_j-3})} \mathcal{U}_{i_j-2})$, or it is of the form $x\bar{l}$, with x denoting a path in the extended graph $\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-2},l_{i_j-2})} \mathcal{U}_{i_j-1}$ starting from some vertex in $\mathcal{F}r(\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-3},l_{i_j-3})} \mathcal{U}_{i_j-2})$ and reaching a vertex u without l-labeled entering edges (in the latter case, we know that v_{i_j} would the last factor of v). Again, this implies that v either denotes a path from the root in the extended graph $\mathcal{T}$ or it is of the form $x\bar{l}$ with x denoting a path in the extended graph $\mathcal{T}$ from the root to a vertex u without l-labeled entering edges. This proves that $\widetilde{\mathcal{T}}$ features only (finite) paths in the extended graph $\mathcal{T}$.

We now prove the converse, that is, all the (finite) paths in the extended graph $\mathcal{T}$ are featured by $\widetilde{\mathcal{T}}$ (and, similarly, by $\bar{\mathcal{T}}$). Let v be a path in the extended tree $\mathcal{T}$ starting from the root and reaching a c-colored vertex. We break v into a finite sequence of words $v_{i_0}, v_{i_1}, \ldots, v_{i_k}$ such that $0 = i_0 < i_1 < \ldots < i_k$, and each v_{i_j} denotes a path in the extended graph $\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-1},l_{i_j-1})} \mathcal{U}_{i_j}$ starting from some vertex in $\mathcal{F}r(\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-2},l_{i_j-2})} \mathcal{U}_{i_j-1})$

(i.e. the root if $j = 0$) and, if $j < k$, reaching some (c_{i_j}, l_{i_j})-colored leaf. For every $j > 0$, we can further break the path v_{i_j} into a finite sequence of words $u_0^{(j)}, \bar{l}_{i_j-1}, w_0^{(j)}, u_1^{(j)}, \bar{l}_{i_j-1}, \ldots$, with $u_h^{(j)}$ denoting a path (i.e. a loop if $u_h^{(j)}$ is not the last factor of v) from the root in the extended graph $\mathcal{U}_{i_j}$ and $w_h^{(j)}$ denoting a path in the extended graph $\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-2}, l_{i_j-2})} \mathcal{U}_{i_j-1}$ starting from some vertex in $\mathcal{F}r(\mathcal{U}_0 \cdot_{(c_0,l_0)} \mathcal{U}_1 \cdot_{(c_1,l_1)} \cdots \cdot_{(c_{i_j-2}, l_{i_j-2})} \mathcal{U}_{i_j-2})$ and, if $w_h^{(j)}$ is not the last factor of v, reaching a (c_{i_j-1}, l_{i_j-1})-colored vertex. By inductive hypothesis, each path $u_h^{(j)}\bar{l}_{i_j-1}$ can be seen as a vertex of $\widetilde{\mathcal{U}}_{i_j}$ which has the same color as the last vertex in the path $u_h^{(j)}$. We further know that, if $u_h^{(j)}$ is not the last factor of v, then $u_h^{(j)}\bar{l}_{i_j-1}$ is a $\widetilde{l}_{i_j-1}$-colored leaf of $\widetilde{\mathcal{U}}_{i_j}$. Similarly, each path $w_h^{(j)}$ can be seen as a vertex of $\widetilde{\mathcal{V}}_{i_j-1}$ which has the same color as the last vertex in the path $w_h^{(j)}$. Moreover, if $w_h^{(j)}$ is not the last factor of v, then $w_h^{(j)}$ is indeed a (c_{i_j-1}, l_{i_j-1})-colored leaf in the tree $\widetilde{\mathcal{V}}_{i_j-1}$. From such observations, it follows that v_{i_j} is vertex of $\widetilde{\mathcal{U}}_{i_j}$, and hence v is a vertex of $\widetilde{\mathcal{T}}$. Similarly, it follows that each (finite) path v in the extended tree $\mathcal{T}$ is a vertex of $\bar{\mathcal{T}}$. Finally, for every label $l \in [k]$ and for every finite path v from the root to a vertex u without l-labeled entering edges in the extended tree $\mathcal{T}$, $v\bar{l}$ is a $\widetilde{l}$-colored (resp. $\perp$-colored) vertex of $\widetilde{\mathcal{T}}$ (resp. $\bar{\mathcal{T}}$). $\square$

6 Conclusions

In this paper we devised an automaton-based method that allows us to solve the decision problem for the MSO theories of several tree structures. First, by taking advantage of well-known results from automata theory, we reduced such a problem to the acceptance problem for Rabin tree automata. Then, we introduced the class of residually regular trees, which extends that of regular trees, and we showed that one can solve the acceptance problem for this family of trees by reducing it to the acceptance problem for equivalent regular trees. We proved the effectiveness of the proposed method by applying it to some meaningful examples of tree structures. In particular, we provided an embedding of distinctive elements of the Caucal hierarchy, namely, tree generators, into our framework.

The proposed method generalizes the one developed by Carton and Thomas to decide the theory of the linear order $(\mathbb{N}, <)$ extended with suitable unary relations, that is, those relations which are encoded by residually ultimately periodic words [7]. Since any ω-word over a finite alphabet Σ can be seen as an infinite 1-ary Σ-colored tree and string concatenation is definable in terms of tree concatenation, the notion of residually regular tree subsumes that of residually ultimately periodic word. Furthermore, some interesting binary relations over the linear order $(\mathbb{N}, <)$ turn out to be MSO definable in terms of residually regular trees. As an example, in [21, 22] we give an alternative decidability proof for the theory of the linear order $(\mathbb{N}, <)$ extended with the flip function [19, 26]. Finally, many trees in the Caucal hierarchy [9] can be embedded into residually regular trees. This last fact hints at the possibility of establishing a connection between our approach and the transformational one developed by Caucal for deciding MSO theories of infinite graphs.

We are currently trying to determine whether or not any deterministic tree in the Caucal hierarchy can be embedded into a suitable residually regular tree. We already know that the converse does not hold, since there exist some deterministic trees, such as, for instance, $\mathcal{T}_{tow}$ (cf. Section 3.4), which do not belong to the Caucal hierarchy, but can be handled by our method. We are also investigating the possibility of extending the proposed automaton-based framework to manage non-deterministic trees. To this end, we are looking for a more general notion of tree equivalence based on more expressive automata, such as, for instance, the automata on tree-like structures used in [27].

References

[1] K. Barthelmann. On equational simple graphs. Technical Report 9, Universität Mainz, Institut für Informatik, 1997.

[2] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science*, pages 1–11. Stanford University Press, 1960.

[3] A. Blumensath. Prefix-recognizable graphs and monadic second-order logic. Technical Report AIB-06-2001, RWTH Aachen, 2001.

[4] A. Blumensath and E. Gradel. Automatic structures. In *Logic in Computer Science*, pages 51–62, 2000.

[5] T. Cachat. Higher order pushdown automata, the Caucal hierarchy of graphs and parity games. In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming*, volume 2719 of *LNCS*, pages 556–569, 2003.

[6] A. Carayol and S. Wöhrle. The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata. In *Proceedings of the 23rd Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 2914 of *LNCS*, pages 112–123. Springer, 2003.

[7] O. Carton and W. Thomas. The monadic theory of morphic infinite words and generalizations. *Information and Computation*, 176:51–65, 2002.

[8] D. Caucal. On the regular structure of prefix rewriting. *Theoretical Computer Science*, 106:61–86, 1992.

[9] D. Caucal. On infinite terms having a decidable monadic theory. In *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science*, volume 2420 of *LNCS*, pages 165–176. Springer, 2002.

[10] D. Caucal. On infinite transition graphs having a decidable monadic theory. *Theoretical Computer Science*, 290:79–115, 2003.

[11] B. Courcelle. The monadic second-order logic of graphs II: Infinite graphs of bounded tree width. *Mathematical Systems Theory*, 21:187–221, 1989.

[12] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 193–242. Elsevier, 1990.

[13] B. Courcelle. Monadic second-order graph transductions: a survey. *Theoretical Computer Science*, 126:53–75, 1994.

[14] B. Courcelle and I. Walukiewicz. Monadic second-order logic, graph coverings, and unfoldings of transition systems. *Annals of Pure and Applied Logic*, 92:35–62, 1998.

[15] C.C. Elgot and M.O. Rabin. Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. *Journal of Symbolic Logic*, 31(2):169–181, 1966.

[16] M. Franceschet and A. Montanari. Temporalized logics and automata for time granularity. *Theory and Practice of Logic Programming, Special Issue on Verification and Computational Logic*, 4(5-6), 2004. To appear.

[17] A. Montanari. *Metric and Layered Temporal Logic for Time Granularity*. ILLC Dissertation Series 1996-02. Institute for Logic, Language and Computation, University of Amsterdam, 1996.

[18] A. Montanari, A. Peron, and A. Policriti. Theories of ω-layered metric temporal structures: Expressiveness and decidability. *Logic Journal of IGPL*, 7(1):79–102, 1999.

[19] A. Montanari, A. Peron, and A. Policriti. Extending Kamp's theorem to model time granularity. *Journal of Logic and Computation*, 12(4):641–678, 2002.

[20] A. Montanari and A. Policriti. Decidability results for metric and layered temporal logics. *Notre Dame Journal of Formal Logic*, 37(2):260–282, 1996.

[21] A. Montanari and G. Puppis. Decidability of MSO theories of tree structures. Research Report 01, Dipartimento di Matematica e Informatica, Università di Udine, Italy, 2004.

[22] A. Montanari and G. Puppis. Decidability of the theory of totally unbounded ω-layered structures. In *Proceedings of the 11th International Symposium in Temporal Representation and Reasoning (TIME)*, pages 156–160, 2004.

[23] D. Muller and P. Schupp. The theory of ends, pushdown automata, and second-order logics. *Theoretical Computer Science*, 37:51–75, 1985.

[24] M.O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[25] W. Thomas. Languages, automata, and logic. In G. Rozemberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 389–455. Springer, 1997.

[26] W. Thomas. Constructing infinite graphs with a decidable MSO-theory. In *Proceedings of the International Symposium on Mathematical Foundations of Computer Science*, LNCS, pages 113–124. Springer, 2003.

[27] I. Walukiewicz. Monadic second-order logic on tree-like structures. *Theoretical Computer Science*, 275:311–346, 2002.

[28] G. Zhang. Automata, boolean matrices, and ultimate periodicity. *Information and Computation*, 152(1):138–154, 1999.