UNIVERSITÀ
DEGLI STUDI
DI UDINE

Università degli studi di Udine

Decidability of MSO Theories of Tree Structures

Original

Availability:
This version is available http://hdl.handle.net/11390/856753 since 2024-07-26T13:55:56Z

Publisher:
Springer Verlag

Published
DOI:10.1007/b104325

Terms of use:
The institutional repository of the University of Udine (http://air.uniud.it) is provided by ARIC services. The aim is to enable open access to all the world.

Publisher copyright

(Article begins on next page)

14 September 2026

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 deterministic 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 deterministic 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 [20]. 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 [19], 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 operations [23], 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]. As a matter of fact, 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 [22], 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 deterministic 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. Successively, we show that the proposed technique can be used 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

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 [21]. 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 $c \in \Sigma$ is 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 $v \in V_i$.

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. The *frontier* $\mathcal{F}r(\mathcal{T})$ of $\mathcal{T}$ is the *prefix-free* language $\{u \in \mathcal{D}om(\mathcal{T}) : \forall\, l \in [k].\ ul \notin \mathcal{D}om(\mathcal{T})\}$. 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 $\mathcal{B}ch(\mathcal{T})$.

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 $L_i, U_i \subseteq S$ [20]. 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 $(\mathcal{R}(u), \mathcal{T}(u), \mathcal{R}(u1), \ldots, \mathcal{R}(uk)) \in E$ for every $u \in \mathcal{D}om(\mathcal{R})$. 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.

3 An Automaton-based Approach to Decidability

In this section, we develop an automaton-based method to decide MSO theories of infinite (complete) deterministic 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 [20] 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, for any given Rabin tree automaton, whether it 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 non-isomorphic 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 introduce some tools for tree manipulation [17] (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, $U \subseteq \mathcal{F}r(\mathcal{T})$, 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 node $u \in U$ by $\mathcal{R}_u$.*

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 $U = \{u \in \mathcal{F}r(\mathcal{T}_1) : \mathcal{T}_1(u) = c\}$.*

It is not difficult to show that the operator $\cdot_c$ is not associative. We assume that it 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 *ultimately periodic* 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 ultimately periodic factorizations to regular trees [17].

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

From Proposition 1, it immediately follows that $Acc(\mathcal{T})$ is decidable for any infinite (complete) deterministic tree $\mathcal{T}$ generated by an ultimately periodic factorization.

3.1 Residually Regular Trees

We now show how to reduce the acceptance problem for a large class of infinite (complete) deterministic 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) partial run of a Rabin tree automaton $M = (S, I, E, AP)$: a *partial run* of M on a full, finite or infinite, (Σ-colored) tree $\mathcal{T}$ is a full (S-colored) tree $\mathcal{P}$ such that (i) $\mathcal{D}om(\mathcal{P}) = \mathcal{D}om(\mathcal{T})$ and (ii) $(\mathcal{P}(v), \mathcal{T}(v), \mathcal{P}(v1), \ldots, \mathcal{P}(vk)) \in E$, for every $v \in \mathcal{D}om(\mathcal{P}) \setminus \mathcal{F}r(\mathcal{P})$.

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 partial run $\mathcal{P}_1$ of M on $\mathcal{T}_1$, there exists a partial run $\mathcal{P}_2$ of M on $\mathcal{T}_2$ (and vice versa) such that*

1. *$\mathcal{T}_1(\varepsilon) = \mathcal{T}_2(\varepsilon)$ and $\mathcal{P}_1(\varepsilon) = \mathcal{P}_2(\varepsilon)$;*
2. *for every $v \in \mathcal{F}r(\mathcal{T}_1)$, there exists $u \in \mathcal{F}r(\mathcal{T}_2)$ such that $\mathcal{T}_1(v) = \mathcal{T}_2(u)$, $\mathcal{P}_1(v) = \mathcal{P}_2(u)$, and $\mathcal{I}mg(\mathcal{P}_1|v)) = \mathcal{I}mg(\mathcal{P}_2|u)$, 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 $\mathcal{I}nf(\mathcal{P}_1|v) = \mathcal{I}nf(\mathcal{P}_2|u)$, and vice versa.*

The equivalence $\equiv_M$ satisfies the following properties [17].

Theorem 1. *It holds that:*

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 and parity tree automata.

Taking advantage of Theorem 1, we identify a large class of deterministic trees, that we call residually regular trees, whose acceptance problem is decidable. We say that an infinite sequence $\mathcal{S} = \mathcal{T}_0 \mathcal{T}_1 \mathcal{T}_2 \ldots$ of finite full trees is 1-residually ultimately periodic if, for every Rabin tree automaton M, one can compute an ultimately periodic sequence $\mathcal{S}' = \mathcal{T}'_0 \mathcal{T}'_1 \mathcal{T}'_2 \ldots$ of finite trees such that $\mathcal{T}_i \equiv_M \mathcal{T}'_i$, for all i. We call 1-residually regular trees those trees that are obtained by concatenating the trees in a 1-residually ultimately periodic sequence. The notion of 1-residually ultimately periodic factorization can be extended to level n, with n being any countable ordinal, by no longer considering finite trees but level $n' < n$ residually regular trees. For every countable 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 a countable ordinal n, a factorization $\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots$ is* n-residually ultimately periodic *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 ultimately periodic 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 ultimately periodic factorization. A* residually ultimately periodic factorization *is an n-residually ultimately periodic factorization, for some countable ordinal n. A* residually regular tree *is a tree enjoying a residually ultimately periodic factorization.*

It is worth noticing that the above definition allows residually ultimately periodic factorizations to encompass residually regular factors of any arbitrary level. For instance, we can start with some factors $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots$ which respectively are level $1, 2, 3, \ldots$ 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 ultimately periodic factorization. Precisely, an $\equiv_M$-regular form of a 1-residually ultimately periodic 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 countable ordinal $n > 1$, an $\equiv_M$-regular form of an n-residually ultimately periodic 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 ultimately periodic 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 ultimately periodic 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 $\prod_{i\in\mathbb{N}}(\mathcal{T}_i)_{c_i}$. Furthermore, the factorization of an $\equiv_M$-regular form is computable from a given n-residually ultimately periodic factorization $\mathcal{S}$, which can be finitely represented by a function mapping an integer $i \in \mathbb{N}$ to (an n'-residually, with $n' < n$, ultimately periodic factorization of) the i-th factor of $\mathcal{S}$. Hence, we have the following theorem [17].

Theorem 2. *Let $\mathcal{T}$ be the infinite (complete) deterministic tree resulting from an n-residually ultimately periodic 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 $\mathcal{T} \in \mathscr{L}(M)$ iff $\mathcal{T}' \in \mathscr{L}(M)$.*

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

3.2 Properties of Residually Ultimately Periodic Factorizations

We now identify some structural properties that allow us to easily build residually ultimately periodic factorizations. The resulting framework somehow generalizes previous results by Zhang [24] 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 ultimately 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 a ultimately periodic sequence.*

We say that a function is *effectively* residually ultimately 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 ultimately periodic. From now on, we restrict our attention to effectively residually ultimately periodic functions, which can be characterized as follows.

Proposition 2. *A function $f : \mathbb{N} \to \mathbb{N}$ is (effectively) residually ultimately periodic iff for all $l \geq 0$ and $r > 0$, one can compute $p \geq 0$ and $q > 0$ such that $[f(n)]_{l,r} = [f([n]_{p,q})]_{l,r}$.*

Definition 6. *A function $f : \mathbb{N} \to \mathbb{N}$* has unbounded infimum *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 ultimately periodic functions [17]. Examples of generated functions are n^2, 2^n, $2^n - n^2$, n^n, $n!$, and the exponential tower $2^{2^{\cdot^{\cdot^{2}}}}$.

Theorem 3. *Let f and g be residually ultimately periodic functions. The following functions are residually ultimately 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. **(Generalized sum)** h *defined by* $h(n) = \sum_{i=0}^{n-1} f(i)$;
8. **(Generalized product)** h *defined by* $h(n) = \prod_{i=0}^{n-1} f(i)$;
9. **(Substitution)** $g \circ f$.

The next theorem shows how one can combine (colored) trees to obtain residually ultimately periodic factorizations [17]. In particular, case *1.* links residually ultimately periodic functions to residually ultimately periodic trees; case *2.* states that by interleaving the factors of residually ultimately periodic factorizations we obtain again a residually ultimately periodic 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 ultimately periodic factorization.

Theorem 4. *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 ultimately periodic 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 ultimately 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 ultimately periodic factorization;*
3. **(Grouping)** *if there is $q > 0$ and there is a residually ultimately periodic 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 ultimately periodic factorization $\prod_{j\in\mathbb{N}}(\mathcal{U}_j)_c$ such that $\mathcal{T}_{i+1} = \mathcal{U}_i \cdot_c (\mathcal{T}_i \cdot_d \mathcal{U}_i) \cdot_c (\mathcal{T}_i \cdot_d \mathcal{U}_i) \cdot_c \ldots$.*

4 Some Applications of the Proposed Method

In the following, we apply the proposed method to decide the theories of two meaningful tree structures. Futhermore, we provide an embedding of some representative graphs of the so-called Caucal hierarchy [9], namely, tree generators for MSO interpretations, into our framework. We first recall the basic notions of unfolding and MSO interpretation [22].

Definition 7. *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 $v_0l_0v_1\ldots l_{n-1}v_n$, and E'_l is the set of all the pairs of paths of the form $(v_0l_0v_1\ldots l_{n-1}v_n, v_0l_0v_1\ldots l_{n-1}v_nlv_{n+1})$.*

Notice that the unfolding of any finite graph is (isomorphic to) a regular tree. Moreover, unfoldings from MSO definable vertices preserve the decidability of the MSO theories of graph structures [14]. Another transformation which preserves decidability is the MSO interpretation, which is defined as follows.

Definition 8. *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.*

As a first example of application of our approach, 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.

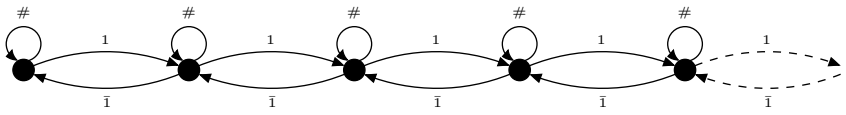

Fig. 1. The semi-infinite line $\mathcal{L}$.

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. By adding an auxiliary third color c to manage concatenation, 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 4 (case *4.*), it follows that such a factorization is residually ultimately periodic. This accounts for the decidability of $MTh(\mathcal{T})$.

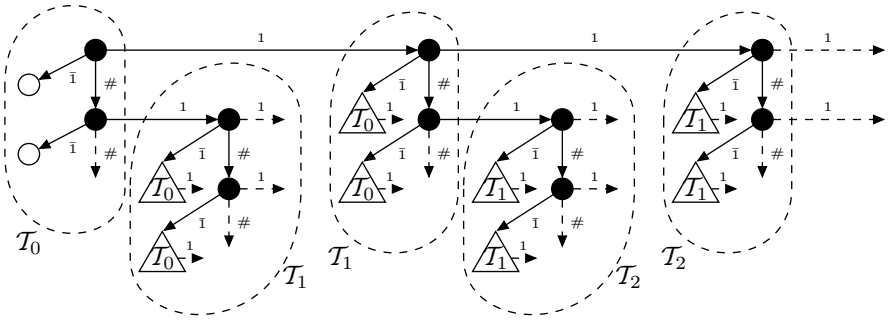

Fig. 2. The tree $\mathcal{T}$ embedding the unfolding of $\mathcal{L}$.

As a second example, consider the infinite binary $\{w, b\}$-colored tree $\mathcal{T}_{tow}$ such that $\mathcal{T}_{tow}(u) = b$ iff $u = 1^n 0^m$, 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 ultimately periodic 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 4 (case *1.*), it follows that such a factorization is residually ultimately periodic, and thus $MTh(\mathcal{T}_{tow})$ is decidable.

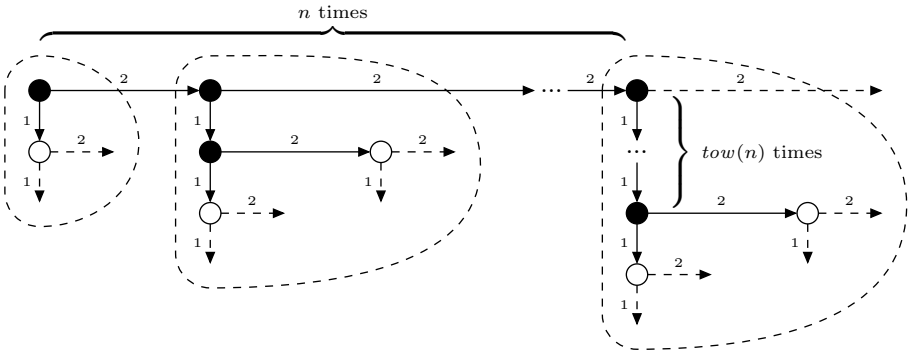

Fig. 3. The residually regular tree $\mathcal{T}_{tow}$.

We conclude the section by showing that residually regular trees allow us to capture a relevant subclass of the graphs in the Caucal hierarchy [9]. 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. 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., $\overline{1}, \overline{2}, \#$), and then by unfolding the resulting graph from its root. $\mathcal{C}_3$ 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. Tree generators can be embedded into residually regular trees as follows. 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* [17]. 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.

5 Conclusions

In this paper we devised an automaton-based method that allows us to solve the decision problem for the MSO theories of several deterministic tree structures. First, by taking advantage of well-known results from automata theory, we reduced the 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. Finally, we applied the proposed method to some meaningful examples of tree structures.

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 [17, 18] we give an alternative decidability proof for the theory of the linear order $(\mathbb{N}, <)$ extended with the flip function [16, 22]. 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 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 [23].

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] 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.

[17] 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.

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

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

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

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

[22] 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.

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

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