Decidability of MSO Theories of Deterministic Tree Structures

Gabriele Puppis

puppis@dimi.uniud.it

(joint work with Angelo Montanari)

Department of Mathematics and Computer Science
University of Udine, Italy

Outline

- MSO logics over tree structures
- The automaton-based approach
- Reduction to acceptance of regular trees
- Structural properties
- Application examples
- Further work

MSO Logics over tree structures (1)

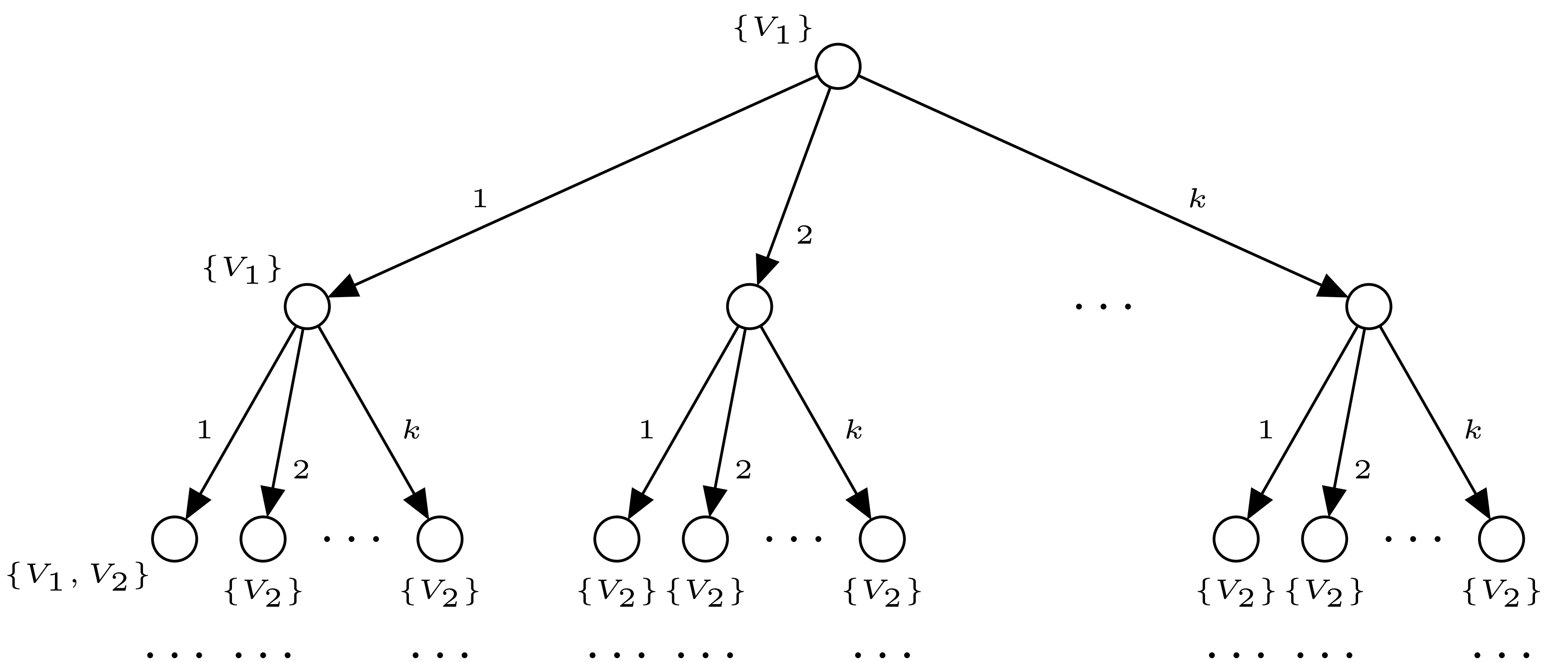
Let $\Lambda = \{1, \ldots, k\}$ be a finite set of **edge labels**.

We consider **infinite deterministic trees** extended with tuples of **unary predicates**: $(\mathcal{T}, \bar{V}) = (\Lambda^*, (E_l)_{l \in \Lambda}, (V_i)_{i \in [1,m]})$

Example.

$V_1 = \{\varepsilon, 1, 11, 111, \ldots\}$

$V_2 = \{11, 12, \ldots, 1k, 21, 22, \ldots, 2k, \ldots\}$



MSO Logics over tree structures (2)

MSO formulas over a tree $\mathcal{T}$ are built up from atoms:

- $E_l(X_i, X_j)$ "X_i, X_j denote singletons $\{u\}, \{v\}$ with (u, v) being an l-labeled edge"
- $X_i \subseteq X_j$ "X_i denotes a subset of X_j"

...through connectives $\vee$, $\neg$ and quantifier $\exists$ over variables.

- Each free variable X_i in a formula $\varphi(\bar{X})$ is interpreted by a designated subset V_i.
- $\mathcal{T} \models \varphi[\bar{V}]$ iff $\varphi(\bar{X})$ holds in $\mathcal{T}$ by interpreting V_i for X_i, for all i.

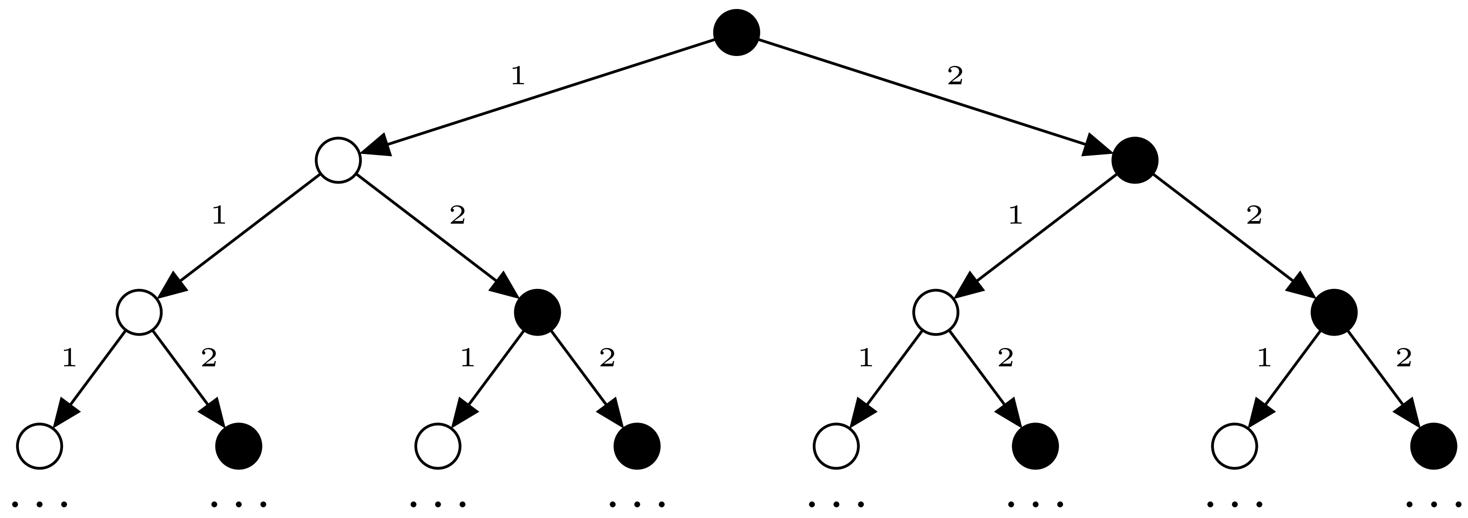
The **model-checking problem** for $(\mathcal{T}, \bar{V})$ is to decide whether $\mathcal{T} \models \varphi[\bar{V}]$, for any given formula $\varphi(\bar{X})$.

MSO Logics over tree structures (3)

Example. The formula

$$\varphi(X) = X(\varepsilon) \;\wedge\; \forall\, x.\, \exists\, y.\, (X(x) \;\wedge\; E_2(x,y) \;\rightarrow\; X(y))$$

holds in the binary tree extended with the predicate V represented by black colored vertices:



Remark. We identify a tree structure $(\mathcal{T}, \bar{V})$ with its **canonical representation** $\mathcal{T}_{\bar{V}}$ (i.e. an infinite complete *vertex-colored* tree).

The automaton based approach (1)

We consider **tree automata** accepting colored trees in a *top-down* fashion:

- they 'spread' states inside the input tree (in accordance to transition relations),
- they 'verify' that suitable acceptance conditions (envisaging occurrences of states) are satisfied for each path in the tree.

We write $\mathcal{T}_{\bar{V}} \in \mathscr{L}(M)$ to say that the tree $\mathcal{T}_{\bar{V}}$ is accepted by automaton M.

Example. (**Rabin acceptance condition**)
Given $AC = \{(L_1, U_1), \ldots, (L_n, U_n)\}$, we require that, for each infinite path, there is a pair $(L_i, U_i) \in AC$ such that *at least one state in* U_i, but *no state in* L_i, is visited infinitely often.

The automaton based approach (2)

Step 1. We reduce the *model checking problem* to an **acceptance problem** by exploiting the correspondence between MSO formulas over tree structures and Rabin tree automata.

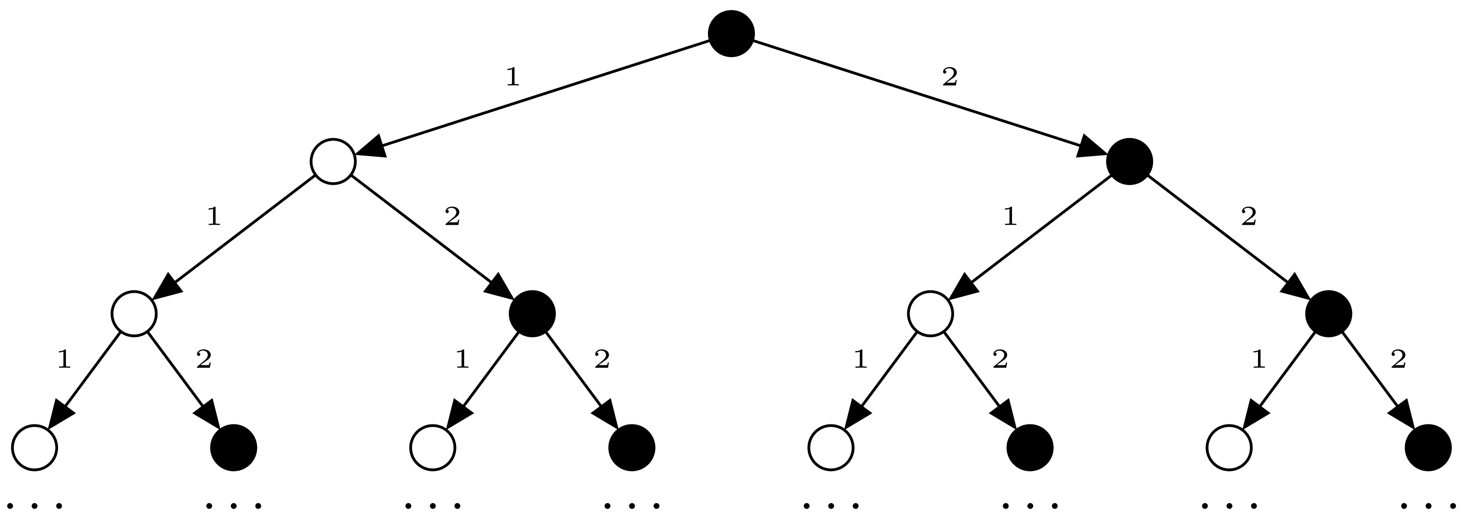
[Rabin '69]
For every formula $\varphi(\bar{X})$, there is a Rabin tree automaton M (and vice versa) such that for every tree structure $(\mathcal{T}, \bar{V})$

$$\mathcal{T} \models \varphi[\bar{V}] \quad \Leftrightarrow \quad \mathcal{T}_{\bar{V}} \in \mathscr{L}(M)$$

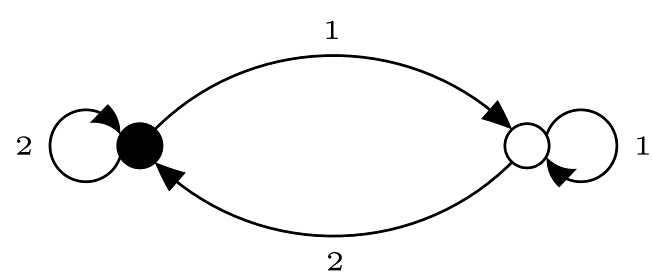
$\Rightarrow$ the decision problem for $MTh(\mathcal{T}, \bar{V})$ reduces to the acceptance problem $Acc(\mathcal{T}_{\bar{V}})$ for Rabin tree automata.

The automaton based approach (3)

Remark. The problem $Acc(\mathcal{T}_{\bar{V}})$ *can be decided* for any **regular tree** $\mathcal{T}_{\bar{V}}$ (i.e. a tree with only finitely many distinct subtrees)...



...by simply considering the intersection with the tree automaton generating $\mathcal{T}_{\bar{V}}$...



The automaton based approach (4)

Step 2. We extend the class of trees for which the acceptance problem turns out to be decidable.

Idea. Given an automaton M, we define an **equivalence** $\cong_M$ that groups together those (finite or infinite) trees on which M *'behaves' in a similar way*.

In particular, for two infinite complete trees $\mathcal{T}, \mathcal{T}'$,
$\mathcal{T} \cong_M \mathcal{T}'$ will imply $\mathcal{T} \in \mathscr{L}(M) \Leftrightarrow \mathcal{T}' \in \mathscr{L}(M)$.

Fact. Many non-regular trees turn out to be equivalent to some (computable) regular trees.

$\Rightarrow$ in such cases we will be able to solve $Acc(\mathcal{T})$ by reducing it to the decidable problem $Acc(\mathcal{T}')$

A digression into Büchi automata (1)

Given a Büchi automaton M, we can define an equivalence $\cong_M$ over finite words s.t. $u \cong_M u'$ iff, for every pair of states r, s,

- $r \xrightarrow{u} s \quad \Leftrightarrow \quad r \xrightarrow{u'} s$
- $r \xrightarrow{u}_{\circ} s \quad \Leftrightarrow \quad r \xrightarrow{u'}_{\circ} s$

Properties:

- $\cong_M$ has *finite index*
- $\cong_M$ is a *congruence* w.r.t. concatenation
- $\cong_M$-equivalent factorizations are *indistinguishable* by M, namely, if $u_i \cong_M u'_i$ for all $i \geq 0$, then

$$u_0 u_1 u_2 \ldots \in \mathscr{L}(M) \quad \Leftrightarrow \quad u'_0 u'_1 u'_2 \ldots \in \mathscr{L}(M)$$

A digression into Büchi automata (2)

[Elgot, Rabin, Carton and Thomas...]

Let w be an infinite word.

If we can provide a factorization $u_0 \cdot u_1 \cdot u_2 \ldots$ of w such that, *for any congruence* $\cong_M$ there are p, q *computable* such that $\forall\, i > p.\ u_i \cong_M u_{i+q}$

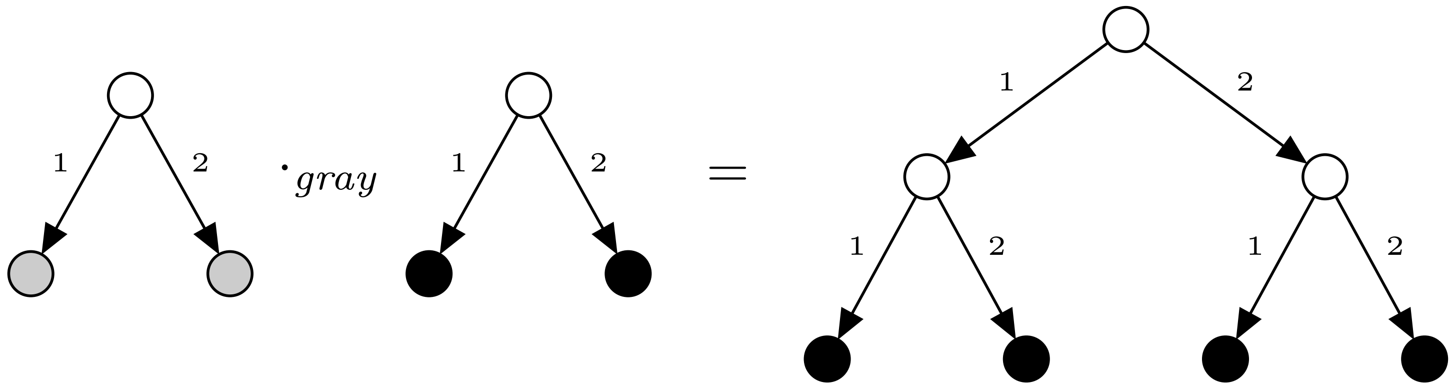
Then:

$$w \in \mathscr{L}(M)$$
$$\Updownarrow$$
$$(u_0 \ldots u_p)(u_{p+1} \ldots u_{p+q})(u_{p+q+1} \ldots u_{p+2q}) \ldots \in \mathscr{L}(M)$$
$$\Updownarrow$$
$$(u_0 \ldots u_p)(u_{p+1} \ldots u_{p+q})(u_{p+1} \ldots u_{p+q}) \ldots \in \mathscr{L}(M)$$
$$\Updownarrow$$
$$(u_0 \ldots u_p) \cdot (u_{p+1} \ldots u_{p+q})^\omega \in \mathscr{L}(M)$$

$\Rightarrow$ we can decide whether M accepts w.

A Reduction to acceptance of regular trees

We define the **tree concatenation** $\mathcal{T}_1 \cdot_c \mathcal{T}_2$ of two (finite or infinite) trees $\mathcal{T}_1$, $\mathcal{T}_2$ as the *substitution of all the c-colored leaves in $\mathcal{T}_1$ by $\mathcal{T}_2$*:



The notion can be extended to *infinite* sequences of trees, henceforth called **factorizations** (e.g. $\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \mathcal{T}_2 \cdot_{c_2} \ldots$).

Proposition. Any **ultimately periodic** factorization consisting of **only regular trees** generates a *regular* tree.

The notion of equivalence

Given an automaton M and a (finite or infinite) tree $\mathcal{T}$, we need to quantify over all the possible **partial runs** of M on $\mathcal{T}$ (i.e. 'run fragments').

Definition. $\mathcal{T}_1 \cong_M \mathcal{T}_2$ iff
$\forall$ partial run $\mathcal{P}_1$ on $\mathcal{T}_1$, $\exists$ a partial run $\mathcal{P}_2$ on $\mathcal{T}_2$ (and vice versa)
such that for $i = 1$ and $i = 2$ we have the *same*

- pair $(\mathcal{T}_i(\varepsilon), \mathcal{P}_i(\varepsilon))$
  (color and state at the *root*)
- set $\{(\mathcal{T}_i(u), \mathcal{P}_i(u))_u\}_{u \text{ leaf}}$
  (pairs color-state at the *frontier*)
- set $\{\mathcal{I}mg(\mathcal{P}_i|\pi)\}_{\pi \text{ fin. path}}$
  (sets of states occurring along *finite full paths*)
- set $\{\mathcal{I}nf(\mathcal{P}_i|\pi)\}_{\pi \text{ inf. path}}$
  (sets of states occurring infinitely often along *infinite paths*)

Properties of $\cong_M$

Properties:

- $\cong_M$ has *finite index*
- $\cong_M$ is a *congruence* w.r.t. concatenations
  namely, if $\mathcal{T}_1 \cong_M \mathcal{T}_1'$ and $\mathcal{T}_2 \cong_M \mathcal{T}_2'$, then
  $$\mathcal{T}_1 \cdot_c \mathcal{T}_2 \cong_M \mathcal{T}_1' \cdot_c \mathcal{T}_2'$$
- $\cong_M$-equivalent factorizations are *indistiguishable* by M
  namely, if $\mathcal{T}_i \cong_M \mathcal{T}_i'$ for all $i \geq 0$, then
  $$\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots \in \mathscr{L}(M) \;\Leftrightarrow\; \mathcal{T}_0' \cdot_{c_0} \mathcal{T}_1' \cdot_{c_1} \ldots \in \mathscr{L}(M)$$

The key ingredient

Let $\mathcal{T}$ be an infinite complete tree.

If we can provide a factorization $\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots$ of $\mathcal{T}$ such that,
for any congruence $\cong_M$ there are p, q *computable* such that
$\forall\, i > p.\ \mathcal{T}_i \cong_M \mathcal{T}_{i+q}$

Then:

$$\mathcal{T} \in \mathscr{L}(M)$$
$$\Updownarrow$$
$$\mathcal{T}_0 \cdot_{c_0} ...\mathcal{T}_p \cdot_{c_p} \mathcal{T}_{p+1} \cdot_{c_{p+1}} ...\mathcal{T}_{p+q} \cdot_{c_{p+q}} \mathcal{T}_{p+q+1} \cdot_{c_{p+q+1}} ... \in \mathscr{L}(M)$$
$$\Updownarrow$$
$$\mathcal{T}_0 \cdot_{c_0} ...\mathcal{T}_p \cdot_{c_p} \mathcal{T}_{p+1} \cdot_{c_{p+1}} ...\mathcal{T}_{p+q} \cdot_{c_{p+q}} \mathcal{T}_{p+1} \cdot_{c_{p+q+1}} ... \in \mathscr{L}(M)$$

Remark. The last factorization is ultimately periodic,
$\Rightarrow$ it generates a (decidable) *regular* tree $\mathcal{T}'$
provided that $\mathcal{T}_0, \mathcal{T}_1, \ldots$ are **regular** trees.

Residually regular trees

Definition. Residually regular trees are defined as follows:

- A tree $\mathcal{T}$ is **level** 1 **residually regular tree** if we can provide a factorization $\mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots$ (with $\mathcal{T}_0, \mathcal{T}_1, \ldots$ *regular* trees) which is *effectively ultimately periodic w.r.t. any congruence* $\cong_M$.

- We extend the notion to **level** $n > 1$ (for n countable ordinal) by allowing the factors to be *level* $n' < n$ *residually regular trees*.

$\Rightarrow$ this gives rise to a hierarchy that is *strictly increasing* at least for the initial (finite ordinal) levels.

The main result

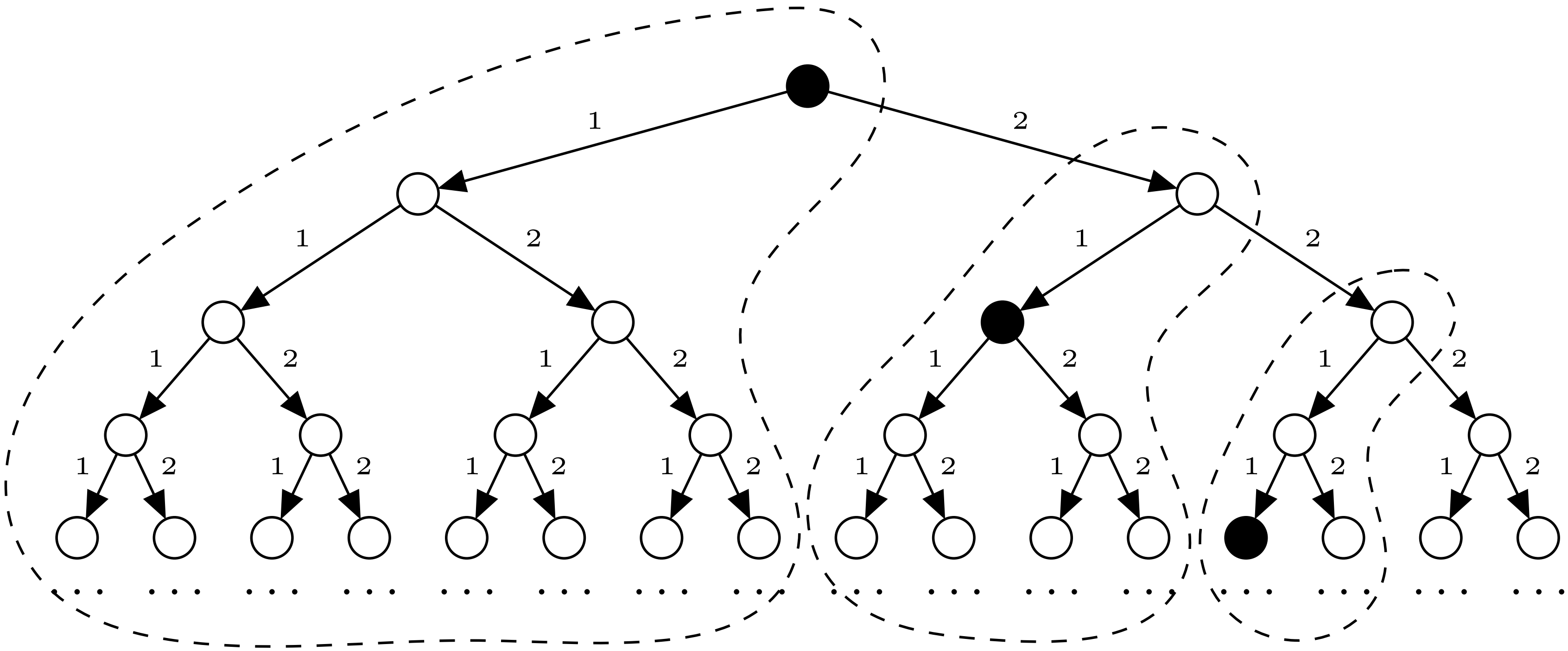
Theorem. $MTh(\mathcal{T}, \bar{V})$ is decidable for every residually regular tree $\mathcal{T}_{\bar{V}}$.

Proof sketch. We decide $MTh(\mathcal{T}, \bar{V})$ as follows:

1. let $\mathcal{S} = \mathcal{T}_0 \cdot_{c_0} \mathcal{T}_1 \cdot_{c_1} \ldots$ be a *level* n *residually ultimately periodic factorization* for $\mathcal{T}_{\bar{V}}$
2. given a formula φ, let M be the corresponding automaton
3. compute the *prefix* p and the *period* q of $\mathcal{S}$ w.r.t. $\cong_M$
4. using induction on n, compute the *ultimately periodic* factorization $\mathcal{S}'$ consisting of only *regular* trees
5. compute the *regular* tree $\mathcal{T}'$ resulting from $\mathcal{S}'$
6. solve $Acc(\mathcal{T}')$ on automaton M
7. accordingly, return Yes or No to the original problem $MTh(\mathcal{T}, \bar{V})$

Structural properties (1)

Residually regular trees are in general non-regular trees which however exhibit a definite pattern in their structure.



- We established some **structural properties** of residually regular trees, such as closure under *recursively defined factorizations*, *iterations*, *periodical groupings*, etc.

Structural properties (2)

Any congruence of finite index $\cong_M$ induces an **homomorphism** from the set T of trees **to a finite groupoid** $(T_{/\cong_M}, \cdot_c)$.

$\Rightarrow$ we can exploit structural properties of finite groupoids (e.g. *Pigeonhole Principle*) to provide residually regular factorizations.

Example. Let $\mathcal{T}$ be a finite tree and recursively define $\mathcal{T}_i$ as $\mathcal{T}_0 = \mathcal{T}$ and $\mathcal{T}_{i+1} = \mathcal{T}_i \cdot_c \mathcal{T}$ for each $i \geq 0$. Then

- for any congruence $\cong_M$, the sequence

  $$[\mathcal{T}_0]_{\cong_M}, [\mathcal{T}_1]_{\cong_M}, [\mathcal{T}_2]_{\cong_M}, \ldots$$

  is (effectively) *ultimately periodic*.
- the tree $\mathcal{T}' = \mathcal{T}_0 \cdot_d \mathcal{T}_1 \cdot_d \mathcal{T}_2 \cdot_d \ldots$
  is *residually regular* and enjoys a decidable theory.

Structural properties (3)

Other examples of structural properties:

- **[Iteration]** Let $\mathcal{T}$ be a residually regular tree. Then the sequence $(\mathcal{T}^{f(i)+1})_{i\in\mathbb{N}}$ is residually periodic provided that f is a '*well behaved*' function (e.g. $f(n) = n^2$, $f(n) = n!$, $f(n) = Fib(n)$, $f(n) = 2^{2^{\cdots^2}}$, etc.)
- **[Grouping]** Let $\mathcal{T}_0 \cdot_c \mathcal{T}_1 \cdot_c \mathcal{T}_2 \cdot_c \mathcal{T}_3 \cdot_c \ldots$ be a residually periodic factorization. Then we can generate another residually regular factorization by periodically grouping the factors, e.g., $(\mathcal{T}_0 \cdot_c \mathcal{T}_1) \cdot_c (\mathcal{T}_2 \cdot_c \mathcal{T}_3) \cdot_c \ldots$
- **[Interleaving]** Let $\mathcal{T}_0^{(j)} \cdot_c \mathcal{T}_1^{(j)} \cdot_c \mathcal{T}_2^{(j)} \cdot_c \ldots$ be a family of residually periodic factorizations, for $j \in [1, n]$. Then we can generate another residually periodic factorization by periodically interleaving the factors from each sequence, e.g., $\mathcal{T}_0^{(1)} \cdot_c \mathcal{T}_0^{(2)} \cdot_c \mathcal{T}_1^{(1)} \cdot_c \mathcal{T}_1^{(2)} \cdot_c \ldots$
- $\ldots$

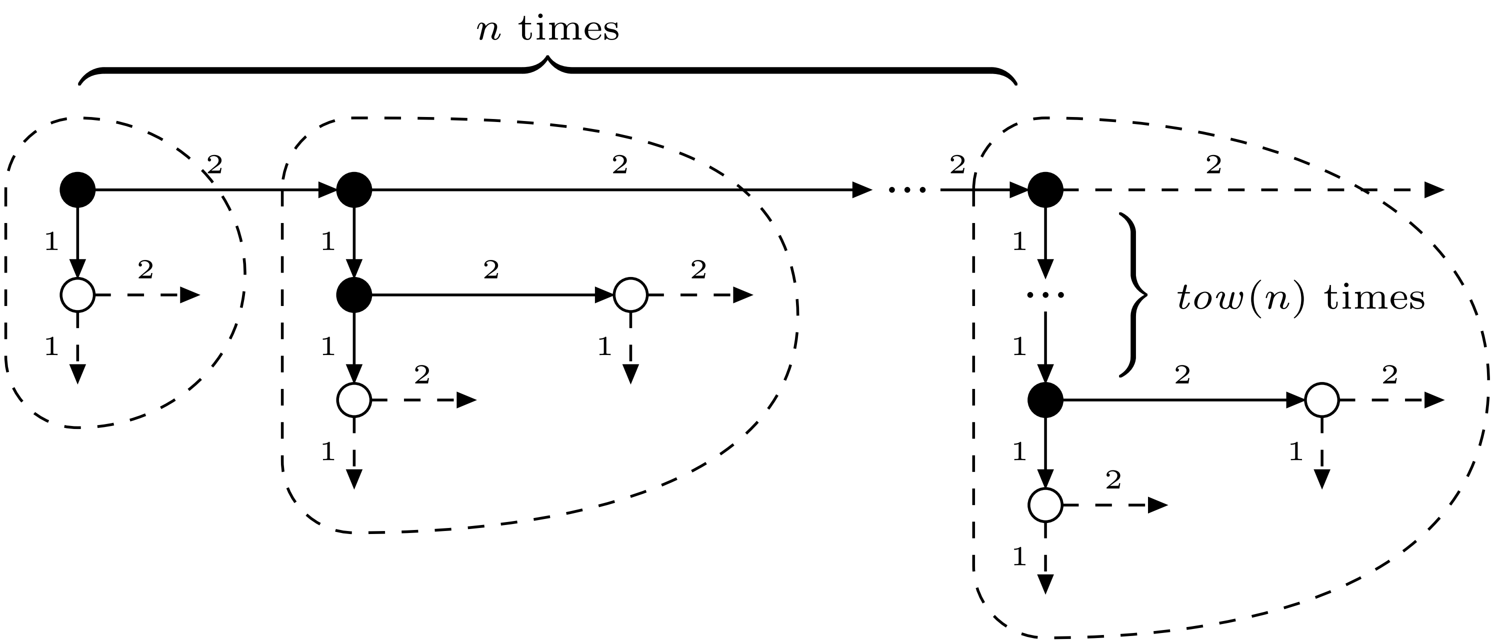
Application examples

We exploited the proposed method to decide the theory of some trees **inside** and **outside** the Caucal hierarchy

Application examples

We exploited the proposed method to decide the theory of some trees inside and **outside** the Caucal hierarchy

- The tree $\mathcal{T}_{tow}$ (see Carayol and Wöhrle '03):

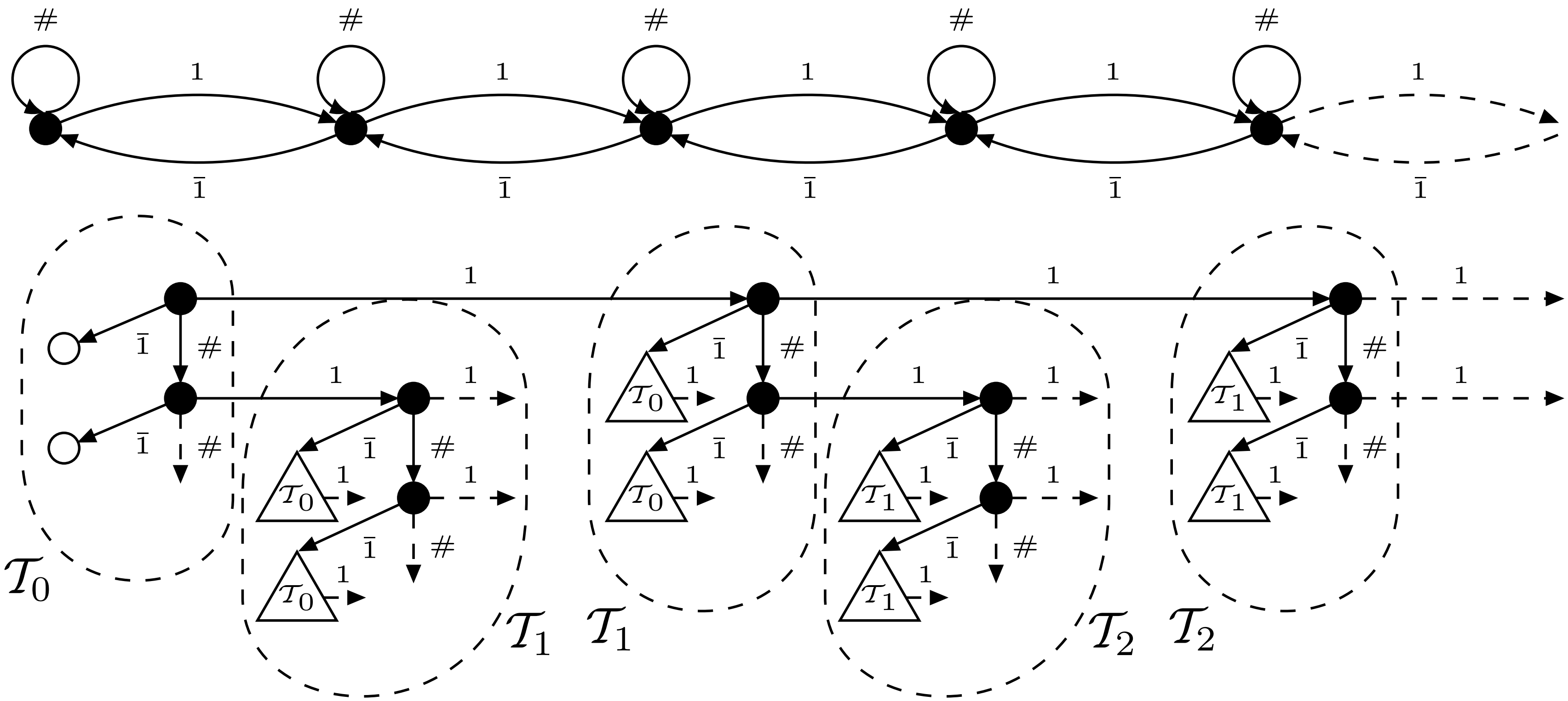


where $tow(n) = \begin{cases} 1 & \text{if } n = 0, \\ 2^{tow(n-1)} & \text{if } n > 0 \end{cases}$

Application examples

We exploited the proposed method to decide the theory of some trees **inside** and outside the Caucal hierarchy

- The **unfolding of the semi-infinite line**:



The factors $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2 \ldots$ can be defined *recursively*.
Thus, by structural properties, the factorization is *residually ultimately periodic*.

Application examples

We exploited the proposed method to decide the theory of some trees **inside** and outside the Caucal hierarchy

- The **tree generators** associated with the levels of the Caucal hierarchy:

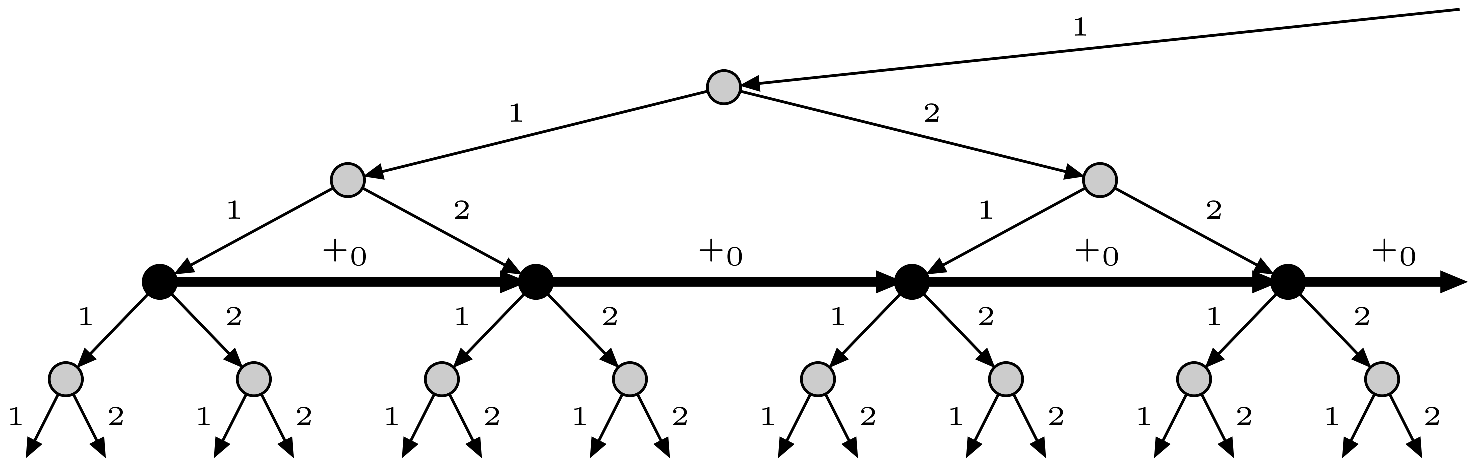
  These trees are obtained by an n-fold application of the unfolding (with backward edges and loops) starting from the infinite binary tree.

  They allow one to obtain **each graph of a level of the Caucal hierarchy** via MSO interpretations.

  As for the case of the unfolding of the semi-infinite line, we proved that they enjoy a *residually ultimately periodic factorization*.

Application examples

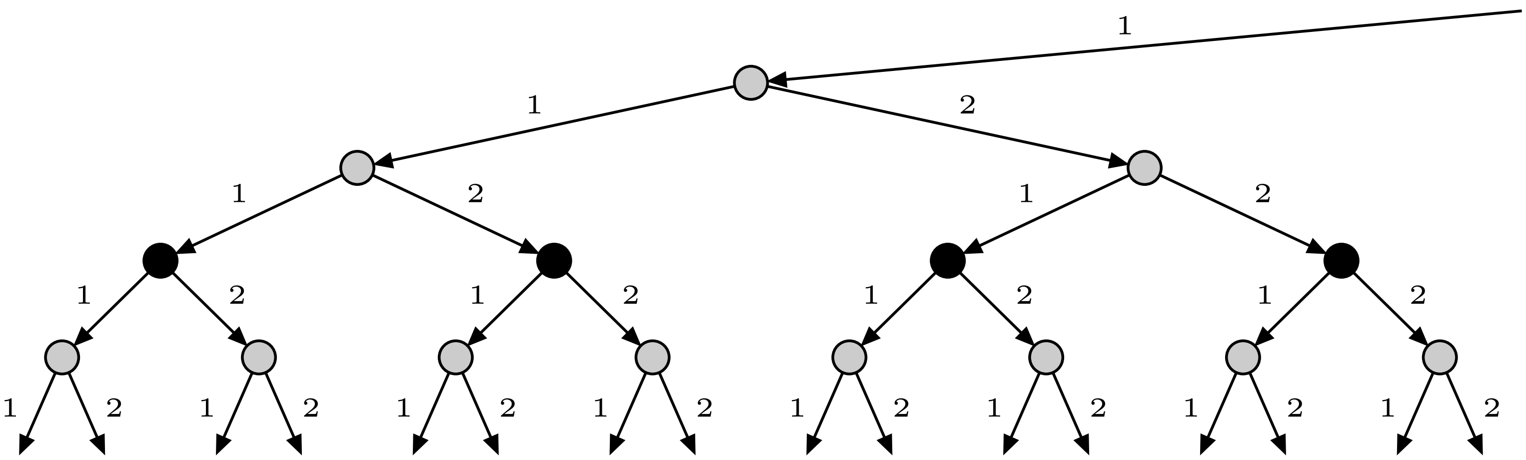
Finally, we exploited the method to decide the theory of the **totally unbounded ω-layered structure**:

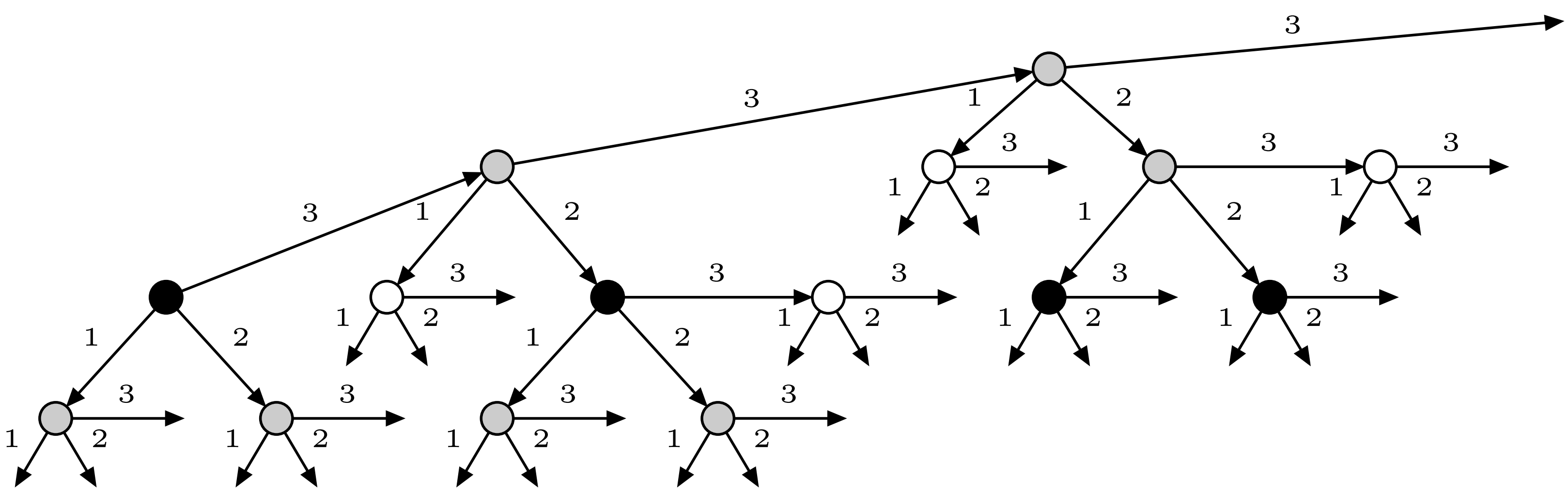


- The structure contains arbitrarily fine/coarse layers
- Arrows map elements of a given layer to elements of the immediately finer layer
- Black vertices denote the elements of a distinguished layer (layer 0) endowed with a (MSO-definable) successor relation $+_0$

Application examples

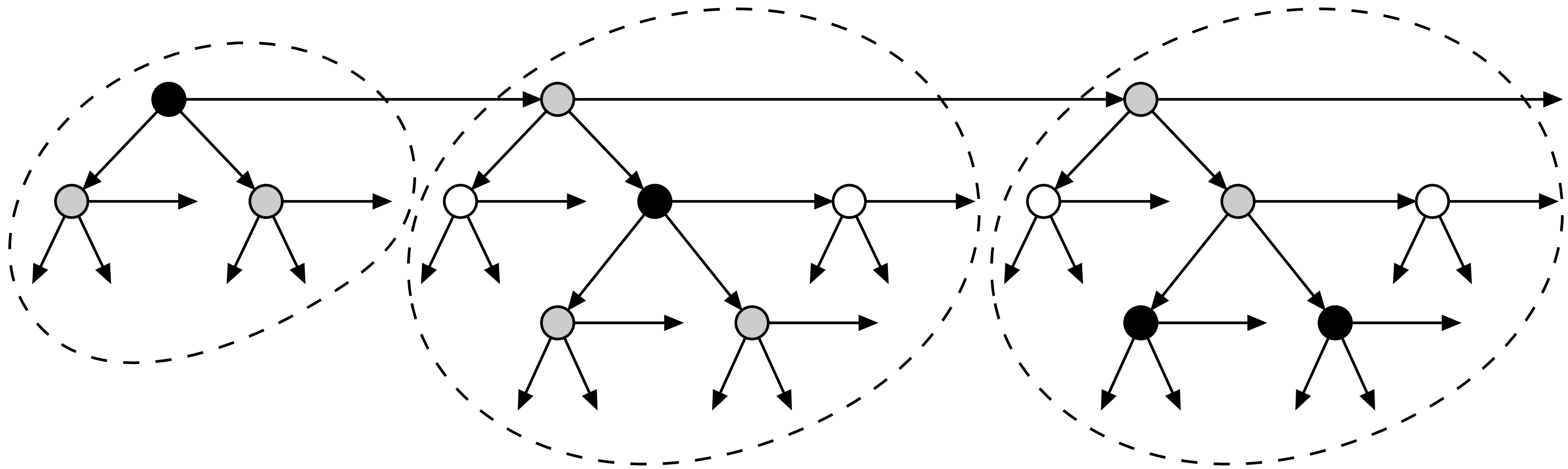
The totally unbounded ω-layered structure can be interpreted into an infinite complete **ternary tree**:





Application examples

The resulting tree can be proved to be *residually regular*:



Dashed regions denote factors, which can be defined *recursively*. Thus, by structural properties, the factorization is *residually ultimately periodic*.

Further work

- Extend the notion of congruence to different, more expressive, classes of automata
(e.g. **automata over tree-like structures**).
- Compare the automaton-based approach with other ones.

  In particular, we are trying to
  - generalize the approach to embed **Courcelle's algebraic trees** and the deterministic trees of the **Caucal hierarchy**,
  - exploit possible connections with the **compositional method** of Shelah.