# Unambiguity and uniformization problems on infinite trees

**Marcin Bilkowski*** **and Michał Skrzypczak**†

**University of Warsaw**
**Banacha 2, 02-097 Warsaw, Poland**
`{m.bilkowski,mskrzypczak}@mimuw.edu.pl`

## Abstract

A nondeterministic automaton is called unambiguous if it has at most one accepting run on every input. A regular language is called unambiguous if there exists an unambiguous automaton recognizing this language. Currently, the class of unambiguous languages of infinite trees is not well-understood. In particular, there is no known decision procedure verifying if a given regular tree language is unambiguous. In this work we study the self-dual class of bi-unambiguous languages — languages that are unambiguous and their complement is also unambiguous. It turns out that thin trees (trees with only countably many branches) emerge naturally in this context.

We propose a procedure $P$ designed to decide if a given tree automaton recognizes a bi-unambiguous language. The procedure is sound for every input. It is also complete for languages recognisable by deterministic automata. We conjecture that $P$ is complete for all inputs but this depends on a new conjecture stating that there is no MSO-definable choice function on thin trees. This would extend a result by Gurevich and Shelah on the undefinability of choice on the binary tree.

We provide a couple of equivalent statements to our conjecture, we also give several related results about uniformizability on thin trees. In particular, we provide a new example of a language that is not unambiguous, namely the language of all thin trees. The main tool in our studies are algebras that can be seen as an adaptation of Wilke algebras to the case of infinite trees.

**1998 ACM Subject Classification** F.4.3 Formal Languages

**Keywords and phrases** nondeterministic automata, infinite trees, MSO logic

**Digital Object Identifier** 10.4230/LIPIcs.CSL.2013.81

## 1 Introduction

Infinite trees form a rich class of models, one infinite tree may encode whole set of finite words or a strategy in an infinite duration game. Therefore, the decidability of Monadic Second-Order (MSO) logic over infinite trees [19] is often called the *mother of all decidability results*. The proof of this decidability result follows a similar line as in the case of finite words [27] — we find a model of automata that are equivalent in expressive power with MSO logic and have decidable emptiness problem.

The proof of Rabin's theorem deals with nondeterministic automata as deterministic ones have strictly smaller expressive power. It is one of the main reasons why many problems about regular languages of infinite trees are very hard. For example, no algorithm is known

* This paper has been partially supported by the Polish Ministry of Science grant no. N206 567840.
† Author supported by ERC Starting Grant "Sosna" no. 239850.


Computer Science Logic 2013 (CSL'13).
Editor: Simona Ronchi Della Rocca; pp. 81–100
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to decide the parity index in the class of all regular tree languages. On the other hand, there are many results for the restricted class of deterministic languages [11, 15, 16, 17, 13]. Unambiguous automata can be seen as a natural intermediate class between deterministic and nondeterministic ones. An automaton is unambiguous if it has at most one accepting run on every input. In some settings [25, 3] unambiguous automata admit faster algorithms than general nondeterministic automata.

The unambiguous automata do not capture the class of all regular languages of infinite trees. As shown in [5], the language $L_b$ of trees containing at least one letter $b$ cannot be recognised by any unambiguous automaton. The proof uses a result by Gurevich and Shelah [8] stating that there is no MSO-definable choice function on the full binary tree (see [5] for a simpler proof of this result). To the authors' best knowledge, the non-definability of choice has been so far the only method to show that a tree language is ambiguous (i.e. not unambiguous).

The class of unambiguous languages of infinite trees is not well-understood. In particular, there is no effective procedure known that decides whether a given nondeterministic automaton recognises an unambiguous language. Additionally, unambiguous languages lack some natural properties. As witnessed by the language $L_b$, a complement of an unambiguous (and even deterministic) language may be ambiguous. Also, as shown in Proposition 2 of this work, a sum of two deterministic languages may be ambiguous.

Due to the above reasons we concentrate on the class of languages such that both the language and its complement are unambiguous. We call these languages *bi-unambiguous*. An easy argument shows that this class is effectively closed under boolean operations. Moreover, the class is rich enough to contain languages beyond the $\sigma$-algebra generated by $\mathbf{\Pi}^1_1$ sets (see [9]). In particular, there are bi-unambiguous languages that are topologically harder than all deterministic languages.

Our motivating problem is to find an effective procedure that verifies if a given regular tree language is bi-unambiguous. Unfortunately, we are unable to solve this problem in full generality. We have a candidate $P$ for such a procedure and we prove that $P$ is sound — if $P$ returns YES then the language is bi-unambiguous. Also, $P$ is complete for deterministic languages — if $L$ is deterministic and bi-unambiguous then $P$ returns YES. The completeness of $P$ in the general case relies on a new conjecture (Conjecture 1 below).

Interestingly, the class of thin trees (trees containing only countably many branches, see [12, 21, 2]) emerges naturally in this context. The crucial technical tool of the procedure $P$ can be seen as an application of the algebra designed for thin trees [10, 2] in the setting of all trees. For this purpose a class of *prophetic thin algebras* is introduced. Basing on algebraic observations we show that $P$ is complete if the following conjecture holds.

▶ Conjecture 1 (Undefinability of a choice function on thin trees). There is no MSO formula in the language of trees $\varphi(x, X)$ such that for every non-empty set $X \subseteq \{l, r\}^*$ that is contained in a thin tree, $\varphi(x, X)$ holds for exactly one vertex $x$ and such a vertex $x$ belongs to $X$.

To the authors' best knowledge the above conjecture is new. It is a strengthening of the result of Gurevich and Shelah [8] as we restrict the class of allowed sets $X$.

We find this conjecture interesting in its own right. A number of equivalent statements is provided. Also, it turns out that, assuming Conjecture 1, the class of finite prophetic thin algebras has many good properties (e.g. it is a pseudo-variety of algebras corresponding exactly to the class of bi-unambiguous languages).

Conjecture 1 can be seen as an instance of a more general problem of uniformization. We provide some related results on uniformizability on thin trees. In particular, we show that there exists some non-uniformizable formula on thin trees. It can be seen as an alternative to [8] answer to Rabin's Uniformization Problem. Also, we show that the language of all thin trees is ambiguous, thus providing an essentially new example of an ambiguous language.

We begin by introducing some basic definitions and notions. In Section 3 we define the procedure $P$ and show its properties. Section 4 is devoted to the analysis of the choice problem on thin trees. In Section 5 we study related uniformization problems on thin trees.

## 2 Basic notions

### 2.1 Trees

For technical reasons we work with ranked alphabets $A = (N, L)$ where $N$ (like nodes) contains binary symbols and $L$ (like leafs) contains nullary symbols. We assume that both sets $N$ and $L$ are finite and nonempty. We say that $t$ is a *tree over the alphabet* $(N, L)$ if $t$ is a function from its nonempty domain $\mathrm{dom}(t) \subseteq \{l, r\}^*$ into $N \cup L$ in such a way that $\mathrm{dom}(t)$ is prefix-closed and for every vertex $w \in \mathrm{dom}(t)$ either:

- $w$ is an *(internal) node of* $t$ (i.e. $wl, wr \in \mathrm{dom}(t)$) and $t(w) \in N$, or
- $w$ is a *leaf of* $t$ (i.e. $wl, wr \notin \mathrm{dom}(t)$) and $t(w) \in L$.

The set of all trees over an alphabet $A$ is denoted as $\mathrm{Tr}_A$. A tree containing no leaf is *full*. If $t \in \mathrm{Tr}_A$ is a tree and $w \in \mathrm{dom}(t)$ is a vertex of $t$ then by $t\restriction_w \in \mathrm{Tr}_A$ we denote the subtree of $t$ rooted in $w$. By $\preceq$ we denote the prefix-order on elements of $\{l, r\}^{\leq\omega}$.

A sequence $\pi \in \{l, r\}^\omega$ is an *infinite branch of a tree* $t$ if for every $w \prec \pi$ we have that $w \in \mathrm{dom}(t)$. An element $d \in \{l, r\}$ is called a *direction*, the opposite direction is denoted as $\bar{d}$. The empty sequence $\epsilon$ is called the *root* of a tree $t$. If $\pi$ is an infinite branch of a tree $t$ and $w \not\prec \pi$ but $w$ is a child of a vertex on $\pi$ then we say that $w$ is *off* $\pi$.

A tree $t \in \mathrm{Tr}_A$ is *thin* if there are only countably many infinite branches of $t$. The set of all thin trees is denoted by $\mathrm{Th}_A$. A tree that is not thin is *thick*. A tree is *regular* if it has only finitely many different subtrees. For $a \in N$ by $a(t_l, t_r) \in \mathrm{Tr}_A$ we denote the tree consisting of the root $\epsilon$ labelled by the letter $a$ and two subtrees $t_l, t_r \in \mathrm{Tr}_A$ respectively.

A *multi-context* over an alphabet $A = (N, L)$ is a tree $c \in \mathrm{Tr}_{(N, L \sqcup \{\square\})}$. A vertex $w \in \mathrm{dom}(c)$ such that $c(w) = \square$ is called a *hole of* $c$. For every tree $t \in \mathrm{Tr}_A$ the *composition of* $c$ *and* $t$, denoted $c \cdot t \in \mathrm{Tr}_A$, is obtained by plugging copies of $t$ in all the holes of $c$.

If a multi-context $c$ has exactly one hole not in the root then it is called a *context*. The set of all contexts over the alphabet $A$ is denoted as $\mathrm{Con}_A$. The set of all contexts over $A$ that are thin as trees is denoted by $\mathrm{ThCon}_A$. For $t \in \mathrm{Tr}_A$ and $w \in \mathrm{dom}(t)$, by $t[\square/w]$ we denote the context obtained from $t$ by putting the hole in $w$.

Let $t_A \in \mathrm{Tr}_A$ and $M$ be a ranked alphabet. We say that $t_M \in \mathrm{Tr}_M$ is a *labelling of* $t_A$ if $\mathrm{dom}(t_M) = \mathrm{dom}(t_A)$. In that case we define the tree $(t_A, t_M) \in \mathrm{Tr}_{A\times M}$ in the natural way.

### 2.2 Automata

A nondeterministic parity tree automaton is a tuple $\mathcal{A} = (Q, A, \delta, I, \Omega)$ where

- $Q$ is a finite set of *states*,
- $A = (N, L)$ is a ranked alphabet,
- $\delta = \delta_2 \sqcup \delta_0$ is a *transition relation*: $\delta_2 \subseteq Q \times Q \times N \times Q$ contains *transitions for nodes* $(q, q_l, a, q_r)$ and $\delta_0 \subseteq Q \times L$ contains *transitions for leafs* $(q, b)$,
- $I \subseteq Q$ is a set of *initial states*,
- $\Omega \colon Q \to \mathbb{N}$ is a *priority function*.

A run of an automaton $\mathcal{A}$ on a tree $t \in \mathrm{Tr}_A$ is a labelling $\rho$ of $t$ over the ranked alphabet $(Q, Q)$ such that for every vertex $w$ of $t$

- if $w$ is a node of $t$ then $(\rho(w), \rho(wl), t(w), \rho(wr)) \in \delta_2$,
- if $w$ is a leaf of $t$ then $(\rho(w), t(w)) \in \delta_0$.

A run $\rho$ is *consistent* if for every infinite branch $\pi$ of $t$ the lim sup of values of $\Omega$ on states along $\pi$ is even: $\limsup_{n\to\infty} \Omega(\rho(\pi\restriction_n)) \equiv 0 \mod 2$. The state $\rho(\epsilon)$ is called the *value of* $\rho$.

Similarly one can define a run $\rho$ on a context $c$ with the hole $w$, the only difference is that there is no constraint on the value $\rho(w)$ in the hole of $c$.

A run $\rho$ is *accepting* if it is consistent and $\rho(\epsilon) \in I$. A tree $t \in \mathrm{Tr}_A$ is *accepted by* $\mathcal{A}$ if there exists an accepting run $\rho$ of $\mathcal{A}$ on $t$. The set of trees accepted by $\mathcal{A}$ is called the *language recognised by* $\mathcal{A}$ and is denoted by $\mathrm{L}(\mathcal{A})$. A language $L \subseteq \mathrm{Tr}_A$ is *regular* if there exists an automaton recognising $L$.

We say that an automaton $\mathcal{A}$ is *deterministic* if $I = \{q_I\}$ and for every state $q \in Q$ and letter $a \in N$ there is at most one transition $(q, q_l, a, q_r) \in \delta_2$. An automaton $\mathcal{A}$ is *unambiguous* if for every tree $t \in \mathrm{L}(\mathcal{A})$ there is exactly one accepting run of $\mathcal{A}$ on $t$. A language $L \subseteq \mathrm{Tr}_A$ is *deterministic* (resp. *unambiguous*) if there exists a deterministic (resp. unambiguous) automaton recognising $L$. A language that is not unambiguous is called *ambiguous*. A deterministic language is, by the definition, unambiguous. A language $L$ is *bi-unambiguous* if both $L$ and $\mathrm{Tr}_A \setminus L$ are unambiguous.

We finish this section with an observation showing that unambiguous languages are not closed under finite union.

▶ Proposition 2. There exist deterministic languages $L_1, L_2$ such that $L_1 \cup L_2$ is ambiguous.

## 2.3 Logic

We use the standard notion of Monadic Second-Order (MSO) logic (see [26]). The syntax of this logic allows quantification over elements and subsets of the domain, boolean connectives, predicates for the letters in a given alphabet, and two relations $l$-child, $r$-child.

For a given MSO formula $\varphi(\vec{P})$ over an alphabet $A = (N, L)$ with $n$ parameters $P_1, \ldots, P_n$ by $\mathrm{L}(\varphi(\vec{P}))$ we denote the set of trees over the alphabet $(N \times \{0,1\}^n, L \times \{0,1\}^n)$ that satisfy $\varphi$ when parameters $P$ are decoded from their characteristic functions.

The crucial property of MSO logic is expressed by the following theorem.

▶ **Theorem 3** (Rabin [19]). *A language $L \subseteq \mathrm{Tr}_A$ is regular if and only if there exists an MSO formula $\varphi$ such that $L = \mathrm{L}(\varphi)$. There are effective procedures translating MSO formulas into equivalent automata and vice versa.*

# 3 Bi-unambiguous languages

In this section we concentrate on the following decision problem.

▶ Problem 4. The input is a nondeterministic parity tree automaton $\mathcal{A}$. The output should be YES if the language $\mathrm{L}(\mathcal{A})$ is bi-unambiguous. Otherwise, the output should be NO.

We construct a procedure $P$ with the following properties.

▶ **Theorem 5.** *Let $\mathcal{A}$ be a nondeterministic tree automaton.*

1. *$P(\mathcal{A})$ terminates.*
2. *If $P(\mathcal{A}) = \mathrm{YES}$ then $\mathrm{L}(\mathcal{A})$ is bi-unambiguous.*

3. *If* $\mathrm{L}(\mathcal{A})$ *is deterministic and* $P(\mathcal{A}) = \mathrm{NO}$ *then* $\mathrm{L}(\mathcal{A})$ *is not bi-unambiguous*[1]*.*
4. *If Conjecture 1 is true and* $P(\mathcal{A}) = \mathrm{NO}$ *then* $\mathrm{L}(\mathcal{A})$ *is not bi-unambiguous.*

Recall that it is decidable whether a given regular tree language is recognisable by a deterministic tree automaton (see [17]). Therefore, the above assumption that $\mathrm{L}(\mathcal{A})$ is deterministic can be effectively checked given some representation of $\mathrm{L}(\mathcal{A})$. The rest of this section is devoted to defining $P$ and showing the above theorem.

## 3.1 Thin algebra

The crucial tool in the construction of the procedure $P$ is a variant of thin forest algebra [2], called *thin algebra*. Thin algebra can be seen as a natural extension of Wilke algebra [28, 30] and Wilke tree algebra [29] to the case of infinite trees.

Let us fix a ranked alphabet $A = (N, L)$. A *thin algebra over* $A$ is a two-sorted algebra $(H, V)$ with a number of operations:

- $u \cdot v \in V$ for $u, v \in V$,
- $v \cdot h \in H$ for $v \in V, h \in H$,
- $v^\infty \in H$ for $v \in V$,
- $\mathrm{Node}(a, d, h) \in V$ for $a \in N$, $d \in \{l, r\}$, and $h \in H$,
- $\mathrm{Leaf}(b) \in H$ for $b \in L$.

Note that the first three operations are the same as in the case of Wilke algebras. The last two operations allow to operate on trees. For simplicity, we write $a(\square, h)$ instead of $\mathrm{Node}(a, l, h)$ and $a(h, \square)$ instead of $\mathrm{Node}(a, r, h)$. Similarly, $b()$ stands for $\mathrm{Leaf}(b)$ and $a(h_l, h_r) \in H$ denotes the result of $a(h_l, \square) \cdot h_r$.

The axioms of thin algebra are axioms of Wilke algebra and one additional axiom: $a(\square, h_r) \cdot h_l = a(h_l, \square) \cdot h_r$.

▶ **Fact 6.** Let $(H, V)$ be a thin algebra and let $(v_i)_{i \in \mathbb{N}}$ be any sequence of elements of $V$. There exists a unique value $\prod_i v_i \in H$ for which: if $j_0 < j_1 < \ldots$ is a sequence of numbers and $s, e \in V$ are types such that:

- $v_0 \cdot \ldots \cdot v_{j_0} = s$,
- for all $i \geq 0$ $v_{j_i+1} \cdot \ldots \cdot v_{j_{i+1}} = e$

then $s \cdot e^\infty = \prod_i v_i$. Also, the following holds $\prod_{i \geq 0} v_i = v_0 \cdot \prod_{i \geq 1} v_i$.

**Proof.** The same as in the case of Wilke algebra, see [18]. ◀

It is easy to verify that the pair $(\mathrm{Tr}_A, \mathrm{Con}_A)$ has a natural structure of a thin algebra. In particular, the operation $c^\infty$ constructs the tree $c^\infty$ from a context $c$ by *looping* the hole of $c$ to the root of $c$. The subalgebra of $(\mathrm{Tr}_A, \mathrm{Con}_A)$ consisting of thin regular trees and thin regular contexts is free in the class of thin algebras over the alphabet $A$. The algebra $(\mathrm{Tr}_A, \mathrm{Con}_A)$ is not free.

A homomorphism $\alpha\colon (H, V) \to (H', V')$ between two thin algebras over the same alphabet $A$ is defined in the usual way: $\alpha$ should be a function mapping elements of $H$ into $H'$ and elements of $V$ into $V'$ that preserves all the operations of thin algebra. Such a homomorphism is *surjective* if $\alpha(H) = H'$ and $\alpha(V) = V'$.

Since $(\mathrm{Tr}_A, \mathrm{Con}_A)$ is not free in the class of thin algebras, we need to define one additional requirement for homomorphisms $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$. Let $A = (N, L)$ and put $A \sqcup H =$

[1] What is equivalent to ambiguity of the complement of $\mathrm{L}(\mathcal{A})$.

$(N, L \sqcup H)$. Consider any tree $c \in \mathrm{Tr}_{A \sqcup H}$ and $t \in \mathrm{Tr}_A$. We say that $t$ *is an extension of* $c$ if $\mathrm{dom}(c) \subseteq \mathrm{dom}(t)$ and for every $w \in \mathrm{dom}(c)$ either:

- $c(w) \in N \cup L$ and $c(w) = t(w)$,
- $c(w) \in H$ and $c(w) = \alpha(t\restriction_w)$.

That is, $t$ is supposed to agree with $c$ on all the letters in $N \cup L$ and whenever $c$ declared some type $h \in H$ in a leaf $w$ then the subtree $t\restriction_w$ has $\alpha$-type $h$ (i.e. $\alpha(t\restriction_w) = h$).

▶ **Definition 7.** We say that $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ *is compositional* if there exists a function $\bar{\alpha}\colon \mathrm{Tr}_{A \sqcup H} \to H$ such that if $t \in \mathrm{Tr}_A$ is an extension of $c \in \mathrm{Tr}_{A \sqcup H}$ then $\bar{\alpha}(c) = \alpha(t)$.

Let $L \subseteq \mathrm{Tr}_A$ be a language of trees. We say that a homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ *recognises* $L$ if $\alpha$ is compositional and there is a set $F \subseteq H$ such that $L = \alpha^{-1}(F)$.

▶ Fact 8. Since every context $c \in \mathrm{Con}_A$ can be obtained as a finite combination of trees $t \in \mathrm{Tr}_A$ using the operation Node, if $\alpha_1, \alpha_2\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ are two homomorphisms that agree on $\mathrm{Tr}_A$ then $\alpha_1 = \alpha_2$.

The following theorem introduces the notion of *syntactic morphism* for a given language. It is an adaptation of a similar theorem for the case of thin forest algebras, see [10] for a deeper explanation. For the sake of completeness, a sketch of a proof is given in Appendix A.

▶ **Theorem 9.** *For every regular tree language $L$ there exists a* syntactic morphism for $L$*: a finite thin algebra $S_L = (H, V)$ (called a* syntactic algebra of $L$*) and a homomorphism $\alpha_L\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to S_L$ such that:*

- *$\alpha_L$ is surjective, compositional, and recognises $L$,*
- *for every $h \in H$ the language $L_h := \alpha_L^{-1}(\{h\})$ is regular,*
- *if $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to S$ is surjective and recognises $L$ then there is exactly one homomorphism $\beta\colon S \to S_L$ such that $\beta \circ \alpha = \alpha_L$.*

*A syntactic algebra $S_L$ and languages $L_h$ can be effectively computed basing on a non-deterministic automaton recognising $L$.*

Note that by the last bullet, all syntactic morphisms for a given language are *isomorphic* — there are homomorphisms $\beta$ that make the respective diagrams commute. Therefore, a syntactic morphism can be seen as a unique representation of a language.

An intermediate step in this proof requires a definition of some finite thin algebra $S_{\mathcal{A}} = (H_{\mathcal{A}}, V_{\mathcal{A}})$ that recognises the language $\mathrm{L}(\mathcal{A})$ for a given automaton $\mathcal{A}$. The constructed algebra is called the *automaton algebra for* $\mathcal{A}$. The definition of $S_{\mathcal{A}}$ is the same as in [10]. The homomorphism into $S_{\mathcal{A}}$ that recognises $\mathrm{L}(\mathcal{A})$ is based on the following operation that will be used later:

$$Q_{\mathcal{A}}(t) = \{q \in Q :\ \exists_\rho\ \rho \text{ is a consistent run of } \mathcal{A} \text{ on } t \text{ with value } q\} \subseteq 2^Q. \tag{1}$$

If $\mathcal{A}$ is known from the context, we write just $Q(t)$. By $\tau_{\mathcal{A}}(t)$ we denote the labelling of $t$ defined $\tau_{\mathcal{A}}(t)(w) = Q_{\mathcal{A}}(t\restriction_w)$.

What is important in Theorem 9 is that we explicitly fix the homomorphism $\alpha_L$. Usually (e.g. in the case of monoids) there is a unique such homomorphism for a fixed interpretation of the alphabet. It turns out that this is not the case for thin algebras and all binary trees. Therefore, to fully describe a given language we need an algebra $S_L$, a set $F \subseteq H$, and a homomorphism $\alpha_L$ (it can be represented by the languages $L_h$).

## 3.2 Prophetic algebras

The situation when there are multiple homomorphisms from all trees into a given thin algebra comes from the fact that the algebra may not be *prophetic*. In this section we formally introduce this notion.

Let $(H, V)$ be a thin algebra over an alphabet $A = (N, L)$. Let $t \in \mathrm{Tr}_A$ be a tree. A labelling $\tau \in \mathrm{Tr}_{(H,H)}$ of $t$ is a *marking of $t$ by types in $H$* if:

- for every node $w$ of $t$ we have $\tau(w) = t(w)(\tau(wl), \tau(wr))$,
- for every leaf $w$ of $t$ we have $\tau(w) = t(w)()$.

A marking $\tau$ is *consistent* if it is consistent on every infinite branch $\pi$ of $t$. Let $\pi = d_0 d_1 \ldots$ and let $w_0 \prec w_1 \prec \ldots$ be the sequence of vertices of $t$ along $\pi$. The sequence of types of contexts $v_i = \mathrm{Node}(t(w_i), d_i, \tau(w_i \bar{d_i}))$ is called the *decomposition of $\tau$ along $\pi$*. Now, $\tau$ is *consistent on $\pi$* if for every $i \in \mathbb{N}$ we have

$$\tau(w_i) = \prod_{j \geq i} v_j. \tag{2}$$

Informally speaking, a marking $\tau$ is consistent along $\pi$ if the types of $\tau$ along $\pi$ agree with the types that can be computed using $\prod$ basing on the types of vertices that are off $\pi$. By the definition of a marking, it is enough to require (2) for infinitely many $i \in \mathbb{N}$ in the definition of consistency.

Note that if a homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to S$ is fixed, for every tree $t \in \mathrm{Tr}_A$ the marking $\tau_\alpha(t)(w) := \alpha(t{\restriction}_w)$ (called the *marking induced by $\alpha$ on $t$*) is consistent.

▶ **Example 10.** Fix the alphabet $A_b = (\{n\}, \{b\})$. Let $L_b \subseteq \mathrm{Tr}_{A_b}$ contain exactly these trees which have at least one leaf. One may verify that the syntactic morphism for $L_b$ can be defined as follows: $H_{L_b} = \{h_a, h_b\}$, $V_{L_b} = \{v_a, v_b\}$, and $\alpha_{L_b}(t) = h_b$ (resp. $\alpha_{L_b}(c) = v_b$) if and only if a tree $t$ (resp. a context $c$) contains any leaf (not counting the hole of $c$).

Let $t_n$ be the full binary tree equal everywhere $n$. Observe that $t_n$ does not belong to $L_b$ and the marking $\tau_{\alpha_{L_b}}(t_n)$ induced by $\alpha_{L_b}$ on $t_n$ equals $h_a$ in every vertex. Consider another marking $\tau'$ of $t_n$ that equals $h_b$ everywhere. Note that $\tau'$ is consistent — it looks like a consistent marking along every branch. Therefore, $t$ has two consistent markings.

Going further, one can construct a compositional homomorphism $\alpha'\colon (\mathrm{Tr}_{A_b}, \mathrm{Con}_{A_b}) \to (H_{L_b}, V_{L_b})$ that assigns $h_b$ to the tree $t_n$. Therefore, there are two distinct compositional homomorphisms from $(\mathrm{Tr}_{A_b}, \mathrm{Con}_{A_b})$ to $(H_{L_b}, V_{L_b})$.

Recall that the language $L_b$ used above is known to be ambiguous, see [14].

The following fact follows from [2], the proof goes via induction on *rank* of thin trees.

▶ Fact 11. If $t \in \mathrm{Tr}_A$ is a thin tree and $(H, V)$ is a finite thin algebra over the alphabet $A$ then there exists exactly one consistent marking $\tau$ of $t$.

The following definition is crucial for the procedure $P$. The term *prophetic* is motivated by [6].

▶ **Definition 12.** We say that a thin algebra $(H, V)$ over an alphabet $A$ is *prophetic* if for every tree $t \in \mathrm{Tr}_A$ there exists at most one consistent marking of $t$ by types in $H$.

Note that if $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to S$ is a homomorphism and $S$ is prophetic then, for every tree $t \in \mathrm{Tr}_A$, the only consistent marking of $t$ is the marking induced by $\alpha$. In particular, there is at most one homomorphism from $(\mathrm{Tr}_A, \mathrm{Con}_A)$ into $S$, see Fact 8.

Since the property that a given finite thin algebra is prophetic can be expressed in MSO over the full binary tree, so we obtain the following fact.

▶ Fact 13. It is decidable whether a given finite thin algebra $(H, V)$ is prophetic.

▶ Fact 14. By the definition, a subalgebra of a prophetic thin algebra is also prophetic. Similarly, a product of two prophetic thin algebras is also prophetic.

## 3.3 Semi-characterisation

The following theorem gives a connection between bi-unambiguous languages and prophetic algebras.

▶ **Theorem 15.** *A language $L \subseteq \mathrm{Tr}_A$ is bi-unambiguous if and only if there exists a surjective homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ that recognises $L$ such that $(H, V)$ is a finite prophetic thin algebra over the alphabet $A$.*

First assume that $L$ is a bi-unambiguous language. Let $\mathcal{A}, \mathcal{B}$ be a pair of unambiguous automata recognising $L$ and $\mathrm{Tr}_A \setminus L$ respectively. We describe how to construct a finite prophetic thin algebra $(H_U, V_U)$ recognising $L$.

The first step can be expressed as the following fact.

▶ Fact 16. Assume that $\mathcal{A}$ is an unambiguous automaton over an alphabet $A$ and $t \in \mathrm{Tr}_A$. Assume that $\tau$ is a consistent marking of $t$ by types in the automaton algebra $S_{\mathcal{A}}$. Then there is at most one run $\rho$ of $\mathcal{A}$ on $t$ such that $\rho(\epsilon) \in I^{\mathcal{A}}$ and $\forall_{w \in \mathrm{dom}(t)}\ \rho(w) \in \tau(w)$.

Using the above observation and properties of the automaton algebra, we can entail that for every consistent marking $\tau$ of a given tree $t$ and for every $q \in \tau_{\mathcal{A}}(t)(\epsilon)$ there is a consistent run of $\mathcal{A}$ on $t$ with value $q$. Therefore, for every consistent marking $\tau$ of $t$ we have $\forall_{w \in \mathrm{dom}(t)}\ \tau(w) \subseteq \tau_{\mathcal{A}}(t)(w)$. Our aim is to put some additional constraints on $\tau$ that imply equality in the above formula. This is obtained by the second step of the reasoning, as expressed in the following lemma. The idea to use pairs of sets of states in this context was suggested by Igor Walukiewicz.

▶ **Lemma 17.** *Let $\mathcal{A}, \mathcal{B}$ be a pair of unambiguous automata recognising $L$ and $\mathrm{Tr}_A \setminus L$ respectively. Let $R = \{(Q_{\mathcal{A}}(t), Q_{\mathcal{B}}(t)) :\ t \in \mathrm{Tr}_A\}$. Then the set $R$ ordered coordinate-wise by inclusion is an antichain.*

Now let $t \in \mathrm{Tr}_A$ and assume that we have consistent markings $\tau_1, \tau_2$ of $t$ with respect to algebras $S_{\mathcal{A}}, S_{\mathcal{B}}$ respectively. Assume that for every $w \in \mathrm{dom}(t)$ we have $(\tau_1(w), \tau_2(w)) \in R$. Then $\tau_1(w) \subseteq \tau_{\mathcal{A}}(t)(w)$, $\tau_2(w) \subseteq \tau_{\mathcal{B}}(t)(w)$, and by the above lemma $\tau = \tau_{\mathcal{A}}(t), \tau' = \tau_{\mathcal{B}}(t)$. This shows that the product of algebras $S_{\mathcal{A}}$ and $S_{\mathcal{B}}$ is prophetic.

The following lemma implies the opposite direction of Theorem 15.

▶ **Lemma 18.** *Let $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ be a compositional homomorphism into a finite prophetic thin algebra $(H, V)$ and $h_0 \in H$. The language $L_{h_0} = \alpha^{-1}(h_0)$ is unambiguous.*

Using this lemma, if $\alpha$ recognises a language $L$ then $L$ and $\mathrm{Tr}_A \setminus L$ are finite disjoint unions of unambiguous languages $L_{h_0}$ so $L$ is bi-unambiguous.

The construction of an unambiguous automaton $\mathcal{C}$ recognising $L$ goes as follows: let $\mathcal{C}$ guess some marking $\tau$ of a given tree $t$ by types in $H$. Then, $\mathcal{C}$ verifies that the root is labelled by $h_0$ and the marking $\tau$ is consistent. Since consistency of a marking is a branch-wise $\omega$-regular condition, so it can be verified by a deterministic top-down automaton. Since $(H, V)$ is prophetic, so the only possible consistent marking of $t$ is the marking induced by $\alpha$. So $\mathcal{C}$ has at most one accepting run on $t$ and it accepts $t$ if and only if $\alpha(t) = h_0$.

Theorem 15 implies the following lemma, that can also be proved without use of algebra.

▶ Remark. The class of bi-unambiguous languages is closed under boolean operations and language quotients $c^{-1} \cdot L = \{t :\ c \cdot t \in L\}$ for contexts $c$.

## 3.4 The procedure P

Now we can formally define our procedure $P$. This procedure consists of three steps:

1. Read a nondeterministic automaton $\mathcal{A}$ recognising a regular tree language $L$.
2. Compute the syntactic thin algebra $S_L$ of $L$.
3. Answer YES if $S_L$ is prophetic, otherwise answer NO.

By Theorem 9 and Fact 13 both operations undertaken by $P$ are effective. Therefore, $P$ is well-defined and always terminates. Note that if $S_L$ is prophetic then, by Theorem 15, the language $L$ is bi-unambiguous. Therefore, Item 5 of Theorem 5 holds. The only remaining possibility of failure of the procedure $P$ is when $L$ is bi-unambiguous but the syntactic algebra $S_L$ is not prophetic. Our aim is to exclude this possibility. In general, Conjecture 1 implies that the syntactic algebra of a bi-unambiguous language is prophetic, see Remark 4.1. This shows that Item 5 of Theorem 5 holds. The following theorem implies Item 5 of Theorem 5.

▶ **Theorem 19.** *If $L$ is deterministic and bi-unambiguous then the algebra $S_L$ is prophetic.*

The rest of this section is devoted to proving this theorem. Let $\mathcal{D}$ be a deterministic tree automaton recognising $L \subseteq \mathrm{Tr}_A$. A state $q \in Q_{\mathcal{D}}$ is *nontrivial* if there is a tree $t$ not accepted by $\mathcal{D}$ from $q$ (i.e. there is no consistent run of $\mathcal{D}$ on $t$ with value $q$). Let $t \in L$ be a tree and $\rho$ be the accepting run of $\mathcal{D}$ on $t$. Let $T_{\mathcal{D}}(t) \subseteq \{l, r\}^*$ be the set of vertices $w \in \mathrm{dom}(t)$ such that $\rho(w)$ is a nontrivial state of $\mathcal{D}$. Note that $T_{\mathcal{D}}(t)$ is a prefix-closed subset of $\mathrm{dom}(t)$. We start with the following lemma.

▶ **Lemma 20.** *If $\mathcal{D}$ is a deterministic tree automaton and $\mathrm{Tr}_A \setminus \mathrm{L}(\mathcal{D})$ is unambiguous then for every tree $t \in \mathrm{L}(\mathcal{D})$ the set $T_{\mathcal{D}}(t)$ is thin.*

**Proof.** Assume contrary and fix a regular tree $t \in L$ such that $T = T_{\mathcal{D}}(t)$ is thick. Let $\rho$ be the run of $\mathcal{D}$ on $t$. Let $\mathcal{A}$ by an unambiguous automaton recognising $\mathrm{Tr}_A \setminus \mathrm{L}(\mathcal{D})$. Now observe that for every $w \in T$ there exists a tree $t_w$ not accepted by $\mathcal{D}$ from the state $\rho(w)$. Let $X \subseteq T$ be any prefix-free set. Let $t(X)$ be the tree obtained from $t$ by plugging simultaneously subtrees $t_w$ under $w$ for every $w \in X$. Note that if $X \neq \emptyset$ then $t(X) \notin \mathrm{L}(\mathcal{D})$ — the run $\rho$ does not extend to accepting run under any $w \in X$. Therefore, we obtain

$$t(\emptyset) \notin \mathrm{L}(\mathcal{A}) \text{ and } \forall_{X \subseteq T}\ (X \text{ is prefix-free and nonempty } \Rightarrow\ t(X) \in \mathrm{L}(\mathcal{A}))\,. \tag{3}$$

Now we construct an automaton $\bar{\mathcal{A}}$ for the language $L_b$ (see Example 10). The transitions of $\bar{\mathcal{A}}$ simulate transitions of $\mathcal{A}$ on $T$. Whenever $\bar{\mathcal{A}}$ reaches a leaf, it simulates the behaviour of $\mathcal{A}$ on the respective tree $t_w$. Since $\mathcal{A}$ is unambiguous, so is $\bar{\mathcal{A}}$. And, by (3) $\mathrm{L}(\bar{\mathcal{A}}) = L_b$. This gives us a contradiction with the fact that $L_b$ is ambiguous. ◀

▶ Fact 21. Let $\mathcal{D}$ be a deterministic automaton and $t \in \mathrm{L}(\mathcal{D}) \subseteq \mathrm{Tr}_A$. Assume that $t' \in \mathrm{Tr}_A$ is a tree satisfying $w \in \mathrm{dom}(t')$ and $t'(w) = t(w)$ for every $w \in T_{\mathcal{D}}(t)$. Then $t' \in \mathrm{L}(\mathcal{D})$.

**Proof.** The accepting run of $\mathcal{D}$ on vertices in $T_{\mathcal{D}}(t)$ can be extended to $t'$ by triviality of the states outside $T_{\mathcal{D}}(t)$. ◀

Now we can finish the proof of Theorem 19.

**Proof.** Assume contrary that the syntactic algebra $S_L$ of $L$ is not prophetic. Let $t$ be a tree and $\tau, \tau'$ be a pair of distinct consistent markings of $t$. Let $h = \tau(\epsilon)$ and $h' = \tau'(\epsilon)$. We can assume that $h \neq h'$ (otherwise instead of $t$ we take $t\restriction_w$ where $w$ is a node for which $\tau(w) \neq \tau'(w)$). Since $h \neq h'$ so there exists a multi-context $c$ such that (by symmetry)

$c \cdot t \in L$ and $c \cdot t' \notin L$. Let $w_0, w_1, \ldots$ be the list of holes of $c$. Since $c \cdot t \in L$ so we can consider the set $T = T_{\mathcal{D}}(c \cdot t) \subseteq \{l, r\}^*$.

By Lemma 20 we know that $T$ is thin, in particular $T_i := T\restriction_{w_i}$ is thin for every $i$. Let $\bar{t}_i$ be the tree obtained from $t$ by substituting some tree of $\alpha_L$-type $\tau'(w)$ instead of $t\restriction_w$ for every minimal $w \notin T_i$. Since $T_i$ is thin and $\alpha_L$-types of subtrees of $\bar{t}_i$ agree with $\tau'$ outside $T_i$ so $\alpha_L(\bar{t}_i) = h'$ — we use the fact that $T_i$ is thin. Let $\bar{t}$ be the tree obtained from $c$ by putting $\bar{t}_i$ instead of the hole $w_i$. Then, by compositionality of $\alpha_L$ we obtain that $\alpha_L(\bar{t}) = \alpha_L(c \cdot t')$, so $\bar{t} \notin L$. But $c \cdot t$ and $\bar{t}$ agree on $T_{\mathcal{D}}(t)$ so by Fact 21 $\bar{t} \in L$, a contradiction. ◀

## 4 (Un)definability of choice on thin trees

In this section we study Conjecture 1, we show a couple of equivalent statements and prove some of its consequences (in particular Item 5 of Theorem 5). We start by formulating the choice problem as a instance of a more general question.

▶ **Definition 22.** Let $\varphi(X, \vec{P})$ be a MSO formula on $A$-labelled trees with monadic parameters $X$ and $\vec{P} = P_1, \ldots, P_n$. We say that $\psi(X, \vec{P})$ is an *uniformization of* $\varphi$ if the following conditions are satisfied for every tree $t$, values of parameters $\vec{P}$, and sets $X_1, X_2 \subseteq \mathrm{dom}(t)$:

$$\begin{aligned}
\Big(\exists_X\ \varphi(X, \vec{P})\Big) &\quad\Leftrightarrow\quad \Big(\exists_X\ \psi(X, \vec{P})\Big)\\
\psi(X_1, \vec{P}) &\quad\Rightarrow\quad \varphi(X_1, \vec{P})\\
\Big(\psi(X_1, \vec{P}) \wedge \psi(X_2, \vec{P})\Big) &\quad\Rightarrow\quad X_1 = X_2
\end{aligned}$$

That is, whenever it is possible to pick some $X$ satisfying $\varphi(X, \vec{P})$ then $\psi$ chooses exactly one such $X$. For simplicity, we allow a (possible empty) list of additional parameters $\vec{P}$ and we assume that the first variable is the one that is supposed to be uniformized.

Now, Conjecture 1 says that the following formula does not have uniformization:

$\mathrm{CHOICE}(x, X) :=$ the given tree is thin and $x \in X$.

A simple interpretation argument shows that Conjecture 1 is equivalent to the non-uniformizability of the following simpler formula.

$\mathrm{LEAF} - \mathrm{CHOICE}(x) :=$ the given tree is thin and $x$ is a leaf.

The following proposition expresses the crucial technical condition, allowing to entail properties of thin algebras using Conjecture 1.

▶ Proposition 23 (assuming Conjecture 1). Assume that $\alpha\colon (H, V) \to (H', V')$ is a surjective homomorphism between two finite thin algebras. Let $t$ be a tree and $\tau'$ be a consistent marking of $t$ by $H'$. Then there exists a consistent marking $\tau$ of $t$ by $H$ such that $\forall_{w \in \mathrm{dom}(t)}\ \alpha(\tau(w)) = \tau'(w)$.

**Sketch of the proof:** assume contrary and fix a regular pair $(t_0, \tau')$ such that there is no marking $\tau$ as above. Consider the standard automaton-pathfinder game, where the automaton proposes a marking $\tau$ and the pathfinder picks directions to show that $\tau$ does not satisfy the above conditions. Since there is no such $\tau$, so pathfinder has a finite memory winning strategy $\sigma$. Now, given a thin tree $t$ we can define the unique consistent marking $\tau$ that satisfies $\alpha(\tau) = \tau'$ on $t$. The play resulting in pathfinder playing $\sigma$ and automaton playing $\tau$ must end in a leaf of $t$. ◀

The second important tool in our analysis enables to make a connection between uniformized relations and induced markings. A formal definition of a transducer and a proof of the following theorem are given in Appendix B.

▶ **Theorem 24.** *Assume that $L_A \subseteq \mathrm{Tr}_A, L_M \subseteq \mathrm{Tr}_{A\times M}$ are regular languages of trees for two ranked alphabets $A, M$ such that $L_A$ is a projection of $L_M$ onto $A$. Assume that $\forall_{t_A \in L_A} \exists!_{t_M \in \mathrm{Tr}_M} (t_A, t_M) \in L_M$. Then, there exist:*

- *a compositional homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to S$ into a finite thin algebra $S$,*
- *a deterministic finite state transducer that reads the marking induced by $\alpha$ on a given tree $t_A$ and outputs the labelling $t_M$ such that $(t_A, t_M) \in L_M$, whenever such $t_M$ exists.*

Now we can present two algebraic statements that are equivalent to Conjecture 1.

▶ **Theorem 25.** *The following conditions are equivalent:*

1. *Conjecture 1 holds.*
2. *For every finite thin algebra $(H, V)$ over an alphabet $A = (N, L)$ and every tree $t \in \mathrm{Tr}_A$ there exists a consistent marking of $t$ by types in $H$.*
3. *For every finite thin algebra $(H, V)$ over the alphabet $A_b = (\{n\}, \{b\})$ there exists a consistent marking of the full tree $t_n \in \mathrm{Tr}_{A_b}$ by types in $H$.*

Note that in the above theorem algebras $(H, V)$ come without any homomorphism from $(\mathrm{Tr}_A, \mathrm{Con}_A)$, so there is no notion of the induced marking.

**Proof.** First we show $1 \Rightarrow 2$. Let $(H, V)$ be a finite thin algebra over an alphabet $A = (N, L)$. Let $(H', V') = (\{h_0\}, \{v_0\})$ be the singleton thin algebra with $b() = h_0$ for every $b \in L$. There is a unique homomorphism $\alpha\colon (H, V) \to (H', V')$. Take any tree $t \in \mathrm{Tr}_A$. Let $\tau'$ be the consistent marking of $t$ that is constant equal $h_0$ on $\mathrm{dom}(t)$. By Proposition 23 there exists a consistent marking of $t$ by types in $H$.

Of course Item 3 follows from Item 2.

For $3 \Rightarrow 1$ we assume that $\psi(x)$ is an MSO formula uniformizing $\mathrm{LEAF}-\mathrm{CHOICE}$. Using Theorem 24 we find a deterministic transducer $\mathcal{T}$ that reads types of subtrees of a given thin tree (with respect to some homomorphism $\alpha$ into a finite thin algebra $(H, V)$) and outputs directions towards the chosen leaf. Let $(H', V')$ be the subalgebra of $(H, V)$ containing $\alpha$-types of $(\mathrm{Th}_A, \mathrm{ThCon}_A)$. By Item 3 there is a consistent marking $\tau$ of the full tree $t_n$ by types in $H'$. We can consider the sequence of directions $\pi$ given by $\mathcal{T}$ on $(t_n, \tau)$. Since $t$ does not have any leaf, so $\pi$ is infinite. Now, we can substitute all subtrees that are not on $\pi$ by thin trees of the respective types given by $\tau$. The result is a thin tree $t'$ such that the directions produced by $\mathcal{T}$ do not reach any leaf of $t'$ — a contradiction. ◀

## 4.1 Prophetic thin algebras

It turns out that (assuming Conjecture 1) the class of finite prophetic thin algebras has a number of nice properties. Most of them can be read as properties of the class of bi-unambiguous languages. To emphasise that we work under the assumption of Conjecture 1, we explicitly put it as a pre-assumption in the statements.

▶ **Theorem 26** (Conjecture 1)**.** *Let $(H, V)$ be a prophetic thin algebra over an alphabet $A$. There exists a unique homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$. Additionally, $\alpha$ is compositional.*

**Proof.** The uniqueness of the homomorphism was observed in Section 3.2. By Theorem 25 and the fact that $(H, V)$ is prophetic, every tree $t \in \mathrm{Tr}_A$ has exactly one consistent marking $\tau_t$ by types in $H$. Let us define $\alpha(t) = \tau_t(\epsilon)$. Clearly $\alpha$ is a compositional homomorphism — if $t$ is an extension of $c$ then the consistent marking $\tau_t$ must agree with the types in the leafs of $c$. ◀

▶ **Theorem 27** (Conjecture 1). *Let $\beta\colon S \to S'$ be a surjective homomorphism between two finite thin algebras. If $S$ is prophetic then $S'$ is also prophetic.*

**Proof.** First fix the homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to S$ given by Theorem 26. Note that $\beta \circ \alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ is a compositional homomorphism. Assume that $S'$ is not prophetic, so there exists a tree $t$ with two consistent markings $\sigma, \sigma'$ by types of $S'$. Without loss of generality we can assume that $\sigma$ is the marking induced by $\beta \circ \alpha$ and $\sigma'(\epsilon) \neq \sigma(\epsilon)$. Let $\tau$ be the marking by types in $S$ induced by $\alpha$ on $t$. Observe that pointwise $\beta(\tau) = \sigma$. By Proposition 23 there exists a consistent marking $\tau'$ of $t$ such that pointwise $\beta(\tau') = \sigma'$. Therefore, $\tau, \tau'$ are two distinct consistent markings of $t$ by types in $H$ — a contradiction. ◀

The following remark ends the proof of Item 5 of Theorem 5.

▶ Remark (Conjecture 1). If $L \subseteq \mathrm{Tr}_A$ is bi-unambiguous then $S_L$ is prophetic.

**Proof.** Since $L$ is bi-unambiguous so by Theorem 15 there exists a surjective homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ that recognises $L$ such that $(H, V)$ is a finite prophetic thin algebra. Since $S_L$ is a syntactic algebra of $L$ so there exists a surjective homomorphism $\beta\colon (H, V) \to S_L$. By Theorem 27 we obtain that $S_L$ is also prophetic. ◀

The next statement shows that prophetic thin algebras form a robust class from the point of view of universal algebra (see [4] for an introduction to this field). The proof follows directly from Theorem 27 and Fact 14.

▶ Remark (Conjecture 1). The class of finite prophetic thin algebras is a pseudo-variety: it is closed under homomorphic images, subalgebras, and finite direct products.

## 5 Uniformizability results on thin trees

In this section we study Conjecture 1 in the context of related uniformization problems on thin trees. One of the notions we concentrate on are *skeletons* of thin trees, introduced in [2].

▶ **Definition 28.** Let $t \in \mathrm{Tr}_A$ be a tree. We say that $\sigma \subseteq \mathrm{dom}(t)$ is a *skeleton of $t$* if $\epsilon \notin \sigma$ and the following conditions are satisfied:
- if $w \in \mathrm{dom}(t)$ is an internal node of $t$ then $\sigma$ contains exactly one of the vertices $wl, wr$,
- if $\pi$ is an infinite branch of $t$ then all but finitely many vertices on $\pi$ belong to $\sigma$.

We identify a set $\sigma \subseteq \mathrm{dom}(t)$ with its characteristic function $\sigma \in \mathrm{Tr}_{(\{0,1\},\{0,1\})}$. By $\mathrm{SKEL}(\sigma)$ we denote the MSO formula expressing the above properties.

The following proposition expresses the crucial property of skeletons, see [2].

▶ Proposition 29 ([2]). A tree $t$ is thin if and only if there exists a skeleton of $t$.

Note that a thin tree may have multiple skeletons. The main idea behind skeletons is that they provide decompositions of thin trees: every skeleton $\sigma$ of a thin tree $t$ defines *the main branch of $\sigma$* that follows $\sigma$ from the root of $t$ and along this branch *simpler thin trees* are plugged. The second bullet in the definition of skeletons means that such a decomposition is well-founded — we can go off the main branch only finitely many times.

## 5.1 Non-uniformizability

In this section we give the following two negative results.

▶ **Theorem 30.** *There is no MSO formula uniformizing* $\mathrm{SKEL}(\sigma)$*.*

▶ **Theorem 31.** *The language* $\mathrm{Th}_{A_b} \subset \mathrm{Tr}_{A_b}$ *of thin trees over the alphabet* $A_b$ *is ambiguous.*

The above theorem can be seen as complementing the following theorem from [2] (adjusted to the case of trees instead of forests).

▶ **Theorem 32** (Theorem 12 from [2])**.** *For every regular language* $L \subseteq \mathrm{Tr}_A$ *that contains only thin trees there exists a nondeterministic automaton* $\mathcal{A}$ *such that* $\mathrm{L}(\mathcal{A}) \cap \mathrm{Th}_A = L$ *and* $\mathcal{A}$ *has at most one accepting run on every thin tree.*

Therefore, every regular tree language containing only thin trees is unambiguous *relatively to thin trees*. But, by Theorem 31, it is the best we can get: even the language of all thin trees is ambiguous among all trees.

The proofs base on two observations, first of them is the existence of transducers, see Theorem 24. The second ingredient is a weaker version of Item 2 in Theorem 25. It is motivated by a similar result on preclones, see [1].

▶ **Theorem 33.** *For every finite thin algebra* $(H, V)$ *over an alphabet* $A = (N, L)$ *there exists a thick tree* $t \in \mathrm{Tr}_A$ *and a consistent marking* $\tau$ *of* $t$ *by types in* $H$*.*

The proof uses Green's relations [7] in the monoid $V$ of a given thin algebra to find an appropriate idempotent $e \in V$ that enables to construct a tree $t$ . The constructed tree is thick but it is not full — many subtrees of $t$ are thin and contain leafs.

Now we can present a sketch of the proof of Theorem 30.

**Proof.** Assume that $\psi(\sigma)$ is a uniformization of $\mathrm{SKEL}(\sigma)$. Using Theorem 24 we find: a homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ info a finite thin algebra and a transducer $\mathcal{T}$. Let $(H', V')$ be the subalgebra of $(H, V)$ that is the image of $(\mathrm{Th}_A, \mathrm{ThCon}_A)$ under $\alpha$.

Using Theorem 33 we construct a thick tree $t$ with a consistent marking $\tau$ by types in $H'$. We run the transducer $\mathcal{T}$ on $(t, \tau)$ what results in a labelling $t_M$ of $t$. Since $t$ is not thin so it has no skeleton. Therefore, one of the conditions for skeletons is not satisfied by $t_M$. Assume that there exists a branch $\pi$ of $t$ such that $t_M$ labels infinitely many vertices on $\pi$ by 0. The other possibility is similar but simpler. Now we can plug thin trees of types given by $\tau$ along $\pi$ obtaining $t'$. By the construction, $t'$ is thin and $\tau$ equals along $\pi$ the marking of $t'$ induced by $\alpha$. Therefore, we can run $\mathcal{T}$ on $(t', \tau_\alpha(t))$ obtaining a result $t'_M$ that agrees with $t_M$ on $\pi$. It is a contradiction since $\mathcal{T}$ is supposed to produce a correct skeleton for every thin tree and $t'_M$ violates assumptions of skeleton on $\pi$. ◀

## 5.2 Degrees of uniformization

In this section we study relationships between uniformization problems on thin trees. The results of this section were found as answers to questions posed by Alexander Rabinovich.

The following definition is motivated by *degrees of selection* studied in [22].

▶ **Definition 34.** We say that a formula $\varphi(X, \vec{P})$ has *higher degree of uniformization* than $\varphi'(Y, \vec{R})$ (denoted $\varphi'(Y, \vec{R}) \preceq_{uni} \varphi(X, \vec{P})$) if there exists a formula $\psi(Y, \vec{R})$ that is defined in MSO extended by an additional predicate $U(X, \vec{P})$ and $\psi(Y, \vec{R})$ uniformizes $\varphi(Y, \vec{R})$ whenever $U$ is interpreted as any relation uniformizing $\varphi(X, \vec{P})$.

▶ Fact 35. The relation $\preceq_{uni}$ is transitive and reflexive. If $\varphi'(X,\vec{P}) \preceq_{uni} \varphi(Y,\vec{R})$ and $\varphi(Y,\vec{R})$ is uniformizable then so is $\varphi'(X,\vec{P})$.

We say that $\varphi$ is *on thin trees* if $\varphi$ is satisfied only on thin trees. The following theorem implies that $\mathrm{SKEL}(\sigma)$ is maximal with respect to $\preceq_{uni}$ among MSO formulas on thin trees.

▶ **Theorem 36.** *For every formula $\varphi(X,\vec{P})$ on thin trees there exists a formula $\varphi'(X,\vec{P},\sigma)$ that uniformizes $\bar{\varphi}(X,\vec{P},\sigma) := \varphi(X,\vec{P}) \wedge \mathrm{SKEL}(\sigma)$.*

The proof is based on the fact that every MSO-definable relation on $\omega$-words is uniformizable, see [24, 12, 20]. Since every skeleton gives a decomposition of a given tree as disjoint branches, so we can uniformize the given formula $\varphi$ independently on these branches. By well-foundedness of skeletons the result is well-defined. The above theorem can also be derived from the proof of Theorem 6.7 in [12] but in a less explicit way.

It turns out that uniformization of $\mathrm{SKEL}(\sigma)$ is connected with definability of well-orderings on thin trees. We say that a formula $\psi(x,y)$ *defines well-order on thin trees* if for every thin tree $t \in \mathrm{Tr}_{A_b}$ the relation $<_\psi$ defined as $(x <_\psi y \Leftrightarrow \psi(x,y))$ is a linear order on $\mathrm{dom}(t)$ and there is no infinite descending sequence of $<_\psi$. In the rest of this section we show that uniformizations of skeletons and definable well-orderings are equivalent — it is possible to define one of them basing on the other.

One direction is simple : the structure of a skeleton gives a natural lexicographic well-order of vertices of a given thin tree. The other direction is a bit more involved: given any definable well-order of a given thin tree $t$ we need to define a skeleton of $t$.

▶ **Theorem 37.** *If there exists an MSO-definable well-order on thin trees then there exists a uniformization of* $\mathrm{SKEL}(\sigma)$.

The following remark follows from Theorem 30 and Theorem 37. It should be connected with a result of [5] stating that the MSO theory of the full binary tree extended with any well-order is undecidable.

▶ Remark. There is no MSO formula $\psi(x,y)$ that defines well-order on thin trees.

**Acknowledgements.** The authors would like to thank Mikołaj Bojańczyk, Henryk Michalewski, Damian Niwiński, Alexander Rabinovich, and Igor Walukiewicz for fruitful discussions on the subject.

## References

1 Mikołaj Bojańczyk. Algebra for trees. A draft version of a chapter that will appear in the AutomathA handbook, 2010.
2 Mikołaj Bojańczyk, Tomasz Idziaszek, and Michał Skrzypczak. Regular languages of thin trees. In *STACS 2013*, volume 20 of *LIPIcs*, pages 562–573, 2013.
3 Nicolas Bousquet and Christof Löding. Equivalence and inclusion problem for strongly unambiguous Büchi automata. In *LATA*, pages 118–129, 2010.
4 Stanley Burris and H. P. Sankappanavar. *A Course in Universal Algebra.* Number 78 in Graduate Texts in Mathematics. Springer-Verlag, 1981.
5 Arnaud Carayol, Christof Löding, Damian Niwiński, and Igor Walukiewicz. Choice functions and well-orderings over the infinite binary tree. *Cent. Europ. J. of Math.*, 8:662–682, 2010.
6 Olivier Carton, Dominique Perrin, and Jean Éric Pin. Automata and semigroups recognizing infinite words. In *Logic and Automata, History and Perspectives*, pages 133–167, 2007.

**7** James Alexander Green. On the structure of semigroups. *Annals of Mathematics*, 54(1):163–172, 1951.

**8** Yuri Gurevich and Saharon Shelah. Rabin's uniformization problem. *J. Symb. Log.*, 48(4):1105–1119, 1983.

**9** Szczepan Hummel. Unambiguous tree languages are topologically harder than deterministic ones. In *GandALF*, pages 247–260, 2012.

**10** Tomasz Idziaszek. *Algebraic methods in the theory of infinite trees.* PhD thesis, University of Warsaw, 2012. unpublished.

**11** Orna Kupferman, Shmuel Safra, and Moshe Y. Vardi. Relating word and tree automata. In *LICS*, pages 322–332. IEEE Computer Society, 1996.

**12** Shmuel Lifsches and Saharon Shelah. Uniformization and skolem functions in the class of trees. *J. Symb. Log.*, 63(1):103–127, 1998.

**13** Filip Murlak. The Wadge hierarchy of deterministic tree languages. *Logical Methods in Computer Science*, 4(4), 2008.

**14** Damian Niwiński and Igor Walukiewicz. Ambiguity problem for automata on infinite trees. unpublished, 1996.

**15** Damian Niwiński and Igor Walukiewicz. Relating hierarchies of word and tree automata. In *STACS*, pages 320–331, 1998.

**16** Damian Niwiński and Igor Walukiewicz. A gap property of deterministic tree languages. *Theor. Comput. Sci.*, 1(303):215–231, 2003.

**17** Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electr. Notes Theor. Comput. Sci.*, 123:195–208, 2005.

**18** Dominique Perrin and Jean Éric Pin. *Infinite Words: Automata, Semigroups, Logic and Games.* Elsevier, 2004.

**19** Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. of the American Math. Soc.*, 141:1–35, 1969.

**20** Alexander Rabinovich. On decidability of monadic logic of order over the naturals extended by monadic predicates. *Information and Computation*, 205(6):870–889, 2007.

**21** Alexander Rabinovich and Sasha Rubin. Interpretations in trees with countably many branches. In *LICS*, pages 551–560. IEEE, 2012.

**22** Alexander Rabinovich and Amit Shomrat. Selection in the monadic theory of a countable ordinal. *J. Symb. Log.*, 73(3):783–816, 2008.

**23** Saharon Shelah. The monadic theory of order. *The Annals of Mathematics*, 102(3):379–419, 1975.

**24** Dirk Siefkes. The recursive sets in certain monadic second order fragments of arithmetic. *Arch. Math. Logik*, 17(1–2):71–80, 1975.

**25** Richard Edwin Stearns and Harry B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM J. Comput.*, 14(3):598–611, 1985.

**26** Wolfgang Thomas. Languages, automata and logics. Technical Report 9607, Institut für Informatik und Praktische Mathematik, Christian-Albsechts-Universität, Kiel, Germany, 1996.

**27** Boris A. Trakhtenbrot. Finite automata and the monadic predicate calculus. *Siberian Mathematical Journal*, 3(1):103–131, 1962.

**28** Thomas Wilke. An algebraic theory for regular languages of finite and infinite words. *Int. J. Alg. Comput.*, 3:447–489, 1993.

**29** Thomas Wilke. An algebraic characterization of frontier testable tree languages. *Theor. Comput. Sci.*, 154(1):85–106, 1996.

**30** Thomas Wilke. Classifying discrete temporal properties. Habilitationsschrift, Universitat Kiel, apr. 1998.

# A Thin algebra

First, let us write explicitly all the axioms of thin algebra (we assume that $h, h_l, h_r \in H$ and $u, v, w \in V$):

1. $(u \cdot v) \cdot w = u \cdot (v \cdot w)$,
2. $(u \cdot v) \cdot h = u \cdot (v \cdot h)$,
3. $(uv)^\infty = u(vu)^\infty$,
4. $(v^n)^\infty = v^\infty$ for every $n \geq 1$,
5. $a(h_l, \square) \cdot h_r = a(\square, h_r) \cdot h_l$.

Let $R_A$ be the set of all regular thin trees over a ranked alphabet $A = (N, L)$. Let $C_A$ be the set of all regular thin contexts over $A$. Note that $(R_A, C_A)$ has the natural structure of a thin algebra over $A$.

▶ Fact 38. $(R_A, C_A)$ is the free algebra in the class of thin algebras over the alphabet $A$.

**Proof.** See [10] for the proof of this fact in the context of forests. ◀

The rest of this section is devoted to showing the following theorem.

▶ **Theorem 9.** *For every regular tree language $L$ there exists a* syntactic morphism *for $L$: a finite thin algebra $S_L = (H, V)$ (called a* syntactic algebra *of $L$) and a homomorphism $\alpha_L : (\mathrm{Tr}_A, \mathrm{Con}_A) \to S_L$ such that:*

- *$\alpha_L$ is surjective, compositional, and recognises $L$,*
- *for every $h \in H$ the language $L_h := \alpha_L^{-1}(\{h\})$ is regular,*
- *if $\alpha : (\mathrm{Tr}_A, \mathrm{Con}_A) \to S$ is surjective and recognises $L$ then there is exactly one homomorphism $\beta : S \to S_L$ such that $\beta \circ \alpha = \alpha_L$.*

*A syntactic algebra $S_L$ and languages $L_h$ can be effectively computed basing on a non-deterministic automaton recognising $L$.*

A syntactic algebra $S_L$ of a given language $L$ can be constructed using standard tools of universal algebra (namely the congruence $\sim_L$). What remains is to show that the constructed algebra is finite. For this purpose we provide some homomorphism $\alpha : (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ that recognises $L$ (see Theorem 41 of [10]) and such that $(H, V)$ is a finite thin algebra. Then, by the universal property of the syntactic algebra, $S_L$ is a surjective image of $(H, V)$, thus $S_L$ is finite.

Let us define a relation $\sim_L$ on the sets $\mathrm{Tr}_A$ and $\mathrm{Con}_A$. We assume that $t, t' \in \mathrm{Tr}_A$, $c, c' \in \mathrm{Con}_A$, and $D$ denotes the set of all multi-contexts over the alphabet $A$.

$$\begin{aligned} t \sim_L t' &\iff \text{for every } d \in D \text{ we have } (d \cdot t \in L \Leftrightarrow d \cdot t' \in L) \\ c \sim_L c' &\iff \text{for every } d \in D \text{ and } s \in \mathrm{Tr}_A \text{ we have } (d \cdot (c \cdot s) \in L \Leftrightarrow d \cdot (c' \cdot s) \in L) \end{aligned}$$

▶ Fact 39. The relation $\sim_L$ is a congruence on $(\mathrm{Tr}_A, \mathrm{Con}_A)$ with respect to the operations of thin algebra. Moreover, if $\alpha : (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ recognises $L$ then (by compositionality of $\alpha$)

$$\alpha(t) = \alpha(t') \Longrightarrow t \sim_L t' \text{ and } \alpha(c) = \alpha(c') \Longrightarrow c \sim_L c'. \tag{4}$$

We define $S_L = (H_L, V_L)$ as the quotient of $(\mathrm{Tr}_A, \mathrm{Con}_A)$ by the relation $\sim_L$ defined above. Since $\sim_L$ is a congruence, so $S_L$ has a structure of thin algebra. We define $\alpha_L$ as the quotient morphism of $\sim_L$.

Now we construct some finite thin algebra recognising $L$. Let $\mathcal{A}$ be a nondeterministic automaton over an alphabet $A$ with states $Q$ such that $\mathcal{A}$ recognises $L$. Assume that $\mathcal{A}$ uses priorities $\{0,\ldots,k\}$. First, recall the definition of $Q_{\mathcal{A}}(t)$ from (1):

$$Q_{\mathcal{A}}(t) = \{q \in Q :\ \exists_\rho\ \rho \text{ is a consistent run of } \mathcal{A} \text{ on } t \text{ with value } q\} \subseteq 2^Q.$$

Similarly, if $c$ is a context over $A$ then let $\Delta_{\mathcal{A}}(c)$ contain those pairs $(q,i,p) \in Q \times \{0,\ldots,k\} \times Q$ such that there exists a consistent run $\rho$ of $\mathcal{A}$ on $c$ with the value $q$, the value in the hole $p$, and the maximal priority on the path from the root to the hole equal $i$.

Now consider the function

$$\alpha_{\mathcal{A}} \colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (2^Q, 2^{Q\times\{0,\ldots,k\}\times Q})$$

that assigns to a tree $t \in \mathrm{Tr}_A$ the set $Q_{\mathcal{A}}(t)$ and assigns to a context $c \in \mathrm{Con}_A$ the set $\Delta_{\mathcal{A}}(c)$.

▶ Fact 40. The function $\alpha_{\mathcal{A}}$ induces uniquely the structure of thin algebra on its image $S_{\mathcal{A}} := (H_{\mathcal{A}}, V_{\mathcal{A}}) \subseteq (2^Q, 2^{Q\times\{0,\ldots,k\}\times Q})$ in such a way that $\alpha_{\mathcal{A}}$ becomes a compositional homomorphism of thin algebras. Moreover, $\alpha_{\mathcal{A}}$ recognises $\mathrm{L}(\mathcal{A})$, since

$$\mathrm{L}(\mathcal{A}) = \alpha_{\mathcal{A}}^{-1}\left(\{h \in H_{\mathcal{A}} :\ h \cap I^{\mathcal{A}} \neq \emptyset\}\right).$$

The algebra $S_{\mathcal{A}}$ along with the homomorphism $\alpha_{\mathcal{A}}$ defined above is called the *automaton algebra for* $\mathcal{A}$. The following lemma presents an important feature of this algebra.

▶ **Lemma 41.** *Assume that $\mathcal{A}$ is a nondeterministic tree automaton over an alphabet $A$, $t \in \mathrm{Tr}_A$ is a tree, and $\tau$ is a consistent marking of $t$ by types in $H_{\mathcal{A}}$. Let $q \in Q^{\mathcal{A}}$ be a state of $\mathcal{A}$. The following conditions are equivalent:*
- $q \in \tau(\epsilon)$
- *There exists a run (possibly not consistent) $\rho$ of $\mathcal{A}$ on $t$ with value $q$ such that for every vertex $w \in \mathrm{dom}(t)$ we have $\rho(w) \in \tau(w)$. Additionally, for every infinite branch $\pi$ of $t$ there exists a run $\rho_\pi$ as above that is consistent on $\pi$.*

**Proof.** First assume that $q \in \tau(\epsilon)$. We inductively show that there exists a run of $\mathcal{A}$ on $t$ satisfying $\rho(w) \in \tau(w)$. Assume that $t = a(t_l, t_r)$ for a pair of trees $t_l, t_r$. Let $h = \tau(\epsilon)$, $h_l = \tau(l)$, and $h_r = \tau(r)$. We need to find a transition $(q, q_l, a, q_r) \in \delta_2^{\mathcal{A}}$ such that $q_l \in h_l$ and $q_r \in h_r$. Let $t'_l, t'_r$ be trees that are mapped by $\alpha_{\mathcal{A}}$ to $h_l, h_r$ respectively. Observe that

$$q \in h = a(h_l, h_r) = \alpha_{\mathcal{A}}\left(a(t'_l, t'_r)\right),$$

therefore there exists a consistent run with value $q$ on $a(t'_l, t'_r)$. The first transition used by this run gives us the states $q_l \in h_l, q_r \in h_r$. Note that if $w$ is a leaf of $t$ and $q \in \tau(w)$ then $(q, t(w)) \in \delta_0$, so the constructed run is also consistent in leafs.

Using the above observation, it is enough to construct a run $\rho$ along $\pi$ that satisfies $\rho(w) \in \tau(w)$ for every $w$ that is off $\pi$ — it will extend to a run on the subtree $t\restriction_w$. The existence of such a run follows from the definition of operations of thin algebra, see Section 4.4.1 of [10] — the fact that $q \in \tau(\epsilon)$ comes from the fact that for every Ramsey decomposition $s \cdot e^\infty$ of the contexts along the branch $\pi$, there is a loop of transitions in $s \cdot e^\infty$ starting in $q$ and satisfying the parity condition.

Now assume that the second bullet of the statement is satisfied. We want to show that $q \in \tau(\epsilon)$. If the tree $t$ is finite then $q \in \tau(\epsilon)$ by induction on the height of $t$. Otherwise, there exists an infinite branch $\pi$ of $t$ and similarly as above, any run $\rho_\pi$ consistent on $\pi$ is a witness that $q \in h$. ◀

▶ **Lemma 42.** *The automaton morphism $\alpha_{\mathcal{A}}\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H_{\mathcal{A}}, V_{\mathcal{A}})$ can be computed effectively basing on $\mathcal{A}$. The syntactic algebra $S_L$ for $L = \mathrm{L}(\mathcal{A})$ and the unique homomorphism $\beta\colon (H_{\mathcal{A}}, V_{\mathcal{A}}) \to S_L$ are computable effectively basing on $\alpha_L$.*

**Proof.** The homomorphism $\alpha_{\mathcal{A}}$ and the structure of thin algebra of $(H_{\mathcal{A}}, V_{\mathcal{A}})$ can be written *by hand*, see Section 4.4.1 from [10].

The homomorphism $\beta$ can be computed using Moore's algorithm, see Lemma 23 of the cited thesis. The construction is similar to the minimisation of a finite deterministic automaton: we mark pairs of elements of $H_{\mathcal{A}}$ and $V_{\mathcal{A}}$ as non-equivalent. We start with all the pairs in $F \times (H_{\mathcal{A}} \setminus F)$ where $\alpha_{\mathcal{A}}^{-1}(F) = L$. Then we iteratively add a pair $(s, s')$ whenever there is an operation of thin algebra (with some parameters fixed) that maps $s, s'$ into $r, r'$ respectively and $(r, r')$ is a marked pair. After a finite number of steps no new pair can be marked and the set of non-marked pairs is a congruence $\sim$ on $(H_{\mathcal{A}}, V_{\mathcal{A}})$. $\beta$ can be defined as the quotient morphism induced by $\sim$. ◀

# B Transducer for an uniformized relation

Let $A = (N, L), M = (M_2, M_0)$ be a pair of ranked alphabets. Let $B = N \sqcup L$. A *transducer from $A$ to $M$* is a deterministic device $\mathcal{T} = (Q, \delta, q_I)$ such that:

1. $Q$ is a finite set of states,
2. $q_I \in Q$ is an initial state,
3. $\delta$ is a pair of functions $\delta_2, \delta_0$,
4. $\delta_2\colon Q \times B \times N \times B \to Q \times M_2 \times Q$ determines transitions in internal nodes,
5. $\delta_0\colon Q \times L \to M_0$ determines transitions in leafs.

Note that a transition in an internal node $w$ takes three letters as the input: the letter in $wl$, $w$, and $wr$.

For every tree $t \in \mathrm{Tr}_A$ a transducer $\mathcal{T}$ defines inductively the labelling $\mathcal{T}(t)$ of $t$ by letters in $M$. The definition is inductive. We start in $w = \epsilon$ in the state $q_I$. Assume that the transducer reached a vertex $w \in \mathrm{dom}(t)$ in a state $q$. If $w$ is a leaf then we put $\mathcal{T}(t)(w) = \delta_0(q, t(w))$. Otherwise, let $a_l, a, a_r$ be letters of $t$ in $wl$, $w$, $wr$ respectively. Then let $\delta_2(q, a_l, a, a_r) = (q_l, m, q_r)$, put $\mathcal{T}(t)(w) = m$, and continue in $wl, wr$ in states $q_l, q_r$ respectively.

▶ Fact 43. The value $\mathcal{T}(t)(w)$ in a vertex $w \in \mathrm{dom}(t)$ depends on the letters of $t$ in vertices of the form $v, vl, vr$ for $v \prec w$. That is, if $t, t'$ agree on all vertices $v, vl, vr$ for $v \prec w$ then $\mathcal{T}(t)(w) = \mathcal{T}(t')(w)$.

▶ **Theorem 24.** *Assume that $L_A \subseteq \mathrm{Tr}_A, L_M \subseteq \mathrm{Tr}_{A\times M}$ are regular languages of trees for two ranked alphabets $A, M$ such that $L_A$ is a projection of $L_M$ onto $A$. Assume that $\forall_{t_A \in L_A} \exists!_{t_M \in \mathrm{Tr}_M} (t_A, t_M) \in L_M$. Then, there exist:*

- *a compositional homomorphism $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to S$ into a finite thin algebra $S$,*
- *a deterministic finite state transducer that reads the marking induced by $\alpha$ on a given tree $t_A$ and outputs the labelling $t_M$ such that $(t_A, t_M) \in L_M$, whenever such $t_M$ exists.*

▶ **Example 44.** Let $\mathcal{A}$ be an unambiguous tree automaton. Let $L_A = \mathrm{L}(\mathcal{A})$ and $L_M$ contain pairs $(t, \rho)$ where $\rho$ is an accepting run of $\mathcal{A}$ on $t \in \mathrm{Tr}_A$. Then, the above theorem states that there exists a transducer that reads the marking induced by some homomorphism $\alpha$ on a given tree $t \in \mathrm{L}(\mathcal{A})$ and produces the accepting run of $\mathcal{A}$ on $t$.

A simple proof of the above theorem can be given using the composition method (see [23]). This proof was suggested by Mikołaj Bojańczyk as a simplification of an earlier proof given by the authors. However, since we are focused on automata, we only sketch it here and give a longer self-contained proof below. Assume that there is an MSO formula defining language $L_M$ that has quantifier depth $n$. Let $|M| = k$ and let $\alpha\colon (\mathrm{Tr}_A, \mathrm{Con}_A) \to (H, V)$ be a homomorphism that recognises all the $(n+k+1)$-types of MSO over $A$. In a vertex $w$ the transducer $\mathcal{T}$ can store in its memory the $(n+m+1)$-type of the currently read context. Then, given $(n+k+1)$-types of both subtrees under $w$, it can compute the $(n+k)$-type of the tree $t[x := w]$ with the current vertex $w$ denoted by an additional variable $x$. The $(n+k)$-type of $t[x := w]$ is enough to ask about the truth value of the following formulas (for every $a \in M_2$):

there is a labelling $t_M \in L_M$ of $t[x := w]$ such that $t_M(x) = a$.

If there is any such labelling $t_M$, then the above formula is true for exactly one letter $a \in M_2$. The transducer $\mathcal{T}$ outputs this letter in $w$ and proceeds in $wl, wr$ updating the type of the context.

The rest of this section is devoted to an automata-based proof of Theorem 24.

Let $\mathcal{A}$ be some nondeterministic tree automaton recognising the language $L_M$. Let $Q$ be the set of states of $\mathcal{A}$. Consider a modification $\bar{\mathcal{A}}$ of the automaton $\mathcal{A}$ where letters of $M$ used in transitions are removed. Formally, $\bar{\mathcal{A}}$ is a projection of $\mathcal{A}$ from the alphabet $A \times M$ to $A$. Note that $\mathrm{L}(\bar{\mathcal{A}}) = L_A$. Let us fix the alphabet $G = (2^Q, 2^Q)$.

Let $\alpha_{\bar{\mathcal{A}}}$ be the automaton morphism into the automaton algebra $(H_{\bar{\mathcal{A}}}, V_{\bar{\mathcal{A}}})$ for $\bar{\mathcal{A}}$. Let $t_A \in \mathrm{Tr}_A$ be a tree. Let $\tau(t_A) = \tau_{\bar{\mathcal{A}}}(t_A)$ be the marking induced by the automaton morphism $\alpha_{\bar{\mathcal{A}}}$ on $t_A$, that is $\tau(t_A)(w) = Q_{\mathcal{A}}(t_A{\restriction_w})$.

The construction goes as follows. The input alphabet is $A \times G$. The set of states $Q^{\mathcal{T}}$ of $\mathcal{T}$ is $2^Q$. The state $\emptyset \in Q^{\mathcal{T}}$ is a sink state reached if the given tree does not belong to $L_A$.

The invariant for non-sink states is: if $\mathcal{T}$ is in a vertex $w$ and it have assigned letters $m_v \in M$ to all vertices $v \prec w$ then the state $S_w$ of $\mathcal{T}$ in $w$ satisfies:

$$S_w = \{q \in Q : \text{exists an accepting run of } \bar{\mathcal{A}} \text{ on } t_A \text{ using letters } m_v \text{ in vertices } v \prec w\}. \quad (5)$$

We will show that the invariant can be preserved. Let us fix a moment during the computation of $\mathcal{T}$: we are in a vertex $w \in \mathrm{dom}(t_A)$. We can assume that $w$ is an internal node of $t_A$. We have already assigned letters $m_v \in M$ to all nodes $v \prec w$. The marking $\tau(t_A)$ gives us sets $Q_{wl}, Q_{wr} \subseteq Q$ in nodes $wl, wr$ respectively. The current state of $\mathcal{T}$ is a set of states $S_w \subseteq Q$.

Consider the following set of letters:

$$P_w = \left\{ m \in M_2 : \exists_{(q, q_l, (t_A(w), m), q_r) \in \delta_2^{\mathcal{A}}}\; q \in S_w \wedge q_l \in Q_{wl} \wedge q_r \in Q_{wr} \right\}.$$

If $P_w = \emptyset$ then let $\mathcal{T}$ fall in a sink state $\emptyset \in 2^Q$ and from that point on output some fixed letters (of arity 2 and 0 respectively) $(m_2, m_0) \in M$. We will show that during the run of $\mathcal{T}$ on any tree $t_A \in L_A$ the sets $P_w$ are nonempty. But first we show the following lemma.

▶ **Lemma 45.** *The set $P_w$ contains at most one letter.*

**Proof.** Let $t(w) = a$. Assume contrary that there are two letters $m, m' \in P_w$. Consider the respective transitions $(q, q_l, (a, m), q_r)$ and $(q, q_l', (a, m'), q_r')$. Since $q, q' \in S_w$ so by (5) there are two accepting runs $\rho, \rho'$ of $\bar{\mathcal{A}}$ on $t_A[\square/w]$ that assign letters $m_v$ to $v \prec w$ and have values $q, q'$ respectively in the hole $w$.

For $d \in \{l, r\}$ let $t_d, t'_d \in \mathrm{Tr}_M$ be trees and $\rho_d, \rho'_d$ be consistent runs of $\mathcal{A}$ that witness that $q_d, q'_d \in Q_{wd}$, i.e. $\rho_d$ is a consistent run of $\mathcal{A}$ on $(t_A\restriction_{wd}, t_d)$ with value $q_d$, similarly for $t'_d$, $\rho'_d$, $q'_d$.

Consider now two trees over the alphabet $A \times M \times Q$:

$$t = (t_A[\square/w], \rho) \cdot (a, m, q)((t_A\restriction_{wl}, t_l, \rho_l), (t_A\restriction_{wr}, t_r, \rho_r)),$$
$$t' = (t_A[\square/w], \rho') \cdot (a, m', q')((t_A\restriction_{wl}, t'_l, \rho'_l), (t_A\restriction_{wr}, t'_r, \rho'_r)).$$

Note that:
- both $t, t'$ equal $t_A$ on the $A$'th coordinate,
- they differ in vertex $w$ on the $M$'th coordinate,
- the $Q$'th coordinate of $t, t'$ denotes an accepting run of $\mathcal{A}$ on the $A \times M$ coordinates.

Therefore, we have a contradiction: $t_A$ has two different labellings $t_M, t'_M$ (one with $m$ and the other with $m'$ in $w$) such that $(t_A, t_M) \in L_M$ and $(t_A, t'_M) \in L_M$. ◀

Let $\mathcal{T}$ select as the letter $m_w$ the only element of $P_w$ whenever $P_w \neq \emptyset$. By the definition of $P_w$, the invariant (5) holds in the vertices $wl, wr$.

Now take any tree $t_A \in L_A$ and consider the result $t_R = \mathcal{T}(t_A, \tau(t_A))$. Let $t_M$ be the unique labelling of $t_A$ such that $(t_A, t_M) \in L_M$. Let $\rho$ be an accepting run of $\mathcal{A}$ on $(t_A, t_M)$. We show inductively that $t_R = t_M$ what finishes the proof. Let $w$ be a node of $t_A$ and assume that for all $v \prec w$ we have $t_R(v) = t_M(v)$. Let $(q, q_l, (a, m), q_r)$ be the transition used by $\rho$ in $w$. By the definition of $P_w$ this transition is a witness that $m \in P_w$. Therefore, $P_w$ is not empty and $t_R(w) = m = t_M(w)$.