Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 377 (2007) 1–24
Theoretical Computer Science
www.elsevier.com/locate/tcs

Fundamental study

# Tree algebras and varieties of tree languages

Saeed Salehi[a], Magnus Steinby[b,c,*]

[a] Department of Mathematics, Institute for Advanced Studies in Basic Sciences, Gava Zang – P.O. Box 45195-1159, Zanjan, Iran
[b] Department of Mathematics, University of Turku, 20014 Turku, Finland
[c] Turku Centre for Computer Science, Finland

Received 27 June 2006; received in revised form 18 January 2007; accepted 4 February 2007

Communicated by A.K. Salomaa

**Abstract**

We consider several aspects of Wilke's [T. Wilke, An algebraic characterization of frontier testable tree languages, Theoret. Comput. Sci. 154 (1996) 85–106] tree algebra formalism for representing binary labelled trees and compare it with approaches that represent trees as terms in the traditional way. A convergent term rewriting system yields normal form representations of binary trees and contexts, as well as a new completeness proof and a computational decision method for the axiomatization of tree algebras. Varieties of binary tree languages are compared with varieties of tree languages studied earlier in the literature. We also prove a variety theorem thus solving a problem noted by several authors. Syntactic tree algebras are studied and compared with ordinary syntactic algebras. The expressive power of the language of tree algebras is demonstrated by giving equational definitions for some well-known varieties of binary tree languages.


*Keywords:* Tree automata; Tree languages; Tree algebras; Binary trees; Varieties of tree languages; Syntactic tree algebras

## 1. Introduction

In algebraic language theory words are usually regarded as elements of the free monoid $X^*$ (or the free semigroup $X^+$ if the empty word is omitted) generated by a given alphabet $X$. In particular, the syntactic monoid (cf. [8,22]) of a language $L \subseteq X^*$ is defined with this interpretation in mind. Similarly, in algebraic treatments of regular tree languages (cf. [7,32,13,14]) trees are often defined as terms, and the syntactic algebra [1,28,29] of a tree language is then a quotient algebra of the appropriate term algebra. However, Wilke [34] has proposed a different framework in which trees are not directly viewed as elements of any algebraic structure but are represented by terms over a signature $\Gamma$ with six operation symbols involving the three sorts **label**, **tree** and **context**. The trees thus represented are binary trees over a given label alphabet. A tree algebra is a $\Gamma$-algebra satisfying certain identities that equate some pairs of $\Gamma$-terms representing the same tree or the same context. The component of sort **tree** of the syntactic tree algebra of a binary tree language $T$ is essentially the syntactic algebra of $T$ in the sense of [1,28,29], while its **context**-component

* Corresponding author at: Department of Mathematics, University of Turku, 20014 Turku, Finland.
*E-mail addresses:* saeed@math.net (S. Salehi), steinby@utu.fi (M. Steinby).

0304-3975/$ - see front matter 
doi:10.1016/j.tcs.2007.02.006

gives the syntactic semigroup of $T$ as defined (as monoids) in [33], and studied further in [27], [21] and [23]. A binary tree language is regular if and only if its syntactic tree algebra is finite [34]. Hence, one may characterize families of binary tree languages by syntactic tree algebras as shown by Wilke [34] in the case of frontier testable (i.e., reverse definite) tree languages.

In this paper we study several aspects of the tree algebra formalism. The theory is formulated in such a way that it

(1) lets us derive the conceptual machinery directly from some general ideas of algebraic language theory,

(2) yields many fundamental results, including the general theorems of [34], in a natural way with transparent algebraic proofs, and

(3) facilitates the comparison with other algebraic approaches to regular tree languages.

A classification theory for binary tree languages based on syntactic tree algebras was called for already in [34], and the lack of an appropriate variety theorem was noted also in [30], [10] and [11]. Here such a theorem is proved. For this, we have to consider varieties of finite tree algebras of a special kind as the direct bijection between varieties of binary tree languages (VBTLs) and all varieties of finite tree algebras fails to hold. We also show that any general variety of tree languages of the kind studied in [30], yields a VBTL when restricted to binary ranked alphabets. That not every VBTL is obtained this way, is due to a subtle difference in the tree homomorphisms used in the definitions of the two kinds of varieties. A similar difference can be noted in the relation between syntactic tree algebras and ordinary syntactic algebras: the syntactic algebra completely determines the syntactic tree algebra, but the converse is only partially true. Anyway, it seems that mostly the same families of binary tree languages are definable in terms of the two syntactic invariants. On the other hand, the language of tree algebras lends itself better for equational definitions of VBTLs.

Let us now review the contents of the paper section by section. In Section 2 we introduce algebras, terms and trees as well as several related notions, fixing at the same time some general notation to be used throughout the paper. In Section 3 Wilke's tree algebras are introduced, and the representations of binary trees and contexts by Wilke's terms are formalized by homomorphisms from the appropriate term algebras to the corresponding tree algebras of binary trees. In Section 4 we turn Wilke's axioms for tree algebras into a convergent term rewriting system, and describe the corresponding normal form representations of binary trees and contexts. The term rewriting system also yields a completeness theorem for Wilke's axioms, proved differently in [34], as well as a computational method to test the equivalence of two tree or context representations.

In Section 5 we define and survey some basic properties of the syntactic congruences and syntactic algebras of subsets of $\Gamma$-algebras, making use of the general many-sorted theory developed in [25]. When these definitions are applied in Section 6 to binary tree languages, regarding these as subsets of sort **tree** of free tree algebras, we obtain Wilke's syntactic tree algebras as well as some basic facts about them. In particular, by noting some relationships between the syntactic tree algebra $STA(T)$ of a binary tree language $T$ and its ordinary syntactic algebra $SA(T)$ and syntactic semigroup $SS(T)$, we get in a new way Wilke's theorem stating that $T$ is regular iff $STA(T)$ is finite. Moreover, we note several general properties of syntactic tree algebras needed in the variety theory.

In Section 7 we introduce varieties of binary tree languages (VBTLs) and varieties of finite tree algebras (VFTAs). However, the natural maps between VBTLs and VFTAs, defined via syntactic algebras, do not yield the complete correspondence one could expect. In Section 8 it is then shown how a Variety Theorem for VBTLs can be obtained by replacing VFTAs with varieties of finite reduced tree algebras; we call a tree algebra reduced if it is generated by its elements of sort **label**, and no two elements of sort **label**, or of sort **context**, are equivalent with respect to the operations of the algebra that yield elements of sort **tree**. All syntactic tree algebras are reduced in this sense. We also show how any tree algebra $\mathcal{M}$ can be transformed to a reduced tree algebra that is maximal among the reduced tree algebras covered by $\mathcal{M}$.

Varieties of binary tree languages are less general than varieties of [1], [28] or [29] in that they involve binary trees only. On the other hand, they are more general in the sense that the alphabet of labels is not fixed. In this they resemble the general varieties of tree languages (GVTLs) of [30] where tree languages over all ranked alphabets and leaf alphabets appear. In Section 9 we show that, when a GVTL is restricted to the ranked alphabets of binary tree languages, a VBTL is obtained. Thus the binary parts of many known families of regular tree languages are VBTLs. However, not every VBTL can be obtained this way from a GVTL. This ultimately depends on the fact that in the binary trees of [34] leaves and inner nodes are labelled with the same symbols. A similar subtle difference surfaces when we study connections between the syntactic tree algebra $STA(T)$, the syntactic algebra $SA(T)$ and the syntactic semigroup $SS(T)$ of a binary tree language $T$. Although $STA(T)$ is completely determined by $SA(T)$, and we can construct $STA(T)$ from $SA(T)$, the converse is not completely true. Nevertheless, it appears that essentially the same families of binary tree languages can be characterized by syntactic tree algebras as by syntactic algebras.

In spite of the above conclusion drawn from the results of Section Section 9, it seems that the language of tree algebras has certain advantages and is very convenient for defining VBTLs by equations. This was first shown by Wilke [34] who gave an elegant equational description of the frontier testable binary tree languages. Wilke also proved that frontier testability is a decidable property for binary tree languages. However, the equational description did not by itself yield a decision method, but a closer analysis of the syntactic tree algebras of frontier testable sets was required. In Section 10 we present, after some relevant general facts, three more examples of equational descriptions of VBTLs.

This paper has been written over a rather long period of time. Hence it both precedes and follows the doctoral dissertation of the first-named author, and some of the results appear already in [24]. However, even in those cases, the presentation may be somewhat different here. The bibliography contains several general references related to the subject matter of this paper. In particular, [31] surveys various algebraic approaches to the classification of regular tree languages and contains many further relevant references.

## 2. Algebras, terms, trees and contexts

In this section we recall some basic notions mainly to fix our notation for later reference. First a word on notation: we shall frequently write $a := b$ to indicate that $a$ is defined to be equal to $b$.

Let $\Sigma$ be a *ranked alphabet*, i.e., a finite set of operation symbols each of which has a given non-negative integer *arity*. For each $m \geq 0$, let $\Sigma_m$ denote the set of $m$-ary symbols in $\Sigma$. A $\Sigma$*-algebra* $\mathcal{D} = (D, \Sigma)$ consists of a non-empty set $D$ (of the *elements* of $\mathcal{D}$) and a $\Sigma$-indexed family of operations such that if $f \in \Sigma_m$, then $f^{\mathcal{D}} : D^m \to D$ is an $m$-ary operation on $D$. In particular, any $c \in \Sigma_0$ fixes a constant $c^{\mathcal{D}} \in D$.

Next we recall the usual definition of trees as terms (cf. [7,32,13,14], for example). Let $X$ be a finite set of symbols disjoint from $\Sigma$, called a *leaf alphabet*. The set $T_\Sigma(X)$ of $\Sigma$*-terms over* $X$ is defined inductively:

(1) $\Sigma_0 \cup X \subseteq T_\Sigma(X)$;
(2) $f(t_1, \ldots, t_m) \in T_\Sigma(X)$ if $m > 0$, $f \in \Sigma_m$ and $t_1, \ldots, t_m \in T_\Sigma(X)$.

We shall view terms in the usual way as (syntactic representations of) trees labelled with symbols in $\Sigma \cup X$, and call them also $\Sigma X$*-trees*.

The *height* $\mathrm{hg}(t)$ of a $\Sigma X$-tree $t$ is defined by setting (1) $\mathrm{hg}(t) = 0$ for any $t \in \Sigma_0 \cup X$, and (2) $\mathrm{hg}(t) = \max\{\mathrm{hg}(t_1), \ldots, \mathrm{hg}(t_m)\} + 1$ for $t = f(t_1, \ldots, t_m)$.

Let $\xi$ be a new symbol that does not appear in $\Sigma$ or $X$. A $\Sigma X$*-context* is a $\Sigma(X \cup \{\xi\})$-tree in which $\xi$ appears exactly once. The set of $\Sigma X$-contexts is denoted by $C_\Sigma(X)$. Furthermore, let $C^+_\Sigma(X) = C_\Sigma(X) \setminus \{\xi\}$ be the set of *non-unit* $\Sigma X$*-contexts*; $\xi$ is the *unit context*. In the special case $X = \emptyset$, we get the sets $T_\Sigma$, $C_\Sigma$ and $C^+_\Sigma$ of $\Sigma$*-trees* (or *ground* $\Sigma$*-terms*), $\Sigma$*-contexts*, and *non-unit* $\Sigma$*-contexts*, respectively.

The $\xi$*-depth* $\mathrm{d}_\xi(p)$ of $p \in C_\Sigma(X)$ is the distance of the $\xi$-labelled node from the root of $p$, that is, (1) $\mathrm{d}_\xi(\xi) = 0$, and (2) if $p = f(t_1, \ldots, t_{i-1}, q, t_{i+1}, \ldots, t_m)$ for some $m > 0$, $1 \leq i \leq m$, $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_m \in T_\Sigma(X)$ and $q \in C_\Sigma(X)$, then $\mathrm{d}_\xi(p) = \mathrm{d}_\xi(q) + 1$.

If $p, q \in C_\Sigma(X)$ and $t \in T_\Sigma(X)$, then $q \cdot p := p(q) \in C_\Sigma(X)$ and $t \cdot p := p(t) \in T_\Sigma(X)$ are obtained from $p$ by replacing the single occurrence of $\xi$ with $q$ and with $t$, respectively. Obviously, $\xi(p) = p(\xi) = p$ and $\xi(t) = t$ for any context $p$ and any tree $t$. Clearly, $(C_\Sigma(X), \cdot, \xi)$ is a monoid for the product $p \cdot q$. Similarly, $(C^+_\Sigma(X), \cdot)$ is a semigroup.

In what follows, we consider especially *binary trees* in which both the inner nodes and the leaves are labelled with symbols from a given finite non-empty alphabet $A$, the *label alphabet*. To obtain compatibility with the term formalism, we define them formally as follows. First of all, we associate with $A$ the ranked alphabet $\Sigma^A = \Sigma^A_0 \cup \Sigma^A_2$, where $\Sigma^A_0 = \{c_a \mid a \in A\}$ and $\Sigma^A_2 = \{f_a \mid a \in A\}$. We shall call $\Sigma^A$-trees and $\Sigma^A$-contexts simply $A$-trees and $A$-contexts, respectively, and the notation is simplified correspondingly. Hence the set $T_A$ of $A$*-trees* and the set $C_A$ of $A$*-contexts* are defined inductively:

(1) $c_a \in T_A$ for every $a \in A$, and $\xi \in C_A$;
(2) $f_a(s, t) \in T_A$ and $f_a(p, t), f_a(t, p) \in C_A$ for all $a \in A$, $s, t \in T_A$ and $p \in C_A$.

Moreover, let $C_A^+ = C_A \setminus \{\xi\}$ be the set of non-unit $A$-contexts.

The $\Sigma^A$*-algebra of* $A$*-trees* $\mathcal{T}_A = (T_A, \Sigma^A)$ is defined by setting

(1) $c_a^{\mathcal{T}_A} = c_a$ for every $a \in A$, and
(2) $f_a^{\mathcal{T}_A}(s,t) = f_a(s,t)$ for every $a \in A$ and all $s, t \in T_A$.

Since $\mathcal{T}_A$ is the $\Sigma^A$-term algebra generated by the empty set, there is for each $\Sigma^A$-algebra $\mathcal{D}$ a unique homomorphism $\varphi_{\mathcal{D}} : \mathcal{T}_A \to \mathcal{D}$ defined by

(1) $c_a\varphi_{\mathcal{D}} = c_a^{\mathcal{D}}$ for $a \in A$, and
(2) $f_a(s,t)\varphi_{\mathcal{D}} = f_a^{\mathcal{D}}(s\varphi_{\mathcal{D}}, t\varphi_{\mathcal{D}})$ for any $a \in A$ and $s, t \in T_A$.

Subsets of $T_A$ we call $A$*-tree languages*, and a *binary tree language* is any set that is an $A$-tree language for some label alphabet $A$.

Let us now introduce Wilke's [34] formalism for representing binary trees by terms over a 3-sorted ranked alphabet. An overview of the theory of many-sorted algebras, as well as many further references, can be found in [19]. In [25] we have developed a general theory of varieties of recognizable subsets of many-sorted algebras, and some of the notions and facts to be presented here could be obtained by a suitable specialization from that theory.

The set of *sorts* is $S = \{\mathbf{label}, \mathbf{tree}, \mathbf{context}\}$. For the sort names we use the abbreviations $\mathbf{l} = \mathbf{label}$, $\mathbf{t} = \mathbf{tree}$ and $\mathbf{c} = \mathbf{context}$. An $S$-sorted set $M$ is a triple $\langle M_{\mathbf{l}}, M_{\mathbf{t}}, M_{\mathbf{c}}\rangle$ in which $M_{\mathbf{l}}$, $M_{\mathbf{t}}$ and $M_{\mathbf{c}}$ are the sets of elements of $M$ of sort **label**, **tree** and **context**, respectively. Although this would not be quite necessary, we shall always assume that the sets $M_{\mathbf{l}}$, $M_{\mathbf{t}}$ and $M_{\mathbf{c}}$ are pairwise disjoint, i.e., that the sort of each element of $M$ is uniquely determined.

Now let $\Gamma = \{\iota, \kappa, \lambda, \rho, \eta, \sigma\}$ be the $S$-sorted ranked alphabet where the *types* of the symbols are as follows:

$$\iota : \mathbf{l} \to \mathbf{t}; \quad \kappa : \mathbf{ltt} \to \mathbf{t}; \quad \lambda, \rho : \mathbf{lt} \to \mathbf{c}; \quad \eta : \mathbf{ct} \to \mathbf{t}; \quad \sigma : \mathbf{cc} \to \mathbf{c}.$$

For example, in any $\Gamma$-algebra the $\lambda$-operation forms an element of sort $\mathbf{c}$ from an element of sort $\mathbf{l}$ and an element of sort $\mathbf{t}$.

For the general notion of many-sorted terms we refer the reader to [19] or [25]. Here we introduce just the kind of $\Gamma$-terms to be used in this paper. The $S$-sorted set $\langle A, T_\Gamma(A), C_\Gamma^+(A)\rangle$ of $\Gamma A$*-terms*, where $T_\Gamma(A)$ is the set of $\Gamma A$*-tree terms*, and $C_\Gamma^+(A)$ the set of non-unit $\Gamma A$*-context terms*, is defined inductively as follows:

(1) if $a \in A$, then $\iota(a) \in T_\Gamma(A)$;
(2) if $a \in A$ and $\mathrm{s}, \mathrm{t} \in T_\Gamma(A)$, then $\kappa(a, \mathrm{s}, \mathrm{t}) \in T_\Gamma(A)$;
(3) if $a \in A$ and $\mathrm{t} \in T_\Gamma(A)$, then $\lambda(a, \mathrm{t}) \in C_\Gamma^+(A)$;
(4) if $a \in A$ and $\mathrm{t} \in T_\Gamma(A)$, then $\rho(a, \mathrm{t}) \in C_\Gamma^+(A)$;
(5) if $\mathrm{p} \in C_\Gamma^+(A)$ and $\mathrm{t} \in T_\Gamma(A)$, then $\eta(\mathrm{p}, \mathrm{t}) \in T_\Gamma(A)$;
(6) if $\mathrm{p}, \mathrm{q} \in C_\Gamma^+(A)$, then $\sigma(\mathrm{p}, \mathrm{q}) \in C_\Gamma^+(A)$.

Hence, the $\Gamma A$-terms are the $\Gamma$-terms over the sorted set of variables $X = \langle A, \emptyset, \emptyset\rangle$ where $A$ is a given label alphabet. To distinguish them from $A$-trees and $A$-contexts, the symbols denoting them are written in Roman type.

**Remark 2.1.** Note that $C_\Gamma^+(A)$ does not include the unit context $\xi$. Similarly, the syntactic tree algebras – to be defined later – do not automatically have a unit element of sort **context**. This means that, in a way, Wilke's [34] theory corresponds to Eilenberg's [8] theory of $+$-varieties and syntactic semigroups. By adding to $\Gamma$ a constant of sort **context** one could obtain a variant of the theory that corresponds to the theory of $*$-varieties and syntactic monoids.

Binary $A$-trees and $A$-contexts are represented by $\Gamma A$-tree terms and $\Gamma A$-context terms as follows. For any $\mathrm{t} \in T_\Gamma(A)$, let $\hat{\mathrm{t}}$ denote the $A$-tree represented by t. Similarly, $\hat{\mathrm{p}}$ denotes the $A$-context represented by a $\Gamma A$-context term $\mathrm{p} \in C_\Gamma^+(A)$. The representations are defined by setting for any $a \in A$, $\mathrm{s}, \mathrm{t} \in T_\Gamma(A)$ and $\mathrm{p}, \mathrm{q} \in C_\Gamma^+(A)$,

(1) $\iota(a)$ represents the $A$-tree $c_a$,
(2) $\kappa(a, \mathrm{s}, \mathrm{t})$ represents the $A$-tree $f_a(\hat{\mathrm{s}}, \hat{\mathrm{t}})$,
(3) $\lambda(a, \mathrm{t})$ represents the $A$-context $f_a(\xi, \hat{\mathrm{t}})$,
(4) $\rho(a, \mathrm{t})$ represents the $A$-context $f_a(\hat{\mathrm{t}}, \xi)$,
(5) $\eta(\mathrm{p}, \mathrm{t})$ represents the $A$-tree $\hat{\mathrm{p}}(\hat{\mathrm{t}})$, and
(6) $\sigma(\mathrm{p}, \mathrm{q})$ represents the $A$-context $\hat{\mathrm{p}}(\hat{\mathrm{q}})$.

The following facts are easy to verify by induction on $A$-trees and $A$-contexts.

**Lemma 2.2.** *Let $A$ be any label alphabet. For any $A$-tree $t$ we can find a $\Gamma A$-tree term $\mathrm{t} \in T_\Gamma(A)$ such that $\hat{\mathrm{t}} = t$, and for any non-unit $A$-context $p$ a $\Gamma A$-context term $\mathrm{p} \in C_\Gamma^+(A)$ such that $\hat{\mathrm{p}} = p$.*

These representations are usually not unique. For example, the $\{a, b\}$-tree terms $\kappa(b, \kappa(a, \iota(b), \iota(a)), \iota(a))$ and $\eta(\lambda(b, \iota(a)), \kappa(a, \iota(b), \iota(a)))$ both represent the same $\{a, b\}$-tree $f_b(f_a(c_b, c_a), c_a)$. In Lemma 3.2 we will formulate this representation relation as a homomorphism.

## 3. Tree algebras

A *$\Gamma$-algebra* $\mathcal{M} = (\langle M_\mathbf{l}, M_\mathbf{t}, M_\mathbf{c}\rangle, \Gamma)$ consists of a nonempty set $M_\mathbf{l}$ of elements of sort **label**, a nonempty set $M_\mathbf{t}$ of elements of sort **tree**, and a nonempty set $M_\mathbf{c}$ of elements of sort **context**, and operations

| | |
|---|---|
| (1) $\iota^\mathcal{M}\colon M_\mathbf{l} \to M_\mathbf{t}$ | (2) $\kappa^\mathcal{M}\colon M_\mathbf{l} \times M_\mathbf{t} \times M_\mathbf{t} \to M_\mathbf{t}$ |
| (3) $\lambda^\mathcal{M}\colon M_\mathbf{l} \times M_\mathbf{t} \to M_\mathbf{c}$ | (4) $\rho^\mathcal{M}\colon M_\mathbf{l} \times M_\mathbf{t} \to M_\mathbf{c}$ |
| (5) $\eta^\mathcal{M}\colon M_\mathbf{c} \times M_\mathbf{t} \to M_\mathbf{t}$ | (6) $\sigma^\mathcal{M}\colon M_\mathbf{c} \times M_\mathbf{c} \to M_\mathbf{c}$, |

defined as realizations of the symbols in $\Gamma$. Usually we write simply $\mathcal{M} = (M, \Gamma)$ with the understanding that $M = \langle M_\mathbf{l}, M_\mathbf{t}, M_\mathbf{c}\rangle$.

The basic algebraic notions, such as subalgebras, congruences, homomorphisms etc., are defined for $\Gamma$-algebras the same way as for many-sorted algebras in general (cf. [19] or [25]). For example, a *homomorphism* $\varphi\colon \mathcal{M} \to \mathcal{N}$ from a $\Gamma$-algebra $\mathcal{M} = (M, \Gamma)$ to a $\Gamma$-algebra $\mathcal{N} = (N, \Gamma)$ is a sorted mapping $\varphi : M \to N$, i.e., an $S$-sorted triple of maps

$$\langle \varphi_\mathbf{l}\colon M_\mathbf{l} \to N_\mathbf{l}, \varphi_\mathbf{t}\colon M_\mathbf{t} \to N_\mathbf{t}, \varphi_\mathbf{c}\colon M_\mathbf{c} \to N_\mathbf{c}\rangle$$

that preserves all the $\Gamma$-operations between $\mathcal{M}$ and $\mathcal{N}$, that is to say, $\iota^\mathcal{M}(a)\varphi_\mathbf{t} = \iota^\mathcal{N}(a\varphi_\mathbf{l})$ for every $a \in M_\mathbf{l}$, $\kappa^\mathcal{M}(a, s, t)\varphi_\mathbf{t} = \kappa^\mathcal{N}(a\varphi_\mathbf{l}, s\varphi_\mathbf{t}, t\varphi_\mathbf{t})$ for all $a \in M_\mathbf{l}$ and $s, t \in M_\mathbf{t}$, etc. A homomorphism $\varphi\colon \mathcal{M} \to \mathcal{N}$ is an *epimorphism* if $\varphi$ is surjective, i.e., $\varphi_\mathbf{i}\colon M_\mathbf{i} \to N_\mathbf{i}$ is surjective for every $\mathbf{i} \in S$. Similarly, $\varphi$ is a *monomorphism* if every $\varphi_\mathbf{i}\colon M_\mathbf{i} \to N_\mathbf{i}$ is injective. Finally, an *isomorphism* is a bijective homomorphism. The fact that $\mathcal{M}$ and $\mathcal{N}$ are isomorphic is denoted by writing $\mathcal{M} \cong \mathcal{N}$.

For any label alphabet $A$, the *$\Gamma$-algebra of $\Gamma A$-terms*

$$\mathcal{T}_\Gamma(A) = (\langle A, T_\Gamma(A), C_\Gamma^+(A)\rangle, \Gamma)$$

is defined by setting

| | |
|---|---|
| (1) $\iota^{\mathcal{T}_\Gamma(A)}(a) = \iota(a)$ | (2) $\kappa^{\mathcal{T}_\Gamma(A)}(a, \mathrm{s}, \mathrm{t}) = \kappa(a, \mathrm{s}, \mathrm{t})$ |
| (3) $\lambda^{\mathcal{T}_\Gamma(A)}(a, \mathrm{t}) = \lambda(a, \mathrm{t})$ | (4) $\rho^{\mathcal{T}_\Gamma(A)}(a, \mathrm{t}) = \rho(a, \mathrm{t})$ |
| (5) $\eta^{\mathcal{T}_\Gamma(A)}(\mathrm{p}, \mathrm{t}) = \eta(\mathrm{p}, \mathrm{t})$ | (6) $\sigma^{\mathcal{T}_\Gamma(A)}(\mathrm{p}, \mathrm{q}) = \sigma(\mathrm{p}, \mathrm{q})$ |

for all $a \in A$, $\mathrm{s}, \mathrm{t} \in T_\Gamma(A)$ and $\mathrm{p}, \mathrm{q} \in C_\Gamma^+(A)$.

Following [34], we call a $\Gamma$-algebra a *tree algebra* if it satisfies the following set of identities *TA*:

(TA1) $\sigma(\sigma(\mathrm{p}, \mathrm{q}), \mathrm{r}) \approx \sigma(\mathrm{p}, \sigma(\mathrm{q}, \mathrm{r}))$
(TA2) $\eta(\sigma(\mathrm{p}, \mathrm{q}), \mathrm{t}) \approx \eta(\mathrm{p}, \eta(\mathrm{q}, \mathrm{t}))$
(TA3) $\eta(\lambda(a, \mathrm{s}), \mathrm{t}) \approx \kappa(a, \mathrm{t}, \mathrm{s})$
(TA4) $\eta(\rho(a, \mathrm{s}), \mathrm{t}) \approx \kappa(a, \mathrm{s}, \mathrm{t})$.

Here, $a$ is a variable of sort **label**, s and t are variables of sort **tree**, and p, q and r variables of sort **context**. Let **TA** denote the equational class of all tree algebras.

For each label alphabet $A$, a tree algebra of special interest is the *$\Gamma$-algebra of $A$-trees* $\mathcal{F}_\mathbf{TA}(A) = (\langle A, T_A, C_A^+\rangle, \Gamma)$, where for any $a \in A$, $s, t \in T_A$ and $p, q \in C_A^+$,

| | |
|---|---|
| (1) $\iota^{\mathcal{F}_\mathbf{TA}(A)}(a) = c_a$ | (2) $\kappa^{\mathcal{F}_\mathbf{TA}(A)}(a, s, t) = f_a(s, t)$ |
| (3) $\lambda^{\mathcal{F}_\mathbf{TA}(A)}(a, t) = f_a(\xi, t)$ | (4) $\rho^{\mathcal{F}_\mathbf{TA}(A)}(a, t) = f_a(t, \xi)$ |
| (5) $\eta^{\mathcal{F}_\mathbf{TA}(A)}(p, t) = p(t)$ | (6) $\sigma^{\mathcal{F}_\mathbf{TA}(A)}(p, q) = p(q)$. |

As shown by Wilke ([34], Proposition 1), and suggested by our notation, $\mathcal{F}_{\mathbf{TA}}(A)$ is the *free tree algebra* generated by $\langle A, \emptyset, \emptyset\rangle$. This means that $\mathcal{F}_{\mathbf{TA}}(A)$ satisfies the identities *TA*, and that if $\mathcal{M} = (M, \Gamma)$ is any tree algebra, then every mapping $\varphi_0 : A \to M_{\mathbf{l}}$ can be extended in a unique way to a homomorphism $\varphi = \langle \varphi_{\mathbf{l}}, \varphi_{\mathbf{t}}, \varphi_{\mathbf{c}}\rangle$ of $\Gamma$-algebras from $\mathcal{F}_{\mathbf{TA}}(A)$ to $\mathcal{M}$ where $\varphi_{\mathbf{l}} = \varphi_0$.

For any label alphabet $A$, an $A$-*instance* of an identity in *TA* is any pair of $\Gamma A$-tree or $\Gamma A$-context terms obtained from the identity by assigning each of the variables $a$, s, t, p, q and r appearing in it a value from the appropriate set $A$, $T_A$ or $C_A^+$. For example, if $b, c \in A$, then

$$\left\langle\ \eta(\lambda(c, \iota(b)), \iota(c)),\ \kappa(c, \iota(b), \iota(c))\ \right\rangle$$

is the $A$-instance of (TA3) obtained by the substitution

$$a \mapsto c,\ \mathrm{s} \mapsto \iota(b),\ \mathrm{t} \mapsto \iota(c).$$

Moreover, let $\equiv^A$ be the fully invariant congruence on $\mathcal{T}_\Gamma(A)$ generated by the set of all $A$-instances of *TA*, i.e., the equational theory in variables $\langle A, \emptyset, \emptyset\rangle$ defined by *TA*.

It is clear that if (u, v) is an $A$-instance of an identity in *TA*, then $\hat{\mathrm{u}} = \hat{\mathrm{v}}$. Furthermore, if (u, v) is obtained from pairs of $\Gamma A$-terms representing the same $A$-tree or the same $A$-context by any inference rule of Birkhoff's equational logic (for the many-sorted version, cf. Section 5.2 in [19]), then again $\hat{\mathrm{u}} = \hat{\mathrm{v}}$. Hence we get at this point the soundness property of Wilke's axiom system *TA*.

**Proposition 3.1.** *Let $A$ be any label alphabet. For any* $\mathrm{s}, \mathrm{t} \in T_\Gamma(A)$ *and* $\mathrm{p}, \mathrm{q} \in C_\Gamma^+(A)$*,*

(a) *if* $\mathrm{s} \equiv_{\mathbf{t}}^A \mathrm{t}$*, then* $\hat{\mathrm{s}} = \hat{\mathrm{t}}$*, and*
(b) *if* $\mathrm{p} \equiv_{\mathbf{c}}^A \mathrm{q}$*, then* $\hat{\mathrm{p}} = \hat{\mathrm{q}}$*.*

As the $\Gamma$-algebras $\mathcal{T}_\Gamma(A)$ and $\mathcal{F}_{\mathbf{TA}}(A)$ both are generated by $\langle A, \emptyset, \emptyset\rangle$, the identity mapping $1_A : A \to A$ (of sort **label**) can be extended in a unique way to an epimorphism $\nu^A : \mathcal{T}_\Gamma(A) \to \mathcal{F}_{\mathbf{TA}}(A)$ of $\Gamma$-algebras that we call the *canonical A-homomorphism*. It is the triple of mappings

$$\langle \nu_{\mathbf{l}}^A : A \to A,\ \nu_{\mathbf{t}}^A : T_\Gamma(A) \to T_A,\ \nu_{\mathbf{c}}^A : C_\Gamma^+(A) \to C_A^+\rangle$$

such that for all $a \in A$, $\mathrm{s}, \mathrm{t} \in T_\Gamma(A)$ and $\mathrm{p}, \mathrm{q} \in C_\Gamma^+(A)$,

(1) $\nu_{\mathbf{l}}^A(a) = a$
(2) $\nu_{\mathbf{t}}^A(\iota(a)) = c_a$
(3) $\nu_{\mathbf{t}}^A(\kappa(a, \mathrm{s}, \mathrm{t})) = f_a(\nu_{\mathbf{t}}^A(\mathrm{s}), \nu_{\mathbf{t}}^A(\mathrm{t}))$
(4) $\nu_{\mathbf{c}}^A(\lambda(a, \mathrm{t})) = f_a(\xi, \nu_{\mathbf{t}}^A(\mathrm{t}))$
(5) $\nu_{\mathbf{c}}^A(\rho(a, \mathrm{t})) = f_a(\nu_{\mathbf{t}}^A(\mathrm{t}), \xi)$
(6) $\nu_{\mathbf{t}}^A(\eta(\mathrm{p}, \mathrm{t})) = \nu_{\mathbf{c}}^A(\mathrm{p})(\nu_{\mathbf{t}}^A(\mathrm{t}))$
(7) $\nu_{\mathbf{c}}^A(\sigma(\mathrm{p}, \mathrm{q})) = \nu_{\mathbf{c}}^A(\mathrm{p})(\nu_{\mathbf{c}}^A(\mathrm{q}))$.

The following lemma is obtained immediately by comparing the above equalities with the clauses defining the $A$-trees and $A$-contexts represented by $\Gamma A$-terms.

**Lemma 3.2.** *For any $\Gamma A$-tree term* $\mathrm{t} \in T_\Gamma(A)$ *and any $\Gamma A$-context term* $\mathrm{p} \in C_\Gamma^+(A)$*, we have* $\nu_{\mathbf{t}}^A(\mathrm{t}) = \hat{\mathrm{t}}$ *and* $\nu_{\mathbf{c}}^A(\mathrm{p}) = \hat{\mathrm{p}}$.

## 4. Normal form representations

We now transform the set of identities *TA* into a convergent (i.e., terminating and confluent) term rewriting system. As noted by many authors, the general theory of term rewriting (as presented in [3,4,6,17], for example) can easily be extended to the many-sorted case simply by requiring that all the reductions preserve the sorts of terms. In our special case this is particularly obvious, and confluence and termination can be proved by standard methods.

**Definition 4.1.** Let $\mathcal{R}$ be the term rewriting system consisting of the rules

(R1) $\sigma(\sigma(\mathrm{p}, \mathrm{q}), \mathrm{r}) \to \sigma(\mathrm{p}, \sigma(\mathrm{q}, \mathrm{r}))$,

(R2) $\eta(\sigma(\mathrm{p}, \mathrm{q}), \mathrm{t}) \to \eta(\mathrm{p}, \eta(\mathrm{q}, \mathrm{t}))$,
(R3) $\eta(\lambda(a, \mathrm{s}), \mathrm{t}) \to \kappa(a, \mathrm{t}, \mathrm{s})$, and
(R4) $\eta(\rho(a, \mathrm{s}), \mathrm{t}) \to \kappa(a, \mathrm{s}, \mathrm{t})$.

**Proposition 4.2.** *The system $\mathcal{R}$ is convergent.*

**Proof.** It is clear that $\mathcal{R}$ is compatible with the lexicographic path ordering induced by any order on $\Gamma$ such that $\eta > \kappa$. Hence, $\mathcal{R}$ is terminating. There are just two critical pairs. The pair

$$\left\langle \eta(\sigma(\mathrm{p}, \sigma(\mathrm{q}, \mathrm{r})), \mathrm{t}) \ , \quad \eta(\sigma(\mathrm{p}, \mathrm{q}), \eta(\mathrm{r}, \mathrm{t})) \right\rangle$$

produced by (R1) and (R2) converges to $\eta(\mathrm{p}, \eta(\mathrm{q}, \eta(\mathrm{r}, \mathrm{t})))$ by applications of (R2), and the other critical pair

$$\left\langle \sigma(\sigma(\mathrm{p}, \sigma(\mathrm{q}, \mathrm{r})), \mathrm{r}') \ , \quad \sigma(\sigma(\mathrm{p}, \mathrm{q}), \sigma(\mathrm{r}, \mathrm{r}')) \right\rangle$$

obtained by overlapping (R1) with itself, converges to $\sigma(\mathrm{p}, \sigma(\mathrm{q}, \sigma(\mathrm{r}, \mathrm{r}')))$ by further applications of (R1). Hence, $\mathcal{R}$ is confluent as well. □

Let ${}_A\!\Rightarrow$ be the ($S$-sorted) reduction relation defined by $\mathcal{R}$ on the $S$-sorted set $\langle A, T_\Gamma(A), C_\Gamma^+(A)\rangle$ of $\Gamma A$-terms, and let ${}_A\!\Leftrightarrow^*$ be its equivalence closure. It follows directly from the definitions of $\mathcal{R}$ and $\equiv^A$ that ${}_A\!\Leftrightarrow^* = \equiv^A$. This suggests the idea to define normal form representations of $A$-trees and $A$-contexts by using $\mathcal{R}$.

Let $IRR(\mathcal{R}, A)_{\mathbf{l}}$, $IRR(\mathcal{R}, A)_{\mathbf{t}}$ and $IRR(\mathcal{R}, A)_{\mathbf{c}}$ be the sets of $\Gamma A$-terms irreducible by $\mathcal{R}$ of sort **label**, **tree** and **context**, respectively. It is clear that $IRR(\mathcal{R}, A)_{\mathbf{l}} = A$. The other two sets are described in the following proposition.

**Proposition 4.3.** *Let A be any label alphabet.*

**a.** *A $\Gamma A$-tree term is irreducible iff it contains the operators $\iota$ and $\kappa$ only, that is to say, $IRR(\mathcal{R}, A)_{\mathbf{t}}$ is the smallest subset of $T_\Gamma(A)$ such that*
  (1) *$\iota(a) \in IRR(\mathcal{R}, A)_{\mathbf{t}}$ for every $a \in A$, and*
  (2) *if $a \in A$ and $\mathrm{s}, \mathrm{t} \in IRR(\mathcal{R}, A)_{\mathbf{t}}$, then $\kappa(a, \mathrm{s}, \mathrm{t}) \in IRR(\mathcal{R}, A)_{\mathbf{t}}$.*

**b.** *$IRR(\mathcal{R}, A)_{\mathbf{c}}$ is the smallest subset of $C_\Gamma^+(A)$ such that*
  (1’) *$\lambda(a, \mathrm{t}), \rho(a, \mathrm{t}) \in IRR(\mathcal{R}, A)_{\mathbf{c}}$ for all $a \in A$ and $\mathrm{t} \in IRR(\mathcal{R}, A)_{\mathbf{t}}$, and*
  (2’) *$\sigma(\lambda(a, \mathrm{t}), \mathrm{p}) \in IRR(\mathcal{R}, A)_{\mathbf{c}}$ and $\sigma(\rho(a, \mathrm{t}), \mathrm{p}) \in IRR(\mathcal{R}, A)_{\mathbf{c}}$ for any $a \in A$, $\mathrm{t} \in IRR(\mathcal{R}, A)_{\mathbf{t}}$ and $\mathrm{p} \in IRR(\mathcal{R}, A)_{\mathbf{c}}$.*

**Proof.** By considering the rules of $\mathcal{R}$ one sees that clauses (1) and (2) define a set of irreducible $\Gamma A$-tree terms. On the other hand, any $\Gamma A$-tree term with a subterm of the form $\eta(\mathrm{p}, \mathrm{t})$ is reducible because the $\Gamma A$-context term p must begin with $\lambda$, $\rho$ or $\sigma$. Hence, all irreducible $\Gamma A$-tree terms are obtained by clauses (1) and (2).

It is clear that no rule of $\mathcal{R}$ applies to any $\Gamma A$-context term obtained by rules (1’) and (2’). That (1’) and (2’) yield all irreducible $\Gamma A$-context terms, is verified by induction on the $\xi$-depth $\mathrm{d}_\xi(\hat{\mathrm{p}})$ of the $A$-context represented by $\mathrm{p} \in IRR(\mathcal{R}, A)_{\mathbf{c}}$. If $\mathrm{d}_\xi(\hat{\mathrm{p}}) = 1$, then p must be a $\Gamma A$-context term given by (1’). If $\mathrm{d}_\xi(\hat{\mathrm{p}}) > 1$, then $\mathrm{p} = \sigma(\mathrm{q}, \mathrm{r})$ for some $\mathrm{q}, \mathrm{r} \in IRR(\mathcal{R}, A)_{\mathbf{c}}$, and because of rule (R1), q must be of the form $\lambda(a, \mathrm{t})$ or $\rho(a, \mathrm{t})$ with $\mathrm{t} \in IRR(\mathcal{R}, A)_{\mathbf{t}}$. Since the inductive assumption applies to r, also p is of the required type. □

As noted in Proposition 3.1, any two $\equiv^A$-congruent $\Gamma A$-tree terms represent the same $A$-tree. Therefore it follows now from Lemma 2.2 and Proposition 4.2 that any $A$-tree is represented by a unique irreducible $\Gamma A$-tree term. By Proposition 4.3 only the operators $\iota$ and $\kappa$ appear in irreducible $\Gamma A$-tree terms, and hence it is clear that if $\mathrm{s}, \mathrm{t} \in IRR(\mathcal{R}, A)_{\mathbf{t}}$ and $\mathrm{s} \neq \mathrm{t}$, then $\hat{\mathrm{s}} \neq \hat{\mathrm{t}}$. Similarly, $A$-contexts are represented by unique irreducible $\Gamma A$-context terms, and since these are of the form

$$\sigma(\mathrm{p}_1, (\sigma(\mathrm{p}_2, \ldots \sigma(\mathrm{p}_{n-1}, \mathrm{p}_n) \ldots),$$

where $n \geq 1$, and each $\mathrm{p}_i$ is of the form $\lambda(a, \mathrm{t})$ or $\rho(a, \mathrm{t})$ with $a \in A$ and $\mathrm{t} \in IRR(\mathcal{R}, A)_{\mathbf{t}}$, it is again clear that $\hat{\mathrm{p}} \neq \hat{\mathrm{q}}$ for any two distinct $\mathrm{p}, \mathrm{q} \in IRR(\mathcal{R}, A)_{\mathbf{c}}$. These observations yield the following proposition that completes the picture.

**Proposition 4.4.** *Let A be any label alphabet. Every A-tree is represented by a unique $\mathcal{R}$-irreducible $\Gamma A$-tree term and hence, if* $\hat{\mathrm{s}} = \hat{\mathrm{t}}$ *for any $\Gamma A$-tree terms* $\mathrm{s}, \mathrm{t} \in T_\Gamma(A)$*, then* $\mathrm{s} \equiv^A \mathrm{t}$*. Similarly, each A-context is represented by a unique $\mathcal{R}$-irreducible $\Gamma A$-context term, and any two $\Gamma A$-context terms that represent the same A-context are $\equiv^A$-congruent.*

By combining this result with Lemma 3.2 and Proposition 3.1, we get the following fact. Recall that the kernel (ker $\varphi$) of a homomorphism $\varphi$ is the set of all pairs $(a, b)$ such that $a\varphi = b\varphi$.

**Corollary 4.5.** *For any label alphabet A,* $\ker \nu^A = \equiv^A$.

Furthermore, we now get Wilke's [34] Proposition 1 in a new way:

**Corollary 4.6.** $\mathcal{F}_{\mathbf{TA}}(A)$ *is the free tree algebra generated by* $\langle A, \emptyset, \emptyset\rangle$.

**Proof.** Since $\equiv^A$ is the fully invariant congruence generated by the $A$-instances of the identities (*TA*), the quotient algebra $\mathcal{T}_\Gamma(A)/\equiv^A$ is freely generated by $\langle A, \emptyset, \emptyset\rangle$ (we identify each $a \in A$ with its $\equiv^A$-class $\{a\}$) over the class **TA**. On the other hand, $\mathcal{F}_{\mathbf{TA}}(A) \cong \mathcal{T}_\Gamma(A))/\ker \nu^A$ by the Homomorphism Theorem (cf. [19] for the many-sorted version). □

By combining Propositions 3.1 and 4.4, we get as a further corollary the following result.

**Proposition 4.7.** *Let A be any label alphabet.*

(a) *For any* $\mathrm{s}, \mathrm{t} \in T_\Gamma(A)$, $\hat{\mathrm{s}} = \hat{\mathrm{t}}$ *if and only if* $\mathrm{s} \equiv^A_{\mathbf{t}} \mathrm{t}$.
(b) *For any* $\mathrm{p}, \mathrm{q} \in C^+_\Gamma(A)$, $\hat{\mathrm{p}} = \hat{\mathrm{q}}$ *if and only if* $\mathrm{p} \equiv^A_{\mathbf{c}} \mathrm{q}$.

This proposition may be regarded as a Completeness Theorem for Wilke's axiomatization (*TA*) with respect to representations of binary trees and contexts. Indeed, it means that any two $\Gamma A$-tree or $\Gamma A$-context terms represent the same $A$-tree or $A$-context, respectively, iff they are *TA*-provably equal.

By Proposition 4.7 the equational theory $\equiv^A$ is trivially decidable: to decide whether $\mathrm{s} \equiv^A_{\mathbf{t}} \mathrm{t}$ holds for any given $\mathrm{s}, \mathrm{t} \in T_\Gamma(A)$, it suffices to construct the $A$-trees $\hat{\mathrm{s}}$ and $\hat{\mathrm{t}}$ and compare them with each other. Similarly, $\mathrm{p} \equiv^A_{\mathbf{c}} \mathrm{q}$ iff $\hat{\mathrm{p}} = \hat{\mathrm{q}}$, for any given $\mathrm{p}, \mathrm{q} \in C^+_\Gamma(A)$. Of course, this fact is implicit also in [34] since it follows from Corollary 4.6 (and also from Corollary 4.5, for that matter). However, let us also note that Proposition 4.2 yields another decision method that does not require forming the trees or contexts: whether any two given $\Gamma A$-tree terms, or two $\Gamma A$-context terms, are $\equiv^A$-equivalent can be decided by computing their respective $\mathcal{R}$-normal forms.

## 5. Syntactic $\Gamma$-algebras

The basic properties of Wilke's [34] syntactic tree algebra congruences and syntactic tree algebras of binary tree languages can be obtained conveniently by considering more generally subsets of arbitrary $\Gamma$-algebras. In [25] we studied these notions for subsets of general many-sorted algebras. Two kinds of subsets were considered, the sorted subsets that have a component of each sort, and the "pure" subsets consisting of elements of one given sort. Since we eventually apply these notions just to binary tree languages, we will focus here on pure subsets. The general theory will be used here by letting the set of sorts be $S = \{\mathbf{label}, \mathbf{tree}, \mathbf{context}\}$ and the ranked alphabet to be $\Gamma = \{\iota, \kappa, \lambda, \rho, \eta, \sigma\}$ as above. In the following section we will then recover Wilke's notions by considering subsets of the free tree algebras $\mathcal{F}_{\mathbf{TA}}(A)$.

A *sorted subset* of an $S$-sorted set $M$ is a triple $\langle L_{\mathbf{l}}, L_{\mathbf{t}}, L_{\mathbf{c}}\rangle$ such that $L_{\mathbf{l}} \subseteq M_{\mathbf{l}}$, $L_{\mathbf{t}} \subseteq M_{\mathbf{t}}$ and $L_{\mathbf{c}} \subseteq M_{\mathbf{c}}$. The inclusion relation and the basic set operations are defined for sorted subsets by the natural sortwise conditions. A *subset of sort* $\mathbf{i} \in S$ of $M$ is any subset of $M_{\mathbf{i}}$. With a subset $T \subseteq M_{\mathbf{i}}$ of sort $\mathbf{i}$ we associate the sorted subset $\langle T\rangle = \langle T_{\mathbf{l}}, T_{\mathbf{t}}, T_{\mathbf{c}}\rangle$ such that $T_{\mathbf{i}} = T$ and $T_{\mathbf{j}} = \emptyset$ for $\mathbf{j} \in S$, $\mathbf{j} \neq \mathbf{i}$. By identifying $T$ with $\langle T\rangle$, we may treat $T$ as a special sorted subset.

Let $\mathcal{M} = (M, \Gamma)$ be a $\Gamma$-algebra. For any $\mathbf{i}, \mathbf{j} \in S$, an *elementary* **ij**-*translation* is any mapping $M_{\mathbf{i}} \to M_{\mathbf{j}}$ obtained from one of the fundamental operations (of positive arity) of sort $\mathbf{j}$ of $\mathcal{M}$ by fixing the values of all arguments save one that is of sort $\mathbf{i}$. Let $\mathrm{ETr}(\mathcal{M}, \mathbf{i}, \mathbf{j})$ denote the set of all elementary **ij**-translations. Thus, for example, $\mathrm{ETr}(\mathcal{M}, \mathbf{l}, \mathbf{t}) = \{\iota^{\mathcal{M}}(\xi_{\mathbf{l}})\} \cup \{\kappa^{\mathcal{M}}(\xi_{\mathbf{l}}, u, v) \mid u, v \in M_{\mathbf{t}}\}$, where $\xi_{\mathbf{l}}$ is a variable of sort **label**. The sets $\mathrm{Tr}(\mathcal{M}, \mathbf{i}, \mathbf{j})$ of **ij**-*translations* ($\mathbf{i}, \mathbf{j} \in S$) are defined inductively by the following:

(1) $\mathrm{ETr}(\mathcal{M}, \mathbf{i}, \mathbf{j}) \subseteq \mathrm{Tr}(\mathcal{M}, \mathbf{i}, \mathbf{j})$ for all $\mathbf{i}, \mathbf{j} \in S$;
(2) for each $\mathbf{i} \in S$, the identity mapping $1_{\mathbf{i}} \colon M_{\mathbf{i}} \to M_{\mathbf{i}}, u \mapsto u$, is in $\mathrm{Tr}(\mathcal{M}, \mathbf{i}, \mathbf{i})$;
(3) if $\alpha(\xi_{\mathbf{i}}) \in \mathrm{Tr}(\mathcal{M}, \mathbf{i}, \mathbf{j})$ and $\beta(\xi_{\mathbf{j}}) \in \mathrm{Tr}(\mathcal{M}, \mathbf{j}, \mathbf{k})$ for some $\mathbf{i}, \mathbf{j}, \mathbf{k} \in S$, then $\beta(\alpha(\xi_{\mathbf{i}})) \in \mathrm{Tr}(\mathcal{M}, \mathbf{i}, \mathbf{k})$.

Let $\theta = \langle \theta_{\mathbf{l}}, \theta_{\mathbf{t}}, \theta_{\mathbf{c}} \rangle$ be a sorted equivalence on $M$, i.e., $\theta_{\mathbf{l}}$, $\theta_{\mathbf{t}}$ and $\theta_{\mathbf{c}}$ are equivalences on $M_{\mathbf{l}}$, $M_{\mathbf{t}}$ and $M_{\mathbf{c}}$, respectively. Then $\theta$ is a *congruence* on $\mathcal{M} = (M, \Gamma)$ if it is invariant with respect to the operations of $\mathcal{M}$. For example, for the $\kappa$-operation this means that for any $a, a' \in M_{\mathbf{l}}$ and $s, t, s', t' \in M_{\mathbf{t}}$, if $a \,\theta_{\mathbf{l}}\, a'$, $s \,\theta_{\mathbf{t}}\, s'$ and $t \,\theta_{\mathbf{t}}\, t'$, then $\kappa^{\mathcal{M}}(a, s, t) \,\theta_{\mathbf{t}}\, \kappa^{\mathcal{M}}(a', s', t')$. The congruences of a $\Gamma$-algebra $\mathcal{M}$ enjoy all the general properties the congruences of usual one-sorted algebras. In particular, every congruence of $\mathcal{M}$ is invariant with respect to every translation of $\mathcal{M}$ and, on the other hand, any sorted equivalence on $M$ that is invariant with respect to all elementary translations of $\mathcal{M}$ is a congruence.

**Definition 5.1.** The *syntactic congruence* $\approx^T$ of a subset $T \subseteq M_{\mathbf{i}}$ of some sort $\mathbf{i} \in S$ is the sorted equivalence $\langle \approx^T_{\mathbf{l}}, \approx^T_{\mathbf{t}}, \approx^T_{\mathbf{c}} \rangle$ on $M$ defined by the condition that for any $\mathbf{j} \in S$ and $u, v \in M_{\mathbf{j}}$,

$$u \approx^T_{\mathbf{j}} v \;\Leftrightarrow\; (\forall \alpha \in \mathrm{Tr}(\mathcal{M}, \mathbf{j}, \mathbf{i}))(\alpha(u) \in T \leftrightarrow \alpha(v) \in T),$$

and its *syntactic algebra* is $\mathcal{M}/T := \mathcal{M}/\approx^T$. For an element $u \in M_{\mathbf{j}}$ of any given sort $\mathbf{j} \in S$, we sometimes use $u/T$ as a shorthand for the congruence class $u/\approx^T_{\mathbf{j}}$ of $u$.

The syntactic congruences and syntactic algebras of (sorted or one-sorted) subsets of $\Gamma$-algebras have the same general properties as the corresponding notions defined for monoids and semigroups [8,22], for general algebras [1, 28–30], and for many-sorted algebras [25]. In fact, the following lemmas are special cases of facts presented in [25].

Recall that an equivalence $\theta$ on a set $U$ *saturates* a subset $L$ of $U$, if $L$ is the union of some $\theta$-classes. Similarly, a sorted equivalence $\theta$ on an $S$-sorted set $M$ *saturates* a subset $T \subseteq M_{\mathbf{i}}$ of some sort $\mathbf{i} \in S$, if $\theta_{\mathbf{i}}$ saturates $T$.

**Lemma 5.2.** *Let $\mathcal{M} = (M, \Gamma)$ be a $\Gamma$-algebra and $\mathbf{i} \in S$ be a sort. For any subset $T \subseteq M_{\mathbf{i}}$, $\approx^T$ is the greatest congruence on $\mathcal{M}$ that saturates $T$.*

Let $\alpha \in \mathrm{Tr}(\mathcal{M}, \mathbf{i}, \mathbf{j})$ be an **ij**-translation of a given $\Gamma$-algebra $\mathcal{M} = (M, \Gamma)$ for some $\mathbf{i}, \mathbf{j} \in S$. If $T \subseteq M_{\mathbf{k}}$ is a subset of some sort $\mathbf{k} \in S$, then let $\alpha^{-1}(T) := \{u \in M_{\mathbf{i}} \mid \alpha(u) \in T\}$ if $\mathbf{k} = \mathbf{j}$, and $\alpha^{-1}(T) := \emptyset$ otherwise. Furthermore, the relation $\varphi \circ \approx^T \circ \varphi^{-1}$ appearing in the following lemma denotes the sorted equivalence

$$\langle \varphi_{\mathbf{l}} \circ \approx^T_{\mathbf{l}} \circ \varphi_{\mathbf{l}}^{-1}, \varphi_{\mathbf{t}} \circ \approx^T_{\mathbf{t}} \circ \varphi_{\mathbf{t}}^{-1}, \varphi_{\mathbf{c}} \circ \approx^T_{\mathbf{c}} \circ \varphi_{\mathbf{c}}^{-1} \rangle$$

on $M$, where for all $\mathbf{j} \in S$ and $u, v \in M_{\mathbf{j}}$, $u \,\varphi_{\mathbf{j}} \circ \approx^T_{\mathbf{j}} \circ \varphi_{\mathbf{j}}^{-1}\, v$ iff $u\varphi_{\mathbf{j}} \approx^T_{\mathbf{j}} v\varphi_{\mathbf{j}}$.

Recall that $T^{\complement}$ is the complement of $T$.

**Lemma 5.3.** *Let $\mathcal{M} = (M, \Gamma)$ and $\mathcal{N} = (N, \Gamma)$ be any $\Gamma$-algebras.*

(a) $\approx^{T^{\complement}} = \approx^T$ *for any subset $T \subseteq M_{\mathbf{i}}$ of any sort $\mathbf{i} \in S$.*
(b) $\approx^T \cap \approx^U \subseteq \approx^{T \cap U}$ *and* $\approx^T \cap \approx^U \subseteq \approx^{T \cup U}$ *for any subsets $T, U \subseteq M_{\mathbf{i}}$ of any sort $\mathbf{i} \in S$.*
(c) *If $\alpha \in \mathrm{Tr}(\mathcal{M}, \mathbf{i}, \mathbf{j})$ is an **ij**-translation of $\mathcal{M}$ for some $\mathbf{i}, \mathbf{j} \in S$, then $\approx^T \subseteq \approx^{\alpha^{-1}(T)}$ for every subset $T \subseteq M_{\mathbf{k}}$ of any sort $\mathbf{k} \in S$.*
(d) *If $\varphi \colon \mathcal{M} \to \mathcal{N}$ is a homomorphism, then $\varphi \circ \approx^T \circ \varphi^{-1} \subseteq \approx^{T\varphi_{\mathbf{i}}^{-1}}$ for every subset $T \subseteq N_{\mathbf{i}}$ of any sort $\mathbf{i} \in S$. If $\varphi$ is an epimorphism, then $\varphi \circ \approx^T \circ \varphi^{-1} = \approx^{T\varphi_{\mathbf{i}}^{-1}}$ holds.*

Let us now formulate the corresponding facts for syntactic algebras. For this we need a couple of definitions.

**Definition 5.4.** A $\Gamma$-algebra $\mathcal{N}$ is said to *cover* a $\Gamma$-algebra $\mathcal{M}$, $\mathcal{M} \preceq \mathcal{N}$ in symbols, if $\mathcal{M}$ is an epimorphic image of a subalgebra of $\mathcal{N}$.

The covering relation generalizes both

- the subalgebra relation: $\mathcal{M} \subseteq \mathcal{N}$ iff $\mathcal{M}$ is (isomorphic to) a subalgebra of $\mathcal{N}$, and
- the image relation: $\mathcal{M} \leftarrow \mathcal{N}$ iff $\mathcal{M}$ is an epimorphic image of $\mathcal{N}$.

**Definition 5.5.** A $\Gamma$-algebra $\mathcal{N}$ is said to *recognize* a subset $T$ of some sort $\mathbf{i} \in I$ of $\mathcal{M}$ if there exist a homomorphism $\varphi\colon \mathcal{M} \to \mathcal{N}$ and subset $F \subseteq N_{\mathbf{i}}$ of sort $\mathbf{i}$ of $\mathcal{N}$ such that $L = F\varphi_{\mathbf{i}}^{-1}$.

**Lemma 5.6.** *Let $\mathcal{M}$ and $\mathcal{N}$ be $\Gamma$-algebras. Then $\mathcal{N}$ recognizes a subset $T \subseteq M_{\mathbf{i}}$ of some sort $\mathbf{i} \in S$ of $\mathcal{M}$ iff $\mathcal{M}/T \preceq \mathcal{N}$.*

The lemma expresses the fact that, in a certain sense, the syntactic algebra is the minimal $\Gamma$-algebra recognizing a given subset.

**Lemma 5.7.** *Let $\mathcal{M} = (M, \Gamma)$ and $\mathcal{N} = (N, \Gamma)$ be any $\Gamma$-algebras.*

(a) *$\mathcal{M}/T^{\complement} = \mathcal{M}/T$ for any subset $T \subseteq M_{\mathbf{i}}$ of any sort $\mathbf{i} \in S$.*
(b) *$\mathcal{M}/T \cap U \preceq \mathcal{M}/T \times \mathcal{M}/U$ and $\mathcal{M}/T \cup U \preceq \mathcal{M}/T \times \mathcal{M}/U$ for any subsets $T, U \subseteq M_{\mathbf{i}}$ of any sort $\mathbf{i} \in S$.*
(c) *If $\alpha \in \mathrm{Tr}(\mathcal{M}, \mathbf{i}, \mathbf{j})$ is an $\mathbf{ij}$-translation of $\mathcal{M}$ (where $\mathbf{i}, \mathbf{j} \in S$), then we have $\mathcal{M}/\alpha^{-1}(T) \leftarrow \mathcal{M}/T$ for every subset $T \subseteq M_{\mathbf{k}}$ of any sort $\mathbf{k} \in S$.*
(d) *If $\varphi\colon \mathcal{M} \to \mathcal{N}$ is a homomorphism, then $\mathcal{M}/T\varphi_{\mathbf{i}}^{-1} \preceq \mathcal{N}/T$ for every subset $T \subseteq N_{\mathbf{i}}$ and any sort $\mathbf{i} \in S$. If $\varphi$ is an epimorphism, then $\mathcal{M}/T\varphi_{\mathbf{i}}^{-1} \cong \mathcal{N}/T$.*

## 6. Syntactic tree algebras

When we define the syntactic congruences and syntactic algebras of binary tree languages by regarding these as subsets of sort **tree** of free tree algebras $\mathcal{F}_{\mathbf{TA}}(A)$, the definitions and facts of the previous section get more explicit forms.

**Definition 6.1.** Let $A$ be a label alphabet. The *syntactic tree algebra congruence*, the *STA-congruence* for short, of an $A$-tree language $T$ is its syntactic congruence as a subset of sort **tree** of the $\Gamma$-algebra $\mathcal{F}_{\mathbf{TA}}(A)$ of $A$-trees. The syntactic algebra $\mathcal{F}_{\mathbf{TA}}(A)/{\approx^T}$ is called the *syntactic tree algebra* of $T$, and it is denoted by $STA(T)$.

Since $\mathcal{F}_{\mathbf{TA}}(A)$ is a tree algebra, the syntactic tree algebras of $A$-tree languages are really tree algebras. To show that the above definition agrees with Wilke's [34] definitions, we need a careful analysis of the translations of the free tree algebras $\mathcal{F}_{\mathbf{TA}}(A)$. The relevant parts of such an analysis are presented in the following lemma.

**Lemma 6.2.** *Let $A$ be any label alphabet.*

(a) *A mapping $\alpha\colon A \to T_A$ is an $\mathbf{lt}$-translation of $\mathcal{F}_{\mathbf{TA}}(A)$ iff either*
  (1) *there is an $A$-context $p \in C_A$ such that $\alpha(a) = p(c_a)$ for every $a \in A$, or*
  (2) *there exist an $A$-context $p \in C_A$ and $A$-trees $s, t \in T_A$ such that $\alpha(a) = p(f_a(s, t))$ for every $a \in A$.*
(b) *A mapping $\alpha\colon T_A \to T_A$ is a $\mathbf{tt}$-translation of $\mathcal{F}_{\mathbf{TA}}(A)$ iff there is an $A$-context $p \in C_A$ such that $\alpha(t) = p(t)$ for every $t \in T_A$.*
(c) *A mapping $\alpha\colon C_A^+ \to T_A$ is a $\mathbf{ct}$-translation of $\mathcal{F}_{\mathbf{TA}}(A)$ iff there exist an $A$-context $r \in C_A^+$ and an $A$-tree $t \in T_A$ such that $\alpha(p) = r(p(t))$ for every $p \in C_A^+$.*

**Proof.** That all translations are expressible as claimed, can be proved by induction following the definition of the sets $\mathrm{Tr}(\mathcal{F}_{\mathbf{TA}}(A), \mathbf{i}, \mathbf{j})$ ($\mathbf{i}, \mathbf{j} \in S$) of translations of $\mathcal{F}_{\mathbf{TA}}(A)$. The complete proof presented in the Appendix of [26] involves numerous cases and also statements about the missing types of translations. Here we just illustrate the idea by some example cases.

For an elementary $\mathbf{lt}$-translation $\alpha(\xi_{\mathbf{l}}) = \kappa^{\mathcal{F}_{\mathbf{TA}}(A)}(\xi_{\mathbf{l}}, s, t)$, where $s, t \in T_A$, we have a case of alternative (2) in statement (a) where $p = \xi$ and $s, t \in T_A$ are the given $A$-trees $s$ and $t$. Indeed, $\alpha(a) = \kappa^{\mathcal{F}_{\mathbf{TA}}(A)}(a, s, t) = f_a(s, t) = \xi(f_a(s, t))$ for every $a \in A$.

Consider now an $\mathbf{lt}$-translation $\beta(\alpha(\xi_{\mathbf{l}}))$ obtained as the composition of an $\mathbf{lt}$-translation $\alpha$ and a $\mathbf{tt}$-translation $\beta$, and assume that there exist $A$-contexts $p, q \in C_A$ such that $\alpha(a) = p(c_a)$ for every $a \in A$, and $\beta(t) = q(t)$ for every $t \in T_A$. Then $q(p)$ is an $A$-context such that $q(p)(c_a) = \beta(\alpha(a))$ for every $a \in A$.

Of course, we should also show that all the mappings obtainable by the constructions mentioned in (a)–(c) really are translations of the appropriate types. For example, we have to prove that for any $p \in C_A$, the mapping $A \to T_A,\ a \mapsto p(c_a)$ is an $\mathbf{lt}$-translation of $\mathcal{F}_{\mathbf{TA}}(A)$. This can be done by induction on the $\xi$-depth of $p$. □

By using Lemma 6.2 and the observation that the **l**- and **c**-components of $\langle T\rangle$ are empty, we obtain a description of the STA-congruence of an $A$-tree language $T$ that is essentially Wilke's definition.

**Proposition 6.3.** *The STA-congruence $\approx^T$ of any $A$-tree language $T \subseteq T_A$ is obtained as follows. For any $a, b \in A$, $s, t \in T_A$ and $p, q \in C_A^+$,*

(a) $a \approx_{\mathbf{l}}^T b$ *iff*
  (1) $(\forall p \in C_A)\big(p(c_a) \in T \leftrightarrow p(c_b) \in T\big)$, *and*
  (2) $(\forall p \in C_A)(\forall s, t \in T_A)\big(p(f_a(s,t)) \in T \leftrightarrow p(f_b(s,t)) \in T\big)$,

(b) $s \approx_{\mathbf{t}}^T t$ *iff* $(\forall p \in C_A)\big(p(s) \in T \leftrightarrow p(t) \in T\big)$, *and*

(c) $p \approx_{\mathbf{c}}^T q$ *iff* $(\forall r \in C_A)(\forall t \in T_A)\big(r(p(t)) \in T \leftrightarrow r(q(t)) \in T\big)$.

Let us now show how syntactic tree algebras are related to the usual syntactic algebras [1,28–30] and the syntactic semigroups (obtained by a natural modification from the syntactic monoids of [33]). Then we obtain new proofs for Wilke's [34] basic results about syntactic tree algebras and recognizable binary tree languages. The following definitions are restricted directly to binary tree languages.

**Definition 6.4.** Let $T \subseteq T_A$ for some label alphabet $A$.

(a) The *syntactic congruence* of $T$ is the relation $\theta_T$ on $T_A$ defined by

$$s\,\theta_T\,t \;\Leftrightarrow\; (\forall p \in C_A)(p(s) \in T \leftrightarrow p(t) \in T) \qquad (s, t \in T_A),$$

and its *syntactic algebra* is the $\Sigma^A$-algebra $SA(T) := \mathcal{T}_A/\theta_T$.

(b) The *syntactic semigroup congruence* of $T$ is the relation $\sigma_T$ on $C_A^+$ defined by the condition that for any $p, q \in C_A^+$,

$$p\,\sigma_T\,q \;\Leftrightarrow\; (\forall t \in T_A)(\forall r \in C_A)(r(p(t)) \in T \leftrightarrow r(q(t)) \in T),$$

and the *syntactic semigroup* of $T$ is $SS(T) := C_A^+/\sigma_T$, where $C_A^+$ is regarded as a semigroup with respect to the product $p \cdot q = q(p)$.

The usual definition of a recognizable subset of an algebra [20] can be applied to a binary tree language $T \subseteq T_A$ either by regarding $T$ as a subset of the $\Sigma^A$-algebra $\mathcal{T}_A = (T_A, \Sigma^A)$ or as a subset of sort **tree** of the tree algebra $\mathcal{F}_{\mathbf{TA}}(A) = (\langle A, T_A, C_A^+\rangle, \Gamma)$. However, as shown by Wilke [34], the two definitions are equivalent. We choose the first alternative since it is immediately clear that it means recognizability by a finite tree recognizer (cf. [7,20,32,13, 14], for example).

**Definition 6.5.** Let $A$ be a label alphabet. An $A$-tree language $T \subseteq T_A$ is said to be *recognizable*, or *regular*, if there exist a finite $\Sigma^A$-algebra $\mathcal{D}$ and a subset $F$ of $\mathcal{D}$ such that $T = F\varphi_{\mathcal{D}}^{-1}$. Let $\mathrm{Rec}_A$ denote the set of all recognizable $A$-tree languages.

The above definition can also be expressed by saying that $T \in \mathrm{Rec}_A$ iff $T$ is saturated by a congruence on $\mathcal{T}_A$ of finite index. The following proposition includes the contents of Wilke's [34] Propositions 2 and 3.

**Proposition 6.6.** *For any binary tree language $T \subseteq T_A$ over any label alphabet $A$, the following conditions are equivalent:*

(1) $T \in \mathrm{Rec}_A$;
(2) *$SA(T)$ is a finite $\Sigma^A$-algebra;*
(3) *$SS(T)$ is a finite semigroup;*
(4) *$STA(T)$ is a finite tree algebra;*
(5) *$T$ is recognized by a finite tree algebra.*

**Proof.** That (1)–(3) are equivalent for tree languages quite generally is well known (cf. [13,14,29,31,33], for example).

Proposition 6.3 shows that $\theta_T = \approx_{\mathbf{t}}^T$ and $\sigma_T = \approx_{\mathbf{c}}^T$, and hence (4) implies (1)–(3). The equivalence of (4) and (5) follows from Lemma 5.6.

That (2) implies (4) follows from the fact that the syntactic congruence $\theta_T$ determines completely the syntactic semigroup congruence $\sigma_T$. Indeed, by comparing the definitions of the two relations, it is easy to see that for any $p, q \in C_A^+$, $p\,\sigma_T\,q$ holds iff $p(t)\,\theta_T\,q(t)$ for every $t \in T_A$. This means, in particular, that if $SA(T)$ is finite, then so is $SS(T)$, and hence also $STA(T)$ is finite as its **l**-component is always finite. □

The next two lemmas, needed in the variety theory, are also well-known in various other forms, and all of them can be derived from the general many-sorted theory of [25]. Here Lemma 6.7 follows from Lemma 5.7 when this is applied to free tree algebras, and Lemma 6.8 follows from Proposition 6.3(b).

Since the **tt**-translations of a free tree algebra $\mathcal{F}_{\mathbf{TA}}(A)$ are defined by $A$-contexts, we define $p^{-1}(T) := \{t \in T_A \mid p(t) \in T\}$ for any binary tree language $T \subseteq T_A$ and any $A$-context $p \in C_A$.

**Lemma 6.7.** *Let $A$ and $B$ be label alphabets. For any $A$-tree languages $T, U \subseteq T_A$,*

(a) $STA(T^{\complement}) = STA(T)$,
(b) $STA(T \cap U), STA(T \cup U) \preceq STA(T) \times STA(U)$,
(c) $STA(p^{-1}(T)) \leftarrow STA(T)$ *for every* $p \in C_A$, *and*
(d) $STA(T\varphi_{\mathbf{t}}^{-1}) \preceq STA(T)$ *for any homomorphism* $\varphi : \mathcal{F}_{\mathbf{TA}}(B) \to \mathcal{F}_{\mathbf{TA}}(A)$.

**Lemma 6.8.** *Let $T \subseteq T_A$ for some label alphabet $A$.*

(a) $T \in \mathrm{Rec}_A$ *iff the set* $\{p^{-1}(T) \mid p \in C_A\}$ *is finite.*
(b) *The* $\approx^T$*-class* $t/T$ *of any $A$-tree* $t \in T_A$ *can be given as*

$$\bigcap\{p^{-1}(T) \mid p \in C_A, p(t) \in T\} \setminus \bigcup\{p^{-1}(T) \mid p \in C_A, p(t) \notin T\}.$$

## 7. Varieties of binary tree languages

In this section we introduce varieties of binary tree languages. Although the general many-sorted theory of [25] yielded all the basic properties of syntactic tree algebras, the variety theorems of [25] are not directly applicable here. Firstly, the free algebras $\mathcal{F}_{\mathbf{TA}}(A)$ are always generated by sorted sets of the special form $\langle A, \emptyset, \emptyset \rangle$, not by arbitrary finite sorted sets. Secondly, we are now concerned just with subsets of sort **tree** while the varieties in [25] consist either of many-sorted sets or one-sorted sets of all possible sorts. In fact, the correspondence one could expect between varieties of binary tree languages and varieties of finite tree algebras fails to hold. The modifications necessary for a true variety theorem are introduced in the following section.

A *family of recognizable binary tree languages* is a mapping $\mathcal{V}$ that assigns to each label alphabet $A$ a set $\mathcal{V}(A) \subseteq \mathrm{Rec}_A$ of regular $A$-tree languages. We write $\mathcal{V} = \{\mathcal{V}(A)\}$ with the understanding that $A$ ranges over all label alphabets. The inclusion relation and various operations on such families are defined in the natural way: if $\mathcal{U} = \{\mathcal{U}(A)\}$ and $\mathcal{V} = \{\mathcal{V}(A)\}$ are families of recognizable binary tree languages, then

- $\mathcal{U} \subseteq \mathcal{V}$ iff $\mathcal{U}(A) \subseteq \mathcal{V}(A)$ for every label alphabet $A$,
- $\mathcal{U} \cap \mathcal{V}$ is the family $\mathcal{W} = \{\mathcal{W}(A)\}$ such that $\mathcal{W}(A) = \mathcal{U}(A) \cap \mathcal{V}(A)$ for every label alphabet $A$, etc.

**Definition 7.1.** A *variety of binary tree languages*, a *VBTL* for short, is a family of recognizable binary tree languages $\mathcal{V} = \{\mathcal{V}(A)\}$ such that for all label alphabets $A$ and $B$,

(1) $\mathcal{V}(A) \neq \emptyset$,
(2) if $T, U \in \mathcal{V}(A)$, then $T^{\complement}, T \cap U \in \mathcal{V}(A)$,
(3) if $T \in \mathcal{V}(A)$, then $p^{-1}(T) \in \mathcal{V}(A)$ for every $p \in C_A$, and
(4) if $\varphi : \mathcal{F}_{\mathbf{TA}}(A) \to \mathcal{F}_{\mathbf{TA}}(B)$ is a homomorphism, then $T\varphi_{\mathbf{t}}^{-1} \in \mathcal{V}(A)$ for every $T \in \mathcal{V}(B)$.

Let **VBTL** denote the class of all VBTLs.

A *variety of finite tree algebras*, a *VFTA* for short, is a nonempty class of finite tree algebras closed under subalgebras, homomorphic images and finite direct products. Let **VFTA** denote the class of all VFTAs.

In terms of the usual class operators $S$ and $H$ and the operator $P_{\mathrm{f}}$ that forms the class of all direct products with finitely many factors from a given class (cf. [5] or [2], for example), we can define a VFTA as a class $\mathbf{K}$ of finite tree algebras such that $S(\mathbf{K}), H(\mathbf{K}), P_{\mathrm{f}}(\mathbf{K}) \subseteq \mathbf{K}$.

It is clear that $(\mathbf{VBTL}, \subseteq)$ and $(\mathbf{VFTA}, \subseteq)$ are complete lattices. Therefore there is for each family of recognizable binary tree languages $\mathcal{V}$ a least VBTL containing $\mathcal{V}$, the *VBTL generated* by $\mathcal{V}$. Similarly, for any class $\mathbf{K}$ of finite tree algebras, the *VFTA generated* by $\mathbf{K}$ is the least VFTA containing $\mathbf{K}$ as a subclass.

The following fact, easy to prove and well-known from other similar situations, is frequently needed. Note that the value $n = 0$ yields the trivial tree algebras.

**Lemma 7.2.** *For any class* **K** *of finite tree algebras, the VFTA generated by* **K** *consists of the tree algebras* $\mathcal{M}$ *such that* $\mathcal{M} \preceq \mathcal{M}_1 \times \cdots \times \mathcal{M}_n$ *for some* $n \geq 0$ *and* $\mathcal{M}_1, \ldots, \mathcal{M}_n \in \mathbf{K}$.

Following the general pattern of various variety theorems we define two maps that connect the classes **VBTL** and **VFTA**.

**Definition 7.3.** For any family of recognizable binary tree languages $\mathcal{V} = \{\mathcal{V}(A)\}$, let $\mathcal{V}^a$ be the VFTA generated by the class of all syntactic tree algebras $STA(T)$ where $T \in \mathcal{V}(A)$ for some label alphabet $A$.

For any class **K** of finite tree algebras, $\mathbf{K}^t$ is the family of recognizable binary tree languages such that $\mathbf{K}^t(A) = \{T \subseteq T_A \mid STA(T) \in \mathbf{K}\}$ for each label alphabet $A$.

In the above definition, and in other similar situations, we tacitly assume that **K** is an abstract class of algebras, i.e., it contains every algebra isomorphic to any of its members. The following proposition shows how close to a variety theorem, that would establish an isomorphism between (**VBTL**, $\subseteq$) and (**VFTA**, $\subseteq$), we get with the above definitions.

**Proposition 7.4.** *Let* $\mathcal{U}$ *and* $\mathcal{V}$ *be families of recognizable binary tree languages, and let* **K** *and* **L** *be classes of finite tree algebras.*

(a) *If* $\mathcal{U} \subseteq \mathcal{V}$*, then* $\mathcal{U}^a \subseteq \mathcal{V}^a$.
(b) *If* $\mathbf{K} \subseteq \mathbf{L}$*, then* $\mathbf{K}^t \subseteq \mathbf{L}^t$.
(c) *If* $\mathcal{V} \in \mathbf{VBTL}$*, then* $\mathcal{V}^a \in \mathbf{VFTA}$.
(d) *If* $\mathbf{K} \in \mathbf{VFTA}$*, then* $\mathbf{K}^t \in \mathbf{VBTL}$.
(e) *If* $\mathcal{V} \in \mathbf{VBTL}$*, then* $\mathcal{V}^{at} = \mathcal{V}$.
(f) *If* $\mathbf{K} \in \mathbf{VFTA}$*, then* $\mathbf{K}^{ta} \subseteq \mathbf{K}$ *but the inclusion may be proper.*

**Proof.** Here we show just that the inclusion in (f) may be proper; the rest can be found in the proof of Proposition 8.7 below.

Let us consider the $\Gamma$-algebra $\mathcal{M} = (\langle \{a, b\}, \{t\}, \{p\} \rangle, \Gamma)$, where the operations are defined in the only possible way, i.e., $\iota^{\mathcal{M}}(a) = \iota^{\mathcal{M}}(b) = \kappa^{\mathcal{M}}(a, t, t) = \kappa^{\mathcal{M}}(b, t, t) = \eta^{\mathcal{M}}(p, t) = t$ and $\lambda^{\mathcal{M}}(a, t) = \lambda^{\mathcal{M}}(b, t) = \rho^{\mathcal{M}}(a, t) = \rho^{\mathcal{M}}(b, t) = \sigma^{\mathcal{M}}(p, p) = p$. Since the **t**- and **c**-components are singletons, it is clear that $\mathcal{M}$ satisfies the identities *TA* and is therefore a tree algebra. Let **K** be the VFTA generated by $\mathcal{M}$. The **t**-component of every member of **K** is also a singleton, and therefore $\mathbf{K}^t(A) = \{\emptyset, T_A\}$ for every label alphabet $A$. This means that $\mathbf{K}^{ta}$ is the class of trivial tree algebras and hence $\mathcal{M} \in \mathbf{K} \setminus \mathbf{K}^{ta}$. □

## 8. Reduced tree algebras and a variety theorem

There are natural reasons why a complete correspondence between the classes **VBTL** and **VFTA** was not obtained in the previous section. Firstly, since the algebras $\mathcal{F}_{\mathbf{TA}}(A)$ are generated by their **l**-components, so are the syntactic tree algebras of all binary tree languages. In fact, Wilke [34] anticipated a variety theorem that would involve varieties of such **l**-generated finite tree algebras. However, that something more is required, is indicated by the counterexample used in the proof of Proposition 7.4; the tree algebra $\mathcal{M}$ is **l**-generated. It turns out that we have to focus on tree algebras that do not have pairs of elements of sort **label** or **context** that are in a sense equivalent.

**Definition 8.1.** For any tree algebra $\mathcal{M} = (M, \Gamma)$, let $\mathcal{M}^l$ denote the subalgebra of $\mathcal{M}$ generated by $\langle M_{\mathbf{l}} \rangle = \langle M_{\mathbf{l}}, \emptyset, \emptyset \rangle$. If $\mathcal{M}^l = \mathcal{M}$, then $\mathcal{M}$ is said to be **l**-*generated*. An **l**-generated tree algebra $\mathcal{M} = (M, \Gamma)$ is *reduced* if it satisfies the following two additional conditions:

(1) For any $a, b \in M_{\mathbf{l}}$, if $\iota^{\mathcal{M}}(a) = \iota^{\mathcal{M}}(b)$ and $\kappa^{\mathcal{M}}(a, s, t) = \kappa^{\mathcal{M}}(b, s, t)$ for all $s, t \in M_{\mathbf{t}}$, then $a = b$.
(2) For any $p, q \in M_{\mathbf{c}}$, if $\eta^{\mathcal{M}}(p, t) = \eta^{\mathcal{M}}(q, t)$ for every $t \in M_{\mathbf{t}}$, then $p = q$.

Any tree algebra $\mathcal{M} = (M, \Gamma)$ can be reduced as follows. Let $\mathcal{M}^l = \mathcal{N} = (N, \Gamma)$, and let $\theta^{\mathcal{M}}$ be the sorted relation on $N$ such that

(1) for any $a, b \in N_{\mathbf{l}}$, $a\, \theta^{\mathcal{M}}_{\mathbf{l}}\, b$ iff $\iota^{\mathcal{N}}(a) = \iota^{\mathcal{N}}(b)$ & $(\forall s, t \in N_{\mathbf{t}})(\kappa^{\mathcal{N}}(a, s, t) = \kappa^{\mathcal{N}}(b, s, t))$,
(2) for any $s, t \in N_{\mathbf{t}}$, $s\, \theta^{\mathcal{M}}_{\mathbf{t}}\, t$ iff $s = t$, and
(3) for any $p, q \in N_{\mathbf{c}}$, $p\, \theta^{\mathcal{M}}_{\mathbf{c}}\, q$ iff $(\forall t \in N_{\mathbf{t}})(\eta^{\mathcal{N}}(p, t) = \eta^{\mathcal{N}}(q, t))$.

It is easy to see that $\theta^{\mathcal{M}}$ is a congruence on $\mathcal{N}$, and let $\mathcal{M}^r$ denote the quotient algebra $\mathcal{N}/\theta^{\mathcal{M}}$.

**Lemma 8.2.** *For any tree algebra $\mathcal{M}$, the tree algebra $\mathcal{M}^r$, as defined above, is reduced. If $\mathcal{M}$ is reduced, then $\mathcal{M}^r \cong \mathcal{M}$.*

**Proof.** Let us write $\mathcal{N} = \mathcal{M}^l$ and $\theta = \theta^{\mathcal{M}}$. Since $\mathcal{N}$ is **l**-generated, so is $\mathcal{M}^r = \mathcal{N}/\theta$. Assume that for some $a, b \in N_{\mathbf{l}}$,

(A) $\iota^{\mathcal{M}^r}(a/\theta_{\mathbf{l}}) = \iota^{\mathcal{M}^r}(b/\theta_{\mathbf{l}})$, and
(B) $(\forall s, t \in N_{\mathbf{t}})(\kappa^{\mathcal{M}^r}(a/\theta_{\mathbf{l}}, s/\theta_{\mathbf{t}}, t/\theta_{\mathbf{t}}) = \kappa^{\mathcal{M}^r}(b/\theta_{\mathbf{l}}, s/\theta_{\mathbf{t}}, t/\theta_{\mathbf{t}}))$.

Condition (A) is equivalent to $\iota^{\mathcal{M}}(a)/\theta_{\mathbf{t}} = \iota^{\mathcal{M}}(b)/\theta_{\mathbf{t}}$, and hence $\iota^{\mathcal{M}}(a) = \iota^{\mathcal{M}}(b)$ by the definition of $\theta_{\mathbf{t}}$. Similarly, $(\forall s, t \in N_{\mathbf{t}})(\kappa^{\mathcal{M}}(a, s, t) = \kappa^{\mathcal{M}}(b, s, t))$ follows from (B). Together (A) and (B) imply $a/\theta_{\mathbf{l}} = b/\theta_{\mathbf{l}}$ by the definition of $\theta_{\mathbf{l}}$. This means that $\mathcal{M}^r$ satisfies (1) of Definition 8.1. Condition (2) follows similarly from the fact that

$$(\forall t \in N_{\mathbf{t}})(\eta^{\mathcal{M}^r}(p/\theta_{\mathbf{c}}, t/\theta_{\mathbf{t}}) = \eta^{\mathcal{M}^r}(q/\theta_{\mathbf{c}}, t/\theta_{\mathbf{t}})) \;\Rightarrow\; p/\theta_{\mathbf{c}} = q/\theta_{\mathbf{c}},$$

for all $p, q \in N_{\mathbf{c}}$. Hence, $\mathcal{M}^r$ is reduced.

If $\mathcal{M}$ is reduced, then $\mathcal{M}^l = \mathcal{M}$, and each component of $\theta^{\mathcal{M}}$ is the identity relation on the respective set. Hence, $\mathcal{M}^r = \mathcal{M}/\theta^{\mathcal{M}} \cong \mathcal{M}$. □

**Lemma 8.3.** *For any tree algebras $\mathcal{M} = (M, \Gamma)$ and $\mathcal{N} = (N, \Gamma)$, if $\mathcal{M} \preceq \mathcal{N}$, then $\mathcal{M}^r \preceq \mathcal{N}^r$.*

**Proof.** The covering relation is transitive as $\mathcal{M} \preceq \mathcal{N}$ iff $\mathcal{M} \in HS(\{\mathcal{N}\})$, and the well-known properties of the class operators $S$ and $H$ by which $HSHS(\mathbf{K}) = HS(\mathbf{K})$ for any class $\mathbf{K}$ of algebras (cf. [5], for example). Therefore it suffices to prove the following special cases of the lemma:

(a) if $\mathcal{M} \subseteq \mathcal{N}$, then $\mathcal{M}^r \preceq \mathcal{N}^r$;
(b) if $\mathcal{M} \leftarrow \mathcal{N}$, then $\mathcal{M}^r \preceq \mathcal{N}^r$.

If $\mathcal{M} \subseteq \mathcal{N}$, then also $\mathcal{M}^l \subseteq \mathcal{N}^l$, and therefore we may assume in (a) that $\mathcal{M}$ and $\mathcal{N}$ are **l**-generated. Let $\mu := \theta^{\mathcal{M}}$ and let $\nu := \theta^{\mathcal{N}} \cap (M \times M)$ be the restriction of $\theta^{\mathcal{N}}$ to $M$. Then $\mathcal{M}/\nu \subseteq \mathcal{N}^r$, and therefore it is enough to show that $\nu \subseteq \mu$ because then $\mathcal{M}^r = \mathcal{M}/\mu \leftarrow \mathcal{M}/\nu$.

For any $a, b \in M_{\mathbf{l}}$,

$$\begin{aligned} a\, \nu_{\mathbf{l}}\, b &\Rightarrow \iota^{\mathcal{N}}(a) = \iota^{\mathcal{N}}(b) \;\&\; (\forall s, t \in N_{\mathbf{t}})(\kappa^{\mathcal{N}}(a, s, t) = \kappa^{\mathcal{N}}(b, s, t)) \\ &\Rightarrow \iota^{\mathcal{M}}(a) = \iota^{\mathcal{M}}(b) \;\&\; (\forall s, t \in M_{\mathbf{t}})(\kappa^{\mathcal{M}}(a, s, t) = \kappa^{\mathcal{M}}(b, s, t)) \\ &\Rightarrow a\, \mu_{\mathbf{l}}\, b, \end{aligned}$$

and hence $\nu_{\mathbf{l}} \subseteq \mu_{\mathbf{l}}$. It is obvious that $\nu_{\mathbf{t}} = \mu_{\mathbf{t}}$ and the inclusion $\nu_{\mathbf{c}} \subseteq \mu_{\mathbf{c}}$ is verified similarly as $\nu_{\mathbf{l}} \subseteq \mu_{\mathbf{l}}$. Hence $\nu \subseteq \mu$.

To prove (b), let $\varphi \colon \mathcal{N} \to \mathcal{M}$ be an epimorphism. Since $N_{\mathbf{l}}\varphi = M_{\mathbf{l}}$, it is clear that the restriction of $\varphi$ to $N^l$ is an epimorphism from $\mathcal{N}^l$ onto $\mathcal{M}^l$. We may therefore again assume that $\mathcal{M}$ and $\mathcal{N}$ themselves are **l**-generated. Let $\mu := \theta^{\mathcal{M}}$ and $\nu := \theta^{\mathcal{N}}$. We show now that the mapping $\psi : N/\nu \to M\mu$ defined by

$$\psi_{\mathbf{l}} \colon a/\nu_{\mathbf{l}} \mapsto a\varphi_{\mathbf{l}}/\mu_{\mathbf{l}}, \quad \psi_{\mathbf{t}} \colon t/\nu_{\mathbf{t}} \mapsto t\varphi_{\mathbf{t}}/\mu_{\mathbf{t}}, \quad \psi_{\mathbf{c}} \colon p/\nu_{\mathbf{c}} \mapsto p\varphi_{\mathbf{c}}/\mu_{\mathbf{c}}$$

(where $a \in N_{\mathbf{l}}, t \in N_{\mathbf{t}}, p \in N_{\mathbf{c}}$) is an epimorphism from $\mathcal{N}^r$ onto $\mathcal{M}^r$.

First we note that $\psi$ is well-defined. For example, for any $a, b \in N_{\mathbf{l}}$,

$$\begin{aligned} &a/\nu_{\mathbf{l}} = b/\nu_{\mathbf{l}} \Rightarrow \\ &\iota^{\mathcal{N}}(a) = \iota^{\mathcal{N}}(b) \;\&\; (\forall s, t \in N_{\mathbf{t}})(\kappa^{\mathcal{N}}(a, s, t) = \kappa^{\mathcal{N}}(b, s, t)) \Rightarrow \\ &\iota^{\mathcal{N}}(a)\varphi_{\mathbf{t}} = \iota^{\mathcal{N}}(b)\varphi_{\mathbf{t}} \;\&\; (\forall s, t \in N_{\mathbf{t}})(\kappa^{\mathcal{N}}(a, s, t)\varphi_{\mathbf{t}} = \kappa^{\mathcal{N}}(b, s, t)\varphi_{\mathbf{t}}) \Rightarrow \\ &\iota^{\mathcal{M}}(a\varphi_{\mathbf{l}}) = \iota^{\mathcal{M}}(b\varphi_{\mathbf{l}}) \;\&\; (\forall s, t \in N_{\mathbf{t}})(\kappa^{\mathcal{M}}(a\varphi_{\mathbf{l}}, s\varphi_{\mathbf{t}}, t\varphi_{\mathbf{t}}) = \kappa^{\mathcal{M}}(b\varphi_{\mathbf{l}}, s\varphi_{\mathbf{t}}, t\varphi_{\mathbf{t}})) \Rightarrow \\ &a\varphi_{\mathbf{l}}/\mu_{\mathbf{l}} = b\varphi_{\mathbf{l}}/\mu_{\mathbf{l}}, \end{aligned}$$

where the last equality depends on the assumption that $\varphi$ is surjective. Similarly, $s/\nu_{\mathbf{t}} = t/\nu_{\mathbf{t}}$ implies $s\varphi_{\mathbf{t}}/\mu_{\mathbf{t}} = t\varphi_{\mathbf{t}}/\mu_{\mathbf{t}}$ for any $s, t \in N_{\mathbf{t}}$, and $p/\nu_{\mathbf{c}} = q/\nu_{\mathbf{c}}$ implies $p\varphi_{\mathbf{c}}/\mu_{\mathbf{c}} = q\varphi_{\mathbf{c}}/\mu_{\mathbf{c}}$ for any $p, q \in N_{\mathbf{c}}$.

It is clear that $\psi$ is surjective. Finally, by routine computations it can be verified that $\psi$ is a homomorphism. For example,

$$\iota^{\mathcal{N}/\nu}(a/\nu_{\mathbf{l}})\psi_{\mathbf{t}} = (\iota^{\mathcal{N}}(a)/\nu_{\mathbf{t}})\psi_{\mathbf{t}} = \iota^{\mathcal{N}}(a)\varphi_{\mathbf{t}}/\mu_{\mathbf{t}} = \iota^{\mathcal{M}}(a\varphi_{\mathbf{l}})/\mu_{\mathbf{t}} = \iota^{\mathcal{M}/\mu}(a\varphi_{\mathbf{l}}/\mu_{\mathbf{l}}) = \iota^{\mathcal{M}/\mu}((a/\nu_{\mathbf{l}})\psi_{\mathbf{l}}),$$

for every $a \in N_{\mathbf{l}}$. □

The following proposition summarizes the main properties of $\mathcal{M}^r$ and shows that it is, up to isomorphism, the greatest reduced tree algebra covered by $\mathcal{M}$.

**Proposition 8.4.** *For any tree algebra $\mathcal{M}$, $\mathcal{M}^r$ is a reduced tree algebra such that $\mathcal{M}^r \preceq \mathcal{M}$. Moreover, if $\mathcal{N} \preceq \mathcal{M}$ for a reduced tree algebra $\mathcal{N}$, then $\mathcal{N} \preceq \mathcal{M}^r$.*

**Proof.** We know already that $\mathcal{M}^r$ is reduced and $\mathcal{M}^r \preceq \mathcal{M}$ follows directly from the definition. If $\mathcal{N}$ is a reduced tree algebra such that $\mathcal{N} \preceq \mathcal{M}$, then $\mathcal{N} \cong \mathcal{N}^r \preceq \mathcal{M}^r$ by Lemmas 8.2 and 8.3. □

Let us now note a couple of important facts about reduced tree algebras and syntactic tree algebras.

**Lemma 8.5.** *The syntactic tree algebra of any binary tree language is reduced. On the other hand, for any finite reduced tree algebra $\mathcal{M}$, there exist a label alphabet $A$ and regular $A$-tree languages $T_1, \ldots, T_n \subseteq T_A$, for some $n \geq 1$, such that $STA(T_j) \preceq \mathcal{M}$ for every $j = 1, \ldots, n$, and $\mathcal{M} \subseteq STA(T_1) \times \cdots \times STA(T_n)$.*

**Proof.** As a quotient algebra of $\mathcal{F}_{\mathbf{TA}}(A)$, the syntactic tree algebra $STA(T)$ of a binary tree language $T \subseteq T_A$ is naturally $\mathbf{l}$-generated. That $STA(T)$ satisfies conditions (1) and (2) of Definition 8.1 can be verified by using Proposition 6.3.

To prove the second claim of the proposition, take a label alphabet $A$ such that $|A| \geq |M_{\mathbf{l}}|$. Since $\mathcal{M}$ is $\mathbf{l}$-generated, there is an epimorphism $\varphi\colon \mathcal{F}_{\mathbf{TA}}(A) \to \mathcal{M}$. Assume that $M_{\mathbf{t}} = \{t_1, \ldots, t_n\}$ for some $n \geq 1$, and let $T_j := t_j\varphi_{\mathbf{t}}^{-1}$ for each $j = 1, \ldots, n$. By using Lemma 5.7(d) we obtain for every $j = 1, \ldots, n$,

$$STA(T_j) = \mathcal{F}_{\mathbf{TA}}(A)/T_j \cong \mathcal{M}/\{t_j\} \preceq \mathcal{M}.$$

To prove that $\mathcal{M} \subseteq STA(T_1) \times \cdots \times STA(T_n)$, it suffices to prove that $\mathcal{M}$ is isomorphic to a subalgebra of $\mathcal{M}/\{t_1\} \times \cdots \times \mathcal{M}/\{t_n\}$. To do this, we consider the mapping

$$\psi = \langle \psi_{\mathbf{l}}, \psi_{\mathbf{t}}, \psi_{\mathbf{c}} \rangle \colon \mathcal{M} \to \mathcal{M}/\{t_1\} \times \cdots \times \mathcal{M}/\{t_n\}$$

that maps each element $u \in M_{\mathbf{i}}$ ($\mathbf{i} \in S$) to $(u/\{t_1\}, \ldots, u/\{t_n\})$. It is clear that $\psi$ is a homomorphism from $\mathcal{M}$ to $\mathcal{M}/\{t_1\} \times \cdots \times \mathcal{M}/\{t_n\}$. Hence, it remains to be shown that $\psi$ is injective.

If $a\psi_{\mathbf{l}} = b\psi_{\mathbf{l}}$ for some $a, b \in M_{\mathbf{l}}$, then $a \approx_{\mathbf{l}}^{\{t_j\}} b$ for every $j = 1, \ldots, n$. In particular, $a \approx_{\mathbf{l}}^{\{\iota^{\mathcal{M}}(a)\}} b$ which implies that $\iota^{\mathcal{M}}(a) = \iota^{\mathcal{M}}(b)$. Similarly, $a \approx_{\mathbf{l}}^{\{\kappa^{\mathcal{M}}(a,s,t)\}} b$ implies $\kappa^{\mathcal{M}}(a, s, t) = \kappa^{\mathcal{M}}(b, s, t)$, for all $s, t \in M_{\mathbf{t}}$. Since $\mathcal{M}$ is reduced, this means that $a = b$.

If $s\psi_{\mathbf{t}} = t\psi_{\mathbf{t}}$ for some $s, t \in M_{\mathbf{t}}$, then $s \approx_{\mathbf{t}}^{\{s\}} t$ yields $s = t$.

Finally, if $p\psi_{\mathbf{c}} = q\psi_{\mathbf{c}}$ for some $p, q \in M_{\mathbf{c}}$, then $p \approx_{\mathbf{c}}^{\{\eta^{\mathcal{M}}(p,t)\}} q$ implies $\eta^{\mathcal{M}}(p, t) = \eta^{\mathcal{M}}(q, t)$ for every $t \in M_{\mathbf{t}}$. Since $\mathcal{M}$ is reduced, this means that $p = q$.

Hence we have shown that $\psi : \mathcal{M} \to \mathcal{M}/\{t_1\} \times \cdots \times \mathcal{M}/\{t_n\}$ is a monomorphism. □

**Definition 8.6.** A *variety of finite reduced tree algebras*, an *rVFTA* for short, is a nonempty class of finite reduced tree algebras $\mathbf{R}$ such that $\mathcal{N} \in \mathbf{R}$ whenever $\mathcal{N}$ is a reduced tree algebra and $\mathcal{N} \preceq \mathcal{M}_1 \times \cdots \times \mathcal{M}_n$ for some $n \geq 1$ and $\mathcal{M}_1, \ldots, \mathcal{M}_n \in \mathbf{R}$. Let **rVFTA** denote the class of all rVFTAs.

An rVFTA contains, in particular, all reduced subalgebras and all reduced images of its members. Since the intersection of any collection of rVFTAs is also an rVFTA, we may speak about the *rVFTA generated* by any given class of finite reduced tree algebras.

We now move towards establishing an isomorphism between the complete lattices (**rVFTA**, $\subseteq$) and (**VBTL**, $\subseteq$) thus obtaining the desired variety theorem. The mapping $\mathbf{R} \mapsto \mathbf{R}^t$ is defined as above but its application is restricted to classes of finite reduced tree algebras. The mapping $\mathcal{V} \mapsto \mathcal{V}^a$ is modified as follows: if $\mathcal{V} = \{\mathcal{V}(A)\}$ is any family of recognizable binary tree languages, then $\mathcal{V}^a$ is the rVFTA generated by the class of all syntactic tree algebras $STA(T)$ where $T \in \mathcal{V}(A)$ for some $A$.

**Proposition 8.7** (*The Variety Theorem*). *Let $\mathcal{U}$ and $\mathcal{V}$ be families of recognizable binary tree languages, and let* **P** *and* **R** *be classes of finite reduced tree algebras.*

(a) *If $\mathcal{U} \subseteq \mathcal{V}$, then $\mathcal{U}^a \subseteq \mathcal{V}^a$.*
(b) *If $\mathbf{P} \subseteq \mathbf{R}$, then $\mathbf{P}^t \subseteq \mathbf{R}^t$.*
(c) *If $\mathcal{V} \in$ **VBTL**, then $\mathcal{V}^a \in$ **rVFTA**.*
(d) *If $\mathbf{R} \in$ **rVFTA**, then $\mathbf{R}^t \in$ **VBTL**.*
(e) *If $\mathcal{V} \in$ **VBTL**, then $\mathcal{V}^{at} = \mathcal{V}$.*
(f) *If $\mathbf{R} \in$ **rVFTA**, then $\mathbf{R}^{ta} = \mathbf{R}$.*

**Proof.** Assertions (a) and (b) are obvious, (c) follows directly from the definition of $\mathcal{V}^a$, and (d) follows from Lemma 6.7.

As to (e), the inclusion $\mathcal{V} \subseteq \mathcal{V}^{at}$ is also obvious, and the less obvious converse inclusion can be shown by adapting suitably Eilenberg's [8] original proof similarly as, for example, in [29] (Proposition 7.3) or in [25] (Proposition 6.3) where the corresponding fact is proved in the general many-sorted case. For completeness and the reader's convenience we present such a proof for the current case, too.

Assume that $T \in \mathcal{V}^{at}$ for some $A$. Then $STA(T) \in \mathcal{V}^a$ implies that $STA(T) \preceq STA(T_1) \times \cdots \times STA(T_n)$, where $T_i \in \mathcal{V}(A_i)$ $(i = 1, \ldots, n)$ for some $n \geq 1$ and label alphabets $A_1, \ldots, A_n$. For each $i = 1, \ldots, n$, let $\varphi^i$ denote the syntactic homomorphism $\mathcal{F}_{\mathbf{TA}}(A_i) \to STA(T_i)$ that maps each element of $\mathcal{F}_{\mathbf{TA}}(A_i)$ to its $\approx^{T_i}$-class. Then there is a homomorphism

$$\beta\colon \mathcal{F}_{\mathbf{TA}}(A_1) \times \cdots \times \mathcal{F}_{\mathbf{TA}}(A_n) \to STA(T_1) \times \cdots \times STA(T_n)$$

such that $\beta\pi^i = \tau^i\varphi^i$ for each $i = 1, \ldots, n$, where

$$\pi^i : STA(T_1) \times \cdots \times STA(T_n) \to STA(T_i) \quad \text{and} \quad \tau^i\colon \mathcal{F}_{\mathbf{TA}}(A_1) \times \cdots \times \mathcal{F}_{\mathbf{TA}}(A_n) \to \mathcal{F}_{\mathbf{TA}}(A_i)$$

are the respective projections. By Lemma 5.6 there exist a homomorphism $\varphi\colon \mathcal{F}_{\mathbf{TA}}(A) \to STA(T_1) \times \cdots \times STA(T_n)$ and a subset $H$ of the product $STA(T_1)_{\mathbf{t}} \times \cdots \times STA(T_n)_{\mathbf{t}}$ such that $T = H\varphi_{\mathbf{t}}^{-1}$. Since $\beta$ is an epimorphism, there is a homomorphism $\psi\colon \mathcal{F}_{\mathbf{TA}}(A) \to \mathcal{F}_{\mathbf{TA}}(A_1) \times \cdots \times \mathcal{F}_{\mathbf{TA}}(A_n)$ such that $\psi\beta = \varphi$. Because $H$ is finite, $T$ is the union of the finitely many sets $u\varphi_{\mathbf{t}}^{-1}$ with $u = (u_1, \ldots, u_n) \in H$. Since $\varphi\pi^i = \psi\tau^i\varphi^i$ for every $i = 1, \ldots, n$, each such set can be expressed as

$$u\varphi_{\mathbf{t}}^{-1} = \bigcap\{u_i(\varphi_{\mathbf{t}}\pi^i_{\mathbf{t}})^{-1} \mid 1 \leq i \leq n\} = \bigcap\{u_i(\varphi^i_{\mathbf{t}})^{-1}(\psi\tau^i)_{\mathbf{t}}^{-1} \mid 1 \leq i \leq n\}.$$

It follows from Lemma 6.8 that $u_i(\varphi^i_{\mathbf{t}})^{-1} \in \mathcal{V}(T_i)$ for every $i = 1, \ldots, n$, and hence also $T \in \mathcal{V}(A)$.

The inclusion $\mathbf{R}^{ta} \subseteq \mathbf{R}$ in (f) follows from the fact that the syntactic tree algebras that generate $\mathbf{R}^{ta}$ are also in **R**. Indeed, if $T \in \mathbf{R}^t(A)$ for some $A$, then $STA(T)$ is in **R** by the definition of $\mathbf{R}^t$.

It remains to show that also $\mathbf{R} \subseteq \mathbf{R}^{ta}$ holds for any rVFTA **R**. Let us consider an $\mathcal{M} \in \mathbf{R}$. Since $\mathcal{M}$ is a finite reduced tree algebra, there exist by Lemma 8.5 a label alphabet $A$ and recognizable $A$-tree languages $T_1, \ldots, T_n \subseteq T_A$ $(n \geq 1)$ such that $STA(T_j) \preceq \mathcal{M}$ $(j = 1, \ldots, n)$ and $\mathcal{M} \subseteq STA(T_1) \times \cdots \times STA(T_n)$. Then $STA(T_j) \in \mathbf{R}$ and so $T_j \in \mathbf{R}^t(A)$ for every $j = 1, \ldots, n$, and hence by the inclusion $\mathcal{M} \subseteq STA(T_1) \times \cdots \times STA(T_n)$ we get $\mathcal{M} \in \mathbf{R}^{ta}$. □

To conclude this section, we shall note that every rVFTA is obtained as the class of reduced members of a VFTA, but that this fact does not establish a bijection between **rVFTA** and **VFTA** because a given rVFTA can be obtained from several VFTAs.

**Proposition 8.8.** *For any VFTA* **K***, the class* $\mathbf{K}^r$ *of all reduced members of* **K** *is an rVFTA. On the other hand, for each rVFTA* **R***, there is a VFTA* **K** *such that* $\mathbf{K}^r = \mathbf{R}$*, but this* **K** *is not necessarily unique for a given* **R**.

**Proof.** It follows easily from the definitions of VFTAs and rVFTAs that $\mathbf{K}^r \in$ **rVFTA** for any $\mathbf{K} \in$ **VFTA**, and also that if $\mathbf{R} \in$ **rVFTA** and **K** is the VFTA generated by **R**, then $\mathbf{K}^r = \mathbf{R}$.

For the last assertion, let **R** be the rVFTA of all trivial tree algebras. Then **R** is itself a VFTA such that $\mathbf{R}^r = \mathbf{R}$. On the other hand, we have $\mathbf{K}^r = \mathbf{R}$ also for the VFTA **K** of all finite tree algebras $\mathcal{M} = (M, \Gamma)$ such that $|M_{\mathbf{t}}| = 1$. □

## 9. VBTLs and general varieties of tree languages

The varieties of binary tree languages considered here are in some sense less general than the varieties of tree languages studied in [29], for example, but at the same time they are in some respect more general. Less general they are because they involve binary trees only and in that there are no separate leaf alphabets. On the other hand, a VBTL is not restricted to one ranked alphabet but contains tree languages over all alphabets of the form $\Sigma^A$. In this respect VBTLs resemble the general varieties of tree languages (GVTL) of [30] and the similar varieties studied in [18]. We shall show that each GVTL becomes a VBTL when restricted to the binary ranked alphabets $\Sigma^A$ considered here. Since many known families of regular tree languages are indeed GVTLs, this fact yields several natural examples of VBTLs. Such examples include the families of nilpotent, definite, reverse definite, generalized definite, locally testable and non-counting tree languages. For showing the connection between GVTLs and VBTLs we have to recall the definition of a GVTL.

Let $\Sigma$ and $\Omega$ be ranked alphabets. A *g-morphism* from a $\Sigma$-algebra $\mathcal{D} = (D, \Sigma)$ to an $\Omega$-algebra $\mathcal{E} = (E, \Omega)$ is a pair of mappings $\alpha\colon \Sigma \to \Omega$ and $\varphi\colon D \to E$ such that

(1) $\alpha(f) \in \Omega_m$ for any $f \in \Sigma_m$ and $m \geq 0$,
(2) $c^{\mathcal{D}}\varphi = \alpha(c)^{\mathcal{E}}$ for every $c \in \Sigma_0$, and
(3) $f^{\mathcal{D}}(d_1, \ldots, d_m)\varphi = \alpha(f)^{\mathcal{E}}(d_1\varphi, \ldots, d_m\varphi)$ for every $m > 0$, $f \in \Sigma_m$ and $d_1, \ldots, d_m \in D$.

It is easy to see (cf. [30]) that a g-morphism $(\alpha, \varphi)\colon \mathcal{T}_\Sigma(X) \to \mathcal{T}_\Omega(Y)$ between term algebras is essentially a relabelling of trees that replaces each label from $\Sigma$ with its $\alpha$-image. That leaves labelled with leaf symbols in $X$ may be replaced by any $\Omega Y$-trees, is of no consequence here because in a VBTL all leaf alphabets are empty (and not shown at all).

A *general variety of tree languages (GVTL)* is a family of regular tree languages $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ such that for all ranked alphabets $\Sigma$ and $\Omega$, and all leaf alphabets $X$ and $Y$,

(G1) $\mathcal{V}(\Sigma, X) \neq \emptyset$,
(G2) if $T, U \in \mathcal{V}(\Sigma, X)$, then $T^{\complement}, T \cap U \in \mathcal{V}(\Sigma, X)$,
(G3) if $T \in \mathcal{V}(\Sigma, X)$ and $p \in C_\Sigma(X)$, then $p^{-1}(T) \in \mathcal{V}(\Sigma, X)$, and
(G4) if $(\alpha, \varphi)\colon \mathcal{T}_\Sigma(X) \to \mathcal{T}_\Omega(Y)$ is a g-morphism, then $T\varphi^{-1} \in \mathcal{V}(\Sigma, X)$ for every $T \in \mathcal{V}(\Omega, Y)$.

If we restrict ourselves to the ranked alphabets $\Sigma^A$ obtained from label alphabets and assume that the leaf alphabets are always empty, then the above definition matches exactly the definition of a VBTL except for the last clauses concerning g-morphisms and homomorphism, respectively. Hence, to determine the relationship between VBTLs and these restricted GVTLs we have to describe the g-morphisms between the term algebras $\mathcal{T}_A$ and the **t**-components of homomorphisms between the free tree algebras $\mathcal{F}_{\mathbf{TA}}(A)$. The following two lemmas follow directly from the appropriate definitions.

**Lemma 9.1.** *Let A and B be any label alphabets. If* $(\alpha, \varphi)\colon \mathcal{T}_A \to \mathcal{T}_B$ *is a g-morphism, then*

(1) $c_a\varphi = \alpha(c_a)$ *for every* $a \in A$, *and*
(2) $f_a(s, t)\varphi = \alpha(f_a)(s\varphi, t\varphi)$ *for any* $a \in A$ *and* $s, t \in T_A$.

The lemma also shows that the mapping $\varphi\colon T_A \to T_B$ in a g-morphism $(\alpha, \varphi)\colon \mathcal{T}_A \to \mathcal{T}_B$ is a relabelling fully determined by $\alpha\colon \Sigma^A \to \Sigma^B$.

**Lemma 9.2.** *Let A and B be label alphabets. If* $\varphi : \mathcal{F}_{\mathbf{TA}}(A) \to \mathcal{F}_{\mathbf{TA}}(B)$ *is a homomorphism, then*

(1) $c_a\varphi_{\mathbf{t}} = c_{a\varphi_{\mathbf{l}}}$ *for every* $a \in A$, *and*
(2) $f_a(s, t)\varphi_{\mathbf{t}} = f_{a\varphi_{\mathbf{l}}}(s\varphi_{\mathbf{t}}, t\varphi_{\mathbf{t}})$ *for all* $a \in A$ *and* $s, t \in T_A$.

Hence, homomorphisms between free tree algebras also are just relabellings of binary trees. Moreover, it is clear that for any homomorphism $\varphi : \mathcal{F}_{\mathbf{TA}}(A) \to \mathcal{F}_{\mathbf{TA}}(B)$ there is a g-morphism $(\alpha, \psi)\colon \mathcal{T}_A \to \mathcal{T}_B$ such that $t\varphi_{\mathbf{t}} = t\psi$ for every $t \in T_A$; we just define $\alpha$ by setting $\alpha(c_a) = c_{a\varphi_{\mathbf{l}}}$ and $\alpha(f_a) = f_{a\varphi_{\mathbf{l}}}$ for every $a \in A$, and this is consistent with the idea that $c_a$ and $f_a$ actually represent the same label $a$. This means that (G4) in the above definition of a GVTL implies the corresponding condition in the definition of a VBTL. The following fact is now obvious.

**Proposition 9.3.** *For any GVTL $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$, the family of recognizable binary tree languages $\mathcal{V}^b = \{\mathcal{V}^b(A)\}$, where $\mathcal{V}^b(A) = \mathcal{V}(\Sigma^A, \emptyset)$ for each label alphabet A, is a VBTL.*

The relabellings defined by homomorphisms between free tree algebras are somewhat less general than the g-morphisms of Lemma 9.1 because of the bindings between the pairs $c_a, f_a$ $(a \in A)$: if $c_a$ maps to $c_b$, $f_a$ has to map to $f_b$, and conversely. This means that (G4) is strictly stronger than clause (4) in Definition 7.1, even when restricted to our binary tree languages, and therefore it is conceivable that not every VBTL is obtained as a restriction of a GVTL. That this is indeed the case is shown by the following example.

**Example 9.4.** For each label alphabet $A$, let $\mathcal{V}(A)$ be the set of all regular $A$-tree languages $T \subseteq T_A$ such that $f_a(c_a, t) \approx_{\mathbf{t}}^T t$ for all $a \in A$ and $t \in T_A$. It is easy to verify that $\mathcal{V} = \{\mathcal{V}(A)\}$ is a VBTL. Assume that $\mathcal{V} = \mathcal{U}^b$ for some GVTL $\mathcal{U} = \{\mathcal{U}(\Sigma, X)\}$. Let $A = \{a, b\}$ and define the $A$-contexts $p_a = f_a(c_a, \xi)$ and $p_b = f_b(c_b, \xi)$. Let $T$ be the least $A$-tree language such that $c_a \in T$ and $p_a(t), p_b(t) \in T$ for every $t \in T$. Then $T \in \mathcal{V}(A) = \mathcal{U}(\Sigma^A, \emptyset)$. Consider the g-morphism $(\alpha, \varphi)\colon \mathcal{T}_A \to \mathcal{T}_B$ defined by the assignment

$$\alpha\colon \Sigma^A \to \Sigma^A,\ c_a \mapsto c_a, c_b \mapsto c_b, f_a \mapsto f_b, f_b \mapsto f_a,$$

and the $A$-tree $t = f_a(c_a, c_a)$. Now $t \in T\varphi^{-1}$ but $p_a(t) \notin T\varphi^{-1}$ because $t\varphi = f_b(c_a, c_a) = p_b(c_a) \in T$ while $p_a(t)\varphi = f_b(c_a, f_b(c_a, c_a)) \notin T$. Hence, $T\varphi^{-1} \notin \mathcal{U}(\Sigma^A, \emptyset)$, a contradiction, and we have shown that $\mathcal{V} = \mathcal{U}^b$ for no GVTL $\mathcal{U}$.

Whether there are more natural examples of varieties of binary tree languages that cannot be obtained from a GVTL remains to be seen.

The relationship between the two theories can be illuminated also by considering the corresponding syntactic algebras. First we show how the syntactic tree algebra of a binary tree language can be obtained from its syntactic algebra.

The set $\mathrm{Tr}^+(\mathcal{D})$ of (non-unit) *translations* of a $\Sigma$-algebra $\mathcal{D} = (D, \Sigma)$ is the least set of unary operations on $D$ that (1) contains every *elementary translation*

$$D \to D,\ x \mapsto f^{\mathcal{D}}(d_1, \ldots, d_{i-1}, x, d_{i+1}, \ldots, d_m) \quad (x \in D),$$

where $m > 0$, $f \in \Sigma_m$, $1 \le i \le m$ and $d_1, \ldots, d_{i-1}, d_{i+1}, \ldots, d_m \in D$ are given, and (2) is closed under composition. Note that $\mathrm{Tr}^+(\mathcal{D})$ contains the identity map of $D$ only if this is the composition of some elementary translations.

**Definition 9.5.** Let $A$ be a label alphabet and $\mathcal{D} = (D, \Sigma^A)$ be any $\Sigma^A$-algebra. Let $\delta_{\mathcal{D}}$ be the equivalence on $A$ defined by

$$a\,\delta_{\mathcal{D}}\,b \ \Leftrightarrow\ c_a^{\mathcal{D}} = c_b^{\mathcal{D}}\ \&\ f_a^{\mathcal{D}} = f_b^{\mathcal{D}} \quad (a, b \in A).$$

Now the $\Gamma$-algebra $\mathcal{D}^\bullet = (\langle A/\delta_{\mathcal{D}}, D, \mathrm{Tr}^+(\mathcal{D})\rangle, \Gamma)$ is defined by setting for all $a \in A$, $d, e \in D$ and $p, q \in \mathrm{Tr}^+(\mathcal{D})$,

(1) $\iota^{\mathcal{D}^\bullet}(a/\delta_{\mathcal{D}}) = c_a^{\mathcal{D}}$ (2) $\kappa^{\mathcal{D}^\bullet}(a/\delta_{\mathcal{D}}, d, e) = f_a^{\mathcal{D}}(d, e)$
(3) $\lambda^{\mathcal{D}^\bullet}(a/\delta_{\mathcal{D}}, d) = f_a^{\mathcal{D}}(\xi, d)$ (4) $\rho^{\mathcal{D}^\bullet}(a/\delta_{\mathcal{D}}, d) = f_a^{\mathcal{D}}(d, \xi)$
(5) $\eta^{\mathcal{D}^\bullet}(p, d) = p(d)$ (6) $\sigma^{\mathcal{D}^\bullet}(p, q) = p(q)$.

The operations of $\mathcal{D}^\bullet$ are well-defined by the definition of $\delta_{\mathcal{D}}$. Moreover, the following holds.

**Lemma 9.6.** *For any label alphabet A and any $\Sigma^A$-algebra $\mathcal{D} = (D, \Sigma^A)$, the $\Gamma$-algebra $\mathcal{D}^\bullet$ is a tree algebra. Furthermore, if $\mathcal{D}$ is generated by the empty set, then $\mathcal{D}^\bullet$ is reduced.*

**Proof.** It is easy to verify that $\mathcal{D}^\bullet$ satisfies the identities *TA*. Suppose $\mathcal{D}$ is generated by $\emptyset$. To see that $\mathcal{D}^\bullet$ is **l**-generated, we apply Definition 9.5:

(a) $A/\delta_{\mathcal{D}}$ generates all of $D$ by (1) and (2), and
(b) all elementary translations of $\mathcal{D}^\bullet$ are obtained from $A/\delta_{\mathcal{D}}$ and $D$ by (3) and (4), and all of their compositions are obtained by (6).

That $\mathcal{D}^\bullet$ satisfies condition (1) of Definition 8.1 follows from the definition of $\delta_{\mathcal{D}}$, and condition (2) follows from the definition of $\eta^{\mathcal{D}^\bullet}$. □

**Proposition 9.7.** *$SA(T)^{\bullet} \cong STA(T)$ for any binary tree language $T$.*

**Proof.** Assume that $T \subseteq T_A$ for some leaf alphabet $A$. Let us compare

$$SA(T)^{\bullet} = (\langle A/\delta_{SA(T)}, T_A/\theta_T, \mathrm{Tr}^+(SA(T))\rangle, \Gamma)$$

with

$$STA(T) = \mathcal{F}_{\mathbf{TA}}(A)/\approx^T = (\langle A/\approx^T_{\mathbf{l}}, T_A/\approx^T_{\mathbf{t}}, C^+_A/\approx^T_{\mathbf{c}}\rangle, \Gamma).$$

First of all, we may replace $\mathrm{Tr}^+(SA(T))$ with $C^+_A/\sigma_T$ since by Lemma 6.2, for every $\alpha \in \mathrm{Tr}^+(SA(T))$, there is a $p \in C^+_A$ such that $\alpha(t/\theta_T) = p(t)/\theta_T$ for every $t \in T_A$, and if $p\,\sigma_T\,q$ for some $p, q \in C^+_A$, then $p(t)/\theta_T = q(t)/\theta_T$ for every $t \in T_A$.

In the proof of Proposition 6.6 we already noted that $\theta_T = \approx^T_{\mathbf{t}}$ and $\sigma_T = \approx^T_{\mathbf{c}}$. That also $\delta_{SA(T)} = \approx^T_{\mathbf{l}}$ holds follows from the definitions of $\delta_{SA(T)}$ and $SA(T)$ by repeated use of Proposition 6.3: for any $a, b \in A$,

$$\begin{aligned} a\,\delta_{SA(T)}\,b \;&\Leftrightarrow\; c_a^{SA(T)} = c_b^{SA(T)} \;\&\; f_a^{SA(T)} = f_b^{SA(T)} \\ &\Leftrightarrow\; c_a \approx^T_{\mathbf{t}} c_b \;\&\; (\forall s, t \in T_A)(f_a(s,t) \approx^T_{\mathbf{t}} f_b(s,t)) \\ &\Leftrightarrow\; a \approx^T_{\mathbf{l}} b. \end{aligned}$$

To show that the sorted identity map defines an isomorphism between the tree algebras $SA(T)^{\bullet}$ and $STA(T)$, we have to verify that the operations of the two $\Gamma$-algebras are the same. This can be done by straightforward, though somewhat tedious, computations directly based on the relevant definitions. As examples, we consider the $\kappa$- and $\lambda$-operations.

For any $a \in A$ and $s, t \in T_A$,

$$\begin{aligned} \kappa^{SA(T)^{\bullet}}(a/\delta_{SA(T)}, s/\theta_T, t/\theta_T) &= f_a^{SA(T)}(s/\theta_T, t/\theta_T) = f_a^{\mathcal{T}_A}(s,t)/\theta_T \\ &= f_a(s,t)/\theta_T = f_a(s,t)/\approx^T_{\mathbf{t}} \\ &= \kappa^{\mathcal{F}_{\mathbf{TA}}(A)}(a,s,t)/\approx^T_{\mathbf{t}} \\ &= \kappa^{STA(T)}(a/\approx^T_{\mathbf{l}}, s/\approx^T_{\mathbf{t}}, t/\approx^T_{\mathbf{t}}). \end{aligned}$$

When considering the operations involving elements of sort **context**, we identify each translation $T_A/\theta_T \to T_A/\theta_T,\ t/\theta_T \mapsto p(t)/\theta_T$ with the $\approx^T_{\mathbf{c}}$-class of any $A$-context $p \in C_A$ that defines it. For example, $\lambda^{SA(T)^{\bullet}} = \lambda^{STA}$ is then seen as follows. For any $a \in A$ and $s, t \in T_A$,

$$\begin{aligned} \lambda^{SA(T)^{\bullet}}(a/\delta_{SA(T)}, t/\theta_T)(s/\theta_T) &= f_a^{SA(T)}(\xi, t/\theta_T)(s/\theta_T) \\ &= f_a^{SA(T)}(s/\theta_T, t/\theta_T) = f_a(s,t)/\theta_T \\ &= f_a(s,t)/\approx^T_{\mathbf{t}} = \lambda^{\mathcal{F}_{\mathbf{TA}}(A)}(a,t)(s)/\approx^T_{\mathbf{t}} \\ &= \lambda^{STA(T)}(a/\approx^T_{\mathbf{l}}, t/\approx^T_{\mathbf{t}})(s/\approx^T_{\mathbf{t}}). \quad \square \end{aligned}$$

**Corollary 9.8.** *Let $A$ be any label alphabet. For any $A$-tree languages $T$ and $U$, if $SA(T) \cong SA(U)$, then $STA(T) \cong STA(U)$.*

Although the syntactic tree algebra of any binary tree language is determined by its ordinary syntactic algebra, there is a subtle point to be observed that explains why not every BVTL is obtained from a GVTL.

In the theory of GVTLs the syntactic invariant used to characterize a tree language $T \subseteq T_\Sigma(X)$ is its *reduced syntactic algebra* $RA(T)$ (cf. [30]). This is obtained from $SA(T)$ by merging equivalent symbols in $\Sigma^A$ similarly as we merged label symbols when $\mathcal{M}^r$ was constructed from a tree algebra $\mathcal{M}$. However, in the case of a binary tree language $T \subseteq T_A$, the construction of $RA(T)$ may merge two symbols $c_a$ and $c_b$ without merging $f_a$ and $f_b$, or conversely, and in such a case $a$ and $b$ are not merged in $STA(T)$.

**Example 9.9.** Let us consider the $A$-tree language $T = \{c_a\}$ for $A = \{a, b\}$. Clearly, $T_A/\theta_T = \{T, T^{\complement}\}$, and

$$c_a^{SA(T)} = T,\ c_b^{SA(T)} = T^{\complement},\ f_a^{SA(T)}(u,v) = f_b^{SA(T)}(u,v) = T^{\complement},$$

for all $u, v \in T_A/\theta_T$. Hence, $f_a$ and $f_b$ are merged when $RA(T)$ is constructed but $c_a$ and $c_b$ are not. Of course, $a$ and $b$ are not merged in the $\mathbf{l}$-component of $STA(T)$.

In the GVTL-theory any two algebras $\mathcal{D} = (D, \Sigma)$ and $\mathcal{E} = (E, \Omega)$, possibly over different ranked alphabets, are in effect equivalent if they are *g-isomorphic*, $\mathcal{D} \cong_g \mathcal{E}$ in symbols; a *g-isomorphism* is a g-morphism in which both mappings are bijective.

**Remark 9.10.** The syntactic tree algebras of two binary tree languages may be non-isomorphic even when their syntactic algebras (or even reduced syntactic algebras) are g-isomorphic. More precisely: there exist a leaf alphabet $A$ and two $A$-tree languages $T$ and $U$ such that $SA(T) \cong_g SA(U)$ and $RA(T) \cong_g RA(U)$, but $STA(T) \ncong STA(U)$.

**Proof.** Let $A = \{a, b\}$ and let us consider the $A$-tree languages

$$T = \{c_a\} \cup \{f_a(s,t) \mid s, t \in T_A\} \text{ and } U = \{c_a\} \cup \{f_b(s,t) \mid s, t \in T_A\}.$$

Now $T_A/\theta_T = \{T, T^{\complement}\}$ and $T_A/\theta_U = \{U, U^{\complement}\}$, and we may let $RA(T) = SA(T)$ and $RA(U) = SA(U)$ because in neither case are there any pairs of equivalent symbols. It is easy to verify that the pair of maps

$$\alpha\colon \Sigma^A \to \Sigma^A,\ c_a \mapsto c_a,\ c_b \mapsto c_b,\ f_a \mapsto f_b,\ f_b \mapsto f_a,$$
$$\varphi\colon T_A/\theta_T \to T_A/\theta_U,\ T \mapsto U,\ T^{\complement} \mapsto U^{\complement},$$

is a g-isomorphism from $RA(T)$ to $RA(U)$. However, $STA(T) \ncong STA(U)$ because $STA(T)$ satisfies the identity $\iota(a) \simeq \kappa(a, \mathrm{s}, \mathrm{t})$ while $STA(U)$ does not. Indeed, for any $s, t \in T_A$,

$$\iota^{STA(T)}(d/\approx_{\mathbf{l}}^T) = c_d/\approx_{\mathbf{t}}^T = f_d(s,t)/\approx_{\mathbf{t}}^T = \kappa^{STA(T)}(d/\approx_{\mathbf{l}}^T, s/\approx_{\mathbf{t}}^T, t/\approx_{\mathbf{t}}^T),$$

for both $d = a$ and $d = b$, while
$$\iota^{STA(U)}(a/\approx_{\mathbf{l}}^U) = c_a/\approx_{\mathbf{t}}^U \neq f_a(s,t)/\approx_{\mathbf{t}}^U = \kappa^{STA(U)}(a/\approx_{\mathbf{l}}^U, s/\approx_{\mathbf{t}}^U, t/\approx_{\mathbf{t}}^U). \quad \square$$

Remark 9.10 suggests that by using syntactic tree algebras we can make some distinctions between binary tree languages that cannot be made by using syntactic algebras or reduced syntactic algebras. However, this depends again on the bond between a leaf label $c_a$ and an interior node label $f_a$ created by the definition of the syntactic tree congruence, and it may be hard to find any natural examples where the difference could be utilized.

To complete the picture, we introduce a partial converse of the construction of Definition 9.5.

**Definition 9.11.** Let $\mathcal{M} = (M, \Gamma)$ be a tree algebra such that $M_{\mathbf{l}}$ is a finite set. Treating $M_{\mathbf{l}}$ as a label alphabet and denoting it by $A$, we let $\mathcal{M}^\circ = (M_{\mathbf{t}}, \Sigma^A)$ be the $\Sigma^A$-algebra such that for any $a \in A$,

(1) $c_a^{\mathcal{M}^\circ} = \iota^{\mathcal{M}}(a)$, and
(2) $f_a^{\mathcal{M}^\circ}(u, v) = \kappa^{\mathcal{M}}(a, u, v)$ for all $u, v \in M_{\mathbf{t}}$.

As a general example, we note that $\mathcal{F}_{\mathbf{TA}}(A)^\circ = \mathcal{T}_A$ for any label alphabet $A$.

Consider now any leaf alphabet $A$ and any $A$-tree language $T$, and set $\bar{A} := A/\approx_{\mathbf{l}}^T$ and $\bar{a} := a/\approx_{\mathbf{l}}^T$ for every $a \in A$. Since $\approx_{\mathbf{t}}^T = \theta_T$ and $\approx_{\mathbf{c}}^T = \sigma_T$, we may then write $STA(T) = (\langle \bar{A}, T_A/\theta_T, C_A^+/\sigma_T\rangle, \Gamma)$. By easy computations, one may verify that for every $a \in A$,

(1) $c_{\bar{a}}^{STA(T)^\circ} = c_a/\theta_T = c_a^{SA(T)}$, and
(2) $f_{\bar{a}}^{STA(T)^\circ}(s/\theta_T, t/\theta_T) = f_a(s,t)/\theta_T = f_a^{SA(T)}(s/\theta_T, t/\theta_T)$ for all $s, t \in T_A$.

Hence, $\alpha\colon A \to \bar{A}$, $a \mapsto \bar{a}$, and the identity map $\varphi\colon t/\theta_T \mapsto t/\theta_T$ of $T_A/\theta_T$, define a g-morphism $(\alpha, \varphi)\colon SA(T) \to STA(T)^\circ$. Since $\alpha$ is surjective and $\varphi$ the identity map, this means that $STA(T)^\circ$ is very similar to $SA(T)$, the only possible difference being that some identical operations of $SA(T)$ may be replaced by one operation in $STA(T)^\circ$. On the other hand, $RA(T)$ is easily seen to be obtained from $STA(T)^\circ$ by possibly further merging some equally realized operators $c_{\bar{a}}$ and $c_{\bar{b}}$, or $f_{\bar{a}}$ and $f_{\bar{b}}$, for which $a \approx_{\mathbf{l}}^T b$ does not hold. Any characterization of $T$ in terms of $SA(T)$, or $RA(T)$, is therefore likely to yield a characterization in terms of $STA(T)$. This is illustrated by some examples in the following section.

## 10. Equational descriptions of VBTLs

Although basically the same families of binary tree languages can be characterized in terms of syntactic tree algebras as by syntactic algebras, or reduced syntactic algebras, in many cases the language of tree algebras appears to be very convenient for defining the class of tree algebras corresponding to a given VBTL. In [34] Wilke gave an effective characterization of *frontier testable*, or *reverse definite*, binary tree languages in terms of syntactic tree algebras and also presented equational definitions of the corresponding class of finite tree algebras. We shall present some further examples of equational descriptions of tree algebras for some well-known families of tree languages. However, first we consider certain special $\Gamma$-terms and identities involving such terms.

Let $\mathrm{p}, \mathrm{q}, \mathrm{r}, \mathrm{p}_1, \mathrm{p}_2, \ldots, \mathrm{q}_1, \mathrm{q}_2, \ldots$ and $\mathrm{s}, \mathrm{t}, \mathrm{s}_1, \mathrm{s}_2, \ldots, \mathrm{t}_1, \mathrm{t}_2, \ldots$ be variables of sort **context** and **tree**, respectively. It is a major advantage of the language of tree algebras that it admits such variables. For any $k \geq 0$, let

- $\mathrm{p}_k \cdots \mathrm{p}_1(\mathrm{t}) := \eta(\mathrm{p}_k, \eta(\mathrm{p}_{k-1}, \ldots, \eta(\mathrm{p}_1, \mathrm{t}) \ldots))$, and
- $\mathrm{p}_k \cdots \mathrm{p}_1 := \sigma(\mathrm{p}_k, \sigma(\mathrm{p}_{k-1}, \ldots, \sigma(\mathrm{p}_2, \mathrm{p}_1) \ldots))$.

For $k = 0$, these expressions stand for t and $\xi$, respectively.

Let $\zeta$ be a valuation of the variables in a given tree algebra $\mathcal{M} = (M, \Gamma)$. If $\zeta(\mathrm{p}_1) = p_1, \ldots, \zeta(\mathrm{p}_k) = p_k$ $(\in M_{\mathbf{c}})$ and $\zeta(\mathrm{t}) = t$ $(\in M_{\mathbf{t}})$, then $\mathrm{p}_k \cdots \mathrm{p}_1(\mathrm{t})^{\mathcal{M}}(\zeta)$ denotes the value $\eta^{\mathcal{M}}(p_k, \ldots, \eta^{\mathcal{M}}(p_1, t) \ldots)$ of the term $\mathrm{p}_k \cdots \mathrm{p}_1(\mathrm{t})$ in $\mathcal{M}$ for the valuation $\zeta$. Similarly, let $\mathrm{p}_k \cdots \mathrm{p}_1^{\mathcal{M}}(\zeta)$ denote $\sigma^{\mathcal{M}}(p_k, \ldots, \sigma^{\mathcal{M}}(p_2, p_1) \ldots)$. In terms of these conventions, we can say that an identity

$$\mathrm{p}_h \cdots \mathrm{p}_1(\mathrm{s}) \approx \mathrm{q}_k \cdots \mathrm{q}_1(\mathrm{t})$$

*holds* in a tree algebra $\mathcal{M}$, or is *satisfied* by $\mathcal{M}$, if

$$\mathrm{p}_h \cdots \mathrm{p}_1(\mathrm{s})^{\mathcal{M}}(\zeta) = \mathrm{q}_k \cdots \mathrm{q}_1(\mathrm{t})^{\mathcal{M}}(\zeta)$$

for all valuations $\zeta$ of the variables in $\mathcal{M}$.

For any label alphabet $A$, in $\mathcal{F}_{\mathbf{TA}}(A)$ variables of sort **context** and variables of sort **tree** range over the set $C_A^+$ of non-unit $A$-contexts and the set $T_A$ of $A$-trees, respectively.

**Lemma 10.1.** *Let $A$ be a label alphabet, $t \in T_A$ and $q \in C_A^+$, and let us consider any terms $\mathrm{p}_h \cdots \mathrm{p}_1(\mathrm{s})$ and $\mathrm{q}_k \cdots \mathrm{q}_1$, where $h \geq 0$ and $k \geq 1$. Then*

(a) $\mathrm{hg}(t) \geq h$ *iff there exists a valuation $\zeta$ of the variables in $\mathcal{F}_{\mathbf{TA}}(A)$ such that $\mathrm{p}_h \cdots \mathrm{p}_1(\mathrm{s})^{\mathcal{F}_{\mathbf{TA}}(A)}(\zeta) = t$, and*
(b) $\mathrm{d}_\xi(q) \geq k$ *iff there exists a valuation $\zeta$ of the variables in $\mathcal{F}_{\mathbf{TA}}(A)$ such that $\mathrm{q}_k \cdots \mathrm{q}_1^{\mathcal{F}_{\mathbf{TA}}(A)}(\zeta) = q$.*

*In other words,*

(a') $\mathrm{hg}(t) \geq h$ *iff $t = p_h(\ldots p_1(s) \ldots) = s \cdot p_1 \cdot \ldots \cdot p_h$ for some $p_1, \ldots, p_h \in C_A^+$ and $s \in T_A$, and*
(b') $\mathrm{d}_\xi(q) \geq k$ *iff $q = q_k(\ldots q_2(q_1) \ldots) = q_1 \cdot q_2 \cdot \ldots \cdot q_k$ for some $q_1, \ldots, q_k \in C_A^+$.*

Consider any label alphabet $A$ and any $\Sigma^A$-algebra $\mathcal{D} = (D, \Sigma^A)$, and let $\varphi_{\mathcal{D}}$ be the unique homomorphism from $\mathcal{T}_A$ to $\mathcal{D}$. Each $A$-context $p \in C_A$ defines a unary operation $p^{\mathcal{D}} : D \to D$ as follows:

(1) $\xi^{\mathcal{D}} : d \mapsto d$ is the identity map $1_D : D \to D$, and
(2) if $p = f_a(q, s)$ or $p = f_a(s, q)$, for some $a \in A$, $q \in C_A$ and $s \in T_A$, then for every $d \in D$, $p^{\mathcal{D}}(d) = f_a^{\mathcal{D}}(q^{\mathcal{D}}(d), s\varphi_{\mathcal{D}})$ or $p^{\mathcal{D}}(d) = f_a^{\mathcal{D}}(s\varphi_{\mathcal{D}}, q^{\mathcal{D}}(d))$, respectively.

It is clear that each $p^{\mathcal{D}}$ is a translation of $\mathcal{D}$, and if $\varphi_{\mathcal{D}}$ is surjective, then every translation of $\mathcal{D}$ is obtained this way. It is also clear that if $p = q(r)$ for some $q, r \in C_A$, then $p^{\mathcal{D}}$ is the composition of $q^{\mathcal{D}}$ and $r^{\mathcal{D}}$, that is to say, $p^{\mathcal{D}}(d) = q^{\mathcal{D}}(r^{\mathcal{D}}(d))$ for every $d \in D$.

The following lemma results from the above discussion.

**Lemma 10.2.** *Let $A$ be any leaf alphabet and let $\mathcal{M} = (M, \Gamma)$ be a tree algebra such that $M_{\mathbf{l}} = A$. Then the following hold for all $h, k \geq 0$.*

(a) *$\mathcal{M}$ satisfies $\mathrm{p}_h \cdots \mathrm{p}_1(\mathrm{s}) \approx \mathrm{q}_k \cdots \mathrm{q}_1(\mathrm{t})$ iff $p^{\mathcal{M}^\circ}(u) = q^{\mathcal{M}^\circ}(v)$ holds for all $u, v \in M_{\mathbf{t}}$ and all $p, q \in C_A$ such that $\mathrm{d}_\xi(p) \geq h$ and $\mathrm{d}_\xi(q) \geq k$.*

(b) $\mathcal{M}$ *satisfies* $\mathrm{p}_k \cdots \mathrm{p}_1(\mathrm{s}) \approx \mathrm{p}_k \cdots \mathrm{p}_1(\mathrm{t})$ *iff* $p^{\mathcal{M}^\circ}(u) = p^{\mathcal{M}^\circ}(v)$ *holds for all* $u \in M_{\mathbf{t}}$ *and all* $p \in C_A$ *such that* $\mathrm{d}_\xi(p) \geq k$.

(c) $\mathcal{M}$ *satisfies* $\mathrm{p}_h \cdots \mathrm{p}_1 \approx \mathrm{q}_k \cdots \mathrm{q}_1$ *iff* $p^{\mathcal{M}^\circ} = q^{\mathcal{M}^\circ}$ *holds for all* $p, q \in C_A$ *such that* $\mathrm{d}_\xi(p) \geq h$ *and* $\mathrm{d}_\xi(q) \geq k$.

Recall that an algebra $\mathcal{D}$ (of any kind) is said to *ultimately satisfy* (cf. [8]) a sequence of identities $u_1 \approx v_1, u_2 \approx v_2, u_3 \approx v_3, \ldots$ if there is an $n \geq 1$ such that $\mathcal{D}$ satisfies $u_k \approx v_k$ for every $k \geq n$.

The *term function* $t^{\mathcal{D}} \colon D^X \to D$ induced by a term $t \in T_\Sigma(X)$ in a given $\Sigma$-algebra $\mathcal{D} = (D, \Sigma)$ is defined as follows. For any assignment $\alpha \colon X \to D$ of values to the variables,

(1) $c^{\mathcal{D}}(\alpha) = c^{\mathcal{D}}$ for every $c \in \Sigma_0$,
(2) $x^{\mathcal{D}}(\alpha) = \alpha(x)$ for every $x \in X$, and
(3) $t^{\mathcal{D}}(\alpha) = f^{\mathcal{D}}(t_1^{\mathcal{D}}(\alpha), \ldots, t_m^{\mathcal{D}}(\alpha))$ if $t = f(t_1, \ldots, t_m)$.

As the first example we consider the GVTL $Nil = \{Nil(\Sigma, X)\}$ where for each pair $\Sigma$ and $X$, $Nil(\Sigma, X)$ is the set of all finite $\Sigma X$-tree languages and their complements in $T_\Sigma(X)$. In [12] a finite algebra $\mathcal{D} = (D, \Sigma)$ was defined to be *nilpotent* if there is an element $d_0 \in D$ and a number $k > 0$ such that for any leaf alphabet $X$, and any $t \in T_\Sigma(X)$ such that $\mathrm{hg}(t) \geq k$, $t^{\mathcal{D}}(\alpha) = d_0$ for every assignment $\alpha \colon X \to D$. In other words, if $\mathcal{D}$ is viewed as a deterministic bottom-up tree automaton, it reaches the root of any tree of height $\geq k$ in state $d_0$ for any assignment $\alpha$ of initial states to the leaf symbols. If $\mathcal{D}$ is nilpotent, the minimal value of $k$ is called its *degree of nilpotency*. In [29] it was noted that for any fixed $\Sigma$, the class $\mathbf{Nil}_\Sigma$ of all nilpotent $\Sigma$-algebras is the variety of finite $\Sigma$-algebras that corresponds to the family $Nil_\Sigma = \{Nil(\Sigma, X)\}$, where $\Sigma$ is now fixed and $X$ ranges over all leaf alphabets. This fact is easily extended to a correspondence between the GVTL $Nil$ and the *generalized variety of finite algebras* (GVFA; cf. [30], p. 13) **Nil** of all nilpotent algebras. This means that for any $\Sigma$ and $X$, a $\Sigma X$-tree language $T$ is in $Nil(\Sigma, X)$ iff $SA(T) \in \mathbf{Nil}_\Sigma$, or equivalently, iff $RA(T) \in \mathbf{Nil}$.

Let $A$ be any leaf alphabet. It is easy to see that a finite $\Sigma^A$-algebra $\mathcal{D} = (D, \Sigma^A)$ is nilpotent if there exist a $k \geq 0$ and an element $d_0 \in D$ such that $p^{\mathcal{D}}(d) = d_0$ for every $d \in D$ whenever $p \in C_A^+$ is an $A$-context with $d_\xi(p) \geq k$. This means by Lemma 10.2(a) that for a finite tree algebra $\mathcal{M} = (M, \Gamma)$ such that $M_{\mathbf{l}} = A$, the algebra $\mathcal{M}^\circ = (M_{\mathbf{t}}, \Sigma^A)$ (defined in the previous section) is nilpotent iff $\mathcal{M}$ satisfies the identity $\mathrm{p}_k \cdots \mathrm{p}_1(\mathrm{s}) \approx \mathrm{q}_k \cdots \mathrm{q}_1(\mathrm{t})$ for some $k \geq 0$. Furthermore, it is clear that the syntactic algebra $SA(T)$ of any regular $A$-tree language $T$ is nilpotent iff $STA(T)^\circ$ is nilpotent. By collecting together the above observations, we obtain the following description of the VBTL $Nil^b$.

**Proposition 10.3.** *If $T$ is any regular $A$-tree language for some label alphabet $A$, then $T \in Nil^b(A)$ iff $STA(T)$ ultimately satisfies the sequence of identities*

$$\mathrm{p}_1(\mathrm{s}) \approx \mathrm{q}_1(\mathrm{t}),\ \mathrm{p}_2 \cdot \mathrm{p}_1(\mathrm{s}) \approx \mathrm{q}_2 \cdot \mathrm{q}_1(\mathrm{t}),\ \mathrm{p}_3 \cdot \mathrm{p}_2 \cdot \mathrm{p}_1(\mathrm{s}) \approx \mathrm{q}_3 \cdot \mathrm{q}_2 \cdot \mathrm{q}_1(\mathrm{t}), \ldots. \tag{N}$$

A couple of remarks are in order here. One can write for any given label alphabet $A$ a sequence of $\Sigma^A$-equations that ultimately defines the class of nilpotent $\Sigma^A$-algebras. For example, if $A = \{a, b\}$, the class of nilpotent $\Sigma^A$-algebras is ultimately defined by a sequence

$$x_1 \approx x_2,\ f_a(x_1, x_2) \approx f_b(x_3, x_4),\ f_a(f_a(x_1, x_2), x_3) \approx f_a(x_4, f_a(x_5, x_6)),$$
$$f_a(f_a(x_1, x_2), x_3) \approx f_a(x_4, f_b(x_5, x_6)),\ \ldots,$$

that for each $k \geq 0$, contains a set of identities defining the class of $\Sigma^A$-algebras of degree of nilpotency $\leq k$. However, this sequence is more complicated than the sequence (Nil) and it also depends on $A$. On the other hand, it has to be noted that a description of a VBTL like the above proposition does not yield automatically a decision method; we still need some bound for the degree of nilpotency of a nilpotent algebra in terms of its size, for example.

As our next example we consider definite tree languages. A tree language is *definite* if there is a bound $k \geq 0$ such that the membership of a tree in the language can be decided by looking at its root-segment of height $k$. Definite tree languages were first studied by Heuter [15,16], their variety properties were noted in [29,30], and in [9] Ésik describes the corresponding algebras and studies their structure. The following formal definitions are from [15,16] as modified in [29].

Let $\Sigma$ be any ranked alphabet, $X$ any leaf alphabet and $k \geq 0$. For any $\Sigma X$-tree $t$, let $\mathrm{root}(t)$ denote the label of the root of $t$. The *$k$-root* $\mathrm{r}_k(t)$ of a $\Sigma X$-tree $t$ is now defined as follows:

(1) $r_0(t) = \epsilon$ (the empty root segment) for every $t \in T_\Sigma(X)$.
(2) $r_1(t) = \mathrm{root}(t)$ for every $t \in T_\Sigma(X)$.
(3) Let $k \geq 2$. If $\mathrm{hg}(t) \leq k$, then $r_k(t) = t$. If $\mathrm{hg}(t) > k$ and $t = f(t_1, \ldots, t_m)$, then $r_k(t) = f(r_{k-1}(t_1), \ldots, r_{k-1}(t_m))$.

A tree language $T \subseteq T_\Sigma(X)$ is *k-definite* ($k \geq 0$) if for all $s, t \in T_\Sigma(X)$, if $r_k(s) = r_k(t)$, then $s \in T$ iff $t \in T$. A tree language is *definite* if it is $k$-definite for some $k \geq 0$. Let $Def_k(\Sigma, X)$ and $Def(\Sigma, X)$ denote the sets of all $k$-definite and all definite $\Sigma X$-tree languages, respectively. For any $k \geq 0$, $Def_k := \{Def_k(\Sigma, X)\}$ is a GVTL, and so is the union $Def := \{Def(\Sigma, X)\}$ of these families (cf. [30]).

A $\Sigma$-algebra $\mathcal{D} = (D, \Sigma)$ is said to be *k-definite* ($k \geq 0$) if for every $X$ and any $s, t \in T_\Sigma(X)$, if $r_k(s) = r_k(t)$, then $s^{\mathcal{D}}(\alpha) = t^{\mathcal{D}}(\alpha)$ for every $\alpha : X \to D$. An algebra is *definite* if it is $k$-definite for some $k \geq 0$.

In [9] it was shown that a tree language is ($k$-)definite iff its syntactic algebra is ($k$-)definite. To turn this fact into an equational tree algebra characterization, we need one more observation: for any $k \geq 0$, $\Sigma$, $X$ and $s, t \in T_\Sigma(X)$, $r_k(s) = r_k(t)$ holds iff $s = p_k(\ldots p_1(s')\ldots)$ and $t = p_k(\ldots p_1(t')\ldots)$ for some $p_1, \ldots, p_k \in C^+_\Sigma(X)$ and $s', t' \in T_\Sigma(X)$.

By applying the above definitions and facts to the alphabets $\Sigma^A$ and binary tree languages, we get by Lemma 10.2(b) the following result.

**Proposition 10.4.** *Let $T$ be a regular A-tree language for some label alphabet A. Then $T \in Def^b(A)$ iff $STA(T)$ ultimately satisfies the sequence of identities*

$$\mathrm{p}_1(\mathrm{s}) \approx \mathrm{p}_1(\mathrm{t}),\ \mathrm{p}_2 \cdot \mathrm{p}_1(\mathrm{s}) \approx \mathrm{p}_2 \cdot \mathrm{p}_1(\mathrm{t}),\ \mathrm{p}_3 \cdot \mathrm{p}_2 \cdot \mathrm{p}_1(\mathrm{s}) \approx \mathrm{p}_3 \cdot \mathrm{p}_2 \cdot \mathrm{p}_1(\mathrm{t}), \ldots. \tag{D}$$

Again we can note that one could apply the equational descriptions of definite algebras given by Ésik [9] to obtain, for each $A$, a sequence that ultimately defines the class of definite $\Sigma^A$-algebras, but such a sequence is more complicated than (D) and it depends on $A$. As shown by Heuter [15,16], and by Ésik [9], it is decidable whether a given finite algebra is definite or not, but this does not follow from the equational descriptions alone.

As a further, somewhat different, example, we consider the aperiodic tree languages introduced by Thomas [33]. A $\Sigma X$-tree language $T$ is *aperiodic*, or *non-counting*, if there is an $n \geq 0$ such that for all $p, q \in C^+_\Sigma(X)$ and $t \in T_\Sigma(X)$, $t \cdot p^n \cdot q \in T$ iff $t \cdot p^{n+1} \cdot q \in T$. If $Ap(\Sigma, X)$ denotes the set of all aperiodic $\Sigma X$-tree languages, then $Ap := \{Ap(\Sigma, X)\}$ is a GVTL (cf. [30]). In [33] it is shown that a tree language $T$ is aperiodic iff its syntactic monoid $SM(T)$ is *aperiodic*, i.e., all of its subgroups are trivial. A semigroup $M$ is known to be aperiodic iff there exists an $n \geq 0$ such that $x^{n+1} = x^n$ for every $x \in M$ (cf. [8] or [22], for example).

The **c**-component $C^+_A/\approx^T_{\mathbf{c}}$ of the syntactic tree algebra of a binary tree language $T \subseteq T_A$ forms a semigroup with respect to the product $p/T \cdot q/T := p \cdot q/T$. This semigroup is isomorphic to the syntactic semigroup $SS(T)$, and differs from the syntactic monoid $SM(T)$ only in that it does not necessarily have a unit element. Hence, $T$ is aperiodic iff $C^+_A/\approx^T_{\mathbf{c}}$ is an aperiodic semigroup. By Lemma 10.2(c) we may now turn Thomas' result into the following equational characterization, where $\mathrm{p}^n$ stands for the $n$-fold product $\mathrm{p} \cdot \mathrm{p} \cdots \mathrm{p}$ ($n \geq 1$).

**Proposition 10.5.** *If $T$ is a regular A-tree language for some label alphabet A, then $T \in Ap^b(A)$ iff $STA(T)$ ultimately satisfies the sequence of identities*

$$\mathrm{p}^2 \approx \mathrm{p},\ \mathrm{p}^3 \approx \mathrm{p}^2,\ \mathrm{p}^4 \approx \mathrm{p}^3, \ldots. \tag{A}$$

## 11. Concluding remarks

We have developed the theory of tree algebras and tree algebra representations of binary trees in a systematic algebraic way, and explored the relationships between this formalism and some other approaches to the classification of regular tree languages. The new results include also a Variety Theorem. Of course, many questions remain to be studied. For example, one could try to extend the formalism in such a way that the restriction to binary trees could be lifted. Alternatively, one could borrow some ideas from the tree algebra theory to other formalisms to increase their expressive power. One would also like to see further effective characterizations of natural families of binary tree languages in terms of syntactic tree algebras. However, in view of our results, it seems that they would, in most cases, be virtually equivalent to characterizations in terms of ordinary syntactic algebras.

## Acknowledgements

The authors want to thank the two referees for their helpful remarks.

## References

[1] J. Almeida, On pseudovarieties, varieties of languages, pseudoidentities and related topics, Algebra Universalis 27 (1990) 333–350.
[2] J. Almeida, Finite Semigroups and Universal Algebra, World Scientific, Singapore, 1994.
[3] J. Avenhaus, Reduktionssysteme, Springer, Berlin, 1995.
[4] T. Baader, T. Nipkow, Term Rewriting and All That, Cambridge University Press, Cambridge, 1998.
[5] S. Burris, H.P. Sankappanavar, A Course in Universal Algebra, Springer, New York, 1981.
[6] N. Dershowitz, J.-P. Jouannaud, Rewrite systems, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, vol. B, Elsevier Science Publisher B.V., Amsterdam, 1990, pp. 243–320.
[7] J. Doner, Tree acceptors and some of their applications, J. Comput. System Sci. 4 (1970) 406–451.
[8] S. Eilenberg, Automata, Languages and Machines, vol. B, Academic Press, New York, 1976.
[9] Z. Ésik, Definite tree automata and their cascade composition, Publ. Math. Debrecen 48 (3–4) (1996) 243–261.
[10] Z. Ésik, A variety theorem for trees and theories, in: Automata and formal languages VIII (Salgótarján, 1996), Publ. Math. Debrecen 54 (Suppl.) (1999) 711–762.
[11] Z. Ésik, P. Weil, On logically defined recognizable tree languages, in: P.K. Pandya, J. Radhakrishnan (Eds.), Foundations of Software Technology and Theoretical Computer Science (Proc. Conf. FSTTCS'03, Bombay 2003), in: Lect. Notes in Comput. Sci, vol. 2914, Springer-Verlag, Heidelberg, 2003, pp. 195–207.
[12] F. Gécseg, B. Imreh, On a special class of tree automata, in: Automata, Languages and Programming Systems (Proc. Conf., Salgótarján, Hungary, 1988), Department of Mathematics, K. Marx University of Economics, Budapest, 1988, pp. 141–152.
[13] F. Gécseg, M. Steinby, Tree Automata, Akadémiai Kiadó, Budapest, 1984.
[14] F. Gécseg, M. Steinby, Tree languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, vol. 3, Springer-Verlag, Berlin, 1997, pp. 1–69.
[15] U. Heuter, Definite tree languages, Bull. EATCS 35 (1988) 137–144.
[16] U. Heuter, Generalized definite tree languages, in: Mathematical Foundations of Computer Science (Proc. Symp. MFCS'89), in: A. Kreczmar, G. Mirkowska (Eds.), Lect. Notes in Comput. Sci., vol. 379, Springer-Verlag, Berlin, 1989, pp. 270–280.
[17] G. Huet, D.C. Oppen, Equations and rewrite rules, in: R.V. Book (Ed.), Formal Language Theory, Perspectives and Open Problems, Academic Press, New York, 1980, pp. 349–405.
[18] E. Jurvanen, A. Potthoff, W. Thomas, Tree languages recognizable by regular frontier check, in: G. Rozenberg, A. Salomaa (Eds.), Developments in Language Theory, World Scientific, Singapore, 1994, pp. 3–17.
[19] K. Meinke, J.V. Tucker, Universal algebra, in: S. Abramsky, D. Gabbay, T.S. Maibaum (Eds.), Handbook of Logic in Computer Science, vol. 1, Clarendon Press, Oxford, 1992, pp. 189–411.
[20] J. Mezei, J.B. Wright, Algebraic automata and context-free sets, Inform. Control 11 (1967) 3–29.
[21] M. Nivat, A. Podelski, Tree monoids and recognizability of sets of finite trees, in: H. Aït-Kaci, M. Nivat (Eds.), Resolution of Equations in Algebraic Structures, vol. 1, Academic Press, Boston, MA, 1989, pp. 351–367.
[22] J.E. Pin, Varieties of Formal Languages, North Oxford Academic Publishers, Oxford, 1986.
[23] S. Salehi, Varieties of tree languages definable by syntactic monoids, Acta Cybernet. 17 (2005) 21–41.
[24] S. Salehi, Varieties of Tree Languages, TUCS Dissertations 64, Turku Centre for Computer Science, Turku, 2005.
[25] S. Salehi, M. Steinby, Varieties of many-sorted recognizable sets, TUCS Technical Reports 629, September 2004. URL: http://www.tucs.fi/publications/insight.php?id=tSaSt04a.
[26] S. Salehi, M. Steinby, Tree algebras and varieties of tree languages, TUCS Technical Reports 761, March 2006. URL: http://www.tucs.fi/publications/insight.php?id=tSaSt06a.
[27] K. Salomaa, Syntactic monoids of regular forests (in Finnish), M.Sc. Thesis, Department of Mathematics, University of Turku, Turku, 1983.
[28] M. Steinby, Syntactic algebras and varieties of recognizable sets, in: M.C. Gaudel, J.P. Jouannaud (Eds.), Les Arbres en Algèbre et en Programmation (Proc. 4th CAAP, Lille 1979), University of Lille, Lille, 1979, pp. 226–240.
[29] M. Steinby, A theory of tree language varieties, in: M. Nivat, A. Podelski (Eds.), Tree Automata and Languages, North-Holland, Amsterdam, 1992, pp. 57–81.
[30] M. Steinby, General varieties of tree languages, Theoret. Comput. Sci. 205 (1998) 1–43.
[31] M. Steinby, Algebraic characterizations of regular tree languages, in: V.B. Kudryavtsev, I.G. Rosenberg (Eds.), Structural Theory of Automata, Semigroups, and Universal Algebra, in: NATO Science Series, II. Mathematics, Physics and Chemistry, vol. 207, Springer, Dordrecht, 2005, pp. 381–432.
[32] J.W. Thatcher, J.B. Wright, Generalized finite automata theory with an application to a decision problem of second-order logic, Math. Systems Theory 2 (1968) 57–81.
[33] W. Thomas, Logical aspects in the study of tree languages, in: B. Courcelle (Ed.), The 9th Colloquium on Trees in Algebra and Programming (Proc. 9th CAAP, Bordeaux 1984), Cambridge University Press, London, 1984, pp. 31–49.
[34] T. Wilke, An algebraic characterization of frontier testable tree languages, Theoret. Comput. Sci. 154 (1996) 85–106.