

On DR Tree Automata, Unary Algebras and Syntactic Path Monoids

Magnus Steinby^a

To the memory of Zoltán Ésik

Abstract

We consider deterministic root-to-frontier (DR) tree recognizers and the tree languages recognized by them from an algebraic point of view. We make use of a correspondence between DR algebras and unary algebras shown by Z. Ésik (1986). We also study a question raised by F. Gécseg (2007) that concerns the definability of families of DR-recognizable tree languages by syntactic path monoids. We show how the families of DR-recognizable tree languages path-definable by a variety of finite monoids (or semigroups) can be derived from varieties of string languages. In particular, the three path-definable families of Gécseg and B. Imreh (2002, 2004) are obtained this way.

Keywords: deterministic root-to-frontier tree automaton, tree language, unary algebra, syntactic path monoid, variety of finite monoids, variety of languages

1 Introduction

The tree languages recognized by deterministic root-to-frontier (top-down) tree recognizers form a proper subfamily $DRec$ of the family Rec of all recognizable tree languages. The members of $DRec$, the DR-recognizable tree languages, are characterized by the fact that they are completely determined by the labeled paths appearing in their trees (cf. [11, 15, 16, 20]). Any path from the root of a tree to one of its leaves is represented as a word over the so-called path alphabet. Each symbol of this alphabet indicates both the label of a node of a tree and the direction taken at that node. If we group together the paths leading to a leaf labeled with a given symbol x of the leaf alphabet X , then all the paths appearing in the trees of a given tree language T form a family $\langle T_x \rangle_{x \in X}$ of languages over the path alphabet, and if T is DR-recognizable, it is completely determined by these languages T_x . This implies that the DR-recognizable tree languages resemble string languages more than general recognizable tree languages do. In particular, while few known

^aDepartment of Mathematics and Statistics, University of Turku, FI-20014 Turku, Finland, E-mail: steinby@utu.fi

families of recognizable tree languages can be characterized by syntactic monoids or semigroups, Gécseg and Imreh [7, 8] could characterize three subfamilies of $DRec$, those of DR nilpotent, DR monotone and DR definite tree languages, in terms of the syntactic path monoids or semigroups introduced in [12]. We shall show that there exist many more such examples: any $*$ - or $+$ -variety of string languages, as defined by Eilenberg [2], yields a subfamily of $DRec$ that can be characterized by syntactic path monoids or semigroups.

If we regard the path alphabet as a unary ranked alphabet, then the path set of a tree language T becomes a unary tree language $\delta(T)$ that carries the same information as the family $\langle T_x \rangle_{x \in X}$. A DR recognizer may be seen as a finite DR algebra equipped with an initial state and a final state assignment. In [3] Zoltán Ésik associated with each DR algebra a unary algebra over the path alphabet, and noted that using this association one may apply ideas of standard general algebra to DR algebras. We complete the bijection between the two types of algebras by the converse transformation from unary algebras to DR algebras. The usefulness of this bijection derives from the fact that it preserves subalgebras, homomorphisms, congruences and direct products. In particular, we may refer to varieties of finite unary algebras when considering varieties of finite DR algebras. By extending this correspondence to tree recognizers, we study the connections between DR-recognizable tree languages and their unary path languages.

We shall recall or introduce all the special concepts used here. The basic universal algebra needed can be found in the first two chapters of [1], for example. For tree automata and tree languages, the reader may consult [11] and for the theory of varieties of (string) languages the books [2] and [17].

This paper is dedicated to the memory of Zoltán Ésik whom I learned know already in the 1970s. He has made many important contributions in several areas of theoretical computer science, and all his work is characterized by mathematical elegance and precision. It is a pleasure to acknowledge the inspiration I got from one of his, probably less well known, papers.

2 Preliminaries

For any integer $n > 0$, let $[n] = \{1, \dots, n\}$. Let A be any set. For any $i \in [n]$, let $\pi_i : A^n \rightarrow A, (a_1, \dots, a_n) \mapsto a_i$ be the i^{th} projection map. The power-set of A is denoted by $\wp(A)$. If $\varphi : A \rightarrow B$ is a mapping, the image $\varphi(a)$ of an element $a \in A$ may be denoted also by $a\varphi$. Especially homomorphisms will be written this way as right operators. For any equivalence $\theta \in \text{Eq}(A)$ on A , we write $a\theta$ for the θ -class of an element $a \in A$, and $A/\theta := \{a\theta \mid a \in A\}$ is the quotient set. For any alphabet X , the set of (finite) words over X is denoted by X^* and the empty word by ε .

Let Σ be a *ranked alphabet*, i.e., a finite set of operation symbols, which does not contain nullary symbols. For each $m \geq 1$, Σ_m denotes the set of m -ary symbols in Σ . The *rank type* of Σ is the set $r(\Sigma) := \{m \mid \Sigma_m \neq \emptyset\}$. In what follows, Σ is always a ranked alphabet of rank type R and X is an ordinary finite non-empty alphabet, called a *leaf alphabet*, disjoint from Σ . The set $T_\Sigma(X)$ of ΣX -trees is the

least set such that $X \subseteq T_\Sigma(X)$, and $f(t_1, \dots, t_m) \in T_\Sigma(X)$, for all $m \in R$, $f \in \Sigma_m$ and $t_1, \dots, t_m \in T_\Sigma(X)$. A ΣX -tree language is any subset of $T_\Sigma(X)$. Often we speak about *trees* and *tree languages* without specifying the alphabets.

A Σ -algebra \mathcal{A} consists of a nonempty set A and a Σ -indexed family of operations on A such that if $f \in \Sigma_m$ ($m \in R$), then $f^{\mathcal{A}} : A^m \rightarrow A$ is an m -ary operation. We write $\mathcal{A} = (A, \Sigma)$, and call \mathcal{A} *finite* if A is a finite set. Subalgebras, homomorphisms, congruences etc. are defined as usual. For any class \mathbf{K} of Σ -algebras, let $S(\mathbf{K})$ consist of all algebras isomorphic to a subalgebra of a member of \mathbf{K} , $H(\mathbf{K})$ be the class of all images of members of \mathbf{K} , and $P_f(\mathbf{K})$ be the class of algebras isomorphic to a finite direct product of members of \mathbf{K} . A class \mathbf{K} of finite Σ -algebras is a *variety of finite Σ -algebras* (a Σ -VFA for short) if $S(\mathbf{K}), H(\mathbf{K}), P_f(\mathbf{K}) \subseteq \mathbf{K}$. The Σ -VFA generated by a class \mathbf{K} of finite Σ -algebras is denoted by $V_f(\mathbf{K})$.

The ΣX -term algebra $\mathcal{T}_\Sigma(X) = (T_\Sigma(X), \Sigma)$ is defined by setting, for all $m \in R$, $f \in \Sigma_m$ and $t_1, \dots, t_m \in T_\Sigma(X)$, $f^{T_\Sigma(X)}(t_1, \dots, t_m) = f(t_1, \dots, t_m)$.

A ΣX -recognizer $\mathbf{A} = (\mathcal{A}, \alpha, F)$ consists of a finite Σ -algebra $\mathcal{A} = (A, \Sigma)$, an *initial assignment* $\alpha : X \rightarrow A$, and a set $F \subseteq A$ of *final states*. The ΣX -tree language *recognized* by \mathbf{A} is the set $T(\mathbf{A}) := \{t \in T_\Sigma(X) \mid t\alpha_{\mathcal{A}} \in F\}$ where $\alpha_{\mathcal{A}} : \mathcal{T}_\Sigma(X) \rightarrow \mathcal{A}$ is the homomorphic extension of α . A ΣX -tree language is called *recognizable*, or *regular*, if it is recognized by a ΣX -recognizer. Let $Rec(\Sigma, X)$ denote the set of all recognizable ΣX -tree languages.

If $\Sigma = \Sigma_1$, we call Σ a *unary alphabet* and Σ -algebras are *unary algebras*. A unary alphabet Σ may also be treated as an ordinary alphabet and we write Σ -terms as expressions ξu , where ξ is a variable and $u \in \Sigma^*$. The *term functions* induced by such terms ξu in a Σ -algebra $\mathcal{C} = (C, \Sigma)$ are mappings $u^{\mathcal{C}} : C \rightarrow C$, $c \mapsto cu^{\mathcal{C}}$, defined by $ce^{\mathcal{C}} = c$ and $cu^{\mathcal{C}} = g^{\mathcal{C}}(cv^{\mathcal{C}})$ for $u = vg$ ($v \in \Sigma^*$, $g \in \Sigma$).

On the other hand, we may view an ordinary (finite) alphabet Z as a unary ranked alphabet and define *Z-automata* as unary algebras $\mathcal{A} = (A, Z)$ in which each letter $z \in Z$ induces a unary operation $z^{\mathcal{A}} : A \rightarrow A$, $a \mapsto az^{\mathcal{A}}$. For any word $w = z_1 \dots z_k$ in Z^* ($k \geq 0$, $z_1, \dots, z_k \in Z$), the operation $w^{\mathcal{A}} : A \rightarrow A$ is the composition of $z_1^{\mathcal{A}}, \dots, z_k^{\mathcal{A}}$. A *Z-recognizer* is now a system $\mathbf{A} = (\mathcal{A}, a_0, F)$, where $\mathcal{A} = (A, Z)$ is a finite Z -algebra, $a_0 \in A$ is the initial state, and $F \subseteq A$ is the set of final states. The *language recognized* by \mathbf{A} is the set $L(\mathbf{A}) := \{w \in Z^* \mid a_0 w^{\mathcal{A}} \in F\}$. A language $L \subseteq Z^*$ is *regular* if it is recognized by a Z -recognizer.

Often we will say that a string or tree language is *recognized by an algebra \mathcal{A}* if it is recognized by a recognizer of an appropriate kind based on \mathcal{A} .

3 DR algebras and unary algebras

In what follows, the frequently recurring phrase *deterministic root-to-frontier* is abbreviated to DR. The following basic algebraic notions for DR algebras were defined by Virágh [20], but most of them are obtained from those defined for DR tree recognizers in [10]. To simplify the notation, we extend mappings and equivalence relations to m -tuples: if $\mathbf{a} = (a_1, \dots, a_m) \in A^m$, then for any map $\varphi : A \rightarrow B$, let $\mathbf{a}\varphi := (a_1\varphi, \dots, a_m\varphi)$, and for any $\theta \in \text{Eq}(A)$, let $\mathbf{a}\bar{\theta} := (a_1\theta, \dots, a_m\theta)$.

A (finite) DR Σ -algebra consists of a nonempty (finite) set A and a Σ -indexed family of root-to-frontier operations $f_A : A \rightarrow A^m$ ($f \in \Sigma$), where the arity m is that of f ($\in \Sigma_m$). Again we write simply $\mathcal{A} = (A, \Sigma)$.

Let $\mathcal{A} = (A, \Sigma)$ and $\mathcal{B} = (B, \Sigma)$ be any DR Σ -algebras. Then \mathcal{A} is a subalgebra of \mathcal{B} if $A \subseteq B$ and $f_A(a) = f_B(a)$ for all $f \in \Sigma$ and $a \in A$. A mapping $\varphi : A \rightarrow B$ is a homomorphism from \mathcal{A} to \mathcal{B} , and we write $\varphi : \mathcal{A} \rightarrow \mathcal{B}$, if $f_A(a)\bar{\varphi} = f_B(a\varphi)$ for all $f \in \Sigma$ and $a \in A$. If φ is also bijective, it is an isomorphism, and we write $\mathcal{A} \cong \mathcal{B}$ if \mathcal{A} and \mathcal{B} are isomorphic. The direct product of \mathcal{A} and \mathcal{B} is the DR Σ -algebra $\mathcal{A} \times \mathcal{B} = (A \times B, \Sigma)$ such that for all $m \in R$, $f \in \Sigma_m$ and $(a, b) \in A \times B$, if $f_A(a) = (a_1, \dots, a_m)$ and $f_B(b) = (b_1, \dots, b_m)$, then $f_{\mathcal{A} \times \mathcal{B}}((a, b)) = ((a_1, b_1), \dots, (a_m, b_m))$. The general finite direct product $\mathcal{A}_1 \times \dots \times \mathcal{A}_n$ ($n \geq 0$) is defined the same way.

A congruence on \mathcal{A} is an equivalence θ on A such that for any $a, a' \in A$ and $f \in \Sigma$, if $a\theta a'$, then $f_A(a)\bar{\theta} = f_A(a')\bar{\theta}$. Let $\text{Con}(\mathcal{A})$ denote the set of all congruences on \mathcal{A} . For any $\theta \in \text{Con}(\mathcal{A})$, the quotient DR algebra $\mathcal{A}/\theta = (A/\theta, \Sigma)$ is defined by $f_{\mathcal{A}/\theta}(a\theta) := f_A(a)\bar{\theta}$ for all $a \in A$ and $f \in \Sigma$.

All the usual facts about subalgebras, homomorphisms, congruences, etc. hold for DR algebras, too. For example, the kernel of any homomorphism $\varphi : \mathcal{A} \rightarrow \mathcal{B}$ is a congruence on \mathcal{A} , and $\mathcal{A}/\ker \varphi \cong \mathcal{B}$ if φ is surjective.

The tree languages recognized by deterministic root-to-frontier recognizers are characterized by the labeled paths appearing in their trees. The paths are described using the path alphabet $\widehat{\Sigma} := \bigcup_{m \in R} \Sigma_m \times [m]$. A pair $(f, i) \in \widehat{\Sigma}$ is written simply as f_i . We regard $\widehat{\Sigma}$ either as a unary ranked alphabet or as an ordinary alphabet.

Following Ésik [3] we associate with any DR Σ -algebra $\mathcal{A} = (A, \Sigma)$ a unary algebra $\mathcal{A}^u = (A, \widehat{\Sigma})$ such that $f_i^{\mathcal{A}^u}(a) = f_A(a)\pi_i$ for all $a \in A$, $m \in R$, $f \in \Sigma_m$ and $i \in [m]$. Let us also introduce a converse transformation: for any $\widehat{\Sigma}$ -algebra $\mathcal{C} = (C, \widehat{\Sigma})$, let $\mathcal{C}^d = (C, \Sigma)$ be the DR Σ -algebra with $f_{\mathcal{C}^d}(c) = (f_1^c(c), \dots, f_m^c(c))$ for all $c \in C$, $m \in R$ and $f \in \Sigma_m$. Since $\mathcal{A}^{ud} = \mathcal{A}$ for any DR Σ -algebra \mathcal{A} and $\mathcal{C}^{du} = \mathcal{C}$ for any $\widehat{\Sigma}$ -algebra \mathcal{C} , there is a natural bijective correspondence between DR Σ -algebras and $\widehat{\Sigma}$ -algebras.

Lemma 3.1. *Let $\mathcal{A} = (A, \Sigma)$ and $\mathcal{B} = (B, \Sigma)$ be any DR Σ -algebras.*

- (a) \mathcal{A} is a subalgebra of \mathcal{B} if and only if \mathcal{A}^u is a subalgebra of \mathcal{B}^u .
- (b) A mapping $\varphi : A \rightarrow B$ is a homomorphism from \mathcal{A} to \mathcal{B} if and only if it is a homomorphism from \mathcal{A}^u to \mathcal{B}^u .
- (c) $(\mathcal{A} \times \mathcal{B})^u = \mathcal{A}^u \times \mathcal{B}^u$.
- (d) $\text{Con}(\mathcal{A}) = \text{Con}(\mathcal{A}^u)$.
- (e) $(\mathcal{A}/\theta)^u = \mathcal{A}^u/\theta$ for any $\theta \in \text{Con}(\mathcal{A})$.

Proof. All five statements follow directly from the appropriate definitions, and (c) was noted already in [3]. Let us verify (e) as an example.

Firstly, $(\mathcal{A}/\theta)^u$ and \mathcal{A}^u/θ are $\widehat{\Sigma}$ -algebras with the same set A/θ of elements. Moreover, for any $a \in A$, $m \in R$, $f \in \Sigma_m$ and $i \in [m]$,

$$f_i^{(\mathcal{A}/\theta)^u}(a\theta) = f_{\mathcal{A}/\theta}(a\theta)\pi_i = f_A(a)\bar{\theta}\pi_i = f_A(a)\pi_i\bar{\theta} = f_i^{\mathcal{A}^u}(a)\bar{\theta} = f_i^{\mathcal{A}^u/\theta}(a\theta),$$

so also their operations are the same. \square

4 DR tree recognizers and unary recognizers

Let us now extend the correspondence between DR algebras and unary algebras to recognizers. A *DR ΣX -recognizer* $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ consists of a finite DR Σ -algebra $\mathcal{A} = (A, \Sigma)$, an *initial state* $a_0 \in A$, and a *final state assignment* $\alpha : X \rightarrow \wp(A)$. To accept or reject an input tree $t \in T_\Sigma(X)$, \mathbf{A} starts at the root of t in state a_0 , and if it has reached a node ν of t labeled with $f \in \Sigma_m$ in state a and $f_{\mathcal{A}}(a) = (a_1, \dots, a_m)$, then it continues its working at the i^{th} immediate successor node of ν in state a_i ($i \in [m]$). The tree is accepted if \mathbf{A} reaches each leaf in a state $a (\in A)$ matching the label $x (\in X)$ of that leaf, i.e., $a \in \alpha(x)$. For a formal definition, we extend α to a mapping $\tilde{\alpha} : T_\Sigma(X) \rightarrow \wp(A)$ by setting $\tilde{\alpha}(x) = \alpha(x)$ for each $x \in X$, and $\tilde{\alpha}(t) = \{a \in A \mid f_{\mathcal{A}}(a) \in \tilde{\alpha}(t_1) \times \dots \times \tilde{\alpha}(t_m)\}$ for $t = f(t_1, \dots, t_m)$. Then $T(\mathbf{A}) := \{t \in T_\Sigma(X) \mid a_0 \in \tilde{\alpha}(t)\}$ is the tree language *recognized* by \mathbf{A} , and the ΣX -tree language $T(\mathbf{A})$ is said to be *DR-recognizable*. Let $DRec(\Sigma, X)$ denote the set of DR-recognizable ΣX -tree languages. Two DR ΣX -recognizers \mathbf{A} and \mathbf{B} are *equivalent* if $T(\mathbf{A}) = T(\mathbf{B})$.

The set $\delta(t) \subseteq T_\Sigma(X)$ of *paths* in a ΣX -tree t is defined by $\delta(x) = \{x\}$ for $x \in X$, and $\delta(t) = f_1\delta(t_1) \cup \dots \cup f_m\delta(t_m)$ for $t = f(t_1, \dots, t_m)$. Thus $\delta(t)$ is a set of unary trees in Polish form. The *path language* of a ΣX -tree language T is the set $\delta(T) := \bigcup\{\delta(t) \mid t \in T\}$. The *path closure* $\Delta(T) := \delta^{-1}(\delta(T))$ of $T \subseteq T_\Sigma(X)$ consists of all ΣX -trees t such that $\delta(t) \subseteq \delta(T)$, and T is *path closed* if $T = \Delta(T)$. Quite generally, for any $U \subseteq T_\Sigma(X)$, the set $\delta^{-1}(U) := \{t \in T_\Sigma(X) \mid \delta(t) \subseteq U\}$ is path-closed. As shown in [16], a regular tree language is DR-recognizable if and only if it is path closed. For properties of the operators δ and Δ , cf. [15, 20].

Remark 4.1. Let Σ be unary. Then $\widehat{\Sigma} = \{f_1 \mid f \in \Sigma\}$ and we may use Σ itself as the path alphabet. Furthermore, we may regard any DR Σ -algebra $\mathcal{A} = (A, \Sigma)$ also as a Σ -algebra by identifying any 1-tuple (a) with the element $a (\in A)$, but as a DR ΣX -recognizer and as a ΣX -recognizer \mathcal{A} reads the input trees in opposite directions. Nevertheless, it is clear that $DRec(\Sigma, X) = Rec(\Sigma, X)$ for every X .

Let us now regard $\widehat{\Sigma}$ as a usual alphabet. For each $x \in X$, the set of x -paths in a ΣX -tree t is $g_x(t) := \{u \in \widehat{\Sigma}^* \mid ux \in \delta(t)\}$. For any $T \subseteq T_\Sigma(X)$ and $x \in X$, let T_x denote the set $\bigcup\{g_x(t) \mid t \in T\}$ of x -paths appearing in T . Obviously, $\delta(T)$ can be recovered from the family $\langle T_x \rangle_{x \in X}$.

Next we recall a few notions from [10, 11]. Let $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ be a DR ΣX -recognizer and $\mathcal{A} = (A, \Sigma)$. For any $a \in A$, let $\mathbf{A}_a := (\mathcal{A}, a, \alpha)$. A state a is a *0-state* if $T(\mathbf{A}_a) = \emptyset$, and it is *reachable* if $a_0 \Rightarrow_{\mathcal{A}}^* a$ for the reflexive transitive closure $\Rightarrow_{\mathcal{A}}^*$ of the relation $\Rightarrow_{\mathcal{A}} \subseteq A \times A$, where for any $a, b \in A$, $a \Rightarrow_{\mathcal{A}} b$ if and only if $b = f_{\mathcal{A}}(a)\pi_i$ for some $m \in R$, $f \in \Sigma_m$ and $i \in [m]$. The recognizer \mathbf{A} is *normalized*, if for all $m \in R$, $f \in \Sigma_m$ and $a \in A$, either every component in $f_{\mathcal{A}}(a) = (a_1, \dots, a_m)$ is a 0-state or no a_i is a 0-state, *reduced* if $T(\mathbf{A}_a) = T(\mathbf{A}_b)$ implies $a = b$ ($a, b \in A$), *connected* if all of its states are reachable, and it is *minimal*

if it is connected and reduced. In [10] it was shown that any DR ΣX -recognizer can be converted into an equivalent normalized minimal DR ΣX -recognizer, and this is also minimal with respect to the number of states and unique up to isomorphism (the isomorphism of DR ΣX -recognizers is defined in the natural way, cf. [10]).

Let us associate with any DR ΣX -recognizer $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ the DR $\widehat{\Sigma}X$ -recognizer $\mathbf{A}^u = (\mathcal{A}^u, a_0, \alpha)$, and with any DR $\widehat{\Sigma}X$ -recognizer $\mathbf{C} = (\mathcal{C}, c_0, \gamma)$ the DR ΣX -recognizer $\mathbf{C}^d = (\mathcal{C}^d, c_0, \gamma)$. Obviously, $\mathbf{A}^{ud} = \mathbf{A}$ and $\mathbf{C}^{du} = \mathbf{C}$.

Proposition 4.1. $T(\mathbf{A}) = \delta^{-1}(T(\mathbf{A}^u))$ for any DR ΣX -tree recognizer \mathbf{A} .

Proof. Let $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ with $\mathcal{A} = (A, \Sigma)$. We show by induction on t that for all $t \in T_{\widehat{\Sigma}}(X)$ and $a \in A$, $t \in T(\mathbf{A}_a)$ if and only if $t \in \delta^{-1}(T(\mathbf{A}_a^u))$. The case $t \in X$ is obvious, so let $t = f(t_1, \dots, t_m)$. If $f_{\mathcal{A}}(a) = (a_1, \dots, a_m)$, then

$$\begin{aligned} t \in T(\mathbf{A}_a) & \text{ iff } t_1 \in T(\mathbf{A}_{a_1}), \dots, t_m \in T(\mathbf{A}_{a_m}) \\ & \text{ iff } \delta(t_1) \subseteq T(\mathbf{A}_{a_1}^u), \dots, \delta(t_m) \subseteq T(\mathbf{A}_{a_m}^u) \\ & \text{ iff } f_1\delta(t_1), \dots, f_m\delta(t_m) \subseteq T(\mathbf{A}_a^u) \\ & \text{ iff } \delta(t) \subseteq T(\mathbf{A}_a^u) \\ & \text{ iff } t \in \delta^{-1}(T(\mathbf{A}_a^u)). \end{aligned}$$

Hence, $T(\mathbf{A}_a) = \delta^{-1}(T(\mathbf{A}_a^u))$ and, in particular, $T(\mathbf{A}) = \delta^{-1}(T(\mathbf{A}^u))$. \square

Corollary 4.1. $T(\mathbf{C}^d) = \delta^{-1}(T(\mathbf{C}))$ for any DR $\widehat{\Sigma}X$ -recognizer \mathbf{C} .

Proof. $T(\mathbf{C}^d) = \delta^{-1}(T(\mathbf{C}^{du})) = \delta^{-1}(T(\mathbf{C}))$ by Proposition 4.1 and $\mathbf{C}^{du} = \mathbf{C}$. \square

Proposition 4.2. $T(\mathbf{A}^u) = \delta(T(\mathbf{A}))$ for any normalized DR ΣX -recognizer \mathbf{A} .

Proof. Let $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ with $\mathcal{A} = (A, \Sigma)$. The inclusion $\delta(T(\mathbf{A})) \subseteq T(\mathbf{A}^u)$ follows from Proposition 4.1 and the fact that $\delta(\delta^{-1}(U)) \subseteq U$ for any $U \subseteq T_{\widehat{\Sigma}}(X)$.

For the converse inclusion we need the assumption that \mathbf{A} is normalized. It is enough to show that for all $a \in A$ and $r \in T_{\widehat{\Sigma}}(X)$, if $r \in T(\mathbf{A}_a^u)$, then $r \in \delta(t)$ for some $t \in T(\mathbf{A}_a)$. This we do by induction on r . For $r \in X$, we may let $t := r$. Next, let $r = f_i s$ for some $f \in \Sigma_m$, $m \in R$, $i \in [m]$ and $s \in T_{\widehat{\Sigma}}(X)$, and assume that the claim holds for s . If $f_{\mathcal{A}}(a) = (a_1, \dots, a_m)$, then $s \in T(\mathbf{A}_{a_i}^u)$ implies that $s \in \delta(t_i)$ for some $t_i \in T(\mathbf{A}_{a_i})$. Since a_i is not a 0-state, there is for every $j \in [m]$, $j \neq i$ a tree $t_j \in T(\mathbf{A}_{a_j})$. Clearly, $t := f(t_1, \dots, t_m) \in T(\mathbf{A}_a)$ and $r \in \delta(t)$. \square

The following fact appears, in a different form, already in [16].

Corollary 4.2. If $T \in D\text{Rec}(\Sigma, X)$, then $\delta(T) \in D\text{Rec}(\widehat{\Sigma}, X)$.

Proposition 4.3. A normalized DR ΣX -recognizer \mathbf{A} is minimal if and only if \mathbf{A}^u is a minimal DR $\widehat{\Sigma}X$ -recognizer of $\delta(T(\mathbf{A}))$.

Proof. Let $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ with $\mathcal{A} = (A, \Sigma)$. Consider any two states $a, b \in A$. If $T(\mathbf{A}_a) = T(\mathbf{A}_b)$, then $T(\mathbf{A}_a^u) = \delta(T(\mathbf{A}_a)) = \delta(T(\mathbf{A}_b)) = T(\mathbf{A}_b^u)$ by Proposition 4.2. On the other hand, if $T(\mathbf{A}_a^u) = T(\mathbf{A}_b^u)$, then Proposition 4.2 yields $\delta(T(\mathbf{A}_a)) =$

$\delta(T(\mathbf{A}_b))$. Since $T(\mathbf{A}_a)$ and $T(\mathbf{A}_b)$ are path-closed, this means that $T(\mathbf{A}_a) = T(\mathbf{A}_b)$. Hence, \mathbf{A} is reduced if and only if \mathbf{A}^u is reduced.

It is obvious that the reachability relations $\Rightarrow_{\mathcal{A}}^*$ and $\Rightarrow_{\mathcal{A}^u}^*$ of \mathbf{A} and \mathbf{A}^u are identical. Hence, \mathbf{A} is connected if and only if \mathbf{A}^u is connected. \square

Next we show how a DR Σ -algebra recognizing a ΣX -tree language T yields a DR $\widehat{\Sigma}$ -algebra recognizing the $\widehat{\Sigma}$ -languages T_x ($x \in X$), and how a DR Σ -algebra recognizing T is obtained from DR $\widehat{\Sigma}$ -algebras recognizing the $\widehat{\Sigma}$ -languages T_x .

Proposition 4.4. *Let T be a DR-recognizable ΣX -tree language.*

- (a) *If a finite DR Σ -algebra $\mathcal{A} = (A, \Sigma)$ recognizes T , then $\mathcal{A}^u = (A, \widehat{\Sigma})$ recognizes every language T_x ($x \in X$).*
- (b) *For each $x \in X$, let $\mathcal{A}_x = (A_x, \widehat{\Sigma})$ be a finite $\widehat{\Sigma}$ -algebra that recognizes T_x . Then the direct product $\prod(\mathcal{A}_x^d \mid x \in X)$ recognizes T .*

Proof. Let $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ be a DR ΣX -recognizer of T . It is easy to see that, for each $x \in X$, the $\widehat{\Sigma}$ -recognizer $\mathbf{A}_x = (\mathcal{A}^u, a_0, \alpha(x))$ recognizes T_x .

To prove (b), consider for each $x \in X$ a $\widehat{\Sigma}$ -recognizer $\mathbf{A}_x = (\mathcal{A}_x, a_{x0}, F_x)$ of T_x . The direct product $\mathcal{A} := \prod_{x \in X} \mathcal{A}_x^d$ simulates the computation of \mathcal{A}_x by its x -component along every path of a given tree $t \in T_\Sigma(X)$. Hence, started in state $(a_{x0})_{x \in X}$, \mathcal{A} should accept t if and only if it reaches, for each $y \in X$, every y -labeled leaf in a state $(a_x)_{x \in X}$ such that $a_y \in F_y$. This means that $T = T(\mathbf{A})$ for $\mathbf{A} = (\mathcal{A}, (a_{x0})_{x \in X}, \alpha)$ if we define α by $\alpha(y) = \prod_{x \in X} G_y(x)$, where $G_y(y) = F_y$ and $G_y(x) = A_x$ for all $x \in X, x \neq y$. \square

5 Definability by syntactic monoids

Let us first recall (cf. [2, 17]) that the *syntactic congruence* of a string language $L \subseteq Z^*$ is the relation θ_L on Z^* defined by

$$u \theta_L v \quad \text{iff} \quad (\forall w, w' \in Z^*)(uw w' \in L \leftrightarrow v w w' \in L),$$

and that the *syntactic monoid* of L is the quotient monoid $M(L) := Z^*/\theta_L$.

Next we define the syntactic monoids and syntactic path monoids of tree languages introduced in [19] and [12], respectively.

Let ξ be a symbol that does not appear in our alphabets Σ or X . A ΣX -context is a $\Sigma(X \cup \{\xi\})$ -tree in which ξ occurs exactly once. The set of all ΣX -contexts is denoted by $C_\Sigma(X)$. If $p, q \in C_\Sigma(X)$ and $t \in T_\Sigma(X)$, then $p \cdot q = q(p)$ and $t \cdot q = q(t)$ are the ΣX -context and the ΣX -tree obtained by replacing the ξ in q by p or t , respectively. Then $C_\Sigma(X)$ forms for the product $p \cdot q$ a monoid in which ξ is the identity element. If Σ is unary, no X -symbols appear in ΣX -contexts, and hence we write C_Σ for $C_\Sigma(X)$.

The *syntactic monoid congruence* μ_T of a ΣX -tree language T is the relation on $C_\Sigma(X)$ is defined by

$$p \mu_T q \quad \text{iff} \quad (\forall t \in T_\Sigma(X))(\forall r \in C_\Sigma(X))(t \cdot p \cdot r \in T \leftrightarrow t \cdot q \cdot r \in T),$$

and the *syntactic monoid* of T is the quotient monoid $\text{SM}(T) := C_{\widehat{\Sigma}}(X)/\mu_T$. The *syntactic path congruence* $\widehat{\mu}_T$ is the relation on $C_{\widehat{\Sigma}}$ defined by

$$p \widehat{\mu}_T q \quad \text{iff} \quad (\forall s \in T_{\widehat{\Sigma}}(X))(r \in C_{\widehat{\Sigma}})(s \cdot p \cdot r \in \delta(T) \leftrightarrow s \cdot q \cdot r \in \delta(T)),$$

and the *syntactic path monoid* of T is the quotient monoid $\text{PM}(T) := C_{\widehat{\Sigma}}/\widehat{\mu}_T$.

In [19] it was shown that T is regular if and only if $\text{SM}(T)$ is finite, and in [12] that a path closed T is DR-recognizable if and only if $\text{PM}(T)$ is finite.

In [12] $\text{PM}(T)$ was defined as the quotient $\widehat{\Sigma}^*/\theta_T$ where θ_T is the intersection of the congruences θ_{T_x} ($x \in X$). It is easy to see that $C_{\widehat{\Sigma}}/\widehat{\mu}_T \cong \widehat{\Sigma}^*/\theta_T$, and hence the next lemma follows from the fact that $\theta_T = \bigcap \{\theta_{T_x} \mid x \in X\}$. To see this, combine Theorem II.6.2 and Lemma II.8.2 of [1]. In [5] the corresponding fact about transition monoids was used.

Lemma 5.1. *For any $T \in \text{DRec}(\Sigma, X)$, $\text{PM}(T)$ is a subdirect product of the monoids $\text{M}(T_x)$ ($x \in X$).*

If Σ is unary and we use Σ itself as the path alphabet, then $T_{\widehat{\Sigma}}(X) = T_{\Sigma}(X)$ and $C_{\widehat{\Sigma}} = C_{\Sigma}$. Moreover, $\delta(U) = U$ for any $U \subseteq T_{\Sigma}(X)$, and $\widehat{\mu}_U$ and μ_U become identical. Hence, $\text{PM}(U) \cong \text{SM}(U)$ for any unary tree language U . Similarly, for any ranked alphabet Σ , we may use $\widehat{\Sigma}$ as its own path alphabet, and then $\delta(\delta(T)) = \delta(T)$ for any $T \subseteq T_{\widehat{\Sigma}}(X)$, which implies $\widehat{\mu}_T = \widehat{\mu}_{\delta(T)}$. By combining these observations, we obtain the following result.

Proposition 5.1. *$\text{PM}(T) \cong \text{PM}(\delta(T)) \cong \text{SM}(\delta(T))$ for any ΣX -tree language T .*

In [5] Gécseg poses the following question. Assume that some property \mathcal{P} of regular string languages is determined by a class \mathbf{M} of finite monoids in the sense that a language has property \mathcal{P} if and only if its syntactic monoid is in \mathbf{M} . Under what conditions can we conclude that the ‘corresponding’ property of regular tree languages is similarly determined by \mathbf{M} ? In [6] the question is also considered for DR tree languages in terms of syntactic path monoids. Let us describe the result of [6] concerning the DR-case.

A class \mathbf{M} of finite monoids is said to be *closed under subdirect products* if any subdirect product of a finite family of members of \mathbf{M} also belongs to \mathbf{M} , and it is *closed under subdirect factors* if whenever a subdirect product of a finite family of monoids belongs to \mathbf{M} , then all the factors are in \mathbf{M} , too¹. A property \mathcal{P} of DR tree languages is *path-defined* by \mathbf{M} if a DR-recognizable tree language T has property \mathcal{P} if and only if $\text{PM}(T) \in \mathbf{M}$. Somewhat reformulated, Theorem 11 of [6] reads as follows.

Proposition 5.2. *(F. Gécseg 2011) Let \mathcal{P} be a property of tree languages that is also defined for string languages, and let \mathbf{M} be a class of finite monoids. Assume that the following three conditions are satisfied.*

- (1) *A DR-recognizable ΣX -tree language T has property \mathcal{P} if and only if T_x has property \mathcal{P} for every $x \in X$.*

¹In [5, 6] this is required of subdirect products of two factors only, but the stronger form is actually used.

(2) For string languages the property \mathcal{P} is defined by \mathbf{M} .

(3) \mathbf{M} is closed under subdirect products and subdirect factors.

Then \mathcal{P} is path-defined for DR-recognizable tree languages by \mathbf{M} .

Condition (3) is explained by Lemma 5.1. If \mathbf{M} is a *variety of finite monoids* (VFM), i.e., if $S(\mathbf{M}), H(\mathbf{M}), P_f(\mathbf{M}) \subseteq \mathbf{M}$, condition (3) is always satisfied. Hence, it is redundant when we consider properties that define varieties of string languages (cf. [2, 17]), and this concerns many of the best-known families of regular languages. In what follows, we replace “properties” by families of tree or string languages.

A *family of tree languages (FTL)* \mathcal{V} assigns to all pairs Σ, X a set $\mathcal{V}(\Sigma, X)$ of ΣX -tree languages. We write $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ with the understanding that Σ and X range over the appropriate alphabets. If Σ is allowed to range over the unary ranked alphabets only, then \mathcal{V} is a *family of unary tree languages (FUTL)*. From any FTL $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ we get a FUTL $\mathcal{V}^u = \{\mathcal{V}^u(\Sigma, X)\}$ by restricting the range of Σ to unary alphabets. We call $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ a *DR family of tree languages (DR-FTL)* if $\mathcal{V}(\Sigma, X) \subseteq DRec(\Sigma, X)$ for all Σ and X . Similarly, a FUTL $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ is a *DR-FUTL* if $\mathcal{V}(\Sigma, X) \subseteq DRec(\Sigma, X)$ for every unary Σ and every X .

Let \mathbf{M} be a class of finite monoids. For any Σ and X , let

$$\mathbf{M}^p(\Sigma, X) := \{T \in DRec(\Sigma, X) \mid \text{PM}(T) \in \mathbf{M}\}.$$

Then $\mathbf{M}^p = \{\mathbf{M}^p(\Sigma, X)\}$ is the *DR-FTL path-defined* by \mathbf{M} , and $\mathbf{M}^u := (\mathbf{M}^p)^u$ is the *DR-FUTL path-defined* by \mathbf{M} .

Note that owing to Proposition 5.1, the third condition could be dropped in the following variant of Proposition 5.2.

Proposition 5.3. *Let $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ be a DR-FTL, and let \mathbf{M} be a class of finite monoids. If*

(1) $\mathcal{V}^u = \mathbf{M}^u$, and

(2) $T \in \mathcal{V}(\Sigma, X)$ if and only if $\delta(T) \in \mathcal{V}^u(\widehat{\Sigma}, X)$ for all Σ, X and $T \subseteq T_\Sigma(X)$,

then $\mathcal{V} = \mathbf{M}^p$.

Proof. Consider any Σ and X . For every $T \in DRec(\Sigma, X)$,

$$\begin{aligned} T \in \mathbf{M}^p(\Sigma, X) &\text{ iff } \text{PM}(T) \in \mathbf{M} &\text{ iff } \text{PM}(\delta(T)) \in \mathbf{M} \\ &\text{ iff } \delta(T) \in \mathbf{M}^u(\widehat{\Sigma}, X) &\text{ iff } \delta(T) \in \mathcal{V}^u(\widehat{\Sigma}, X) \\ &\text{ iff } T \in \mathcal{V}(\Sigma, X), \end{aligned}$$

where we used the definition of \mathbf{M}^p , Proposition 5.1, the definition of \mathbf{M}^u , assumption (1), and finally assumption (2). Hence $\mathcal{V} = \mathbf{M}^p$. \square

Let us now show a way to get all DR-FTLs path-defined by a VFM. For this, recall that Eilenberg [2] defines a **-variety* as a family of languages with certain closure properties and shows that every *-variety $\mathcal{L} = \{\mathcal{L}(Z)\}$ is defined by a unique

VFM \mathbf{M} in the sense that for any $L \subseteq Z^*$, $L \in \mathcal{L}(Z)$ if and only if $M(L) \in \mathbf{M}$. For any $*$ -variety \mathcal{L} , let $\mathcal{V}_{\mathcal{L}} = \{\mathcal{V}_{\mathcal{L}}(\Sigma, X)\}$ be the FTL where

$$\mathcal{V}_{\mathcal{L}}(\Sigma, X) = \{T \in DRec(\Sigma, X) \mid (\forall x \in X) T_x \in \mathcal{L}(\widehat{\Sigma})\},$$

for all Σ and X .

Proposition 5.4. *If $\mathcal{L} = \{\mathcal{L}(Z)\}$ is a $*$ -variety and \mathbf{M} is the corresponding VFM, then $\mathcal{V}_{\mathcal{L}} = \{\mathcal{V}_{\mathcal{L}}(\Sigma, X)\}$ is a DR-FTL path-defined by \mathbf{M} .*

Proof. Let $T \in DRec(\Sigma, X)$. Since $PM(T)$ is a subdirect product of the monoids $M(T_x)$ ($x \in X$) and \mathbf{M} is a VFM, we conclude that $PM(T) \in \mathbf{M}$ if and only if $M(T_x) \in \mathbf{M}$ for every $x \in X$. Because $T \in \mathcal{V}_{\mathcal{L}}(\Sigma, X)$ if and only if $T_x \in \mathcal{L}(\widehat{\Sigma})$ for every $x \in X$, and $T_x \in \mathcal{L}(\widehat{\Sigma})$ if and only if $M(T_x) \in \mathbf{M}$ ($x \in X$), this implies that $T \in \mathcal{V}_{\mathcal{L}}(\Sigma, X)$ if and only if $PM(T) \in \mathbf{M}$. Hence, $\mathcal{V}_{\mathcal{L}} = \mathbf{M}^p$. \square

Next we show that every DR-FTL path-defined by a VFM is obtained this way.

Proposition 5.5. *If a DR-FTL $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ is path-defined by a VFM \mathbf{M} and $\mathcal{L} = \{\mathcal{L}(Z)\}$ is the $*$ -variety defined by \mathbf{M} , then $\mathcal{V} = \mathcal{V}_{\mathcal{L}}$.*

Proof. Consider any Σ and X . For every $T \in DRec(\Sigma, X)$,

$$\begin{aligned} T \in \mathcal{V}(\Sigma, X) &\text{ iff } PM(T) \in \mathbf{M} \text{ iff } PM(T_x) \in \mathbf{M} \text{ for every } x \in X \\ &\text{ iff } T_x \in \mathcal{L}(\widehat{\Sigma}) \text{ for every } x \in X \text{ iff } T \in \mathcal{V}_{\mathcal{L}}(\Sigma, X), \end{aligned}$$

where we used the assumptions of the proposition and Lemma 5.1. \square

Many families of regular languages are characterized by syntactic semigroups rather than by syntactic monoids, and in [6] Gécseg gives the corresponding versions of his results. Let us modify our Propositions 5.3, 5.4 and 5.5 in the same manner.

Firstly, all languages to be considered are ε -free, and $*$ -varieties are replaced by $+$ -varieties and VFMs by varieties of finite semigroups (VFSs). Also, instead of the syntactic monoid $M(L)$ of a language $L \subseteq Z^+ (:= Z^* \setminus \{\varepsilon\})$ we use its syntactic semigroup $S(T)$. For these notions, cf. [2] or [17]. Furthermore, the syntactic path monoid $PM(T)$ of a ΣX -tree language is replaced by the syntactic path semigroup $PS(T) := C_{\Sigma}^+ / \widehat{\sigma}_T$, where $C_{\Sigma}^+ := C_{\Sigma} \setminus \{\xi\}$ and $\widehat{\sigma}_T$ is $\widehat{\mu}_T$ restricted to C_{Σ}^+ .

Let $T_{\Sigma}^+(X) := T_{\Sigma}(X) \setminus X$, and let us call a ΣX -tree language T ε -free if $T \subseteq T_{\Sigma}^+(X)$. Obviously T is ε -free if and only if every T_x ($x \in X$) is ε -free. Furthermore, we call a DR-FTL or a DR-FUTL $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ ε -free if every tree language in \mathcal{V} is ε -free. The ε -free DR-FTL $\mathbf{S}^p = \{\mathbf{S}^p(\Sigma, X)\}$ path-defined by a class of finite semigroups \mathbf{S} is defined by

$$\mathbf{S}^p(\Sigma, X) := \{T \in DRec(\Sigma, X) \mid T \subseteq T_{\Sigma}^+(X), PS(T) \in \mathbf{S}\},$$

and the ε -free DR-FUTL path-defined by \mathbf{S} is $\mathbf{S}^u := (\mathbf{S}^p)^u$. Finally, for any $+$ -variety $\mathcal{L} = \{\mathcal{L}(Z)\}$, let $\mathcal{V}_{\mathcal{L}} = \{\mathcal{V}_{\mathcal{L}}(\Sigma, X)\}$ be the ε -free DR-FTL defined by $\mathcal{V}_{\mathcal{L}}(\Sigma, X) := \{T \in DRec(\Sigma, X) \mid T \subseteq T_{\Sigma}^+(X), (\forall x \in X) T_x \in \mathcal{L}(\widehat{\Sigma})\}$.

We may now state the following variants of Propositions 5.3, 5.4 and 5.5.

Proposition 5.6. *Let $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ be an ε -free DR-FTL, and let \mathbf{S} be a class of finite semigroups. If (1) $\mathcal{V}^u = \mathbf{S}^u$, and (2) $T \in \mathcal{V}(\Sigma, X)$ if and only if $\delta(T) \in \mathcal{V}(\widehat{\Sigma}, X)$ for all Σ, X and $T \subseteq T_{\Sigma}^+(X)$, then $\mathcal{V} = \mathbf{S}^p$.*

Proposition 5.7. *If $\mathcal{L} = \{\mathcal{L}(Z)\}$ is a $+$ -variety and \mathbf{S} is the corresponding VFS, then $\mathcal{V}_{\mathcal{L}} = \{\mathcal{V}_{\mathcal{L}}(\Sigma, X)\}$ is an ε -free DR-FTL path-defined by \mathbf{S} .*

Proposition 5.8. *If an ε -free DR-FTL $\mathcal{V} = \{\mathcal{V}(\Sigma, X)\}$ is path-defined by a VFS \mathbf{S} and $\mathcal{L} = \{\mathcal{L}(Z)\}$ is the $+$ -variety defined by \mathbf{S} , then $\mathcal{V} = \mathcal{V}_{\mathcal{L}}$.*

6 Some examples

Let us now apply the above results to the families of DR-recognizable tree languages considered by Gécseg and Imreh [5, 6, 7, 8]. In [7] they studied monotone string and tree automata and languages. Since monotonicity is basically a property of the underlying algebras, we begin by defining monotone algebras. A Z -algebra $\mathcal{A} = (A, Z)$ is *monotone* if there is a partial order \leq on A such that $a \leq az^{\mathcal{A}}$ for all $a \in A$ and $z \in Z$. A language is *monotone* if it is recognized by a finite monotone algebra. Let $Mon = \{Mon(Z)\}$ be the family monotone languages.

A DR Σ -algebra \mathcal{A} is *monotone* if \mathcal{A}^u is monotone. A ΣX -tree language is *DR monotone* if it is recognized by a finite monotone DR Σ -algebra. Let $DMon = \{DMon(\Sigma, X)\}$ be the DR-FTL of DR monotone tree languages. These definitions are equivalent to the ones of Gécseg and Imreh [7] and the next lemma is an immediate consequence of their corresponding results.

Lemma 6.1. *A DR Σ -algebra $\mathcal{A} = (A, \Sigma)$ is monotone if and only if the reachability relation $\Rightarrow_{\mathcal{A}}^*$ is a partial order on A . Moreover, if \mathcal{A} is monotone for some partial order \leq , then $a \Rightarrow_{\mathcal{A}}^* b$ implies $a \leq b$ ($a, b \in A$).*

Let \mathbf{M}_{cld} be the class of the finite monoids in which all *right-unit submonoids* are *closed under divisors*. These notions, introduced in [7], can be defined by saying that a finite monoid M belongs to \mathbf{M}_{cld} if $s(r_1 r_2) = s$ implies $sr_1 = s$ for all $s, r_1, r_2 \in M$; each $s \in M$ has its right-unit submonoid $\text{RU}(s) := \{r \in M \mid sr = s\}$. In [7] it was shown that that a regular string language is monotone if and only if its syntactic monoid is in \mathbf{M}_{cld} , and in [5] Proposition 5.2 was used for showing that the DR monotone tree languages are path-defined by \mathbf{M}_{cld} . That \mathbf{M}_{cld} is closed under subdirect products and factors is also implied by the following fact.

Proposition 6.1. *\mathbf{M}_{cld} is a VFM.*

Proof. It is clear that $S(\mathbf{M}_{cld}), P_f(\mathbf{M}_{cld}) \subseteq \mathbf{M}_{cld}$. To prove $H(\mathbf{M}_{cld}) \subseteq \mathbf{M}_{cld}$, let $\varphi : M \rightarrow M'$ be an epimorphism and assume that $M \in \mathbf{M}_{cld}$. If $M' \notin \mathbf{M}_{cld}$, there are elements $a', b', c' \in M'$ such that $a'b'c' = a'$, but $a'b' \neq a'$. Let $a, b, c \in M$ be elements for which $a\varphi = a'$, $b\varphi = b'$ and $c\varphi = c'$. Since M is finite, there are numbers $k \geq 0, m \geq 1$ such that $(bc)^{k+m} = (bc)^k$. Let $d := a(bc)^k$. Then $d \cdot (b \cdot (c(bc)^{m-1})) = a(bc)^{k+m} = d$ but $d \cdot b \neq d$ because $(d \cdot b)\varphi = a'b'$ while $d\varphi = a'$. This would mean that $b \notin \text{RU}(d)$ although $b \cdot (c(bc)^{m-1}) \in \text{RU}(d)$. Hence, $M' \in \mathbf{M}_{cld}$ must hold. \square

Since Mon is defined by \mathbf{M}_{cld} , it follows from Eilenberg's Variety Theorem that $Mon = \{Mon(Z)\}$ is a $*$ -variety.

Proposition 6.2. $DMon = \mathcal{V}_{Mon}$.

Proof. Let $T \in DRec(\Sigma, X)$. If $T \in DMon(\Sigma, X)$, then T is recognized by a finite monotone DR Σ -algebra \mathcal{A} . By Proposition 4.4, every T_x ($x \in X$) is recognized by the monotone $\widehat{\Sigma}$ -algebra \mathcal{A}^u . Hence, $T \in \mathcal{V}_{Mon}(\Sigma, X)$.

Conversely, if $T_x \in Mon(\widehat{\Sigma})$ for every $x \in X$, then each T_x is recognized by a monotone $\widehat{\Sigma}$ -algebra \mathcal{A}_x . By Proposition 4.4 T is recognized by the direct product of the algebras \mathcal{A}_x^d , and this DR algebra is monotone (cf. Proposition 7.2 below). \square

By Proposition 5.4, Theorem 22 of [7] follows from Proposition 6.2.

Corollary 6.1. $DMon$ is path-defined by the VFM \mathbf{M}_{cld} .

As a second example we consider nilpotent DR algebras and tree languages. Since nilpotent string languages are characterized by their syntactic semigroups, we shall use Proposition 5.7. The finite and co-finite ε -free languages form a $+$ -variety $Nil = \{Nil(Z)\}$ which corresponds to the VFS \mathbf{Nil} of finite nilpotent semigroups: a semigroup S is *nilpotent* if it has a zero-element 0 and there is a number $k \geq 1$ such that $s_1 \cdot \dots \cdot s_k = 0$ for all $s_1, \dots, s_k \in S$ (cf. [2, 17]). A Z -algebra $\mathcal{A} = (A, Z)$ is *nilpotent* if there exist an element $\tilde{a} \in A$ and a number $k \geq 0$ such that $av^A = \tilde{a}$ for all $a \in A$ and every $v \in Z^*$ of length k . It is easy to see that the ε -free languages recognized by these algebras are exactly the members of Nil (cf. [9], p. 125).

Let us call a DR Σ -algebra \mathcal{A} *nilpotent* if the $\widehat{\Sigma}$ -algebra \mathcal{A}^u is nilpotent. It is clear that this definition is equivalent to the one of [8]. A ΣX -tree language is *DR nilpotent* if it is recognized by a finite nilpotent DR Σ -algebra. Let $DNil = \{DNil(\Sigma, X)\}$ be the FTL of ε -free DR nilpotent tree languages.

Proposition 6.3. $DNil = \mathcal{V}_{Nil}$.

Proof. The proposition follows from Proposition 4.4 and the facts mentioned above. Firstly, if $T \subseteq T_{\Sigma}^+(X)$ is recognized by a finite nilpotent DR Σ -algebra \mathcal{A} , then every T_x is recognized by the finite nilpotent $\widehat{\Sigma}$ -algebra \mathcal{A}^u , and therefore belongs to $Nil(\widehat{\Sigma})$. On the other hand, if each T_x is recognized by a finite nilpotent $\widehat{\Sigma}$ -algebra \mathcal{A}_x , then T is recognized by the finite nilpotent DR Σ -algebra $\prod(\mathcal{A}_x^d \mid x \in X)$. \square

Proposition 5.7 yields the following result proved in [8].

Corollary 6.2. $DNil$ is path-defined by the VFS \mathbf{Nil} .

As our third example we discuss the DR definite tree languages considered in [6, 8]. Let us first recall that a Z -algebra $\mathcal{A} = (A, Z)$ is *definite* if there is a $k \geq 0$ such that $av^A = bv^A$ for all $a, b \in A$ and every $v \in Z^*$ of length k . The languages recognized by definite algebras are also called *definite*, and they are the languages of the form $E \cup Z^*F$ where, for some $k \geq 0$, E is a set of words of length $< k$ and F is a set of words of length k (cf. [9], for example). The ε -free definite languages form

a $+$ -variety $Def = \{Def(Z)\}$ characterized by the VFS \mathbf{D} of all finite semigroups S such that $Se = \{e\}$ for every idempotent $e \in S$ (cf. [2]).

Let us call a DR Σ -algebra \mathcal{A} *definite* if the $\widehat{\Sigma}$ -algebra \mathcal{A}^u is definite, and say that a ΣX -tree language is *DR definite* if it is recognized by a finite definite DR Σ -algebra. Again, these definitions are equivalent to those of [8]. Let $DDef = \{DDef(\Sigma, X)\}$ be the DR-FTL of all ε -free DR definite tree languages.

It follows directly from our definitions and Proposition 4.4 that $DDef = \mathcal{V}_{Def}$. Hence, Proposition 5.7 yields the following result proved in [8].

Proposition 6.4. *$DDef$ is path-defined by the VFS \mathbf{D} .*

Thus all three families of DR-recognizable tree languages shown in [6, 8] to be path-definable by a class of finite monoids or semigroups are obtained from a $*$ - or a $+$ -variety using Proposition 5.4 or Proposition 5.7. By Propositions 5.5 and 5.8 this can be expected once we know that the corresponding family of string languages is a $*$ - or a $+$ -variety. Indeed, any $*$ - and $+$ -variety will yield a DR-FTL or an ε -free DR-FTL path-definable by a VFM or a VFS, respectively.

7 Varieties of finite DR algebras

In this section we discuss varieties of finite DR Σ -algebras for an arbitrarily fixed ranked alphabet Σ . The class operators S , H , P_f and V_f are defined for DR Σ -algebras in the natural way, and we call a class \mathbf{K} of finite DR Σ -algebras a *variety of finite DR Σ -algebras* (DR Σ -VFA) if $S(\mathbf{K})$, $H(\mathbf{K})$, $P_f(\mathbf{K}) \subseteq \mathbf{K}$.

In [3] Ésik defines $\mathbf{K}^u := \{\mathcal{A}^u \mid \mathcal{A} \in \mathbf{K}\}$ for any class \mathbf{K} of DR Σ -algebras. He noted that \mathbf{K} is a variety of DR Σ -algebras if and only if \mathbf{K}^u is a variety of $\widehat{\Sigma}$ -algebras, and that notions from the theory of varieties of algebras may therefore be extended to DR algebras. We shall apply the operation $\mathbf{K} \mapsto \mathbf{K}^u$ to classes of finite DR Σ -algebras, and we also introduce the converse operation that associates with each class \mathbf{U} of finite $\widehat{\Sigma}$ -algebras the class $\mathbf{U}^d := \{\mathcal{C}^d \mid \mathcal{C} \in \mathbf{U}\}$ of finite DR Σ -algebras. Obviously, $\mathbf{K}^{ud} = \mathbf{K}$ and $\mathbf{U}^{du} = \mathbf{U}$.

Lemma 7.1. *Let \mathbf{K} be a class of finite DR Σ -algebras and \mathbf{U} be a class of finite $\widehat{\Sigma}$ -algebras. Then*

- (a) $H(\mathbf{K})^u = H(\mathbf{K}^u)$, $S(\mathbf{K})^u = S(\mathbf{K}^u)$ and $P_f(\mathbf{K})^u = P_f(\mathbf{K}^u)$, and
- (b) $H(\mathbf{U})^d = H(\mathbf{U}^d)$, $S(\mathbf{U})^d = S(\mathbf{U}^d)$ and $P_f(\mathbf{U})^d = P_f(\mathbf{U}^d)$.

Hence, \mathbf{K} is a DR Σ -VFA if and only if \mathbf{K}^u is a $\widehat{\Sigma}$ -VFA, and \mathbf{U} is a $\widehat{\Sigma}$ -VFA if and only if \mathbf{U}^d is a DR Σ -VFA.

Proof. The equalities in (a) and (b) are immediate consequences of Lemma 3.1, and the remaining claims follow from them. \square

An easy modification of Tarski's well-known HSP-theorem (cf. [1], p. 61) yields $V_f(\mathbf{K}) = HSP_f(\mathbf{K})$ for any ranked alphabet Σ and any class \mathbf{K} of finite Σ -algebras (cf. [18], for example). Let us derive the corresponding representation for finite DR Σ -algebras.

Proposition 7.1. $V_f(\mathbf{K}) = HSP_f(\mathbf{K})$ for any class \mathbf{K} of finite DR Σ -algebras.

Proof. Clearly, $\mathbf{K} \subseteq HSP_f(\mathbf{K})$. As a special case of the fact noted above, we get $V_f(\mathbf{K}^u) = HSP_f(\mathbf{K}^u)$. This means, in particular, that $HSP_f(\mathbf{K}^u)$ is a $\widehat{\Sigma}$ -VFA. Since $HSP_f(\mathbf{K})^u = HSP_f(\mathbf{K}^u)$, this implies by Lemma 7.1 that $HSP_f(\mathbf{K})$ is a DR Σ -VFA. If \mathbf{L} is a DR Σ -VFA with $\mathbf{K} \subseteq \mathbf{L}$, then $HSP_f(\mathbf{K}) \subseteq HSP_f(\mathbf{L}) = \mathbf{L}$. Hence, $HSP_f(\mathbf{K})$ is the DR Σ -VFA generated by \mathbf{K} . \square

Lemma 7.1 and Proposition 7.1 yield the following fact.

Corollary 7.1. $V_f(\mathbf{K}) = V_f(\mathbf{K}^u)^d$ for any class \mathbf{K} of finite DR Σ -algebras.

Let \mathbf{DMon}_Σ , \mathbf{DNil}_Σ and \mathbf{DDef}_Σ denote the classes of all finite monotone, nilpotent and definite DR Σ -algebras, respectively.

Proposition 7.2. \mathbf{DMon}_Σ , \mathbf{DNil}_Σ and \mathbf{DDef}_Σ are DR Σ -VFAs.

Proof. By Lemma 7.1, it suffices to verify that the corresponding classes \mathbf{DMon}_Σ^u , \mathbf{DNil}_Σ^u and \mathbf{DDef}_Σ^u of unary algebras are $\widehat{\Sigma}$ -VFAs.

Gécseg and Imreh [7] proved that all finite direct products and homomorphic images of monotone finite automata are monotone. These results apply directly to unary algebras, and hence $P_f(\mathbf{DMon}_\Sigma^u) \subseteq \mathbf{DMon}_\Sigma^u$ and $H(\mathbf{DMon}_\Sigma^u) \subseteq \mathbf{DMon}_\Sigma^u$. Since it is clear that subalgebras of monotone algebras are monotone, we may conclude that \mathbf{DMon}_Σ^u is a $\widehat{\Sigma}$ -VFA.

In [18] it was noted that for any ranked alphabet Σ , the finite nilpotent Σ -algebras form an Σ -VFA, and in [4] Ésik proved the corresponding fact about finite definite Σ -algebras. Hence, also \mathbf{DNil}_Σ^u and \mathbf{DDef}_Σ^u are $\widehat{\Sigma}$ -VFAs. \square

Let us now consider equational definitions of DR Σ -VFAs. The terms appearing in $\widehat{\Sigma}$ -identities are written as expressions ξu , where ξ is a variable and $u \in \widehat{\Sigma}^*$. There are two types of $\widehat{\Sigma}$ -identities, the *regular identities* $\xi u \approx \xi v$ and the *irregular identities* $\xi u \approx \xi' v$, in which ξ and ξ' are two distinct variables. A $\widehat{\Sigma}$ -algebra $\mathcal{C} = (C, \widehat{\Sigma})$ satisfies $\xi u \approx \xi v$ if $u^c = v^c$, and it satisfies $\xi u \approx \xi' v$, where $\xi \neq \xi'$, if $cu^c = dv^c$ for all $c, d \in C$. Furthermore, \mathcal{C} ultimately satisfies an ω -sequence

$$E = \langle \ell_0 \approx r_0, \ell_1 \approx r_1, \ell_2 \approx r_2, \dots \rangle$$

of $\widehat{\Sigma}$ -identities if there exists an $n_0 \geq 0$ such that \mathcal{C} satisfies $\ell_n \approx r_n$ for every $n \geq n_0$. The class E^u of finite $\widehat{\Sigma}$ -algebras ultimately defined by E consists of the finite $\widehat{\Sigma}$ -algebras ultimately satisfying E . By a well-known theorem by Eilenberg and Schützenberger (cf. [2, 17]), a class \mathbf{K} of finite $\widehat{\Sigma}$ -algebras is a $\widehat{\Sigma}$ -VFA if and only if $\mathbf{K} = E^u$ for some ω -sequence E of $\widehat{\Sigma}$ -identities.

Following Ésik [3] we say that a DR Σ -algebra \mathcal{A} satisfies a $\widehat{\Sigma}$ -identity $\ell \approx r$ if \mathcal{A}^u satisfies $\ell \approx r$. Naturally, \mathcal{A} ultimately satisfies an ω -sequence E of $\widehat{\Sigma}$ -identities if \mathcal{A}^u ultimately satisfies E . The class of finite DR Σ -algebras ultimately satisfying E is denoted by E^d . Thus E^u is a class of $\widehat{\Sigma}$ -algebras and $E^{ud} := (E^u)^d$ is the corresponding class of DR Σ -algebras. Similarly, $E^{du} := (E^d)^u$ is the class of $\widehat{\Sigma}$ -algebras corresponding to the class E^d of DR Σ -algebras. Since $E^{ud} = E^d$ and

$E^{du} = E^u$, the next proposition follows from the Eilenberg-Schützenberger theorem and Lemma 7.1.

Proposition 7.3. *A class \mathbf{K} of finite DR Σ -algebras is a DR Σ -VFA if and only if $\mathbf{K} = E^d$ for some ω -sequence E of $\widehat{\Sigma}$ -identities.*

8 Concluding remarks

We have considered some aspects of DR tree recognizers and DR-recognizable tree languages. Our algebraic approach is in a large part based on the connection between DR algebras and unary algebras put forward by Ésik [3]. It was used for describing the relationship between a DR-recognizable tree language and its path language as well as in the discussion of varieties of finite DR algebras.

In Section 5 we showed that any $*$ - or $+$ -variety defines a family of DR-recognizable tree languages path-definable by a variety of finite monoids or a variety of finite semigroups, respectively. Hence there are many families of DR-recognizable tree languages that could be investigated similarly as the families *DMon*, *DNil* and *DDef* were studied in the papers [7, 8, 13, 14]. On a more general level, one should describe the common properties of all such families. In particular, it is conceivable that they are characterized by some closure properties. It is also natural to consider the families of DR-recognizable tree languages that correspond to a varieties of finite DR algebras. Such questions belong to a variety theory of DR-recognizable tree languages still to be developed.

References

- [1] S. Burris and H.P. Sankappanavar, *A Course in Universal Algebra*, Springer-Verlag, New York, 1981.
- [2] S. Eilenberg, *Automata, Languages, and Machines*. Vol. B, Academic Press, New York 1976.
- [3] Z. Ésik, Varieties and general products of top-down algebras, *Acta Cybernetica* **7** (1986), 293-298.
- [4] Z. Ésik, Definite tree automata and their cascade composition, *Publicationes Mathematicae Debrecen* **48** 3-4 (1996), 243-261.
- [5] F. Gécseg, Classes of tree languages determined by classes of monoids, *International Journal of Foundations of Computer Science* **18** (2007), 1237-1246.
- [6] F. Gécseg, Classes of tree languages and DR tree languages given by classes of semigroups, *Acta Cybernetica* **20** (2011), 253-267.
- [7] F. Gécseg and B. Imreh, On monotone automata and monotone languages, *Journal of Automata, Languages and Combinatorics* **7** (2002), 71-82.

- [8] F. Gécseg and B. Imreh, On definite and nilpotent DR tree languages, *Journal of Automata, Languages and Combinatorics* **9** (2004), 55-60.
- [9] F. Gécseg and I. Peák: *Algebraic theory of automata*, Akadémiai Kiadó, Budapest 1972.
- [10] F. Gécseg and M. Steinby, Minimal ascending tree automata, *Acta Cybernetica* **4** (1978), 37-44.
- [11] F. Gécseg and M. Steinby, *Tree Automata*, Akadémiai Kiadó, Budapest, 1984. 2. ed. downloadable from arXiv.org as arXiv:1509.06233, September 2015.
- [12] F. Gécseg and M. Steinby, Minimal recognizers and syntactic monoids of DR tree languages, *Words, Semigroups and Transductions* (eds. M. Ito, G. Paun and S. Yu), World Scientific, Singapore 2001, 155-167.
- [13] Gy. Gyurica, On monotone languages and their characterization by regular expressions, *Acta Cybernetica* **18** (2007), 117-134.
- [14] Gy. Gyurica, On nilpotent languages and their characterization by regular expressions, *Acta Cybernetica* **19** (2009), 231-244.
- [15] E. Jurvanen, *On Tree Languages Defined by Deterministic Root-to-frontier Recognizers*, Dissertation, Department of Mathematics, University of Turku, Turku 1995.
- [16] M. Magidor and G. Moran, *Finite Automata over Finite Trees*, Technical Report 30, Department of Mathematics, Hebrew University, Jerusalem 1969.
- [17] J.E. Pin, *Varieties of Formal Languages*, North Oxford Academic Publishers, London 1986.
- [18] M. Steinby, A theory of tree language varieties, *Tree Automata and Languages* (eds. M. Nivat and A. Podelski), North-Holland, Amsterdam 1992, 57-81.
- [19] W. Thomas, Logical aspects in the study of tree languages, *9th Colloquium on Trees in Algebra and Programming* (ed. B. Courcelle), Proc. 9th CAAP, Bordeaux, 1984, Cambridge University Press, London, 1984, pp. 31-49.
- [20] J. Virágh, Deterministic tree automata I, *Acta Cybernetica* **5** (1980), 33-42; II, *ibid* **6** (1983), 291-301.