Acta Cybernetica **20** (2011) 253–267.

Classes of Tree Languages and DR Tree Languages Given by Classes of Semigroups

Ferenc Gécseg*

Abstract

In the first section of the paper we give general conditions under which a class of recognizable tree languages with a given property can be defined by a class of monoids or semigroups defining the class of string languages having the same property. In the second part similar questions are studied for classes of (DR) tree languages recognized by deterministic root-to-frontier tree recognizers.

Keywords: recognizable tree languages, DR recognizable tree languages, syntactic semigroups, syntactic monoids

1 Introduction

In [3] we characterized the class of recognizable monotone string languages and that of recognizable monotone tree languages by means of syntactic monoids. It turned out that both classes can be defined by the class **M** of monoids whose right unit submonoids are closed under divisors, i.e. a recognizable string or tree language is monotone if and only if its syntactic monoid is in **M**. This was the observation which motivated the writing of paper [1], where such characterizations from more general classes of string languages have been lifted to classes of (frontier-to-root) tree languages.

In [4] we obtained results for the classes of definite and nilpotent deterministic root-to-frontier (DR) tree languages similar to those in [3]. The aim of this paper is to strengthen the main result of [1], on one hand, and to give general conditions under which a class of DR tree languages with a given property can be defined by a class of monoids or semigroups defining the class of string languages having the same property, on the other hand. The proofs are based on the observation that the syntactic monoids (syntactic semigroups) of recognizable tree languages and the syntactic path monoids (syntactic path semigroups) of DR tree languages can be given as subdirect products of the syntactic monoids (syntactic semigroups)

*Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary

DOI: 10.14232/actacyb.20.2.2011.3

of suitable recognizable string languages. We shall show for the classes of DR-monotone, DR-nilpotent and DR-definite tree languages that they satisfy these conditions.

It should be noted that the classes of tree languages considered in this paper are not necessarily varieties. For readers interested in varieties of recognizable tree languages, we refer to the fundamental papers [11] and [13].

2 Notions and Notation

Sets of operational symbols will be denoted by Σ. If Σ is finite and nonvoid, then it is called a *ranked alphabet*. For the subset of Σ consisting of all m-ary operational symbols from Σ we shall use the notation Σ_m $(m \geq 0)$. By a Σ-*algebra* we mean a pair $\mathcal{A} = (A, \{\sigma^{\mathcal{A}} | \sigma \in \Sigma\})$, where $\sigma^{\mathcal{A}}$ is an m-ary operation on A if $\sigma \in \Sigma_m$. If there will be no danger of confusion then we omit the superscript $\mathcal{A}$ in $\sigma^{\mathcal{A}}$ and simply write $\mathcal{A} = (A, \Sigma)$. Finally, all algebras considered in this paper will be finite, i.e. A is finite and Σ is a ranked alphabet.

Take a Σ-algebra $\mathcal{A} = (A, \Sigma)$, a $\sigma \in \Sigma_m$ $(m > 0)$, an i $(1 \leq i \leq m)$ and $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_m \in A$. Then $\sigma(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_m)$ is an *elementary translation symbol* of $\mathcal{A}$. The set of all elementary translation symbols of $\mathcal{A}$ will be denoted by $\mathrm{ETS}(\mathcal{A})$. In the sequel elementary translation symbols will be considered as unary operational symbols. Moreover, $\mathrm{ETalg}(\mathcal{A})$ will denote the unary algebra $(A, \mathrm{ETS}(\mathcal{A}))$ with

$$\sigma(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_m)^{\mathrm{ETalg}(\mathcal{A})}(a) = \sigma^{\mathcal{A}}(a_1, \ldots, a_{i-1}, a, a_{i+1}, \ldots, a_m)$$
$$(\sigma(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_m) \in \mathrm{ETS}(\mathcal{A}),\ a \in A).$$

Let X be a set of variables. The set $T_\Sigma(X)$ of ΣX-*trees* (or Σ-*trees over* X) is defined as follows:

(i) $X \subseteq T_\Sigma(X)$,

(ii) $\sigma(p_1, \ldots, p_m) \in T_\Sigma(X)$ if $m \geq 0$, $\sigma \in \Sigma_m$ and $p_1, \ldots, p_m \in T_\Sigma(X)$, and

(iii) every ΣX-tree can be obtained by applying the rules (i) and (ii) a finite number of times.

In the sequel X will stand for the countable set $\{x_1, x_2, \ldots\}$, and for every $n \geq 0$, X_n will denote the subset $\{x_1, \ldots, x_n\}$ of X. A subset of $T_\Sigma(X_n)$ is called a ΣX_n-*language*. If Σ or X_n is not specified then we speak of a *tree language*.

Take a Σ-algebra $\mathcal{A} = (A, \Sigma)$ and a tree $p \in T_\Sigma(X_n)$. Let us define the mapping $p^{\mathcal{A}} : A^n \to A$ in the following way: for any $\mathbf{a} = (a_1, \ldots, a_n) \in A^n$,

(i) if $p = x_i \in X_n$, then $p^{\mathcal{A}}(\mathbf{a}) = a_i$,

(ii) if $p = \sigma(p_1, \ldots, p_m)$ $(\sigma \in \Sigma_m,\ p_1, \ldots, p_m \in T_\Sigma(X_n))$, then

$$p^{\mathcal{A}}(\mathbf{a}) = \sigma^{\mathcal{A}}(p_1^{\mathcal{A}}(\mathbf{a}), \ldots, p_m^{\mathcal{A}}(\mathbf{a})).$$

If there is no danger of confusion, then we omit $\mathcal{A}$ in $p^{\mathcal{A}}$.

A ΣX_n-*recognizer* is a system $\mathbf{A} = (\mathcal{A}, \mathbf{a}, A')$, where

(i) $\mathcal{A} = (A, \Sigma)$ is an algebra,

(ii) $\mathbf{a} = (a^{(1)}, \ldots, a^{(n)})$ $(a^{(1)}, \ldots, a^{(n)} \in A)$ is the *initial* vector,

(iii) $A' \subseteq A$ is the set of *final* states.

If $n = 1$, then we usually write $a^{(1)}$ for $(a^{(1)})$. Moreover, it is said that $\mathbf{A}$ is *connected* if $\{p(\mathbf{a}) \mid p \in T_\Sigma(X_n)\} = A$.

If Σ and X_n are not specified then we speak of a *tree recognizer*. Furthermore, if $\Sigma = \Sigma_1$ and $n = 1$, then $\mathbf{A}$ is a *finite state recognizer*, shortly *recognizer*. If we are dealing with recognizers, then (unary) trees are sometimes written as words: for a tree $\sigma_1(\ldots(\sigma_k(x_1))\ldots)$ we may write $\sigma_k \ldots \sigma_1$.

The *tree language* $T(\mathbf{A})$ *recognized* by the ΣX_n-recognizer $\mathbf{A} = (\mathcal{A}, \mathbf{a}, A')$ is given by

$$T(\mathbf{A}) = \{p \in T_\Sigma(X_n) \mid p(\mathbf{a}) \in A'\}.$$

The class of recognizable tree languages will be denoted by **Treelang**, and **Lang** is its subclass consisting of all tree languages recognizable by finite state recognizers.

Let **Prop** be a property of recognizable tree languages. The best way is to define **Prop** as a subclass of **Treelang**. If **K** is a subclass of **Treelang**, then **Prop(K)** will denote the class of all tree languages which are simultaneously in **Prop** and **K**.

If not otherwise specified, $\mathbf{A}$ will be the ΣX_n-recognizer $(\mathcal{A}, \mathbf{a}, A')$. Here $\mathcal{A}$ is a Σ-algebra (A, Σ), $\mathbf{a} = (a^{(1)}, \ldots, a^{(n)})$ and $A' \subseteq A$. Consider a ΣX_n-recognizer $\mathbf{A}$. For each $x \in X_n \cup \Sigma_0$, define the finite state ETS($\mathcal{A}$)-recognizer $\mathbf{A}_x = (\mathcal{A}_x, a_x, A'_x)$ in the following way:

(1) $a_x = \begin{cases} a^{(i)}, & \text{if } x = x_i \ (1 \le i \le n), \\ \sigma^{\mathcal{A}}, & \text{if } x = \sigma \in \Sigma_0. \end{cases}$

(2) $A_x = \{p^{\mathrm{ETalg}(\mathcal{A})}(a_x) \mid p \in T_{\mathrm{ETS}(\mathcal{A})}(X_1)\}$.

(3) $\mathcal{A}_x = (A_x, \mathrm{ETS}(\mathcal{A}))$ is a subalgebra of ETalg($\mathcal{A}$).

(4) $A'_x = A_x \cap A'$.

These $\mathbf{A}_x$ are called *translation recognizers* of $\mathbf{A}$.

Let $\hat{T}_\Sigma(X_n)$ denote the set of all Σ-trees over $X_n \cup \{*\}$ $(* \notin X_n)$ in which $*$ occurs exactly once. Elements in $\hat{T}_\Sigma(X_n)$ are *special trees* of Thomas [14] and Heuter [9]. Let us define the product $q{\cdot}p$ of $q \in T_\Sigma(X_n) \cup \hat{T}_\Sigma(X_n)$ and $p \in \hat{T}_\Sigma(X_n)$ by $q{\cdot}p = p(q)$. (Here and in the sequel, for any $p \in \hat{T}_\Sigma(X_n)$ and $q \in T_\Sigma(X_n) \cup \hat{T}_\Sigma(X_n)$, $p(q)$ is obtained by replacing the occurrence of $*$ in p by q $(p(q) = p(* \leftarrow q))$.) Obviously, under this multiplication $\hat{T}_\Sigma(X_n)$ is a monoid with the identity element $*$.

Let $T \subseteq T_\Sigma(X_n)$ be a tree language. Define the binary relation μ_T on $\hat{T}_\Sigma(X_n)$ in the following way: for any $p, q \in \hat{T}_\Sigma(X_n)$,

$$p \equiv q(\mu_T) \iff$$

$$(\forall p', p'' \in \hat{T}_\Sigma(X_n),\ x \in X_n \cup \Sigma_0)((p' \cdot p \cdot p'')(x) \in T \iff (p' \cdot q \cdot p'')(x) \in T).$$

This μ_T is a congruence of the monoid $\hat{T}_\Sigma(X_n)$, which is called the *syntactic congruence* of T. Moreover, the quotient monoid $\hat{T}_\Sigma(X_n)/\mu_T$ is the *syntactic monoid* of T, which will be denoted by $\mathrm{Syntm}(T)$.

The restriction of μ_T to $\hat{T}_\Sigma(X_n) \setminus \{*\}$ will be denoted by the same μ_T. The quotient semigroup $\hat{T}_\Sigma(X_n) \setminus \{*\}/\mu_T$ is the *syntactic semigroup* of T. The syntactic semigroup of T will be denoted by $\mathrm{Synts}(T)$.

We say that a property **Prop** of recognizable tree languages *can be defined by a class* **M** *of monoids,* if for all $T \in$ **Treelang**, $T \in$ **Prop** $\iff \mathrm{Syntm}(T) \in$ **M**. Similarly, a property **Prop** of recognizable tree languages *can be defined by a class* **S** *of semigroups,* if for all $T \in$ **Treelang**, $T \in$ **Prop** $\iff \mathrm{Synts}(T) \in$ **S**. For any ΣX_n-recognizer **A** and $p \in \hat{T}_\Sigma(X_n)$, let $p(\mathbf{a})$ stand for $p(x_1 \leftarrow a^{(1)}, \ldots, x_n \leftarrow a^{(n)})$ and $p(\mathbf{a})(a)$ for $p(\mathbf{a})(* \leftarrow a)$ $(a \in A)$, i.e. $p(\mathbf{a})(a)$ is obtained from p by replacing the occurrences of x_i by $a^{(i)}$ and that of $*$ by a.

Let Y be an ordinary alphabet, Y^* the free semigroup generated by Y and $L \subseteq Y^*$ a language over Y. Furthermore, let μ_L be the binary relation on Y^* given by $u \equiv v(\mu_L)$ $(u, v \in Y^*)$ iff for any $u', u'' \in Y^*$ the equivalence $u'uu'' \in L \iff u'vu'' \in L$ holds. As it is well known, μ_L is a congruence relation on the free monoid Y^*, and the quotient monoid Y^*/μ_L is called the *syntactic monoid* of L. Let the same μ_L denote the restriction of μ_L to the semigroup $Y^+ = Y^* \setminus \{e\}$, where e is the empty word. The quotient semigroup Y^+/μ_L is the *syntactic semigroup* of L. It is obvious, if finite state recognizers are taken as special tree recognizers, then the above two definitions of syntactic monoids coincide. The same is true for syntactic semigroups.

For notions and notation not defined in this paper, see [6] and [7].

3 Tree languages

Let **A** be an arbitrary connected ΣX_n-recognizer. Define the mapping $\epsilon_{\mathbf{A}} : \hat{T}_\Sigma(X_n) \to T_{\mathrm{ETS}(\mathcal{A})}(*)$ in the following way:

1) $\epsilon_{\mathbf{A}}(*) = *$.

2) If $p = \sigma(p_1, \ldots, p_{i-1}, p_i, p_{i+1}, \ldots, p_m)$ $(\sigma \in \Sigma_m,\ p_j \in T_\Sigma(X_n),\ j \in \{1, \ldots, i-1, i+1, \ldots, m\},\ p_i \in \hat{T}_\Sigma(X_n))$, then

$$\epsilon_{\mathbf{A}}(p) = \sigma(p_1^{\mathcal{A}}(\mathbf{a}), \ldots, p_{i-1}^{\mathcal{A}}(\mathbf{a}), \epsilon_{\mathbf{A}}(p_i), p_{i+1}^{\mathcal{A}}(\mathbf{a}), \ldots, p_m^{\mathcal{A}}(\mathbf{a})).$$

Since **A** is connected, $\epsilon_{\mathbf{A}}$ is an onto mapping. If there is no danger of confusion we shall omit **A** in $\epsilon_{\mathbf{A}}$.

Let $T \subseteq T_\Sigma(X_n)$ be a tree language. For each $x \in X_n \cup \Sigma_0$, define the binary relation $\mu_{T,x}$ on $\hat{T}_\Sigma(X_n)$ in the following way: for any $p, q \in \hat{T}_\Sigma(X_n)$,

$$p \equiv q(\mu_{T,x}) \Longleftrightarrow$$

$$(\forall p', p'' \in \hat{T}_\Sigma(X_n))((p' \cdot p \cdot p'')(x) \in T \Longleftrightarrow (p' \cdot q \cdot p'')(x) \in T).$$

Clearly, these relations $\mu_{T,x}$ are congruences of the monoid $\hat{T}_\Sigma(X_n)$.

By the definitions of the syntactic monoid and the syntactic semigroup of a ΣX_n-language T we obviously have the following two results.

Lemma 1. *The syntactic monoid* Syntm(T) *is isomorphic to a subdirect product of the monoids $\hat{T}_\Sigma(X_n)/\mu_{T,x}$, $x \in X_n \cup \Sigma_0$.* ♢

Lemma 2. *The syntactic semigroup* Synts(T) *is isomorphic to a subdirect product of the semigroups $\hat{T}_\Sigma(X_n) \setminus \{*\}/\mu_{T,x}$, $x \in X_n \cup \Sigma_0$, where the restriction of $\mu_{T,x}$ to $\hat{T}_\Sigma(X_n) \setminus \{*\}$ is denoted by the same $\mu_{T,x}$.* ♢

We now show

Lemma 3. *Let* **A** *be an arbitrary connected ΣX_n-recognizer. Then for all $x \in X_n \cup \Sigma_0$,*

$$\hat{T}_\Sigma(X_n)/\mu_{T,x} \cong \mathrm{Syntm}(\mathrm{T}(\mathbf{A}_\mathrm{x})).$$

Proof. It is obvious that for any two $p, q \in \hat{T}_\Sigma(X_n)$ we have $\epsilon(p \cdot q) = \epsilon(p) \cdot \epsilon(q)$. We show that for all $p, q \in \hat{T}_\Sigma(X_n)$,

$$p \equiv q(\mu_{T,x}) \Longleftrightarrow \epsilon(p) \equiv \epsilon(q)(\mu_{T(\mathbf{A}_x)}).$$

Remember that ϵ is an onto mapping since **A** is connected. Thus,

$$\begin{gathered} p \equiv q(\mu_{T,x}) \\ \Updownarrow \\ (\forall r, s \in \hat{T}_\Sigma(X_n))((r \cdot p \cdot s)(x) \in T \Longleftrightarrow (r \cdot q \cdot s)(x) \in T) \\ \Updownarrow \\ (\forall r, s \in \hat{T}_\Sigma(X_n))((r \cdot p \cdot s)(\mathbf{a})(a_x) \in A' \Longleftrightarrow (r \cdot q \cdot s)(\mathbf{a})(a_x) \in A') \\ \Updownarrow \\ (\forall r, s \in \hat{T}_\Sigma(X_n))(\epsilon(r \cdot p \cdot s)(a_x) \in A'_x \Longleftrightarrow \epsilon(r \cdot q \cdot s)(a_x) \in A'_x) \\ \Updownarrow \\ (\forall r, s \in \hat{T}_\Sigma(X_n))((\epsilon(r) \cdot \epsilon(p) \cdot \epsilon(s)) \in A'_x \Longleftrightarrow (\epsilon(r) \cdot \epsilon(q) \cdot \epsilon(s)) \in A'_x) \\ \Updownarrow \\ \epsilon(p) \equiv \epsilon(q)(\mu_{T(\mathbf{A}_x)}). \end{gathered}$$

Therefore, $p/\mu_{T,x} \to \epsilon(p)/\mu_{T(\mathbf{A}_x)}$ $(p \in \hat{T}_\Sigma(X_n))$ is an isomorphic mapping of $\hat{T}_\Sigma(X_n)/\mu_{T,x}$ onto Syntm(T($\mathbf{A}_\mathrm{x}$)). ♢

The following lemma can be proved in a similar way.

Lemma 4. *Let* **A** *be an arbitrary connected* ΣX_n*-recognizer. Then for all* $x \in X_n \cup \Sigma_0$,

$$\hat{T}_\Sigma(X_n) \setminus \{*\}/\mu_{T,x} \cong \mathrm{Synts}(\mathrm{T}(\mathbf{A}_\mathrm{x})).$$

$\diamondsuit$

Let **S** be a class of semigroups. We say that **S** is *closed under subdirect products,* if all subdirect products of semigroups from **S** with finitely many factors are in **S**. Moreover, **S** is *closed under subdirect factors,* if whenever a subdirect product of two semigroups is in **S**, then both of them are in **S**.

In this paper all classes of semigroups will contain only finite semigrouups.

We are now ready to state and prove

Theorem 1. *Let* **Prop** *be a property of recognizable tree languages. Assume that the following conditions are satisfied:*

(1) *For every* ΣX_n*-language* T *there exists a connected* ΣX_n*-recognizer* **A** *with* $T(\mathbf{A}) = T$ *such that*

$$T \in \mathbf{Prop} \Longleftrightarrow (\forall x \in X_n \cup \Sigma_0)(T(\mathbf{A}_x) \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) **Prop**(**Lang**) *can be defined by a class* **M** *of monoids.*

(3) **M** *is closed under subdirect products and subdirect factors.*

Then **Prop** *can be defined by* **M**.

Proof. Assume that the conditions of our theorem are satisfied.

First take a $T \in$ **Treelang** with $\mathrm{Syntm}(T) \in \mathbf{M}$, and let **A** be a connected ΣX_n-recognizer such that $T = T(\mathbf{A})$ satisfies (1). By Lemma 1 and 3, $\mathrm{Syntm}(T)$ is isomorphic to a subdirect product of the monoids $\mathrm{Syntm}(T(\mathbf{A}_x))$ $(x \in X_n \cup \Sigma_0)$. From this, by (3), we obtain that $\mathrm{Syntm}(T(\mathbf{A}_x)) \in \mathbf{M}$, and thus, by (2), $T(\mathbf{A}_x) \in$ **Prop(Lang)** for all $x \in X_n \cup \Sigma_0$, which, by (1), implies that $T = T(\mathbf{A}) \in \mathbf{Prop}$.

Conversely, assume that $T \in \mathbf{Prop}$, and let **A** be a connected tree recognizer with $T = T(\mathbf{A})$ satisfying (1). Then, for each $x \in X_n \cup \Sigma_0$, $T(\mathbf{A}_x) \in$ **Prop(Lang)**. Thus, by (2), $\mathrm{Syntm}(T(\mathbf{A}_x)) \in \mathbf{M}$. Again, by Lemma 1 and 3, $\mathrm{Syntm}(T)$ is isomorphic to a subdirect product of $\mathrm{Syntm}(T(\mathbf{A}_x))$ $(x \in X_n \cup \Sigma_0)$. Moreover, by (3), **M** is closed under subdirect products. Therefore, $\mathrm{Synt}(T) \in \mathbf{M}$. $\diamondsuit$

The next theorem can be proved in a similar way.

Theorem 2. *Let* **Prop** *be a property of recognizable tree languages. Assume that the following conditions are satisfied:*

(1) *For every* ΣX_n*-language* T *there exists a connected* ΣX_n*-recognizer* **A** *with* $T(\mathbf{A}) = T$ *such that*

$$T \in \mathbf{Prop} \Longleftrightarrow (\forall x \in X_n \cup \Sigma_0)(T(\mathbf{A}_x) \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) **Prop**(**Lang**) *can be defined by a class* **S** *of semigroups.*

(3) **S** *is closed under subdirect products and subdirect factors.*

Then **Prop** *can be defined by* **S**. ♢

In [1] we proved

Theorem 3. *Let* **Prop** *be a property of recognizable tree languages. Assume that the following conditions are satisfied.*

(1) *For all minimal tree recognizers* **A**,

$$T(\mathbf{A}) \in \mathbf{Prop} \Longleftrightarrow (\forall x \in X_n \cup \Sigma_0)(T(\mathbf{A}_x) \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) **Prop**(**Lang**) *can be defined by a class* **M** *of monoids.*

(3) **M** *is closed under subdirect products and subdirect factors.*

Then **Prop** *can be defined by* **M**. ♢

It is easy to show that the previous theorem is true for properties defined by semigroups:

Theorem 4. *Let* **Prop** *be a property of recognizable tree languages. Assume that the following conditions are satisfied.*

(1) *For all minimal tree recognizers* **A**,

$$T(\mathbf{A}) \in \mathbf{Prop} \Longleftrightarrow (\forall x \in X_n \cup \Sigma_0)(T(\mathbf{A}_x) \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) **Prop**(**Lang**) *can be defined by a class* **S** *of semigroups.*

(3) **S** *is closed under subdirect products and subdirect factors.*

Then **Prop** *can be defined by* **S**. ♢

We shall need

Lemma 5. *If* **A** *and* **B** *are equivalent connected* ΣX_n*-recognizers then for all* $x \in X_n \cup \Sigma_0$, $\mathrm{Syntm}(T(\mathbf{A}_x)) \cong \mathrm{Syntm}(T(\mathbf{B}_x))$.

Proof. For a $p \in \hat{T}_\Sigma(X_n)$ set $\overline{p} = \epsilon_{\mathbf{A}}(p)$ and $\overline{\overline{p}} = \epsilon_{\mathbf{B}}(p)$. Let $p, q \in \hat{T}_\Sigma(X_n)$ and $x \in X_n$ be arbitrary. We have

$$(\forall r, s \in \hat{T}_\Sigma(X_n))(((r \cdot p \cdot s)(x))^{\mathcal{A}}(\mathbf{a}) \in A' \Leftrightarrow ((r \cdot q \cdot s)(x))^{\mathcal{A}}(\mathbf{a}) \in A') \overset{T(\mathbf{A})=T(\mathbf{B})}{\Longleftrightarrow}$$
$$(\forall r, s \in \hat{T}_\Sigma(X_n))(((r \cdot p \cdot s)(x))^{\mathcal{B}}(\mathbf{b}) \in B' \Leftrightarrow ((r \cdot q \cdot s)(x))^{\mathcal{B}}(\mathbf{b}) \in B')$$
$$\Updownarrow$$
$$(\forall r, s \in \hat{T}_\Sigma(X_n))\overline{r \cdot p \cdot s}^{\mathcal{A}_x}(a_x) \in A' \Leftrightarrow \overline{r \cdot q \cdot s}^{\mathcal{A}_x}(a_x) \in A') \Longleftrightarrow$$
$$(\forall r, s \in \hat{T}_\Sigma(X_n))\overline{\overline{r \cdot p \cdot s}}^{\mathcal{B}_x}(b_x) \in B' \Leftrightarrow \overline{\overline{r \cdot q \cdot s}}^{\mathcal{B}_x}(b_x) \in B')$$
$$\Updownarrow$$
$$(\forall \overline{r}, \overline{s} \in \hat{T}_\Sigma(X_n))((\overline{r} \cdot \overline{p} \cdot \overline{s})^{\mathcal{A}_x}(a_x) \in A' \Leftrightarrow (\overline{r} \cdot \overline{q} \cdot \overline{s})^{\mathcal{A}_x}(a_x) \in A') \Longleftrightarrow$$
$$(\forall \overline{\overline{r}}, \overline{\overline{s}} \in \hat{T}_\Sigma(X_n))((\overline{\overline{r}} \cdot \overline{\overline{p}} \cdot \overline{\overline{s}})^{\mathcal{B}_x}(b_x) \in B' \Leftrightarrow (\overline{\overline{r}} \cdot \overline{\overline{q}} \cdot \overline{\overline{s}})^{\mathcal{B}_x}(b_x) \in B')$$
$$\Updownarrow$$
$$\overline{p} \equiv \overline{q}(\mu_{\mathbf{A}_x}) \Leftrightarrow \overline{\overline{p}} \equiv \overline{\overline{q}}(\mu_{\mathbf{B}_x}).$$

Therefore the mapping $\overline{p}/\mu_{\mathbf{A}_x} \to \overline{\overline{p}}/\mu_{\mathbf{B}_x}$ $(p \in \hat{T}_\Sigma(X_n))$ is an isomorphism between the monoids $\text{Syntm}(T(\mathbf{A}_x))$ and $\text{Syntm}(T(\mathbf{B}_x))$. ♢

The following result can be proved in a similar way.

Lemma 6. *If* $\mathbf{A}$ *and* $\mathbf{B}$ *are equivalent connected* ΣX_n*-recognizers then for all* $x \in X_n \cup \Sigma_0$, $\text{Synts}(T(\mathbf{A}_x)) \cong \text{Synts}(T(\mathbf{B}_x))$. ♢

Now we show

Theorem 5. *Theorems 1 and 3 are equivalent.*

Proof. It is obvious that Theorem 1 implies Theorem 3.

To prove the opposite direction, suppose that Theorem 3 is valid and the conditions of Theorem 1 are satisfied. Take a recognizable ΣX_n-tree T and let $\mathbf{A}$ be the minimal ΣX_n-recognizer for T.

First assume that $T \in \mathbf{Prop}$. Let $\mathbf{B}$ be a connected ΣX_n-recognizer recognizing T such that $T(\mathbf{B}_x) \in \mathbf{Prop(Lang)}$ for all $x \in \hat{T}_\Sigma(X_n)$. Therefore, by (2) in Theorem 1, $\text{Syntm}(T(\mathbf{B}_x)) \in \mathbf{M}$. By Lemma 5, $\text{Syntm}(T(\mathbf{A}_x)) \cong \text{Syntm}(T(\mathbf{B}_x))$, thus $\text{Syntm}(T(\mathbf{A}_x)) \in \mathbf{M}$, which by (2) in Theorem 1 implies $T(\mathbf{A}_x) \in \mathbf{Prop(Lang)}$.

Conversely, suppose that $T(\mathbf{A}_x) \in \mathbf{Prop(Lang)}$ for all $x \in \hat{T}_\Sigma(X_n)$. By (2) in Theorem 1, $\text{Syntm}(T(\mathbf{A}_x)) \in \mathbf{M}$. Let $\mathbf{B}$ be a connected ΣX_n-recognizer with $T(\mathbf{B}) = T(\mathbf{A})$ which satisfies (1) in Theorem 1. By Lemma 5, $\text{Syntm}(T(\mathbf{A}_x))$ and $\text{Syntm}(T(\mathbf{B}_x))$ are isomorphic. Then $\text{Syntm}(T(\mathbf{B}_x)) \in \mathbf{M}$. Therefore, by (2) in Theorem 1, $T(\mathbf{B}_x) \in \mathbf{Prop(Lang)}$. From this, using (1) in Theorem 1, we obtain that $T(\mathbf{A}) = T(\mathbf{B}) \in \mathbf{Prop}$.

We have obtained that the conditions of Theorem 3 are also satisfied. Therefore, **Prop** can be defined by $\mathbf{M}$. ♢

Using a similar proof, one can show

Theorem 6. *Theorems 2 and 4 are equivalent.* ♢

We now show that Theorem 3 is equivalent to

Theorem 7. *Let* **Prop** *be a property of recognizable tree languages and* $\mathbf{M}$ *a class of monoids. Assume that the following conditions are satisfied:*

(1) *For every* ΣX_n*-language* T *and all connected* ΣX_n*-recognizers* $\mathbf{A}$ *with* $T(\mathbf{A}) = T$ *we have*

$$T \in \mathbf{Prop} \Longleftrightarrow (\forall x \in X_n \cup \Sigma_0)(T(\mathbf{A}_x) \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) $\mathbf{Prop}(\mathbf{Lang})$ *can be defined by* $\mathbf{M}$.

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

Then **Prop** *can be defined by* $\mathbf{M}$.

Proof. It is obvious that Theorem 3 implies Theorem 7.

The opposite direction can be shown by the same idea as the second part of the proof of Theorem 5. Suppose that Theorem 7 is valid and the conditions of Theorem 3 are satisfied. Take a recognizable ΣX_n-tree T.

First assume that $T \in \mathbf{Prop}$. Let $\mathbf{B}$ be a connected ΣX_n-recognizer recognizing T. Moreover, let $\mathbf{A}$ be the minimal ΣX_n-recognizer for T. By our assumption, $T(\mathbf{A}_x) \in \mathbf{Prop(Lang)}$ for all $x \in \hat{T}_\Sigma(X_n)$. Then, by (2) in Theorem 3, Syntm$(T(\mathbf{A}_x)) \in \mathbf{M}$. By Lemma 5, Syntm$(T(\mathbf{A}_x)) \cong$ Syntm$(T(\mathbf{B}_x))$, thus Syntm$(T(\mathbf{B}_x)) \in \mathbf{M}$, which by (2) in Theorem 3 implies $T(\mathbf{B}_x) \in \mathbf{Prop(Lang)}$.

Conversely, let $\mathbf{B}$ be a connected ΣX_n-recognizer with $T(\mathbf{B}) = T$ such that $T(\mathbf{B}_x) \in \mathbf{Prop(Lang)}$ for all $x \in \hat{T}_\Sigma(X_n)$. By condition (2) in Theorem 3, Syntm$(T(\mathbf{B}_x)) \in \mathbf{M}$, and thus Syntm$(T(\mathbf{A}_x)) \in \mathbf{M}$ since Syntm$(T(\mathbf{A}_x))$ and Syntm$(T(\mathbf{B}_x))$ are isomorphic. Therefore, again by (2) in Theorem 3, $T(\mathbf{A}_x) \in \mathbf{Prop(Lang)}$. From this, using (1) in Theorem 3, we obtain that $T(\mathbf{B}) = T(\mathbf{A}) \in \mathbf{Prop}$.

We have obtained that the conditions of Theorem 7 are satisfied. Therefore, **Prop** can be defined by **M**. ♢

Summarizing our equivalence results, we have

Theorem 8. *Theorems 1, 3 and 7 are equivalent.* ♢

Using the same technique as in the proof of Theorem 5, one can show that Theorems 2 and 4 are equivalent to

Theorem 9. *Let* **Prop** *be a property of recognizable tree languages. Assume that the following conditions are satisfied:*

(1) *For every ΣX_n-language T and all connected ΣX_n-recognizers $\mathbf{A}$ with $T = T(\mathbf{A})$ we have*

$$T \in \mathbf{Prop} \Longleftrightarrow (\forall x \in X_n \cup \Sigma_0)(T(\mathbf{A}_x) \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) **Prop(Lang)** *can be defined by a class* **S** *of semigroups.*

(3) **S** *is closed under subdirect products and subdirect factors.*

Then **Prop** *can be defined by* **S**.

4 DR tree languages

First of all, we recall several well known concepts from the theory of root-to-frontier tree recognizers.

In what follows, the frequently recurring phrase *deterministic root-to-frontier* is usually abbreviated directly to DR. As before, Σ is a ranked alphabet and X is a (nonempty) frontier alphabet. As usual, in this section we shall suppose that $\Sigma_0 = \emptyset$.

In the study of DR tree languages, which form a proper subclass of all recognizable tree languages, a natural counterpart of syntactic semigroups are syntactic path semigroups introduced in [8]. Thus, for defining classes of DR recognizable tree languages we shall use path semigroups. We have also changed the definition of properties of tree languages defined by tree automata (monotonicity, nilpotency etc) in such a way which is more natural for DR recognizers. To distinguish them from the general definition, we shall use the prefix DR.

A finite *DR Σ-algebra* consists of a non-empty finite set A and a Σ-indexed family of *root-to-frontier operations*

$$\sigma^{\mathcal{A}} : A \longrightarrow A^m \quad (\sigma \in \Sigma_m).$$

Again we write simply $\mathcal{A} = (A, \Sigma)$. A *DR ΣX_n-recognizer* is now defined as a system $\mathbf{A} = (\mathcal{A}, a_0, \mathbf{a})$, where $\mathcal{A} = (A, \Sigma)$ is a finite DR Σ-algebra, $a_0 \in A$ is the *initial state*, and $\mathbf{a} = (A^{(1)}, \ldots, A^{(n)}) \in (\wp A)^n$ is the *final state vector*. ($\wp A$ denotes the power-set of a set A.)

To define the tree language recognized by $\mathbf{A}$, we introduce a mapping $\alpha_{\mathbf{A}}$ of $T_\Sigma(X_n)$ into $\wp A$:

(1) $\alpha_{\mathbf{A}}(x_i) = A^{(i)}$ for $x_i \in X_n$,

(2) $\alpha_{\mathbf{A}}(p) = \{a \in A \mid \sigma^{\mathcal{A}}(a) \in \alpha_{\mathbf{A}}(p_1) \times \ldots \times \alpha_{\mathbf{A}}(p_m)\}$ for $p = \sigma(p_1, \ldots, p_m)$ $(\sigma \in \Sigma_m,\ p_1, \ldots, p_m \in T_\Sigma(X_n))$.

The tree language *recognized* by $\mathbf{A}$ is now defined as the set

$$T(\mathbf{A}) = \{p \in T_\Sigma(X_n) \mid a_0 \in \alpha_{\mathbf{A}}(p)\}.$$

A ΣX_n-tree language is *DR recognizable* if it is recognized by some DR ΣX_n-recognizer. Such tree languages are called *DR tree languages*.

All DR Σ-algebras considered in this paper are supposed to be finite.

Set $\hat{\Sigma} = \bigcup(\{\sigma_1, \ldots, \sigma_m\} \mid \sigma \in \Sigma_m,\ m > 0)$. For any $x \in X_n$ the set $g_x(p) \subseteq \hat{\Sigma}^*$ of *x-paths* in a given ΣX_n-tree p is defined as follows:

(1) $g_x(x) = e$,

(2) $g_x(y) = \emptyset$ for $y \in X_n,\ y \neq x$,

(3) $g_x(p) = \sigma_1 g_x(p_1) \cup \ldots \cup \sigma_m g_x(p_m)$ for $p = \sigma(p_1, \ldots, p_m)$.

For $T \subseteq T_\Sigma(X_n)$ and $x \in X_n$, set $T_x = \bigcup(g_x(p) \mid p \in T)$.

For a DR ΣX_n-recognizer $\mathbf{A} = (\mathcal{A}, a_0, \mathbf{a})$ and $x \in X_n$, now define the recognizer $\mathbf{A}_x = (\hat{\Sigma}, A, a_0, \delta, A^{(i)})$ by $\delta(a, \sigma_j) = \pi_j(\sigma(a))$ $(a \in A,\ \sigma \in \Sigma)$, where $x = x_i$ and π_j is the jth projection of a vector. (Since $\mathbf{A}_x$ are used to recognize words (paths) they are written in the standard form of finite state recognizers.)

We shall use the following obvious result.

Lemma 7. *For all DR recognizable ΣX_n-tree language T and $x \in X_n$, T_x is a recognizable language.* $\diamondsuit$

The *syntactic path congruence* of a ΣX_n-tree language T is the relation on $\hat{\Sigma}^*$ defined by the following condition. For any $w_1, w_2 \in \hat{\Sigma}^*$,

$$w_1 \, \hat{\mu}_T \, w_2 \iff (\forall x \in X)(\forall u, v \in \hat{\Sigma}^*)(uw_1v \in T_x \iff uw_2v \in T_x).$$

The *syntactic path monoid* Synpm(T) of T is $\hat{\Sigma}^*/\hat{\mu}_T$. Denote by the same $\hat{\mu}_T$ the restriction of $\hat{\mu}_T$ to $\hat{\Sigma}^+$. Then $\hat{\Sigma}^+/\hat{\mu}_T$ is called the *syntactic path semigroup* of T and it is denoted by Synps(T).

The following facts are obvious since in both cases $\hat{\mu}_T$ is the intersection of the usual syntactic congruences of the languages T_x $(x \in X)$.

Lemma 8. *For any DR ΣX_n-tree language T, $\hat{\Sigma}^*/\hat{\mu}_T$ is isomorphic to a subdirect product of the syntactic monoids $\hat{\Sigma}^*/\hat{\mu}_{T_x}$ $(x \in X_n)$. Similarly, $\hat{\Sigma}^+/\hat{\mu}_T$ is isomorphic to a subdirect product of the syntactic semigroups $\hat{\Sigma}^+/\hat{\mu}_{T_x}$ $(x \in X_n)$.* ♢

A *DR property* is a class of DR tree languages. We say that a DR property **Prop** *can be path-defined by a class* **M** *of monoids,* if for all DR tree languages T, $T \in$ **Prop** $\iff$ Syntpm$(T) \in$ **M**. Moreover, a DR-property **Prop** *can be path-defined by a class* **S** *of semigroups,* if for all DR tree languages T, $T \in$ **Prop** $\iff$ Syntps$(T) \in$ **S**.

Using Lemma 8, the next result can be proved in the same way as Theorem 1.

Theorem 10. *Let* **Prop** *be a DR property. Assume that the following conditions are satisfied:*

(1) *For every DR ΣX_n-language T there exists a DR ΣX_n-recognizer* **A** *with* $T(\mathbf{A}) = T$ *such that*

$$T \in \mathbf{Prop} \iff (\forall x \in X_n)(T(\mathbf{A}_x) \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) **Prop**(**Lang**) *can be defined by a class* **M** *of monoids.*

(3) **M** *is closed under subdirect products and subdirect factors.*

Then **Prop** *can be path-defined by* **M**. ♢

By the proof of Lemma 7, the above result can be formulated as follows.

Theorem 11. *Let* **Prop** *be a DR property. Assume that the following conditions are satisfied:*

(1) *For every DR ΣX_n-language T,*

$$T \in \mathbf{Prop} \iff (\forall x \in X_n)(T_x \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) **Prop**(**Lang**) *can be defined by a class* **M** *of monoids.*

(3) **M** *is closed under subdirect products and subdirect factors.*

Then **Prop** *can be path-defined by* **M**. ♢

One can also show

Theorem 12. *Let* **Prop** *be a DR property. Assume that the following conditions are satisfied:*

(1) *For every DR* ΣX_n*-language* T,

$$T \in \mathbf{Prop} \iff (\forall x \in X_n)(T_x \in \mathbf{Prop}(\mathbf{Lang})).$$

(2) **Prop(Lang)** *can be defined by a class* **S** *of semigroups.*

(3) **S** *is closed under subdirect products and subdirect factors.*

Then **Prop** *can be path-defined by* **S**.

4.1 DR monotone tree languages

It is said that a DR ΣX_n-recognizer $\mathbf{A} = (\mathcal{A}, a_0, \mathbf{a})$ is *DR monotone* if there exists a partial ordering $\leq$ on A such that $\pi_i(\sigma(a)) \geq a$ for all $\sigma \in \Sigma_m$, $1 \leq i \leq m$ and $a \in A$. Moreover, a tree language $T \subseteq T_\Sigma(X_n)$ is *DR monotone*, if $T = T(\mathbf{A})$ for a DR monotone ΣX_n-recognizer $\mathbf{A}$.

Let S be a semigroup and $s \in S$ an arbitrary element. It is said that $r \in S$ is a *divisor* of s if $s = rt$ or $s = tr$ for some $t \in S$. A subsemigroup S' of S is *closed under divisors* if S' contains all divisors of each of its elements. Moreover, we say that a subsemigroup S' of S is a *right-unit subsemigroup* if there exists an $s \in S$ such that $S' = \{r \in S \mid s = sr\}$. More precisely, in this case S' is called the right-unit subsemigroup of S *belonging* to s.

The class of monoids whose all right-unit subsemigroups are closed under divisors will be denoted by $\mathbf{M}_{\text{cld}}$.

The following result from [3] gives a semigroup-theoretic characterization of monotone languages.

Theorem 13. *A recognizable language* L *is monotone iff every right-unit subsemigroup of the syntactic monoid of* L *is closed under divisors.* ♢

Thus the class of monotone languages is defined by the class $\mathbf{M}_{\text{cld}}$ of monoids. The next result is from [1].

Theorem 14. *The class of all monotone tree languages together with* $\mathbf{M}_{\text{cld}}$ *satisfies the conditions of Theorem 3.* ♢

For DR monotone tree languages we have

Theorem 15. *The class* **Prop** *of all DR monotone tree languages with* $\mathbf{M} = \mathbf{M}_{\text{cld}}$ *satisfies the conditions of Theorem 11.*

Proof. It has been shown in [1] that (2) and (3) are true.

For showing (1), take a DR monotone ΣX_n-language T. For all $x \in X_n$, T_x are monotone. Therefore, $T_x \in \mathbf{Prop}(\mathbf{Lang})$.

Conversely, assume that for all $x_i \in X_n$, T_{x_i} are monotone. Therefore, there are monotone recognizers $\mathbf{B}_i = (\hat{\Sigma}, B_i, b_{i_0}, \delta_i, B_i')$ with partial orderings $\leq_i$ on B_i such that $T(\mathbf{B}_i) = T_{x_i}$. Define the DR ΣX_n-recognizer $\mathbf{B} = (\mathcal{B}, b_0, \mathbf{b})$ in the following way:

(1) $\mathcal{B} = (B, \Sigma)$ where $B = B_1 \times \ldots \times B_n$ and for all $(b_1, \ldots, b_n) \in B$ and $\sigma \in \Sigma_m$,

$$\begin{aligned}&\sigma^{\mathcal{B}}(b_1, \ldots, b_n) = \\ &\qquad ((\delta_1(b_1, \sigma_1), \ldots, \delta_n(b_n, \sigma_1)), \ldots, (\delta_1(b_1, \sigma_m), \ldots, \delta_n(b_n, \sigma_m))).\end{aligned}$$

(2) $b_0 = (b_{1_0} \ldots, b_{n_0})$.

(3) $B^{(i)} = B_1 \times \ldots \times B_{i-1} \times B_i{}' \times B_{i+1} \times \ldots \times B_n$.

It is a routine work to show that $T(\mathbf{B}) = T$. Define the relation $\leq$ on B by

$$((b_1, \ldots, b_n) \leq (b_1', \ldots, b_n')) \Longleftrightarrow ((\forall i \in \{1, \ldots, n\})(b_i \leq_i b_i'))$$

$$((b_1, \ldots, b_n), (b_1', \ldots, b_n') \in B).$$

Easy to show that $\leq$ is a partial ordering and $(b_1, \ldots, b_n) \leq \pi_j(\sigma(b_1, \ldots, b_n))$ for all $(b_1, \ldots, b_n) \in B$, $\sigma \in \Sigma$ and $j \in \{1, \ldots, n\}$. Thus, $T \in \mathbf{Prop}$. $\diamondsuit$

Since the class of DR tree languages is a proper subclass of the class of all tree languages both the class of all monotone tree languages and the class of DR monotone tree languages can be defined by the same class $\mathbf{M}_{\mathrm{cld}}$ of monoids, one could come to the hypothesis that the class of all DR monotone tree languages is the restriction of the class of all monotone tree languages to the class of DR tree languages. However, in [3] it was shown that the class of DR monotone tree languages and that of monotone tree languages are incomparable.

4.2 DR nilpotent tree languages

Let $\mathcal{A} = (A, \Sigma)$ be a DR Σ-algebra, $a \in A$ an element and $p \in T_\Sigma(X_n)$ a tree. Define the word $\overline{\mathrm{fr}}(ap) \in A^*$ in the following way:

1) if $p = x \in X_n$, then $\overline{\mathrm{fr}}(ap) = a$,

2) if $p = \sigma(p_1, \ldots, p_m)$ and $(a_1, \ldots, a_m) = \sigma^{\mathcal{A}}(a)$, then

$$\overline{\mathrm{fr}}(ap) = \overline{\mathrm{fr}}(a_1p_1) \ldots \overline{\mathrm{fr}}(a_mp_m).$$

A DR ΣX_n-algebra $\mathcal{A} = (A, \Sigma)$ is *DR nilpotent* if there are an integer $k \geq 0$ and an element $\bar{a} \in A$ such that for all $a \in A$ and $p \in T_\Sigma(X_n)$ with $\mathrm{mh}(p) \geq k$, $\overline{\mathrm{fr}}(ap) = \bar{a}^l$ for a natural number l. ($\bar{a}$ is called the *nilpotent element* of $\mathcal{A}$ and $\mathrm{mh}(p)$

is the length of the shortest path of p.) A DR ΣX_n-recognizer $\mathbf{A} = (\mathcal{A}, a_0, \mathbf{a})$ is *DR nilpotent* if $\mathcal{A}$ is DR nilpotent. Moreover, a ΣX_n-tree language T is *DR nilpotent* if it can be recognized by a DR nilpotent ΣX_n-recognizer.

A semigroup S is nilpotent if it has a zero-element 0 and there is a non-negative integer k such that $s_1 \dots s_k = 0$ for all $s_1, \dots, s_k \in S$.

The class of all nilpotent semigroups will be denoted by $\mathbf{S}_{\text{nil}}$.

The following result from [12] gives a semigroup-theoretic characterization of nilpotent languages. (See, also [5].)

Theorem 16. *A recognizable language L is nilpotent iff the syntactic semigroup of L is nilpotent.* ♢

Thus the class of nilpotent languages can be defined by the class $\mathbf{S}_{\text{nil}}$ of semigroups.

Theorem 17. *The class of all DR nilpotent tree languages with $\mathbf{S} = \mathbf{S}_{\text{nil}}$ satisfy the conditions of Theorem 12.*

Proof. It has been proved in [1] that (2) and (3) are true.

Condition (1) can be shown in a similar way as (1) in the proof of Theorem 15 by replacing "DR monotone" with "DR nilpotent", taking $(b_1, \dots, b_k)$ to be the nilpotent element if b_i is the nilpotent element of $\mathbf{B}_i$, and disregarding the partial ordering.

4.3 DR definite tree languages

Let S be a semigroup. It is said that S is *right regular* if the equality $ss_I = s_I$ holds in S for any element s and idempotent s_I. The class of all right regular semigroups will be denoted by $\mathbf{S}_{\text{rr}}$

Let $k \geq 0$ be an arbitrary integer. A DR Σ-algebra $\mathcal{A} = (A, \Sigma)$ is *DR k-definite* if $\overline{\text{fr}}(ap) = \overline{\text{fr}}(a'p)$ for all $a, a' \in A$ and $p \in T_\Sigma(X_n)$ with $\text{mh}(p) \geq k$. A DR ΣX_n-recognizer $\mathbf{A} = (\mathcal{A}, a_0, \mathbf{a})$ is *DR k-definite* if $\mathcal{A}$ is DR k-definite. Moreover, a ΣX_n-tree language T is *DR k-definite* if it can be recognized by a DR k-definite ΣX_n-recognizer. Finally, T is *DR definite* if it is DR k-definite for some k.

It is well known that the class of all definite languages can be defined by the class of all right regular semigroups. Thus, condition (2) of Theorem 12 is satisfied by $\mathbf{S}_{\text{rr}}$. We now show

Theorem 18. *The class of all DR definite tree languages with $\mathbf{S} = \mathbf{S}_{\text{rr}}$ satisfies the conditions of Theorem 12.*

Proof. Condition (3) of Theorem 12 is obviously satisfied by $\mathbf{S}_{\text{rr}}$.

It is obvious that if $T \subseteq T_\Sigma(X_n)$ is DR k-definite then so are T_x for all $x \in X_n$. Conversely, assume that T_{x_i} are k_i-definite. There are k_i-definite recognizers $\mathbf{B}_i = (\hat{\Sigma}, B_i, b_{i_0}, \delta_i, B'_i)$ such that $T(\mathbf{B}_i) = T_{x_i}$. Again take the DR ΣX_n-recognizer $\mathbf{B} = (\mathcal{B}, b_0, \mathbf{b})$ obtained by the construction used in the proof of Theorem 15. Let $k = \max(k_i \mid i = 1, \dots, n)$. It is easy to show that $\mathbf{B}$ is k-definite and $T(\mathbf{B}) = T$.

References

[1] Gécseg, F. Classes of tree languages determined by classes of monoids. *International Journal of Foundations of Computer Science,* **18** (2007), 1237-1246.

[2] Gécseg, F. and Imreh, B. On a special class of tree automata. In *2nd Conf. on Automata, Languages and Programming Systems* (Salgótarján, 1988), 141-152.

[3] Gécseg, F. and Imreh, B. On momotone automata and monotone languages. *Journal of Automata, Languages and Combinatorics,* **7** (2002), 71-82.

[4] Gécseg, F. and Imreh, B. On definite and nilpotent DR tree languages. *Journal of Automata, Languages and Combinatorics,* **9** (2004), 55-60.

[5] Gécseg, F. and Peák, I. *Algebraic theory of automata.* (Akadémiai Kiadó, Budapest, 1972).

[6] Gécseg, F. and Steinby, M. *Tree automata.* (Akadémiai Kiadó, Budapest, 1984).

[7] Gécseg, F. and Steinby, M. Tree languages. In *Handbook of Formal Languages, Vol. 3*, eds. G. Rozenberg and A. Salomaa (Springer-Verlag, Berlin, 1997), 1-68.

[8] Gécseg, F. and Steinby, M. Minimal Recognizers and Syntactic Monoids of DR Tree Languages. In Words, Semigroups, & Transductions, World Scientifics, (2001), 155-167.

[9] Heuter, U. Definite tree languages. *Bulletin of the EATCS,* **35** (1988), 137-142.

[10] Pin, J.-E. Syntactic semigroups. In *Handbook of Formal Languages, Vol. 1*, eds. G. Rozenberg and A. Salomaa (Springer-Verlag, Berlin, 1997), 679-746.

[11] Salehi, S. Varieties of tree languages definable by syntactic monoids. *Acta Cybernetica,* **17** (2005), 21-41.

[12] Ševrin, L. N. On some classes of abstract automata (Russian). *Uspehi matem. nauk,* **17:6(108)** (1962), 219.

[13] Steinby, M. A theory of tree language varieties. In *Tree Automata and Languages,* eds. M. Nivat and A. Podelski (Elsevier, Amsterdam, 1992), 5781.

[14] Thomas, W. Logical aspects in the study of tree languages. In *Ninth Colloquium on Trees in Algebra and in Programming* (Cambridge University Press, Cambridge 1984), 270-280.

Received 2nd November 2010