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 $\mathbf{M}$ 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 $\mathbf{M}$. 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 ≥ 0). By a Σ-algebra we mean a pair A = (A, {σ^A | σ ∈ Σ}), where σ^A is an m-ary operation on A if σ ∈ Σ_m. If there will be no danger of confusion then we omit the superscript A and simply write A = (A, Σ). 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 A = (A, Σ), a σ ∈ Σ_m (m > 0), an i (1 ≤ i ≤ m) and a_1, a_{i-1}, a_{i+1}, ..., a_m ∈ A. Then σ(a_1, ..., a_{i-1}, x, a_{i+1}, ..., a_m) is an elementary translation symbol of A. The set of all elementary translation symbols of A will be denoted by ETS(A). In the sequel elementary translation symbols will be considered as unary operational symbols. Moreover, ETalg(A) will denote the unary algebra (A, ETS(A)) with

\[ σ(a_1, ..., a_{i-1}, x, a_{i+1}, ..., a_m)^{ETalg(A)}(a) = σ^A(a_1, ..., a_{i-1}, a, a_{i+1}, ..., a_m) \]

\[ (σ(a_1, ..., a_{i-1}, x, a_{i+1}, ..., a_m) ∈ ETS(A), a ∈ A). \]

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

(i) \( X ⊆ T_Σ(X) \),
(ii) \( σ(p_1, ..., p_m) ∈ T_Σ(X) \) if \( m ≥ 0, σ ∈ Σ_m \) and \( p_1, ..., p_m ∈ T_Σ(X) \), 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, ..., \}, and for every \( n ≥ 0 \), \( X_n \) will denote the subset \{x_1, ..., x_n\} of X. A subset of \( T_Σ(X_n) \) 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 A = (A, Σ) and a tree \( p ∈ T_Σ(X_n) \). Let us define the mapping \( p^A : A^n → A \) in the following way: for any \( a = (a_1, ..., a_n) ∈ A^n \),

(i) if \( p = x_i ∈ X_n \), then \( p^A(a) = a_i \),
(ii) if \( p = σ(p_1, ..., p_m) (σ ∈ Σ_m, p_1, ..., p_m ∈ T_Σ(X_n)) \), then

\[ p^A(a) = σ^A(p_1^A(a), ..., p_m^A(a)). \]
If there is no danger of confusion, then we omit \( A \) in \( pA \).

A \( \Sigma X_n \)-recognizer is a system \( A = (A, \sigma, A') \), where

(i) \( A = (A, \Sigma) \) is an algebra,

(ii) \( \sigma = (a^{(1)}, \ldots, a^{(n)}) \) \((a^{(1)}, \ldots, a^{(n)} \in A) \) 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 \( A \) is connected if \( \{p(a) \mid p \in T_\Sigma(X_n)\} = A \).

If \( \Sigma \) and \( X_n \) are not specified then we speak of a tree recognizer. Furthermore, if \( \Sigma = \Sigma_1 \) and \( n = 1 \), then \( A \) 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 \( T(A) \) recognized by the \( \Sigma X_n \)-recognizer \( A = (A, \sigma, A') \) is given by

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

The class of recognizable tree languages will be denoted by \( \text{Treelang} \), and \( \text{Lang} \) is its subclass consisting of all tree languages recognizable by finite state recognizers.

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

If not otherwise specified, \( A \) will be the \( \Sigma X_n \)-recognizer \((A, \sigma, A')\). Here \( A \) is a \( \Sigma \)-algebra \((A, \Sigma)\), \( \sigma = (a^{(1)}, \ldots, a^{(n)}) \) and \( A' \subseteq A \). Consider a \( \Sigma X_n \)-recognizer \( A \).

For each \( x \in X_n \cup \Sigma_0 \), define the finite state \( \text{ETS}(A) \)-recognizer \( A_x = (A_x, a_x, A'_x) \) in the following way:

\(
\begin{align*}
(1) \quad & a_x = \begin{cases} \ a^{(i)}, & \text{if } x = x_i \ (1 \leq i \leq n), \\ \sigma A', & \text{if } x = \sigma \in \Sigma_0. \end{cases} \\
(2) \quad & A_x = \{p^{\text{ETalg}(A)}(a_x) \mid p \in T_{\text{ETS}(A)}(X_1)\}. \\
(3) \quad & A_x = (A_x, \text{ETS}(A)) \text{ is a subalgebra of } \text{ETalg}(A). \\
(4) \quad & A'_x = A_x \cap A'.
\end{align*}
\)

These \( A_x \) are called translation recognizers of \( A \).

Let \( T_\Sigma(X_n) \) denote the set of all \( \Sigma \)-trees over \( X_n \cup \{\ast\} \) \( (\ast \not\in X_n) \) in which \( \ast \) occurs exactly once. Elements in \( T_\Sigma(X_n) \) are special trees of Thomas [14] and Heuter [9].

Let us define the product \( q \cdot p \) of \( q \in T_\Sigma(X_n) \) and \( p \in T_\Sigma(X_n) \) by \( q \cdot p = p(q) \). (Here and in the sequel, for any \( p \in T_\Sigma(X_n) \) and \( q \in T_\Sigma(X_n) \cup T_\Sigma(X_n), p(q) \) is obtained by replacing the occurrence of \( \ast \) in \( p \) by \( q \) \((p(q) = p(\ast \leftarrow q)) \).) Obviously, under this multiplication \( T_\Sigma(X_n) \) is a monoid with the identity element \( \ast \).
Let $T \subseteq T_\Sigma(X_n)$ be a tree language. Define the binary relation $\mu_T$ on $T_\Sigma(X_n)$ in the following way: for any $p,q \in T_\Sigma(X_n)$,

$$p \equiv q(\mu_T) \iff (\forall p',p'' \in T_\Sigma(X_n), x \in X_n \cup \Sigma_0)((p' \cdot p'')(x) \in T \iff (p' \cdot p'')(x) \in T).$$

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

The restriction of $\mu_T$ to $T_\Sigma(X_n) \setminus \{\ast\}$ will be denoted by the same $\mu_T$. The quotient semigroup $T_\Sigma(X_n) \setminus \{\ast\}/\mu_T$ is the syntactic semigroup of $T$. The syntactic semigroup of $T$ will be denoted by $\text{Synts}(T)$.

We say that a property $\text{Prop}$ of recognizable tree languages can be defined by a class $\text{M}$ of monoids, if for all $T \in \text{Treelang}$, $T \in \text{Prop} \iff \text{Syntm}(T) \in \text{M}$. Similarly, a property $\text{Prop}$ of recognizable tree languages can be defined by a class $\text{S}$ of semigroups, if for all $T \in \text{Treelang}$, $T \in \text{Prop} \iff \text{Synts}(T) \in \text{S}$. For any $\Sigma X_n$-recognizer $A$ and $p \in T_\Sigma(X_n)$, let $p(a)\equiv p(x_1 \leftarrow a^{(1)}, \ldots, x_n \leftarrow a^{(n)})$ and $p(a)(\ast \leftarrow a)$ ($a \in A$), i.e. $p(a)(\ast)$ is obtained from $p$ by replacing the occurrences of $x_i$ by $a^{(i)}$ and that of $\ast$ 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 $\mu_L$ 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 $u'u'' \in L \iff u'vu'' \in L$ holds. As it is well known, $\mu_L$ is a congruence relation on the free monoid $Y^*$, and the quotient monoid $Y^*/\mu_L$ is called the syntactic monoid of $L$. Let the same $\mu_L$ denote the restriction of $\mu_L$ to the semigroup $Y^+ = Y^* \setminus \{e\}$, where $e$ is the empty word. The quotient semigroup $Y^+ / \mu_L$ is the syntactic semigroup of $L$. It is obvious, if finite state recognizers are taken as special tree recognizers, then the 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 $\Sigma X_n$-recognizer. Define the mapping $\epsilon_A : T_\Sigma(X_n) \to T_{\text{ETS}(A)}(\ast)$ in the following way:

1) $\epsilon_A(\ast) = \ast$.

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

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

Since $A$ is connected, $\epsilon_A$ is an onto mapping. If there is no danger of confusion we shall omit $A$ in $\epsilon_A$. 
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 $p, q \in \hat{T}_\Sigma(X_n)$,

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

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 $\Sigma X_n$-language $T$ we obviously have the following two results.

**Lemma 1.** The syntactic monoid $\text{Syntm}(T)$ 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 $\text{Synts}(T)$ is isomorphic to a subdirect product of the semigroups $\hat{T}_\Sigma(X_n) \setminus \{\epsilon\}/\mu_{T, x}$, $x \in X_n \cup \Sigma_0$, where the restriction of $\mu_{T, x}$ to $\hat{T}_\Sigma(X_n) \setminus \{\epsilon\}$ is denoted by the same $\mu_{T, x}$.

We now show

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

$$\hat{T}_\Sigma(X_n)/\mu_{T, x} \cong \text{Syntm}(T(A_x)).$$

**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 $p, q \in \hat{T}_\Sigma(X_n)$,

$$p \equiv q(\mu_{T, x}) \iff \epsilon(p) \equiv \epsilon(q)(\mu_{T(A_x)}).$$

Remember that $\epsilon$ is an onto mapping since $A$ is connected. Thus,

$$p \equiv q(\mu_{T, x}) \implies (\forall r, s \in \hat{T}_\Sigma(X_n))((r \cdot p \cdot s)(x) \in T) \iff (r \cdot q \cdot s)(x) \in T)$$

$$((\forall r, s \in \hat{T}_\Sigma(X_n))((r \cdot p \cdot s)(a_x) \in A') \iff (r \cdot q \cdot s)(a_x) \in A')$$

$$((\forall r, s \in \hat{T}_\Sigma(X_n))((\epsilon(r) \cdot p \cdot s)(a_x) \in A'_\epsilon) \iff (\epsilon(r) \cdot q \cdot s)(a_x) \in A'_\epsilon)$$

$$((\forall r, s \in \hat{T}_\Sigma(X_n))((\epsilon(r) \cdot \epsilon(p) \cdot \epsilon(s))(a_x) \in A'_\epsilon) \iff (\epsilon(r) \cdot \epsilon(q) \cdot \epsilon(s))(a_x) \in A'_\epsilon)$$

$$\epsilon(p) \equiv \epsilon(q)(\mu_{T(A_x)}).$$

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

The following lemma can be proved in a similar way.
Lemma 4. Let \( A \) be an arbitrary connected \( \Sigma X_n \)-recognizer. Then for all \( x \in X_n \cup \Sigma_0 \),
\[
\tilde{T}_\Sigma(X_n) \setminus \{ \ast \}/\mu_{T,x} \cong \text{Synts}(T(A_x)).
\]

Let \( S \) be a class of semigroups. We say that \( S \) is \textit{closed under subdirect products}, if all subdirect products of semigroups from \( S \) with finitely many factors are in \( S \). Moreover, \( S \) is \textit{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 semigroups.

We are now ready to state and prove

Theorem 1. Let \( \text{Prop} \) be a property of recognizable tree languages. Assume that the following conditions are satisfied:

1. For every \( \Sigma X_n \)-language \( T \) there exists a connected \( \Sigma X_n \)-recognizer \( A \) with \( T(A) = T \) such that \( T \in \text{Prop} \iff (\forall x \in X_n \cup \Sigma_0) (T(A_x) \in \text{Prop} \text{(Lang)}) \).

2. \( \text{Prop} \text{(Lang)} \) can be defined by a class \( M \) of monoids.

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

Then \( \text{Prop} \) can be defined by \( M \).

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

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

Conversely, assume that \( T \in \text{Prop} \), and let \( A \) be a connected tree recognizer with \( T = T(A) \) satisfying (1). Then, for each \( x \in X_n \cup \Sigma_0 \), \( T(A_x) \in \text{Prop} \text{(Lang)} \).

Thus, by (2), \( \text{Syntm}(T(A_x)) \in M \). Again, by Lemma 1 and 3, \( \text{Syntm}(T) \) is isomorphic to a subdirect product of \( \text{Syntm}(T(A_x)) \) \( (x \in X_n \cup \Sigma_0) \). Moreover, by (3), \( M \) is closed under subdirect products. Therefore, \( \text{Synt}(T) \in M \). ◊

The next theorem can be proved in a similar way.

Theorem 2. Let \( \text{Prop} \) be a property of recognizable tree languages. Assume that the following conditions are satisfied:

1. For every \( \Sigma X_n \)-language \( T \) there exists a connected \( \Sigma X_n \)-recognizer \( A \) with \( T(A) = T \) such that \( T \in \text{Prop} \iff (\forall x \in X_n \cup \Sigma_0) (T(A_x) \in \text{Prop} \text{(Lang)}) \).
(2) \( \text{Prop}(\text{Lang}) \) can be defined by a class \( S \) of semigroups.

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

Then \( \text{Prop} \) can be defined by \( S \).

In [1] we proved

**Theorem 3.** Let \( \text{Prop} \) be a property of recognizable tree languages. Assume that the following conditions are satisfied.

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

\[ T(A) \in \text{Prop} \iff (\forall x \in X_n \cup \Sigma_0)(T(A_x) \in \text{Prop}(\text{Lang})). \]

(2) \( \text{Prop}(\text{Lang}) \) can be defined by a class \( M \) of monoids.

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

Then \( \text{Prop} \) can be defined by \( M \).

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

**Theorem 4.** Let \( \text{Prop} \) be a property of recognizable tree languages. Assume that the following conditions are satisfied.

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

\[ T(A) \in \text{Prop} \iff (\forall x \in X_n \cup \Sigma_0)(T(A_x) \in \text{Prop}(\text{Lang})). \]

(2) \( \text{Prop}(\text{Lang}) \) can be defined by a class \( S \) of semigroups.

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

Then \( \text{Prop} \) can be defined by \( S \).

We shall need

**Lemma 5.** If \( A \) and \( B \) are equivalent connected \( \Sigma X_n \)-recognizers then for all \( x \in X_n \cup \Sigma_0 \), \( \text{Syntm}(T(A_x)) = \text{Syntm}(T(B_x)) \).

**Proof.** For a \( p \in T_{\Sigma}(X_n) \) set \( \overline{p} = \epsilon_A(p) \) and \( \overline{p} = \epsilon_B(p) \). Let \( p, q \in T_{\Sigma}(X_n) \) and \( x \in X_n \) be arbitrary. We have

\[
(\forall r, s \in T_{\Sigma}(X_n))(((r \cdot s)(x))A_x(a) \in A' \iff ((r \cdot s)(x))A_x(a) \in A') \quad (T(A) = T(B))

(\forall r, s \in T_{\Sigma}(X_n))(((r \cdot s)(x))B_x(b) \in B' \iff ((r \cdot s)(x))B_x(b) \in B') \quad (T(A) = T(B))

(\forall r, s \in T_{\Sigma}(X_n))r \cdot s\cdot A_x(a_x) \in A' \iff r \cdot s\cdot A_x(a_x) \in A')

(\forall r, s \in T_{\Sigma}(X_n))r \cdot s\cdot B_x(b_x) \in B' \iff r \cdot s\cdot B_x(b_x) \in B')

(\forall \overline{r}, \overline{s} \in T_{\Sigma}(X_n))((\overline{r} \cdot \overline{s})A_x(a_x) \in A' \iff (\overline{r} \cdot \overline{s})A_x(a_x) \in A')

(\forall \overline{r}, \overline{s} \in T_{\Sigma}(X_n))((\overline{r} \cdot \overline{s})B_x(b_x) \in B' \iff (\overline{r} \cdot \overline{s})B_x(b_x) \in B')

\[
\overline{p} \equiv \overline{q}(\mu_{A_x}) \iff \overline{p} \equiv \overline{q}(\mu_{B_x}).
\]
Therefore the mapping $\varphi_{A_x} : T \mapsto \varphi_{B_x}$ is an isomorphism between
the monoids $\text{Syntm}(T(A_x))$ and $\text{Syntm}(T(B_x))$. ♦

The following result can be proved in a similar way.

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

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 $\Sigma X_n$-tree $T$ and let $A$ be the minimal $\Sigma X_n$-recognizer for $T$.

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

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

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

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

1. For every $\Sigma X_n$-language $T$ and all connected $\Sigma X_n$-recognizers $A$ with $T(A) = T$ we have

   $T \in \text{Prop} \iff (\forall x \in X_n \cup \Sigma_0)(T(A_x) \in \text{Prop}(\text{Lang})).$

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

3. $M$ is closed under subdirect products and subdirect factors.

Then $\text{Prop}$ can be defined by $M$. 
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 \( \Sigma X_n \)-tree \( T \).

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

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

We have obtained that the conditions of Theorem 7 are satisfied. Therefore, \( \text{Prop} \) 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 \( \text{Prop} \) be a property of recognizable tree languages. Assume that the following conditions are satisfied:

1. For every \( \Sigma X_n \)-language \( T \) and all connected \( \Sigma X_n \)-recognizers \( A \) with \( T = T(A) \) we have
   
   \[ T \in \text{Prop} \iff (\forall x \in X_n \cup \Sigma_0)(T(A_x) \in \text{Prop}(\text{Lang})). \]

2. \( \text{Prop}(\text{Lang}) \) can be defined by a class \( S \) of semigroups.

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

Then \( \text{Prop} \) 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, \( \Sigma \) 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

\[ \sigma^A : A \rightarrow A^m \quad (\sigma \in \Sigma_m). \]

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

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

1. \( \alpha^A(x_i) = A^{(i)} \) for \( x_i \in X_n \).
2. \( \alpha^A(p) = \{ a \in A \mid \sigma^A(a) \in \alpha^A(p_1) \times \ldots \times \alpha^A(p_m) \} \) 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 \( \mathcal{A} \) is now defined as the set

\[ T(\mathcal{A}) = \{ p \in T_\Sigma (X_n) \mid a_0 \in \alpha^A(p) \}. \]

A \( \Sigma \times \Sigma \text{-tree language} \) is DR recognizable if it is recognized by some DR \( \Sigma \times \Sigma \text{-recognizer} \). Such tree languages are called DR tree languages.

All DR \( \Sigma \)-algebras considered in this paper are supposed to be finite.

Set \( \Sigma = \bigcup \{ (\sigma_1, \ldots , \sigma_m) \mid \sigma \in \Sigma_m, m > 0 \} \). For any \( x \in X_n \) the set \( g_x (p) \subseteq \Sigma^* \) of \( x \)-paths in a given \( \Sigma X_n \)-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 \( \Sigma \times \Sigma \text{-recognizer} \( \mathcal{A} = (A, a_0, a) \) and \( x \in X_n \), now define the recognizer \( \mathcal{A}_x = (\Sigma, A, a_0, \delta, A^{(i)}) \) by \( \delta(a, \sigma_j) = \pi_j(\sigma(a)) \) \( (a \in A, \sigma \in \Sigma) \), where \( x = x_i \) and \( \pi_j \) is the \( j \)th projection of a vector. (Since \( \mathcal{A}_x \) 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 \( \Sigma X_n \)-tree language \( T \) and \( x \in X_n \), \( T_x \) is a recognizable language. \( \diamond \)
The syntactic path congruence of a \( \Sigma X_n \)-tree language \( T \) is the relation on \( \hat{\Sigma}^* \) defined by the following condition. For any \( w_1, w_2 \in \hat{\Sigma}^* \),

\[
    w_1 \hat{\mu}_T w_2 \iff (\forall x \in X)(\forall u, v \in \hat{\Sigma}^*)(uw_1 v \in T_x \iff uw_2 v \in T_x).
\]

The syntactic path monoid \( \text{Synpm}(T) \) 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 \( \text{Synps}(T) \).

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 \( \Sigma X_n \)-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 \( \hat{\Sigma}^+/\hat{\mu}_{T_x} \) (\( x \in X_n \)). \( \diamond \)

A DR property is a class of DR tree languages. We say that a DR property \( \text{Prop} \) can be path-defined by a class \( \text{M} \) of monoids, if for all DR tree languages \( T, T \in \text{Prop} \iff \text{Synpm}(T) \in \text{M} \). Moreover, a DR-property \( \text{Prop} \) can be path-defined by a class \( \text{S} \) of semigroups, if for all DR tree languages \( T, T \in \text{Prop} \iff \text{Synps}(T) \in \text{S} \).

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

Theorem 10. Let \( \text{Prop} \) be a DR property. Assume that the following conditions are satisfied:

1. For every DR \( \Sigma X_n \)-language \( T \) there exists a DR \( \Sigma X_n \)-recognizer \( A \) with \( T(A) = T \) such that

\[
    T \in \text{Prop} \iff (\forall x \in X_n)(T(A_x) \in \text{Prop(Lang)}).
\]

2. \( \text{Prop(Lang)} \) can be defined by a class \( \text{M} \) of monoids.

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

Then \( \text{Prop} \) can be path-defined by \( \text{M} \). \( \diamond \)

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

Theorem 11. Let \( \text{Prop} \) be a DR property. Assume that the following conditions are satisfied:

1. For every DR \( \Sigma X_n \)-language \( T \),

\[
    T \in \text{Prop} \iff (\forall x \in X_n)(T_x \in \text{Prop(Lang)}).
\]

2. \( \text{Prop(Lang)} \) can be defined by a class \( \text{M} \) of monoids.

3. \( \text{M} \) 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 $\Sigma X_n$-language $T$,
   \[ T \in \text{Prop} \iff (\forall x \in X_n) (T_x \in \text{Prop}(\text{Lang})). \]

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 $\Sigma X_n$-recognizer $A = (A, a_0, a)$ 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 $T = T(A)$ for a DR monotone $\Sigma X_n$-recognizer $A$.

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 $M_{\text{cld}}$.

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 $M_{\text{cld}}$ of monoids.

The next result is from [1].

**Theorem 14.** The class of all monotone tree languages together with $M_{\text{cld}}$ 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 $M = M_{\text{cld}}$ 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 $\Sigma X_n$-language $T$. For all $x \in X_n$, $T_x$ are monotone. Therefore, $T_x \in \text{Prop} (\text{Lang})$.

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

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

$$\sigma^B(b_1, \ldots, b_n) = ((\delta_1(b_1, \sigma_1), \ldots, \delta_n(b_n, \sigma_1)), \ldots, (\delta_1(b_1, \sigma_m), \ldots, \delta_n(b_n, \sigma_m))).$$

2) $b_0 = (b_{i_0}, \ldots, b_{i_n})$.

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 $T(B) = T$. Define the relation $\leq$ on $B$ by

$$((b_1, \ldots, b_n) \leq (b'_1, \ldots, b'_n)) \iff ((\forall i \in \{1, \ldots, n\})(b_i \leq_i b'_i))$$

$$((b_1, \ldots, b_n), (b'_1, \ldots, b'_n) \in B).$$

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, $T \in \text{Prop}$.  

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 $\text{M}_{\text{clal}}$ 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 $A = (A, \Sigma)$ be a DR $\Sigma$-algebra, $a \in A$ an element and $p \in T_\Sigma(X_n)$ a tree. Define the word $\overline{\text{tr}}(ap) \in A^*$ in the following way:

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

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

$$\overline{\text{tr}}(ap) = \overline{\text{tr}}(a_1p_1) \ldots \overline{\text{tr}}(a_mp_m).$$

A DR $\Sigma X_n$-algebra $A = (A, \Sigma)$ 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 $\text{mh}(p) \geq k$,

$$\overline{\text{tr}}(ap) = \bar{a}^l$$

for a natural number $l$. ($\bar{a}$ is called the nilpotent element of $A$ and $\text{mh}(p)$
is the length of the shortest path of \( p \). A DR \( \Sigma X_n \)-recognizer \( A = (A, a_0, a) \) is DR \( \text{nilpotent} \) if \( A \) is DR nilpotent. Moreover, a \( \Sigma X_n \)-tree language \( T \) is DR \( \text{nilpotent} \) if it can be recognized by a DR nilpotent \( \Sigma X_n \)-recognizer.

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

The class of all nilpotent semigroups will be denoted by \( S_{\text{nil}} \).

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 \( S_{\text{nil}} \) of semigroups.

**Theorem 17.** The class of all DR nilpotent tree languages with \( S = S_{\text{nil}} \) 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 \((b_1, \ldots, b_k)\) to be the nilpotent element if \( b_i \) is the nilpotent element of \( B_i \), 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 \( ssI = sI \) holds in \( S \) for any element \( s \) and idempotent \( sI \). The class of all right regular semigroups will be denoted by \( S_{rr} \).

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

**Theorem 18.** The class of all DR definite tree languages with \( S = S_{rr} \) satisfies the conditions of Theorem 12.

**Proof.** Condition (3) of Theorem 12 is obviously satisfied by \( S_{rr} \).

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 \( T_x \) are \( k_i \)-definite. There are \( k_i \)-definite recognizers \( B_i = (\Sigma, B_i, b_{i0}, \delta_i, B'_i) \) such that \( T(B_i) = T_x \). Again take the DR \( \Sigma X_n \)-recognizer \( B = (B, b_0, b) \) obtained by the construction used in the proof of Theorem 15. Let \( k = \max(k_i \mid i = 1, \ldots, n) \). It is easy to show that \( B \) is \( k \)-definite and \( T(B) = T \).
Classes of Tree Languages and DR Tree Languages

References


Received 2nd November 2010