Acta Cybernetica **17** (2006) 449–457.

On the closedness of nilpotent DR tree languages under Boolean operations*

F. Gécseg† and Gy. Gyurica†

To Professor László Leindler on his 70th birthday

Abstract

This note deals with the closedness of nilpotent deterministic root-to-frontier tree languages with respect to the Boolean operations union, intersection and complementation. Necessary and sufficient conditions are given under which the union of two deterministic tree languages is also deterministic. The paper ends with a characterization of the largest subclass of the class of nilpotent deterministic root-to-frontier tree languages closed under the formation of complements.

1 Introduction

In [3] we introduced nilpotent DR tree languages and characterized them by means of syntactic monoids. For string languages there is another well known characterization of nilpotency: a language L is nilpotent if and only if L or the complement of L is finite. Obviously, this implies that the complements of nilpotent languages are also nilpotent. This result is true for tree languages recognized by nilpotent frontier-to-root tree recognizers (see, [2]), but it does not hold for nilpotent DR tree languages. In this note we study the closedness of nilpotent DR tree languages under the Boolean operations: union, intersection and complementation. We introduce the concepts of union and intersection direct products of DR tree recognizers, which turn out to be very useful in studying unions and intersections of deterministic tree languages. We give necessary and sufficient conditions under which the union of two deterministic tree languages is also deterministic. Moreover, we determine that subclass of the class of nilpotent DR tree languages which is closed under complementation. It will turn out that unary tree languages play an important role in these classes.

Deterministic tree languages have been intensively studied by E. Jurvanen. In [5] she gives several counter examples, among others, for the closedness of deterministic tree languages under union.

*This work has been supported by the Hungarian National Foundation for Scientific Research, Grant T 048786.

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

2 Notions and notations

In this paper Σ is always a *ranked alphabet*, i.e. a finite nonempty set of operation symbols, and for each $m \geq 1$, we denote by Σ_m the set of m-ary symbols in Σ. We assume that there are no nullary symbols in Σ, but instead a finite non-empty *leaf alphabet* X is used. The set $T_\Sigma(X)$ of Σ-*terms* over X is the least set such that

(1) $X \subseteq T_\Sigma(X)$, and

(2) $\sigma(p_1, \ldots, p_m) \in T_\Sigma(X)$, whenever $m \geq 1$, $\sigma \in \Sigma_m$ and $p_1, \ldots, p_m \in T_\Sigma(X)$.

Such terms are regarded as trees in the usual way and we call them ΣX-*trees* (or just *trees*). A ΣX-*tree language* is any subset of $T_\Sigma(X)$.

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

$$\sigma^{\mathcal{A}} : A \longrightarrow A^m \quad (\sigma \in \Sigma),$$

where the arity m is that of $\sigma (\in \Sigma_m)$. We write simply $\mathcal{A} = (A, \Sigma)$. A *DR* ΣX-*recognizer* is a system $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$, where $\mathcal{A} = (A, \Sigma)$ is a finite DR Σ-algebra, $a_0 \in A$ is the *initial state*, and $\alpha : X \to \wp A$ is the *final state assignment*. ($\wp A$ denotes the power set of A.)

We extend α to a mapping $\alpha_{\mathcal{A}} : T_\Sigma(X) \to \wp A$ in the following way:

(1) $x\alpha_{\mathcal{A}} = x\alpha$ for each $x \in X$,

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

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

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

A ΣX-tree language is *DR-recognizable* if it is recognized by some DR ΣX-recognizer.

The *path alphabet* $\hat{\Sigma}$ associated with a ranked alphabet Σ is defined by

$$\hat{\Sigma} = \bigcup_{m>0} \Sigma_m \times \{1, \ldots, m\}.$$

Any element (σ, i) of $\hat{\Sigma}$ is regarded as a letter of an ordinary alphabet, and for convenience we write it as σ_i. Words over $\hat{\Sigma}$ are used for representing paths leading from the root to a leaf in a ΣX-tree. In a letter σ_i appearing in such a representation, the component σ gives the label of a node while the i indicates the direction taken at that node.

For any $x \in X$, the set $g_x(p)$ of *x-paths* in a given ΣX-tree p is defined as follows:

(1) $g_x(x) = \{e\}$, where e is the empty word;

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

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

The mappings g_x are extended to ΣX-tree languages in the natural way, and for any $T \subseteq T_\Sigma(X)$ and $x \in X$, we write $T_x = g_x(T)$. These sets $T_x \subseteq \hat{\Sigma}^*$ are called the *path languages* of T. A ΣX-tree language T is said to be *closed* if $p \in T$ for any ΣX-tree p such that $g_x(p) \subseteq T_x$ for every $x \in X$. As shown in [1] and in [7], a regular tree language is DR-recognizable iff it is closed.

The following result is from [4].

Theorem 1. *For any closed ΣX-tree language the following conditions are equivalent:*

(1) $T \in \mathrm{D}Rec_\Sigma(X)$;

(2) *there is a congruence on $\hat{\Sigma}^*$ of finite index saturating all of the path languages T_x $(x \in X)$;*

(3) μ_T *is of finite index.*

The quotient monoid $PM(T) = \hat{\Sigma}^*/\mu_T$ is called the *syntactic path monoid* of $T(\subseteq T_\Sigma(X))$. As usual, set $\hat{\Sigma}^+ = \hat{\Sigma}^* \setminus \{e\}$, and denote by the same μ_T the restriction of μ_T to $\hat{\Sigma}^+$. Then $PS(T) = \hat{\Sigma}^+/\mu_T$ is called the *syntactic path semigroup* of T. Immediately from Theorem 1 one gets

Corollary 2. *A closed tree language is DR-recognizable iff its syntactic path monoid is finite.*

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

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

2) if $p = \sigma(p_1, \ldots, p_l)$ then $\overline{\mathrm{fr}}(ap) = \overline{\mathrm{fr}}(a_1 p_1) \ldots \overline{\mathrm{fr}}(a_l p_l)$, where $(a_1, \ldots, a_l) = \sigma^{\mathcal{A}}(a)$.

For a ΣX-tree p set $\mathrm{mh}(p) = \min\{|u| : u \in \bigcup(g_x(p) : x \in X)\}$, where $|u|$ denotes the length of u. In words, $\mathrm{mh}(p)$ is the length of the shortest path leading from the root of t to a leaf.

A DR Σ-algebra $\mathcal{A} = (A, \Sigma)$ is *nilpotent* if there are an integer $k \geq 0$ and an element $\bar{a} \in A$ such that for all $a \in A$ and $p \in T_\Sigma(X)$ with $\mathrm{mh}(p) \geq k$, $\overline{\mathrm{fr}}(ap) = \bar{a}^l$ for a natural number l. This $\bar{a}$ is the *nilpotent element* of $\mathcal{A}$ and k is called the *degree of nilpotency* of $\mathcal{A}$. A DR ΣX-recognizer $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ is *nilpotent* if $\mathcal{A}$ is nilpotent. Moreover, a ΣX-tree language T is *nilpotent* if it can be recognized by a nilpotent DR ΣX-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 integer k is the *degree of nilpotency* of S.

For notions and notations not defined here, see [3] and [4].

3 Union

Let $\mathcal{A} = (A, \Sigma)$ and $\mathcal{B} = (B, \Sigma)$ be DR Σ-algebras. Their *direct product* $\mathcal{A} \times \mathcal{B} = (A \times B, \Sigma)$ is given by

$$\sigma^{\mathcal{A}\times\mathcal{B}}((a,b)) = ((\pi_1(\sigma^{\mathcal{A}}(a)), \pi_1(\sigma^{\mathcal{B}}(b))), \ldots, (\pi_m(\sigma^{\mathcal{A}}(a)), \pi_m(\sigma^{\mathcal{B}}(b))))$$

($\sigma \in \Sigma_m,\ (a,b) \in A \times B$). Take two DR ΣX recognizers $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ and $\mathbf{B} = (\mathcal{B}, b_0, \beta)$. Their *union direct product* is

$$\mathbf{A} \times^{\cup} \mathbf{B} = (\mathcal{A} \times \mathcal{B}, (a_0, b_0), \alpha \times^{\cup} \beta),$$

where $(\alpha \times^{\cup} \beta)(x) = (\alpha(x) \times B) \cup (A \times \beta(x))$ $(x \in X)$.

Theorem 3. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two normalized DR* ΣX*-recognizers. Then* $T(\mathbf{A}) \cup T(\mathbf{B})$ *is deterministic if and only if* $T(\mathbf{A}) \cup T(\mathbf{B}) = T(\mathbf{A} \times^{\cup} \mathbf{B})$.

Proof. Assume that $T(\mathbf{A}) \cup T(\mathbf{B})$ is deterministic. Observe that

$$T(\mathbf{A}) \cup T(\mathbf{B}) \subseteq T(\mathbf{A} \times^{\cup} \mathbf{B})$$

holds for arbitrary DR ΣX-recognizer $\mathbf{A}$ and $\mathbf{B}$. Take a $p \in T(\mathbf{A} \times^{\cup} \mathbf{B})$. Then, by the definition of the union product, using the assumption that $\mathbf{A}$ and $\mathbf{B}$ are normalized, we obtain that $g_x(p) \subseteq g_x(T(\mathbf{A}) \cup T(\mathbf{B}))$ for each $x \in X$. Therefore, p is in the closure of $T(\mathbf{A}) \cup T(\mathbf{B})$. However, since $T(\mathbf{A}) \cup T(\mathbf{B})$ is deterministic, it is closed, i.e it coincides with its closure. Therefore, $p \in T(\mathbf{A}) \cup T(\mathbf{B})$.

The converse statement is obvious, since $\mathbf{A} \times^{\cup} \mathbf{B}$ is deterministic. □

We show that in order to study whether the union of two given nilpotent DR ΣX-languages is nilpotent, it is enough to check if their union is deterministic. For this, we need

Lemma 4. *Let* $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ *be a nilpotent DR* ΣX*-recognizer. There exists a normalized nilpotent DR* ΣX*-recognizer* $\mathbf{B} = (\mathcal{B}, b_0, \beta)$ *with* $T(\mathbf{A}) = T(\mathbf{B})$.

Proof. Assume that $\mathbf{A}$ is nilpotent of degree k with the nilpotent element $\bar{a}$. Normalize $\mathbf{A}$ in the following way: if $\sigma(a)$ $(\sigma \in \Sigma,\ a \in A)$ contains a 0-state and $\bar{a}$ is a 0-state then replace it by $\sigma(a) = (\bar{a}, \ldots, \bar{a})$. (Observe that none of the states is a 0-state if $\bar{a}$ is not a 0-state.) Let us denote by $\mathbf{A}^* = (\mathcal{A}^*, a_0, \alpha)$ the resultant recognizer. Then $\mathbf{A}^*$ is normalized, deterministic and $T(\mathbf{A}^*) = T(\mathbf{A})$ (see, p. 115 in [4]). It remains to show that $\mathbf{A}^*$ is nilpotent. It is enough to deal with the case when $\bar{a}$ is a 0-state. Let $a \in A$ be a state and p a tree with $\mathrm{mh}(p) \geq k$. The computing of p in $\mathbf{A}$ and $\mathbf{A}^*$ starting in a coincides up to the point when $\mathbf{A}$ arrives at a 0-state. At this node $\mathbf{A}^*$ will be in state $\bar{a}$ and it remains there during the computing of the subtree belonging to this node. Therefore, $\mathbf{A}^*$ is nilpotent also of degree k with the nilpotent element $\bar{a}$. □

Theorem 5. *Let* $S, T \subseteq T_{\Sigma}(X)$ *be two nilpotent DR tree languages. Then* $S \cup T$ *is nilpotent if and only if it is deterministic.*

Proof. If $S \cup T$ is nilpotent then it is deterministic by definition.

Conversely, assume that $S \cup T$ is deterministic. Let $S = T(\mathbf{A})$ and $T = T(\mathbf{B})$ where $\mathbf{A}$ and $\mathbf{B}$ are normalized nilpotent DR recognizers such that the degree of nilpotency of $\mathbf{A}$ is k and that of $\mathbf{B}$ is l. By Lemma 4, such $\mathbf{A}$ and $\mathbf{B}$ exist. Moreover, by Theorem 3, $S \cup T = T(\mathbf{A} \times^{\cup} \mathbf{B})$. It can be checked in an obvious way that $\mathbf{A} \times^{\cup} \mathbf{B}$ is nilpotent with degree of nilpotency $\max\{k, l\}$. □

Next we give necessary and sufficient conditions under which the union of two deterministic tree languages is not deterministic.

Theorem 6. *Let S and T be DR ΣX-languages. Then $S \cup T$ is not deterministic if and only if there are a tree $p \in T_\Sigma(X)$, two variables $x, y \in X$ and two different paths $u \in g_x(p)$ and $v \in g_y(p)$ such that $u \in g_x(S)$ and $u \notin g_x(T)$, and $v \in g_y(T)$ and $v \notin g_y(S)$.*

Proof. Assume that $S \cup T$ is not deterministic. Let $\mathbf{A}$ and $\mathbf{B}$ be normalized DR ΣX-recognizers with $S = T(\mathbf{A})$ and $T = T(\mathbf{B})$. Since $S \cup T$ is not deterministic, there is a tree $p \in T_\Sigma(X)$ such that $p \in T(\mathbf{A} \times^{\cup} \mathbf{B})$, $p \notin T(\mathbf{A})$ and $p \notin T(\mathbf{B})$. Therefore, for some $x, y \in X$, $u \in g_x(p)$ and $v \in g_y(p)$ we have $u \in g_x(S) \setminus g_x(T)$, $v \in g_y(T) \setminus g_y(S)$ and $u \neq v$.

Conversely, assume that the conclusions of the theorem hold. Denote by w the maximal initial segment of u and v. Then u and v are of form $u = w\sigma_i u'$ and $v = w\sigma_j v'$, and $i \neq j$. Since $u \in g_x(S)$, there is a $q \in S$ with $u \in g_x(q)$. Similarly, there is a $q' \in T$ with $v \in g_y(q')$. Let r be the tree which is obtained from q' by replacing its subtree at $w\sigma_i$ by the subtree of q at $w\sigma_i$. Obviously, r is not in $S \cup T$, however it is in the closure of $S \cup T$. Therefore, $S \cup T$ is not deterministic. □

Obviously, the trees p satisfying the conditions of the previous theorem are not unary. Therefore, from Theorem 6 we directly obtain

Corollary 7. *Let S and T be two DR ΣX-languages. If $S \setminus T \subseteq T_{\Sigma_1}(X)$ or $T \setminus S \subseteq T_{\Sigma_1}(X)$, then $S \cup T$ is deterministic.* □

This corollary, by Theorem 5, implies

Corollary 8. *Let S and T be two nilpotent DR ΣX-languages. If one of them differs from the other one only in unary trees then $S \cup T$ is nilpotent.* □

Let $p \in T_\Sigma(X) \setminus X$ be a tree. Then $\mathrm{root}(p) = \sigma$ if $p = \sigma(p_1, \ldots, p_m)$. For a tree language $T \subseteq T_\Sigma(X)$, set $\mathrm{root}(T) = \{\mathrm{root}(p) | p \in T \setminus X\}$.

The following result directly follows from Theorems 6 and 5.

Corollary 9. *Let S and T be nilpotent DR ΣX-languages. If*

$$\mathrm{root}(S) \cap (\mathrm{root}(T) = \emptyset,$$

then $S \cup T$ is nilpotent. □

Later we shall use the following obvious result.

Lemma 10. *Let $S, T \subseteq T_\Sigma(X)$ be nilpotent DR tree languages. Then for each $x \in X$, if both $g_x(S)$ and $g_x(T)$ are infinite then $g_x(S) \setminus g_x(T)$ and $g_x(T) \setminus g_x(S)$ are finite.* □

Corollary 11. *Let S and T be nilpotent DR ΣX-languages such that $S \setminus T \not\subseteq T_{\Sigma_1}(X)$. If for an $x \in X$, $g_x(T) \setminus g_x(S)$ is infinite, then $S \cup T$ is not nilpotent.*

Proof. Take a $p \in S \setminus T$ with $p \notin T_{\Sigma_1}(X)$. Then there exist a variable $y \in X$ and a path $u \in g_y(p)$ such that $u \notin g_y(T)$. Assume that the degree of nilpotency of S is k and that of T is l. Then for the variable x satisfying the condition of the corollary, all the trees $r \in T_\Sigma(X)$ of the form $\bar{\mathrm{fr}}(r) = x^t$ with $\mathrm{mh}(p) \geq l$ are in T. Let q be the tree which is obtained from p by replacing each leaf, except for the leaf at u, by a tree r for which $\mathrm{mh}(r) \geq \max\{k, l\}$ and $\bar{\mathrm{fr}}(r) = x^t$ under some t. By Lemma 10, $g_x(S)$ is finite, thus q obviously satisfies the conditions of Theorem 6. Therefore, $S \cup T$ is not nilpotent. □

From Corollary 11 we directly obtain

Corollary 12. *Let S and T be two nilpotent ΣX-languages. If S is finite, T is infinite and $S \setminus T \not\subseteq T_{\Sigma_1}(X)$, then $S \cup T$ is not nilpotent.*

4 Intersection

Let $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ and $\mathbf{B} = (\mathcal{B}, b_0, \beta)$ be DR ΣX-recognizers. Their *intersection direct product* is

$$\mathbf{A} \times^\cap \mathbf{B} = (\mathcal{A} \times \mathcal{B}, (a_0, b_0), \alpha \times^\cap \beta),$$

where $(\alpha \times^\cap \beta)(x) = \alpha(x) \times \beta(x)$ $(x \in X)$.

Theorem 13. *Let $\mathbf{A}$ and $\mathbf{B}$ be DR ΣX-recognizers. Then $T(\mathbf{A}) \cap T(\mathbf{B}) = T(\mathbf{A} \times^\cap \mathbf{B})$.*

Proof. Obvious. □

It is also obvious that the intersection direct product of nilpotent DR tree recognizers is nilpotent. Thus, from Theorem 13, we obtain

Theorem 14. *The class of the nilpotent DR ΣX-languages is closed under intersection.*

5 Complementation

Let $T \subseteq T_\Sigma(X)$ be a tree language. The complement $T_\Sigma(X) \setminus T$ of T will be denoted by $\mathrm{c}(T)$. Moreover, for all tree languages $T \subseteq T_\Sigma(X)$ and variables $x \in X$, $T(x)$ will stand for $T \cap T_{\Sigma_1}(\{x\})$, i.e. $T(x)$ consists of all (unary) trees from T whose leaves are x.

Lemma 15. *Assume that $\Sigma_i = \emptyset$ for all $i > 1$. Then a tree language $T \subseteq T_\Sigma(X)$ is nilpotent if and only if $T(x)$ or $\mathrm{c}(T)(x)$ is finite for each $x \in X$.*

Proof. If T is nilpotent, then obviously, each $T(x)$ $(x \in X)$ is nilpotent. Since in the unary case we can apply the well known characterization of nilpotent string languages, $T(x)$ or $\mathrm{c}(T)(x)$ is finite.

Conversely, assume that the conclusions of our lemma hold. Observe that in this special case the path language $g_x(S)$ of S and the path language $g_x(S(x))$ of $S(x)$ coincide for all $x \in X$ and $S \subseteq T_\Sigma(X)$. Moreover, again by a well known classical characterization of nilpotent string languages (see, [6]), the syntactic semigroups of nilpotent string languages are nilpotent. Therefore, the syntactic semigroup of $g_x(T)$ and that of $g_x(\mathrm{c}(T))$ are nilpotent. This, by the proof of Theorem 5 in [3], implies that both T and $\mathrm{c}(T)$ are nilpotent. □

From the above theorem we directly obtain

Corollary 16. *If $\Sigma_i = \emptyset$ for all $i > 1$, then a tree language $T \subseteq T_\Sigma(X)$ is nilpotent if and only if its complement $\mathrm{c}(T)$ is nilpotent.* □

Lemma 17. *Suppose that Σ contains at least one operational symbol with arity greater than 1, and let $T \subseteq T_{\Sigma_1}(X)$ be a tree language. Then T is nilpotent if and only if it is finite. Moreover, if T is nilpotent then so is its complement $\mathrm{c}(T)$.*

Proof. Assume that T is infinite and nilpotent, and that it is recognized by the nilpotent DR ΣX-recognizer $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ with degree of nilpotency k. Let $\bar{a}$ be the nilpotent element of $\mathbf{A}$. Since T is infinite, there exists a $p(x) \in T$ with $\mathrm{h}(p) \geq k$. Therefore, $\bar{a} \in \alpha(x)$. Thus, all trees $q \in T_\Sigma(\{x\})$ with $\mathrm{mh}(q) \geq k$ are also in T, which contradicts the assumption that $T \subseteq T_{\Sigma_1}(X)$.

Conversely, assume that $T \subseteq T_{\Sigma_1}(X)$ is finite. Construct a DR ΣX-recognizer $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ in the following way. Let $k = \max\{\mathrm{h}(p) \mid p \in T\}$. Set

$$A = \{u \in \hat{\Sigma}^* \mid |u| \leq k\} \cup \{*\}.$$

Moreover, for all $m > 0$, $\sigma \in \Sigma_m$ and $u \in \hat{\Sigma}^*$, let

$$\sigma^{\mathcal{A}}(u) = (u\sigma_1, \ldots, u\sigma_m),$$

if $|u| < k$, and

$$\sigma^{\mathcal{A}}(u) = \sigma^{\mathcal{A}}(*) = (*, \ldots, *),$$

otherwise. Finally, let $a_0 = e$, and $\alpha(x) = g_x(T)$ $(x \in X)$. It is obvious that $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ is nilpotent and $T = T(\mathbf{A})$. It is also clear that $\mathbf{A'} = (\mathcal{A}, a_0, \alpha')$ with $\alpha'(x) = A \setminus \alpha(x)$ $(x \in X)$ recognizes $\mathrm{c}(T)$. □

Lemma 18. *Suppose that Σ contains at least one operational symbol with arity greater than 1, and let $T \subseteq T_\Sigma(X)$ be an infinite nilpotent tree language. If $\mathrm{c}(T) \not\subseteq T_{\Sigma_1}(X)$, then $\mathrm{c}(T)$ is not nilpotent.*

Proof. Suppose that T can be recognized by a nilpotent DR ΣX-recognizer $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ with degree of nilpotency k. Let $\bar{a}$ be the nilpotent element of $\mathbf{A}$. Since T is infinite, there is a $\bar{z} \in X$ such that $\bar{a} \in \alpha(\bar{z})$. Assume that $\mathrm{c}(T) \nsubseteq T_{\Sigma_1}(X)$ is nilpotent and can be recognized by the nilpotent DR ΣX-recognizer $\mathbf{B} = (\mathcal{B}, b_0, \beta)$ with the nilpotent element $\bar{b}$. Suppose that the degree of nilpotency of $\mathbf{B}$ l. Take a tree $p \in \mathrm{c}(T)$ with $p \notin T_{\Sigma_1}(X)$. Then for some (not necessarily different) variables $x, y \in X$, there are different paths $u \in g_x(p)$ and $v \in g_y(p)$ such that $a_0u \notin \alpha(x)$ or $a_0v \notin \alpha(y)$. Assume that $a_0u \notin \alpha(x)$. Suppose that $l \geq k$. Replace in p the variable y at v with an arbitrary $r \in T_\Sigma(\{z\})$ $(z \in X)$ of height greater than or equal to l, and denote the resultant tree by q. Obviously, $q \in \mathrm{c}(T)$. Then $\bar{b} \in \beta(z)$ for all $z \in X$. Thus, for every tree $t \in T_\Sigma(X)$ with $\mathrm{mh}(t) \geq l$ we have $t \in \mathrm{c}(T)$. Moreover, by our assumptions, every tree $t \in T_\Sigma(\{\bar{z}\})$ with $\mathrm{mh}(t) \geq l$ is also in T, which is a contradiction. The case $k > l$ can be treated in a similar way. □

Using Lemma 18, we give a simple example showing that there exists a nilpotent tree language whose complement is not nilpotent.

Example 19. Let $\Sigma = \sigma_2 = \{\sigma\}$ and $X = \{x, y\}$. Take the DR Σ-algebra $\mathcal{A} = (A, \Sigma)$ with $A = \{a_0\}$ and $\sigma^{\mathcal{A}}(a_0) = (a_0, a_0)$. Finally, let $\mathbf{A} = (\mathcal{A}, a_0, \alpha)$ be the ΣX-recognizer, where $\alpha(x) = \{a_0\}$ and $\alpha(y) = \emptyset$. Obviously, $\mathbf{A}$ is nilpotent and $T(\mathbf{A}) = T_\Sigma(\{x\})$. Since $T(\mathbf{A})$ is infinite and $\Sigma_1 = \emptyset$, by Lemma 18, $\mathrm{c}(T(\mathbf{A}))$ is not nilpotent.

We now state a result characterizing those DR tree languages T for which both T and $\mathrm{c}(T)$ are nilpotent. The case $\Sigma = \Sigma_1$ is settled by Lemma 15.

Theorem 20. *Suppose that Σ contains at least one operational symbol with arity greater than 1, and let $T \subseteq T_\Sigma(X)$ be a tree language. Then T and $\mathrm{c}(T)$ are simultaneously nilpotent if and only if one of the following two statements is true:*

(i) $T \subseteq T_{\Sigma_1}(X)$ *and* T *is finite.*

(ii) $\mathrm{c}(T) \subseteq T_{\Sigma_1}(X)$ *and* $\mathrm{c}(T)$ *is finite.*

Proof. If (i) holds, then, by Lemma 17, both T and $\mathrm{c}(T)$ are nilpotent. Case (ii) can be treated in the same way.

Conversely, assume that T and $\mathrm{c}(T)$ are simultaneously nilpotent. If T is finite, then $\mathrm{c}(T)$ is infinite. Thus, by Lemma 18, $T \subseteq T_{\Sigma_1}(X)$. Therefore (i) holds. If T is infinite, then $\mathrm{c}(T) \subseteq T_{\Sigma_1}(X)$ by Lemma 18. From this, using the assumption that $\mathrm{c}(T)$ is nilpotent, by Lemma 17, we obtain that $\mathrm{c}(T)$ is finite. Therefore, (ii) holds. □

References

[1] Courcelle, B.: A representation of trees by languages I. *Theoretical Computer Science* **6** (1978), 255-279.

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

[3] Gécseg, F. and Imreh, B.: On definite and nilpotent DR tree languages, *Journal of Automata, Languages and Combinatorics,* **9:1** (2004), 55-60.

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

[5] Jurvanen, E.: The Boolean closure of DR-recognizable tree languages, *Acta Cybernetica,* **10** (1992), 255-272.

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

[7] Virágh, J.: Deterministic ascending tree automata I. *Acta Cybernetica* **5** (1980), 33-42.

Received April, 2005