Acta Cybernetica **19** (2009) 231–244.

On Nilpotent Languages and Their Characterization by Regular Expressions

György Gyurica*

Abstract

Tree languages recognized by deterministic root-to-frontier recognizers are also called DR-languages. The concept of generalized R-chain languages was introduced by the author in his paper *On monotone languages and their characterization by regular expressions* (Acta Cybernetica, **18** (2007), 117-134.) and it has turned out that the monotone DR-languages are exactly those languages that can be given by generalized R-chain languages. In this paper we give a similar characterization for nilpotent DR-languages by means of plain R-chain languages. Also a regular expression based characterization is given for nilpotent string languages.

1 Introduction

Monotone string and DR-languages were characterized by means of regular expressions in [4] and [9]. For string languages it was shown in [4] that the class of monotone languages and the class of languages represented by finite unions of seminormal chain languages are the same. In case of DR-languages it is turned out in [9] that the class of monotone DR-languages and the class of languages represented by generalized R-chain languages are the same. In this paper our goal was to find a similar characterization for both nilpotent string and DR-languages, however our main focus was directed towards nilpotent DR-languages because nilpotent string languages were already studied in the past intensively.

After introducing the necessary concepts we brought in the concept of plain chain languages by which we characterized nilpotent string languages. Later, a similar chain-like structure was introduced for DR-languages, that were given the name plain R-chain languages. It has turned out that a DR-language is nilpotent if and only if it can be given as a plain R-chain language. The proof required some additional results, among which one states a condition by which the class of DR-languages is closed under x-product. We have also defined when a DR-language is path complete or x-terminating. These concepts turned out to be very handy if we want to characterize the x-product of nilpotent DR-languages.

*Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary E-mail: gyurica@inf.u-szeged.hu

DOI: 10.14232/actacyb.19.1.2009.15

For notions and notation not defined in this paper we refer the reader to [8] and [9].

2 Nilpotent string languages

Let X be a finite nonempty alphabet. X^* denotes the set of all words over X. The *length* of a word $u \in X^*$ is denoted by $|u|$ which is the number of occurrences of letters from X in u. The empty word is denoted by e. As usual, N will denote the set of natural numbers, i.e. $N = \{1, 2, \ldots\}$. $X^+(= X^* \setminus \{e\})$ denotes the set of words with length greater than 0. The set of words no longer than $k \in N$ is $X^{*,k} = \{u \in X^* : |u| \leq k\}$. A word $w \in X^*$ is a *prefix* of a word $u \in X^*$ if there is a word $v \in X^*$ for which $u = wv$. Moreover, we say that $w \in X^*$ is a *proper* prefix of $u \in X^*$ if w is a prefix of u and $|w| < |u|$.

A system $\mathbf{A} = (A, X, \delta, a_0, A')$ is called an *X-recognizer*, if A is a finite set of *states*, X is the *input alphabet*, $\delta : A \times X \to A$ is the *next-state function*, $a_0 \in A$ is the *initial state*, and $A' \subseteq A$ is the set of *final states*. The next-state function can be extended to a function $\delta^* : A \times X^* \to A$, where $\delta^*(a, e) = a$, and $\delta^*(a, xu) = \delta^*(\delta(a, x), u)$ for every $a \in A,\ x \in X,\ u \in X^*$. If there is no danger of confusion, instead of $\delta^*(a, u)$ we can use the notation $\delta(a, u)$ or simply au.

The *language* $L(\mathbf{A})$ *recognized by* $\mathbf{A}$ is given by

$$L(\mathbf{A}) = \{u \in X^* \mid a_0u \in A'\}.$$

A language $L \subseteq X^*$ is called *regular* or *recognizable* if it can be recognized by an X-recognizer.

An X-recognizer $\mathbf{A} = (A, X, \delta, a_0, A')$ is *nilpotent* if there is an integer $k \geq 0$ and a state $\bar{a} \in A$ such that $au = \bar{a}$ for all $a \in A,\ u \in X^*$ with $|u| \geq k$. The state $\bar{a}$ is called the *nilpotent element* of $\mathbf{A}$, and the least k for which the above condition holds is called the *degree of nilpotency* of $\mathbf{A}$. A language $L \subseteq X^*$ is *nilpotent* if there is a nilpotent X-recognizer $\mathbf{A}$ for which $L(\mathbf{A}) = L$.

Remark 1. Nilpotent element has various names in the terminology like *absorbent element* (see [2]), or *trap state*, etc. We will use the term *nilpotent element* in the rest of this paper.

The complement of a language $L \subseteq X^*$ is defined as $X^* \setminus L$ and will be denoted by $c(L)$ in the sequel. The following lemma is well-known (see [5]).

Lemma 1. *A language $L \subseteq X^*$ is nilpotent if and only if L or $c(L)$ is finite.*

We introduce some chain-like languages that were used in [4] and [9]. A language $L \subseteq X^*$ is *fundamental*, if $L = Y^*$ for a $Y \subseteq X$. A language $L \subseteq X^*$ is a *chain language*, if L can be given in the form $L = L_0x_1L_1x_2 \ldots x_{k-1}L_{k-1}x_kL_k$, where $x_1, \ldots, x_k \in X$ and every L_i $(0 \leq i \leq k)$ is a product of fundamental languages. We will call a chain language $L = L_0x_1L_1x_2 \ldots x_{k-1}L_{k-1}x_kL_k$ *plain* , if $L_i = \{e\}$ for every $0 \leq i < k$ and $L_k = Y^*$, where $Y = \emptyset$ or $Y = X$. We will use small Greek letters like $\zeta, \eta, \theta, \ldots$ to denote plain chain languages.

Let $\zeta = x_1x_2 \dots x_k L_k$ be a plain chain language. Clearly, ζ is finite if $L_k = \{e\}$, and ζ is infinite if $L_k = X^*$. Furthermore, the *length* of ζ is defined as $|\zeta| = k$. We consider the plain chain language $\zeta' = x_1x_2 \dots x_j$ as a *prefix* of ζ if either $j > k$ with $x_{k+1} \dots x_j \in L_k$ or $1 \leq j \leq k$. Note that every word of X^* can be considered as a finite plain chain language.

The following two remarks are trivial.

Remark 2. The languages $\{e\}$ and X^* are nilpotent.

Remark 3. Every finite language can be given as a union of finitely many plain chain languages.

Lemma 2. *Let $L \subseteq X^*$ be an infinite language that is given as a union of plain chain languages $\zeta_1, \dots, \zeta_l$ $(l \in N)$. If for all $u \in X^*$ there is an $i \in \{1, \dots, l\}$ such that u is a prefix of ζ_i, then L is nilpotent.*

Proof. Let us suppose that the conditions of the lemma hold. We construct a recognizer $\mathbf{A} = (X^{*,k} \cup \{\bar{a}\}, X, \delta, \{e\}, A')$ where $k = max\{|\zeta_i| : 1 \leq i \leq l\}$. For every $a \in A$ and $x \in X$, we define the next-state function δ as follows:

$$\delta(a, x) = \begin{cases} ax, & \text{if } a \in X^{*,k-1}, \\ \bar{a}, & \text{if } a \in X^{*,k} \text{ and } |a| = k, \\ \bar{a}, & \text{if } a = \bar{a}. \end{cases}$$

To define the set of final states A', let $\bar{a} \in A'$. Moreover, let $x_1 \dots x_j \in A'$ for every $x_1 \dots x_j L_j \in \{\zeta_1, \dots, \zeta_l\}$, and if $L_j = X^*$ then for every $u \in X^{*,k-j}$ let $x_1 \dots x_j u \in A'$.

Let us now take a word $u \in L$ and let $\zeta \in \{\zeta_1, \dots, \zeta_l\}$ be the shortest plain chain language for which u is a prefix of ζ. If $|u| > k$ then $a_0 u = \bar{a}$ hence $u \in L(\mathbf{A})$. If $|u| \leq k$ then $|\zeta| \leq |u|$ holds since u is represented by ζ in L, therefore by the construction of A' we get $u \in L(\mathbf{A})$. Let us now take a word $w \in L(\mathbf{A})$. The construction of A' implies that there is a plain chain language $\zeta \in \{\zeta_1, \dots, \zeta_l\}$ for which w is a prefix of ζ and $|\zeta| \leq |w|$. Since ζ takes part in the representation of L, we get $w \in L$. Thus $L = L(\mathbf{A})$. It is obvious that $ev = \bar{a}$ for any word $v \in X^*$ with $|v| > k$, therefore L is nilpotent with the nilpotent element $\bar{a}$ and with the degree of nilpotency of $k + 1$. □

Let $\mathbf{A} = (A, X, \delta_{\mathbf{A}}, a_0, A')$ and $\mathbf{B} = (B, X, \delta_{\mathbf{B}}, b_0, B')$ be X-recognizers. The *direct product* of $\mathbf{A}$ and $\mathbf{B}$ is defined as the X-recognizer $\mathbf{A} \times \mathbf{B} = (A \times B, X, \delta, (a_0, b_0), F)$, where $F \subseteq A \times B$ and δ is defined as $\delta((a, b), x) = (\delta_{\mathbf{A}}(a, x), \delta_{\mathbf{B}}(b, x))$ for all $a \in A, b \in B$ and $x \in X$. Let τ be a mapping from A onto B. We say that τ is *homomorphism* of $\mathbf{A}$ onto $\mathbf{B}$ if $\tau(a_0) = b_0$, $\tau^{-1}(B') = A'$ and $\tau(\delta_{\mathbf{A}}(a, x)) = \delta_{\mathbf{B}}(\tau(a), x)$ for every $a \in A$ and $x \in X$. In this case $\mathbf{B}$ is the *homomorphic image* of $\mathbf{A}$ beside τ.

The following properties of nilpotent recognizers are well-known.

Lemma 3. *The direct products and homomorphic images of nilpotent recognizers are also nilpotent.*

Proof. It is obvious that the class of nilpotent recognizers is closed under direct products (see [5]).

To prove that the class of nilpotent recognizers is closed under the homomorphic images, let the recognizer $\mathbf{B} = (B, X, \delta_{\mathbf{B}}, b_0, B')$ be the homomorphic image of the nilpotent recognizer $\mathbf{A} = (A, X, \delta_{\mathbf{A}}, a_0, A')$ under a homomorphism $\tau : A \to B$. Let the state $\bar{a} \in A$ be the nilpotent element of $\mathbf{A}$ and let k be the degree of nilpotency of $\mathbf{A}$. Let $\tau(\bar{a}) = \bar{b}$. Taking any word $u \in X^*$ for which $|u| \geq k$ and an element $a \in A$ we get that $au = \bar{a}$, and thus $\tau(a)u = \tau(\bar{a})$. As every element of B has an inverse image in A we can state that $\mathbf{B}$ is nilpotent with the degree of nilpotency of k and with the nilpotent element $\bar{b}$. □

It is a widely used fact that the unions and the intersections of languages recognized by given recognizers can be recognized by direct products of these recognizers (see [5]). Also it is clear that the complement of a nilpotent language is also nilpotent (again, see [5]). Thus we get

Corollary 1. *The family of nilpotent languages is closed under union, intersection and complement.*

Let $\mathbf{A} = (A, X, \delta_{\mathbf{A}}, a_0, A')$ be an X-recognizer. The X-recognizer $\mathbf{B} = (B, X, \delta_{\mathbf{B}}, b_0, B')$ is the *connected subrecognizer* of $\mathbf{A}$, if $B = \{a_0 u | u \in X^*\}$, $B' = A' \cap B$, $a_0 = b_0$ and $\delta_{\mathbf{B}}(b, x) = \delta_{\mathbf{A}}(b, x)$ for every $b \in B, x \in X$.

Lemma 4. *The connected subrecognizer of a nilpotent recognizer is nilpotent.*

Proof. Let $\mathbf{A} = (A, X, \delta_{\mathbf{A}}, a_0, A')$ be a nilpotent X-recognizer with the nilpotent element of $\bar{a}$ and with the degree of nilpotency of k. Also let $\mathbf{B} = (B, X, \delta_{\mathbf{B}}, b_0, B')$ be the connected subrecognizer of $\mathbf{A}$. By the definitions of connected subrecognizer and nilpotent X-recognizer we get that $\bar{a} \in B$. Again, by the definition of connected subrecognizer we obtain that $b_0 u = \bar{a}$ for every word $u \in X^*$ with $|u| \geq k$. Thus $\mathbf{B}$ is nilpotent. □

It is also a well-known fact that the minimal recognizer recognizing a language L is a homomorphic image of the connected subrecognizer of any recognizer recognizing L (see [8]). Thus we have

Corollary 2. *A language is nilpotent iff the minimal recognizer recognizing it is nilpotent.*

Before we continue studying nilpotent languages, we observe some basic correlations between nilpotent and monotone languages.

An X-recognizer $\mathbf{A} = (A, X, \delta, a_0, A')$ is *monotone* if there is a partial ordering $\leq$ on A such that $a \leq \delta(a, x)$ holds for all $a \in A$ and $x \in X$. It is obvious that for all $a \in A$ and $u \in X^*$, $a \leq au$ holds, too. A language $L \subseteq X^*$ is *monotone* if $L = L(\mathbf{A})$ for a monotone X-recognizer $\mathbf{A}$. Later we will use the fact that every partial ordering on a finite set can be extended to a linear ordering. For more details on monotone languages we refer the reader to [4].

The following property of nilpotent languages is well-known (see [10]).

Lemma 5. *Every nilpotent language is monotone.*

Proof. Let $L \subseteq X^*$ be a nilpotent language and let $\mathbf{A} = (A, X, \delta, a_0, A')$ be a nilpotent recognizer for which $L = L(\mathbf{A})$. Moreover, let us suppose that L is not monotone. This means that there are states $a, b \in A$ with $a \neq b$ and words $u, v \in X^*$ such that $au = b$ and $bv = a$. Let k be the degree of nilpotency of $\mathbf{A}$. Then $a(uv)^k = a$ and $b(vu)^k = b$, which contradicts the assumption that $\mathbf{A}$ is nilpotent with degree of nilpotency of k. □

Remark 4. In the proof of Lemma 5 we relied on the assumption that there are states a and b and words u and v such that $a \neq b$, $au = b$ and $bv = a$. If there is no such a pair of states a and b, then the relation defined by $a' \leq a'u$ for every $a' \in A$ and $u \in X^*$, is a monotone order.

From Lemma 5 we easily get

Corollary 3. *Let $\mathbf{A} = (A, X, \delta, a_0, A')$ be a nilpotent recognizer. There is a linear ordering $\leq$ on A such that $a \leq ax$ holds for all $a \in A$ and $x \in X$.*

The converse of Lemma 5 does not hold as the following example shows.

Example 1. It is easy to see that $L = ab^*$ is monotone, but is not nilpotent.

In the sequel, we will characterize nilpotent languages by means of plain chain languages.

Lemma 6. *Every nilpotent language $L \subseteq X^*$ can be given as a union of finitely many plain chain languages $\zeta_1, \ldots, \zeta_l$ $(l \in N)$, where in case of infinite L it holds that for all $u \in X^*$ there is an $i \in \{1, \ldots, l\}$ such that u is a prefix of ζ_i.*

Proof. Let $L \subseteq X^*$ be a nilpotent language. If L is finite, then by Remark 3 L can be given as a union of finitely many plain chain languages. If L is infinite, then let $\mathbf{A} = (A, X, \delta, a_0, A')$ be the minimal nilpotent recognizer for which $L = L(\mathbf{A})$. If A is singleton, then $L(\mathbf{A}) = X^*$, thus L is a plain chain language. Let us now assume that $A = \{a_0, \ldots, a_n\}$ $(n > 0)$, and let the state a_n be the nilpotent element of $\mathbf{A}$. Using Corollary 3, we have a linear ordering $\leq$ on A such that $a_0 \leq \ldots \leq a_n$ holds. Since $\mathbf{A}$ is nilpotent and minimal, $a \neq a_n$ implies $ax \neq a$ for all $a \in A$ and $x \in X$. Let us define the recognizer $\mathbf{A}_{i,j} = (A, X, \delta, a_i, \{a_j\})$ for all $0 \leq i, j \leq n$. Furthermore, for all $x \in X$, let us define the recognizer $\mathbf{A}_x = (A, X, \delta, a_0, A_x)$, where $A_x = \{a \in A \setminus \{a_n\} \mid ax = a_n\}$. Using the recognizers defined above we can write $L(\mathbf{A}_{0,n}) = \bigcup_{x \in X} L(\mathbf{A}_x)xL(\mathbf{A}_{n,n})$. Since $L(\mathbf{A}_x)$ is finite and $L(\mathbf{A}_{n,n}) = X^*$, we get that $L(\mathbf{A}_{0,n})$ is given as a finite union of plain chain languages. Using the languages $L(\mathbf{A}_{i,j})$, we can give L as

$$L = L(\mathbf{A}) = \bigcup_{a_m \in A'} L(\mathbf{A}_{0,m}) = \bigcup_{a_m \in A' \setminus \{a_n\}} L(\mathbf{A}_{0,m}) \cup \bigcup_{x \in X} L(\mathbf{A}_x)xL(\mathbf{A}_{n,n}),$$

where every $L(\mathbf{A}_{0,m})$ is finite $(0 \leq m < n)$, hence by Remark 3 we gave L as the union of finitely many plain chain languages. Let us now take a word $u \in X^*$ and let $w \in X^*$ be any word for which $|uw| \geq n$. Since $a_0uw = a_n$ there is a plain chain language ζ in the above representation of $L(\mathbf{A}_{0,n})$ for which u is the prefix of ζ. $\square$

From Lemma 1, Lemma 2 and Lemma 6 we directly obtain

Theorem 1. *Let $L \subseteq X^*$ be a regular language. L is nilpotent iff L can be given as a union of finitely many plain chain languages $\zeta_1, \ldots, \zeta_l$ $(l \in N)$, where in case of infinite L it holds that for all $u \in X^*$ there is an $i \in \{1, \ldots, l\}$ such that u is a prefix of ζ_i.*

3 Nilpotent DR-languages

A finite nonempty set of operational symbols is called *ranked alphabet* and will be denoted by Σ in this paper. The subset of all m-ary operational symbols will be denoted by Σ_m $(\subseteq \Sigma)$, $m > 0$. We shall exclude the case $m = 0$, so it is supposed that $\Sigma_0 = \emptyset$ in the sequel.

Let X be a set of variables. The set $T_\Sigma(X)$ of ΣX*-trees* 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 $p_1, \ldots, p_m \in T_\Sigma(X)$,

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

In the rest of this paper X will stand for the countable set $\{x_1, x_2, \ldots\}$, and for every $n \geq 0$, X_n will denote the subset $\{x_1, \ldots, x_n\} \subseteq X$. The power set of the set S will be denoted by $\mathfrak{p}(S)$.

A *deterministic root-to-frontier* Σ*-algebra* (or *DR* Σ*-algebra* for short) is a pair $\mathcal{A} = (A, \Sigma)$, where A is a nonempty set and Σ is a ranked alphabet. Every $\sigma \in \Sigma_m$ is represented as a mapping $\sigma^{\mathcal{A}} : A \to A^m$. We call $\mathcal{A}$ *finite*, if A is finite.

A system $\mathfrak{A} = (\mathcal{A}, a_0, \mathbf{a})$ will represent a *deterministic root-to-frontier* ΣX_n*-recognizer* (or a *DR* ΣX_n*-recognizer* for short), where $\mathcal{A} = (A, \Sigma)$ is a finite DR Σ-algebra, $a_0 \in A$ is the *initial state*, and $\mathbf{a} = (A^{(1)}, \ldots, A^{(n)}) \in \mathfrak{p}(A)^n$ is the *final state vector*. If Σ or X_n is not specified, we speak of *DR-recognizers*.

Let $\mathfrak{A} = (\mathcal{A}, a_0, \mathbf{a})$ be a DR ΣX_n-recognizer. Let us define the mapping $\alpha : T_\Sigma(X_n) \to \mathfrak{p}(A)$ as usual. For every $p \in T_\Sigma(X_n)$

(i) if $p = x_i \in X_n$, then $\alpha(p) = A^{(i)}$,

(ii) if $p = \sigma(p_1, \ldots, p_m)$, then $\alpha(p) = \{a \in A \mid \sigma^{\mathcal{A}}(a) \in \alpha(p_1) \times \ldots \times \alpha(p_m)\}$.

The *tree language recognized* by $\mathfrak{A}$ is denoted by $T(\mathfrak{A})$ and is given by

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

Tree languages recognized by DR-recognizers are also called *DR-languages.*

Let $\mathfrak{A}$ be a DR ΣX_n-recognizer and $a \in A$ one of its states. The tree language recognized by $\mathfrak{A}$ from the state a is defined by

$$T(\mathfrak{A}, a) = \{\ p \in T_\Sigma(X_n) \mid a \in \alpha(p)\}.$$

A state a is called 0-*state* if $T(\mathfrak{A}, a) = \emptyset$. $\mathfrak{A}$ is called *normalized* if for all $\sigma \in \Sigma_m$ and $a \in A$ it holds that each component of $\sigma^{\mathcal{A}}(a)$ is a 0-state or no component of $\sigma^{\mathcal{A}}(a)$ is a 0-state. Moreover, $\mathfrak{A}$ is called *reduced* if for all states $a, b \in A$ it holds that $a \neq b$ implies $T(\mathfrak{A}, a) \neq T(\mathfrak{A}, b)$. It is a well-known fact that every DR-language can be recognized by a normalized and reduced DR-recognizer. For more details we refer the reader to [6], [7] and [8].

Now we define the ordinary alphabet $\hat{\Sigma}$ corresponding to the ranked alphabet Σ. For all $\sigma, \tau \in \Sigma$, let

(i) $\hat{\Sigma}_\sigma = \{\sigma_1, \ldots, \sigma_m\}$, if $\sigma \in \Sigma_m$ $(m > 0)$, and

(ii) $\hat{\Sigma}_\sigma \cap \hat{\Sigma}_\tau = \emptyset$, if $\sigma \neq \tau$.

We define $\hat{\Sigma}$ as $\hat{\Sigma} = \bigcup\, (\hat{\Sigma}_\sigma \mid \sigma \in \Sigma)$.

For each $x \in X_n$, the set $g_x(t)$ of x-*paths* of a tree $t \in T_\Sigma(X_n)$ is defined as follows:

(i) $g_x(x) = \{e\}$,

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

(iii) $g_x(t) = \sigma_1 g_x(t_1) \cup \ldots \cup \sigma_m g_x(t_m)$ for $t = \sigma(t_1, \ldots, t_m)$, $\sigma \in \Sigma_m$, $t_i \in T_\Sigma(X_n)$, $1 \leq i \leq m$, $m > 0$.

The mappings g_x are extended to ΣX_n-tree languages in the natural way, that is, for any tree language $T \subseteq T_\Sigma(X_n)$ and variable $x \in X_n$, let $g_x(T) = \bigcup_{t \in T} g_x(t)$. The sets $g_x(T) \subseteq \hat{\Sigma}^*$ are also denoted by T_x and are called the *path languages* of T. Moreover, let us define $g(T)$ as $g(T) = \bigcup_{x \in X} T_x$. A tree language $T \subseteq T_\Sigma(X_n)$ is said to be *closed* if a tree $t \in T_\Sigma(X_n)$ is in T if and only if $g_x(t) \subseteq T_x$ for all $x \in X_n$. It is a well-known result that a regular tree language is DR-recognizable if and only if it is closed (cf. [1] and [11]).

For any integer $n \in N$ and sets $S_1, \ldots, S_n$, let $\pi_i : S_1 \times \ldots \times S_n \to S_i$ be the i-th projection, that is, $\pi_i(s_1, \ldots, s_i, \ldots, s_n) = s_i$ for all $s_i \in S_i$ and $1 \leq i \leq n$. Let Σ be a ranked alphabet, and let $\hat{\Sigma}$ be the alphabet corresponding to it. Let $\mathcal{A} = (A, \Sigma)$ be a DR Σ-algebra. For every $u \in \hat{\Sigma}^*$, the mapping $u^{\mathcal{A}} : A \to A$ is defined as follows:

(i) If $u = e$, then $au^{\mathcal{A}} = a$, and

(ii) if $u = \sigma_j v$, then $au^{\mathcal{A}} = \pi_j(\sigma(a))v^{\mathcal{A}}$ for all $a \in A$, $\sigma \in \Sigma_m$, $v \in \hat{\Sigma}^*$, and $j \in \{1, \ldots, m\}$.

The mapping defined above can be extended to subsets of $\hat{\Sigma}^*$ in the natural way. In the rest of this paper we will omit the superscript $\mathcal{A}$ in $u^{\mathcal{A}}$ if the DR Σ-algebra $\mathcal{A}$ inducing $u^{\mathcal{A}}$ is obvious.

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 $au = \bar{a}$ for all $a \in A$ and $u \in \hat{\Sigma}^*$ with $|u| \geq k$. The state $\bar{a}$ is called the *nilpotent element* of $\mathcal{A}$ and the least k for which the above condition holds is called the *degree of nilpotency* of $\mathcal{A}$. A DR ΣX_n-recognizer $\mathfrak{A} = (\mathcal{A}, a_0, \mathbf{a})$ is *nilpotent* if the underlying DR Σ-algebra $\mathcal{A}$ is nilpotent. Finally, a ΣX_n-tree language T is *nilpotent* if it can be recognized by a nilpotent DR ΣX_n-recognizer.

Remark 5. A different definition of nilpotent DR Σ-algebras and a typical characterization can be found in [3]. We will show that the two definitions define the same set of DR Σ-algebras.

Let $\mathcal{A} = (A, \Sigma)$ be a DR Σ-algebra, let $a \in A$ be an element and let $t \in T_\Sigma(X_n)$ be a tree. We define the word $\overline{\mathrm{fr}}(at) \in A^*$ as follows:

(i) if $t \in X$, then $\overline{\mathrm{fr}}(at) = a$,

(ii) if $t = \sigma(t_1, \ldots, t_m)$, then $\overline{\mathrm{fr}}(at) = \overline{\mathrm{fr}}(a_1t_1) \ldots \overline{\mathrm{fr}}(a_mt_m)$, where $\sigma \in \Sigma_m$, $\sigma^{\mathcal{A}}(a) = (a_1, \ldots, a_m)$, $t_1, \ldots, t_m \in T_\Sigma(X_n)$, $m > 0$.

For any tree $t \in T_\Sigma(X_n)$, let $\mathrm{mh}(t) = min\{|u| : u \in g(t)\}$, that is, $\mathrm{mh}(t)$ is the length of the shortest path leading from the root of t to a leaf. Now we recall the definition of nilpotent DR Σ-algebra from [3]. 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 $t \in T_\Sigma(X_n)$ with $\mathrm{mh}(t) \geq k$, $\overline{\mathrm{fr}}(at) = \bar{a}^l$ for a natural number l. This $\bar{a}$ is called the *nilpotent element* of $\mathcal{A}$ and the least k for which the above condition holds is called the *degree of nilpotency* of $\mathcal{A}$.

Lemma 7. *The two definitions of nilpotent DR languages above define the same set of DR Σ-algebras.*

Proof. Let $\mathcal{A} = (A, \Sigma)$ be a DR Σ-algebra, let k be an integer, and let $\bar{a} \in A$ be an element such that for any $a \in A$ and $t \in T_\Sigma(X_n)$ with $\mathrm{mh}(t) \geq k$ we have $\overline{\mathrm{fr}}(at) = \bar{a}^l$ for a natural number l. Let us now take a word $u \in \hat{\Sigma}^*$ such that $|u| \geq k$. By taking any tree $p \in T_\Sigma(X_n)$ for which u is the shortest path in $g(p)$ we have $\overline{\mathrm{fr}}(ap) = \bar{a}^{l'}$ for a natural number l'. That means $au = \bar{a}$.

Conversely, let $k \geq 0$ be an integer and $\bar{a} \in A$ a state such that for every $a \in A$ and $u \in \hat{\Sigma}^*$ with $|u| \geq k$, $au = \bar{a}$ holds. Let us now take a tree $p \in T_\Sigma(X_n)$ for which $\mathrm{mh}(p) \geq k$. Since every path in $g(p)$ is at least k long, we have $\overline{\mathrm{fr}}(ap) = \bar{a}^l$ for a natural number l. Therefore the two definitions define the same set of DR Σ-algebras. $\square$

A DR Σ-algebra $\mathcal{A} = (A, \Sigma)$ is called *monotone* if there is a partial ordering $\leq$ on A such that $a \leq \pi_i(\sigma(a))$ for all $a \in A$, $\sigma \in \Sigma_m$ and $1 \leq i \leq m$. Moreover, we say that a DR ΣX_n-recognizer $\mathfrak{A}$ is a *monotone DR ΣX_n-recognizer* if the underlying

DR Σ-algebra $\mathcal{A}$ is monotone. Finally, a language $T \subseteq T_\Sigma(X_n)$ is *monotone*, if $T = T(\mathfrak{A})$ for a monotone DR ΣX_n-recognizer $\mathfrak{A}$.

As in the string case, there is a basic correlation between the nilpotent and monotone DR-languages that we state in

Lemma 8. *Every nilpotent DR-language is monotone.*

Corollary 4. *Let $\mathfrak{A} = (\mathcal{A}, a_0, \mathbf{a})$ be a nilpotent DR ΣX_nrecognizer where $\mathcal{A} = (A, \Sigma)$. There exists a linear ordering $\leq$ on A such that $a \leq \pi_i(\sigma(a))$ holds for all $a \in A$, $\sigma \in \Sigma_m$ and $i \in \{1, \ldots, m\}$.*

Similarly to the string case, the converse of Lemma 8 does not hold.

4 Simple approach

Before we continue, we have to clarify some details regarding some particular operations on tree languages. The *σ-product* of ΣX_n-tree languages $T_1, \ldots, T_m$ is the tree language $\sigma(T_1, \ldots, T_m) = \{\sigma(t_1, \ldots, t_m) \mid t_i \in T_i,\ 1 \leq i \leq m\}$, where $m > 0$ and $\sigma \in \Sigma_m$. We assume that the reader is familiar with the operations of union, x-product and x-iteration. In the sequel, we use the operation of x-product in right-to-left manner, that is, for any tree languages $S, T \subseteq T_\Sigma(X_n)$ the x-product $T \cdot_x S$ is interpreted as a tree language in which the trees are obtained by taking a tree s from S and replacing every leaf symbol x in s by a tree from T. Note that different occurrences of x may be replaced by different trees from T. We will also assume that $T \cdot_y R \cdot_x S$ always means $T \cdot_y (R \cdot_x S)$ for any variables $x, y \in X_n$ and tree languages $S, R, T \subseteq T_\Sigma(X_n)$.

Let Σ be a ranked alphabet and let X_n be a set of variables. The set $RE(\Sigma X_n)$ of all *regular ΣX_n-expressions* and the tree language $T(\eta)$ *represented* by $\eta \in RE(\Sigma X_n)$ are defined in parallel as follows:

- $\emptyset \in RE(\Sigma X_n)$, $\qquad T(\emptyset) = \emptyset$,
- $\forall x \in X_n : x \in RE(\Sigma X_n)$, $\qquad T(x) = \{x\}$,

If $\sigma \in \Sigma_m$, $\eta_1, \eta_2, \ldots, \eta_m \in RE(\Sigma X_n)$, $x \in X_n$, $m > 0$, then

- $(\eta_1) + (\eta_2) \in RE(\Sigma X_n)$, $\qquad T((\eta_1) + (\eta_2)) = T(\eta_1) \cup T(\eta_2)$,
- $(\eta_2) \cdot_x (\eta_1) \in RE(\Sigma X_n)$, $\qquad T((\eta_2) \cdot_x (\eta_1)) = T(\eta_2) \cdot_x T(\eta_1)$,
- $(\eta_1)^{*,x} \in RE(\Sigma X_n)$, $\qquad T((\eta_1)^{*,x}) = T(\eta_1)^{*,x}$,
- $\sigma(\eta_1, \ldots, \eta_m) \in RE(\Sigma X_n)$, $\qquad T(\sigma(\eta_1, \ldots, \eta_m)) = \sigma(T(\eta_1), \ldots, T(\eta_m))$.

Some parentheses can be omitted from regular ΣX_n-expressions, if a precedence relation is assumed between the operations of σ-product, x-iteration, x-product, and union in the given order.

Let $\mathfrak{A} = (\mathcal{A}, a_0, \mathbf{a})$ be a nilpotent DR ΣX_n-recognizer, where $\mathcal{A} = (A, \Sigma)$, $A = \{a_0, \ldots, a_k\}$ and $\mathbf{a} = (A^{(1)}, \ldots, A^{(n)})$. Due to Corollary 4 we assume that $a_0 \leq$

$a_1 \leq \ldots \leq a_k$ holds and we can also suppose that a_k is the nilpotent element of $\mathfrak{A}$. Let $\Xi_k = \{\xi_0, \ldots, \xi_k\}$ be a set of auxiliary variables for which $X_n \cap \Xi_k = \emptyset$ holds, and let $\phi : A \to \Xi_k$ be a bijective mapping defined by $\phi(a_i) = \xi_i$ for every $0 \leq i \leq k$. Since every nilpotent DR-language is monotone, we may recall the *trivial regular expression belonging to* $\mathfrak{A}$ ($\eta_\mathfrak{A}$ for short), which is defined in [9] as follows:

$$\eta_\mathfrak{A} = \eta_k \cdot_{\xi_k} \eta_{k-1} \cdot_{\xi_{k-1}} \cdots \cdot_{\xi_1} \eta_0,$$

where for each $i = 0, \ldots, k$,

$$\eta_i = (p_1^i + \cdots + p_{l_i}^i + y_1^i + \cdots + y_{r_i}^i) \cdot_{\xi_i} (t_1^i + \cdots + t_{j_i}^i)^{*,\xi_i},$$

and where

1) $y_1^i, \ldots, y_{r_i}^i$ are all the elements of the set $\{x_z \in X_n | \; a_i \in A^{(z)}, 1 \leq z \leq n\}$,

2) $p_s^i = \sigma(\xi_{i_1}, \ldots, \xi_{i_m})$ for such $\sigma \in \Sigma_m$ and $\xi_{i_v} \in \Xi_k$ $(1 \leq v \leq m)$ that $\sigma(a_i) = (\phi^{-1}(\xi_{i_1}), \ldots, \phi^{-1}(\xi_{i_m}))$ and $a_i \notin \bigcup_{1 \leq v \leq m} \{\pi_v(\sigma(a_i))\}$ hold for every $s \in \{1, \ldots, l_i\}$,

3) $t_s^i = \sigma(\xi_{i_1}, \ldots, \xi_{i_m})$ for such $\sigma \in \Sigma_m$ and $\xi_{i_v} \in \Xi_k$ $(1 \leq v \leq m)$ that $\sigma(a_i) = (\phi^{-1}(\xi_{i_1}), \ldots, \phi^{-1}(\xi_{i_m}))$ and $a_i \in \bigcup_{1 \leq v \leq m} \{\pi_v(\sigma(a_i))\}$ hold for every $s \in \{1, \ldots, j_i\}$,

4) $| \{p_1^i, \ldots, p_{l_i}^i\} | + | \{t_1^i, \ldots, t_{j_i}^i\} | = | \Sigma |$.

In each η_i the part $(p_1^i + \cdots + p_{l_i}^i + y_1^i + \cdots + y_{r_i}^i)$ is called the *terminating part* of η_i, furthermore, the part $(t_1^i + \cdots + t_{j_i}^i)^{*,\xi_i}$ is called the *iterating part* of η_i. The expressions of the form $\eta_k \cdot_{\xi_k} \ldots \eta_1 \cdot_{\xi_1} \eta_0$ are called *chains.*

Let us now observe the regular $\Sigma(X_n \cup \Xi_k)$-expression $\eta_\mathfrak{A}$ that is detailed above. It is obvious that in each η_i $(0 \leq i < k)$ the iterating part is empty because there is no symbol $\sigma \in \Sigma_m$ and state $a \in A \setminus \{a_k\}$ for which $a \in \bigcup_{1 \leq v \leq m} \{\pi_v(\sigma(a))\}$. Thus we can omit these iterating parts from $\eta_\mathfrak{A}$. By these omissions we simplified the trivial regular expression belonging to $\mathfrak{A}$, and we will call the result the *plain regular expression belonging to* $\mathfrak{A}$ (denoted by $\zeta_\mathfrak{A}$).

5 Characterization

Let $S \subseteq T_\Sigma(X_n)$ be a tree language and let $p \in T_\Sigma(X_n)$ be a tree. The *height* $height(p)$, *root* $root(p)$, *leaves* $leaves(p)$ and the set of *subtrees* $Sub(p)$ of the tree p are defined as follows:

(i) If $p \in X_n$, then $height(p) = 0$, $root(p) = p$, $leaves(p) = \{p\}$, and $Sub(p) = \{p\}$.

(ii) If $p = \sigma(t_1, \ldots, t_m)$, $\sigma \in \Sigma_m$, $t_i \in T_\Sigma(X_n)$, $1 \leq i \leq m$, $m > 0$, then $height(p) = 1 + max\{height(t_i) : 1 \leq i \leq m\}$, $root(p) = \sigma$, $leaves(p) = \bigcup_{1 \leq i \leq m} leaves(t_i)$, and $Sub(p) = \{p\} \cup \bigcup_{1 \leq i \leq m}(Sub(t_i))$.

The above functions (except height) are extended from trees to tree languages as follows: $root(S) = \{root(p) \mid p \in S\}$, $leaves(S) = \bigcup_{p \in S} leaves(p)$, and $Sub(S) = \bigcup_{p \in S} Sub(p)$.

For any tree language $S \subseteq T_\Sigma(X_n)$, let the set of *operational symbols appearing in* S be defined as $root(Sub(S)) \setminus X_n$ and let it be denoted by Σ_S. For any language $S \subseteq T_\Sigma(X_n)$ and variable $x \in X_n$, let $\Sigma_{S,x}$ denote the set $\{\sigma \in \Sigma_m \mid \exists u \in g_x(S),\ \exists v \in \hat{\Sigma}^*,\ \exists i \in \{1, \ldots, m\} \ :\ u\sigma_i v \in g(S),\ m > 0\}$.

Later we will use the following lemma.

Lemma 9. *Every finite DR-language is nilpotent*

Proof. Let T be a finite DR-language and let us assume that the DR ΣX_n-recognizer $\mathfrak{A}$ recognizes T. It is obvious that $\mathfrak{A}$ is nilpotent with the degree of nilpotency of $1 + max\{height(t) : t \in T\}$. □

The following lemma is similar to Theorem 18 in [9] with the only difference that monotonicity is not included.

Lemma 10. *Let* S *and* T *be DR-languages, and let* $x \in X_n$. *If* $root(T) \cap \Sigma_{S,x} = \emptyset$, *then* $T \cdot_x S$ *is deterministic.*

Proof. The proof is the same as the proof of Theorem 18 in [9] except that we have to omit monotonicity from the conditions and conclusion. □

We say that a tree language $S \subseteq T_\Sigma(X_n)$ is *path complete* if for every word $u \in g(S)$ and for every prefix $w = w_1 \ldots w_{l-1} w_l$ of u, the word $w_1 \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(S)$ for all $\bar{w}, w_1, \ldots, w_l \in \hat{\Sigma}$, $l \in N$.

Lemma 11. *Let* $x \in X_n$ *be a variable and let* $S \subseteq T_\Sigma(X_n)$, $T \subseteq T_\Sigma(X_n)$, *and* $T \cdot_x S$ *be DR-languages. If* S *and* T *are path complete, then so is* $T \cdot_x S$.

Proof. Let the conditions of the lemma hold. Let us take a word u from $g(T \cdot_x S)$ and take a prefix $w = w_1 \ldots w_{l-1} w_l$ of u where $w_1, \ldots, w_l \in \hat{\Sigma}$ and $l \in N$. Let $\bar{w} \in \hat{\Sigma}$ be also arbitrarily chosen. If $u \in g(S)$ then $w_1 \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(S)$ because S is path complete, and so $w_1 \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(T \cdot_x S)$. If $u = u_S u_T$ where $u_S \in g_x(S)$ and $u_T \in g(T)$, then we differentiate 3 cases:

(i) If w is a prefix of u_S then $w_1 \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(S)$ because S is path complete. Thus $w_1 \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(T \cdot_x S)$.

(ii) If $u_S = e$ then $w_1 \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(T)$ since T is path complete. Hence $w_1 \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(T \cdot_x S)$.

(iii) If u_S is a prefix of w then there is an integer $i \in N$ such that $u_S = w_1 \ldots w_i$. In this case $w_{i+1} \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(T)$. Since $u_S \in g_x(S)$ $w_1 \ldots w_i w_{i+1} \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(T \cdot_x S)$.

Since in every case $w_1 \ldots w_{l-1}\bar{w}$ is a prefix of a word from $g(T \cdot_x S)$, we proved that $T \cdot_x S$ is path complete. □

For any variable $x \in X_n$, a tree language $S \subseteq T_\Sigma(X_n)$ is said to be *x-terminating* if the following condition holds. For every $u \in g(S)$, if u is not a proper prefix of any $w \in g(S)$, then $u \in g_x(S)$.

Theorem 2. *Let $x_i \in X_n$ be a variable and let $S \subseteq T_\Sigma(X_n)$ and $T \subseteq T_\Sigma(X_n)$ be nilpotent DR-languages. If $root(T) \cap \Sigma_{S,x_i} = \emptyset$ and S is finite, path complete and x_i-terminating, then $T \cdot_{x_i} S$ is nilpotent.*

Proof. Let the conditions of the theorem hold. Let $\mathfrak{A} = (\mathcal{A}, a_0, \mathbf{a})$ and $\mathfrak{B} = (\mathcal{B}, b_0, \mathbf{b})$ be reduced, connected and normalized nilpotent DR ΣX_n-recognizers, where $\mathcal{A} = (A, \Sigma)$, $\mathbf{a} = (A^{(1)}, \ldots, A^{(n)})$, $\mathcal{B} = (B, \Sigma)$, $\mathbf{b} = (B^{(1)}, \ldots, B^{(n)})$ and $A \cap B = \emptyset$ such that $T(\mathfrak{A}) = S$ and $T(\mathfrak{B}) = T$. Let k and l be the degrees of nilpotency of $\mathfrak{A}$ and $\mathfrak{B}$, respectively, and also let $\bar{a}$ and $\bar{b}$ be the nilpotent elements of $\mathfrak{A}$ and $\mathfrak{B}$, respectively.

We construct a nilpotent DR ΣX_n-recognizer $\mathfrak{C} = (\mathcal{C}, c_0, \mathbf{c})$ that recognizes $T \cdot_{x_i} S$. Let $\mathcal{C} = (C, \Sigma)$, $C = (A \cup B) \setminus \{\bar{a}\}$, $c_0 = a_0$ and $\mathbf{c} = (C^{(1)}, \ldots, C^{(n)})$, where $\mathbf{c}$ is defined in the following way:

$$C^{(j)} = \begin{cases} A^{(j)} \cup B^{(j)} \cup A^{(i)}, & \text{if } x_j \in T,\ j \neq i, \\ A^{(j)} \cup B^{(j)}, & \text{if } x_j \notin T,\ j \neq i, \\ B^{(j)} \cup A^{(i)}, & \text{if } x_j \in T,\ j = i, \\ B^{(j)}, & \text{if } x_j \notin T,\ j = i. \end{cases}$$

The elements of Σ in $\mathcal{C}$ are represented as follows. For all $\sigma \in \Sigma$ and $c \in C$, let

$$\sigma^{\mathcal{C}}(c) = \begin{cases} \sigma^{\mathcal{B}}(c), & \text{if } c \in B, \\ \sigma^{\mathcal{B}}(b_0), & \text{if } c \in A^{(i)},\ \sigma \in root(T), \\ \sigma^{\mathcal{B}}(b_0), & \text{if } c \in A^{(i)},\ cu^{\mathcal{A}} = \bar{a} \text{ for any } u \in \hat{\Sigma}, \\ \sigma^{\mathcal{A}}(c), & \text{otherwise.} \end{cases}$$

First we show that $T(\mathfrak{C}) = T \cdot_{x_i} S$. In order to show $T(\mathfrak{C}) \subseteq T \cdot_{x_i} S$, let us consider the definition of $\mathbf{c}$. Obviously, we need to keep $B^{(j)}$ in $C^{(j)}$ in all cases. Then, in case of $j \neq i$ we need to keep $A^{(j)}$ in $C^{(j)}$ to retain all x_j-paths in $T(\mathfrak{C})$ that we had in $T(\mathfrak{A})$. Finally, if $x_j \in T$, then we need to derive x_j in all states of $A^{(i)}$ in $\mathfrak{C}$ since in this case every path from $g_{x_i}(S)$ is in $g_{x_j}(T \cdot_{x_i} S)$ as well. To show $T \cdot_{x_i} S \subseteq T(\mathfrak{C})$, let us consider the definition of $\sigma^{\mathcal{C}}(c)$ which is consisted of four parts. In the first we guarantee a $\mathfrak{B}$-like processing in $\mathfrak{C}$. The second and the third cases ensure that the processing of a word from $T \cdot_{x_i} S$, that is in a state $a \in A^{(i)}$ at the moment, can be continued in $\mathfrak{B}$. This is important because for any variable $x_j \in X_n$ and every path $uv \in g_{x_j}(T \cdot_{x_i} S)$ with $v \in g_{x_j}(T)$ and $u \in g_{x_i}(S)$ we need $c_0 uv \in C^{(j)}$. Finally, the fourth case manages the processing of any path $g(S)$ in $\mathfrak{C}$. Also, the condition $root(T) \cap \Sigma_{S,x_i} = \emptyset$ guarantees us that $\mathfrak{C}$ can determine at every step during the processing of a tree whether the next input symbol is evaluated in $\mathfrak{A}$ or in $\mathfrak{B}$. Thus we have $T(\mathfrak{C}) = T \cdot_{x_i} S$.

Now we show that $\mathfrak{C}$ is nilpotent. It is trivial that $\bar{a}$ is the trap state of $\mathfrak{A}$ since S is finite. Furthermore, $au = \bar{a}$ implies $au' = \bar{a}$ for every $a \in A$ and $u, u' \in \hat{\Sigma}$ because S is path complete. Moreover, since S is x_i-terminating, every state $a \in A \setminus \{\bar{a}\}$ with $av = \bar{a}$ implies that $a \in A^{(i)}$ for any $v \in \hat{\Sigma}$. Thus, knowing that T is nilpotent, we easily get that $cw = \bar{b}$ for every state $c \in C$ and path $w \in \hat{\Sigma}^*$ where $|w| \geq k + l$. Therefore, $\mathfrak{C}$ is nilpotent with the nilpotent element $\bar{b}$ and with the degree of nilpotency of not greater than $k + l$. □

Lemma 12. *The DR-language $T_\Sigma(Y)$ is nilpotent for any $Y \subseteq X_n$.*

Proof. Let $T = T_\Sigma(Y)$ be a DR-language for which $Y \subseteq X_n$. We construct the DR ΣX-recognizer $\mathfrak{A} = (\mathcal{A}, a_0, \mathbf{a})$ where $\mathcal{A} = (\{a_0\}, \Sigma)$, $\sigma^{\mathcal{A}}(a_0) = (a_0, \ldots, a_0)$ for all $\sigma \in \Sigma$, and $\mathbf{a} = (A^{(1)}, \ldots, A^{(n)})$ where $A^{(i)} = \{a_0\}$ if $x_i \in Y$, $\emptyset$ otherwise. Obviously, $\mathfrak{A}$ is nilpotent with the nilpotent element a_0 and with the degree of nilpotency of 0. □

A tree language represented by $\eta = \eta_k \cdot_{\xi_k} \ldots \cdot_{\xi_1} \eta_0$ is called a *plain R-chain language*, if $T(\eta_i)$ is finite and path complete, $leaves(T(\eta_i) \setminus X_n) \subseteq \{\xi_{i+1}, \ldots, \xi_k\}$, $root(T(\eta_{i+1})) \cap \Sigma_{T(\eta_i \cdot_{\xi_i} \cdots \cdot_{\xi_1} \eta_0), \xi_{i+1}} = \emptyset$ for all $i \in \{0, \ldots, k-1\}$, and $T(\eta_k) = Z \cdot_{\xi_k} T_\Sigma(Y \cup \{\xi_k\})$ where $Y, Z \subseteq X_n$.

Theorem 3. *Let $T \subseteq T_\Sigma(X_n)$ be a DR-language. T is nilpotent iff it is a plain R-chain language.*

Proof. Let T be a nilpotent DR-language and let $\mathfrak{A}$ be a reduced and normalized nilpotent DR ΣX_n-recognizer that recognizes T. By constructing the plain regular expression $\zeta_{\mathfrak{A}}$ we represent T as a plain R-chain language.

Conversely, let $\eta = \eta_k \cdot_{\xi_k} \ldots \cdot_{\xi_1} \eta_0$ be a plain R-chain language for which $T(\eta) = T$. Using Lemma 12 it is obvious that $T(\eta_k)$ is nilpotent. By repeated use of Lemma 10 and Lemma 11 we get that $T(\eta_{k-1} \cdot_{\xi_{k-1}} \ldots \cdot_{\xi_1} \eta_0)$ is path complete DR-language and is also nilpotent because of Lemma 9. Moreover, we see that $T(\eta_{k-1} \cdot_{\xi_{k-1}} \ldots \cdot_{\xi_1} \eta_0)$ is ξ_k-terminating because every z-path of $T(\eta_{k-1} \cdot_{\xi_{k-1}} \ldots \cdot_{\xi_1} \eta_0)$ is a proper prefix of a ξ_k-path of $T(\eta_{k-1} \cdot_{\xi_{k-1}} \ldots \cdot_{\xi_1} \eta_0)$, $z \in X_n$. Thus using Theorem 2 we get that $T(\eta_k \cdot_{\xi_k} \ldots \cdot_{\xi_1} \eta_0)$ is nilpotent, hence T is nilpotent, too. □

6 Conclusion

We have characterized nilpotent DR-languages by means of plain R-chain languages. To achieve this result, we have stated among others a condition by which the class of DR-languages is closed under the operation of x-product. Unlike in [9] we did not investigate the possibility of reducing plain R-chain languages nor we have investigated the number of auxiliary variables in them, however similar methods seem possible that we have seen in [9].

7 Acknowledgement

The author is thankful to Professor Ferenc Gécseg for his helpful comments and valuable suggestions. The author is also grateful to the anonymous referee whose remarks improved the quality of this paper considerably.

References

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

[2] Gécseg, F.: On some classes of tree automata and tree languages, *Annales Academiæ Scientiarum Fennicæ, Mathematica.* **25** (2000), 325-336.

[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 Imreh, B.: On monotone automata and monotone languages, *Journal of Automata, Languages, and Combinatorics.* **7** (2002), 71-82.

[5] Gécseg, F. and Peák, I.: *Algebraic Theory of Automata*, Akadémiai Kiadó, Budapest 1972.

[6] Gécseg, F. and Steinby, M.: Minimal ascending tree automata, *Acta Cybernetica,* **4** (1978), 37-44.

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

[8] Gécseg, F. and Steinby, M.: *Tree Automata*, Akadémiai Kiadó, Budapest 1984.

[9] Gyurica, Gy.: On monotone languages and their characterization by regular expressions, *Acta Cybernetica*, **18** (2007), 117-134.

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

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

Received 1st January 2008