# On the Complexity of the Syntax of Tree Languages

**Dedicated to Prof. Werner Kuich on the occasion of his retirement.**

Symeon Bozapalidis[1] and Antonios Kalampakas[2,3]

[1] Aristotle University of Thessaloniki, Department of Mathematics, 54124, Thessaloniki, Greece
bozapali@math.auth.gr
[2] Democritus University of Thrace, Department of Production Engineering and Management, 67100, Xanti, Greece
[3] Technical Institute of Kavala, Department of Exact Sciences, 65404, Kavala, Greece
akalamp@math.auth.gr

**Abstract.** The syntactic complexity of a tree language is defined according to the number of the distinct syntactic classes of all trees with a fixed yield length. This leads to a syntactic classification of tree languages and it turns out that the class of recognizable tree languages is properly contained in that of languages with bounded complexity. A refined syntactic complexity notion is also presented, appropriate exclusively for the class of recognizable tree languages. A tree language is recognizable if and only if it has finitely many refined syntactic classes. The constructive complexity of a tree automaton is also investigated and we prove that for any reachable tree automaton it is equal with the refined syntactic complexity of its behavior.

## 1 Introduction

The notion of graph language recognizability by virtue of magmoids was investigated in [2]. An advantage of this approach is that it is possible to determine the syntactic complexity of graph languages.

We say that two graphs of the same type are equivalent modulo the *syntactic congruence* $\sim_L$, of a graph language $L$, whenever they have the same set of contexts with respect to $L$. A graph language $L$ is recognizable if and only if there are finitely many syntactic classes at every type.

The *syntactic complexity* of a recognizable graph language $L$ is then measured by a function mapping any type $(m, n)$ to the number of syntactic classes at this type. This leads to a classification of graph languages according to their syntax. For instance the syntactic complexity of the set $Con(\Sigma)$ of connected graphs is bellian and also graph languages with constant, polynomial and exponential complexity are displayed (cf. [2]). In [6] the language of Eulerian graphs is shown to be syntactically more complicated than that of connected graphs. On the other hand the notion of syntactic complexity in the setup of pictures is discussed in [1].

S. Bozapalidis and G. Rahonis (Eds.): CAI 2009, LNCS 5725, pp. 189–203, 2009.

In the present paper we develop a similar descriptive complexity theory in order to investigate and classify tree languages according to their syntactic structure. Let us denote by $T_\Gamma$ the set of all trees over the ranked alphabet $\Gamma$ and by $P_\Gamma$ the monoid of all trees with just one occurrence of the variable $x$ in their yield. $P_\Gamma$ acts on $T_\Gamma$ via substitution at $x$

$$P_\Gamma \times T_\Gamma \to T_\Gamma, \quad (\tau, t) \mapsto \tau \cdot t = \tau[t/x].$$

Two notions of derivative, with respect to a tree language $L \subseteq T_\Gamma$, arise: for $\tau \in P_\Gamma$ and $t \in T_\Gamma$,

$$\tau^{-1}L = \{t \mid t \in T_\Gamma, \tau \cdot t \in L\}, \quad Lt^{-1} = \{\tau \mid \tau \in P_\Gamma, \tau \cdot t \in L\}.$$

The syntactic congruence associated with $L$ is then

$$t \sim_L t' \text{ if and only if } Lt^{-1} = Lt'^{-1}$$

and the syntactic complexity of $L$ is the function $SC_L : \mathbb{N} \to \mathbb{N}$ which sends every natural number $n$ to the number of distinct $\sim_L$-classes of trees with yield length $n$. It is well known that every recognizable language $L$ (i.e., behavior of a finite tree automaton) has finitely many right derivatives and so its $SC_L$ is bounded. Thus the growth rate of this syntactic measure can only be used for a classification of non-recognizable tree languages (Section 3).

A refined notion of syntactic complexity is introduced in Section 4 in order to define a syntactic hierarchy within the class of recognizable tree languages. Denote by $P_\Gamma^{(n)}$ the set formed by all trees where $x_1, \ldots, x_n$ occur in the yield of the tree (in this order from left to right) exactly once. For $t_1, \ldots, t_n \in T_\Gamma$, we write $\tau[t_1, \ldots, t_n]$ for the tree obtained by substituting in $\tau$ the trees $t_1, \ldots, t_n$ at $x_1, \ldots, x_n$ respectively. There results a function

$$P_\Gamma^{(n)} \times T_\Gamma^n \to T_\Gamma, \quad (\tau, t_1, \ldots, t_n) \mapsto \tau[t_1, \ldots, t_n]$$

according to which two dual notions of derivatives, with respect to a language $L \subseteq T_\Gamma$, arise: for $\tau \in P_\Gamma^{(n)}$ and $t_1, \ldots, t_n \in T_\Gamma$,

$$\tau^{-1}L = \{(t_1, \ldots, t_n) \mid t_1, \ldots, t_n \in T_\Gamma, \tau[t_1, \ldots, t_n] \in L\},$$

$$L(t_1, \ldots, t_n)^{-1} = \{\tau \mid \tau \in P_\Gamma^{(n)}, \tau[t_1, \ldots, t_n] \in L\}.$$

A main result of this paper states that the following conditions are equivalent

**i)** a language $L \subseteq T_\Gamma$ is recognizable;
**ii)** for all $n$, $card\{\tau^{-1}L \mid \tau \in P_\Gamma^{(n)}\} < \infty$;
**iii)** for all $n$, $card\{L(t_1, \ldots, t_n)^{-1} \mid t_1, \ldots, t_n \in T_\Gamma\} < \infty$.

The refined syntactic complexity of a recognizable language $L \subseteq T_\Gamma$ is the function $RSC_L : \mathbb{N} \to \mathbb{N}$ sending every natural number $n$ to the number of distinct

left derivatives $\tau^{-1}L$, where $\tau$ ranges over the set $P_\Gamma^{(n)}$. Two interesting languages with linear and exponential refined syntactic complexity are displayed. The first one is generated by the regular tree grammar

$$G: \quad x \to f(a, x, b), \quad x \to c$$

and $RSC_L(n) = 2(n+1)$, for all $n$. The second is generated by the regular tree grammar

$$G: \quad x_1 \to f(x_1, x_2), \quad x_1 \to a, \quad x_2 \to g(x_1, x_2), \quad x_2 \to b$$

and $RSC_L(n) = 2^n + 1$, for all $n$.

In the last section we present a way to measure how complicated the structure of a tree automaton is. Let $\mathcal{M} = (Q, \mu, F)$ be a (deterministic bottom up) tree automaton, over the input alphabet $\Gamma$, where $Q$ is the state set, $F \subseteq Q$ the final state set and $\mu = (\mu_f : Q^k \to Q)$, $f \in \Gamma_k$, $k \geq 0$, is the table of moves of $\mathcal{M}$. For $\tau \in P_\Gamma^{(n)}$ and $q_1, \ldots, q_n \in Q$ we denote by $\tau[q_1, \ldots, q_n]$ the state obtained by substituting $q_i$ at $x_i$ inside $\tau$, $1 \leq i \leq n$. The constructive complexity of $\mathcal{M}$ is the function $CC_\mathcal{M} : \mathbb{N} \to \mathbb{N}$ defined by the fornmula

$$CC_\mathcal{M}(n) = card\{\tau^{-1}F \mid \tau \in P_\Gamma^{(n)}\}, \text{ for all } n,$$

with $\tau^{-1}F = \{(q_1, \ldots, q_n) \mid q_1, \ldots, q_n \in Q, \tau[q_1, \ldots, q_n] \in F\}$. For reachable tree automata $\mathcal{M}, \mathcal{M}'$ we demonstrate that if $\mathcal{M}$ simulates $\mathcal{M}'$, then $CC_\mathcal{M} = CC_{\mathcal{M}'}$.

Consequently, the constructive complexity of any reachable automaton $\mathcal{M}$, with behavior $L$, coincides with the constructive complexity of the minimal tree automaton $\mathcal{M}_L$ associated with $L$: $CC_\mathcal{M} = CC_{\mathcal{M}_L}$.

As $CC_{\mathcal{M}_L} = RSC_L$ we get that the constructive complexity of any reachable automaton is equal with the refined syntactic complexity of its behavior. As a byproduct we get a bound for the function $RSC_L$, namely

$$RSC_L(n) \leq 2^{(cardQ_L)^n}, \text{ for all } n,$$

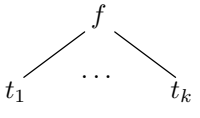
where $Q_L$ is the state set of the minimal automaton $\mathcal{M}_L$.

## 2 Basic Facts

To construct trees we need a (finite) ranked alphabet $\Gamma = \bigcup_{k \geq 0} \Gamma_k$ and a set $X = \{x_1, x_2, \ldots\}$ of variables. Let $X_n = \{x_1, x_2, \ldots, x_n\}$, $X_0 = \emptyset$. The set of trees over $\Gamma$ and $X$ is the smallest set of $T_\Gamma(X)$ inductively defined by the items

- $\Gamma_0 \cup X \subseteq T_\Gamma(X)$
- $t_1, \ldots, t_k \in T_\Gamma(X)$ and $f \in \Gamma_k$ implies $f(t_1, \ldots, t_k) \in T_\Gamma(X)$.

Often $f(t_1, \ldots, t_k)$ is depicted as



hence the denomination tree. We write $T_\Gamma$ instead of $T_\Gamma(\emptyset)$. The height of a tree $t \in T_\Gamma(X)$ is the length of its longest branch. Formally the function $height : T_\Gamma(X) \to \mathbb{N}$ is inductively defined by

- $height(\alpha) = 0$, for $a \in \Gamma_0 \cup X$;
- $height(f(t_1, \ldots, t_k)) = 1 + max\{height(t_1), \ldots, height(t_k)\}$, $f \in \Gamma_k$ and $t_1, \ldots, t_k \in T_\Gamma(X)$.

Subsets of $T_\Gamma(X)$ are refereed to as *tree languages*.

The basic operation on trees is substitution. Given $t, t_1, \ldots, t_n \in T_\Gamma(X_n)$, we denote by $t[t_1, \ldots, t_n]$ the result of substituting $t_i$ at every occurrence of $x_i$, inside $t$, $1 \leq i \leq n$. Denote by $P_\Gamma$ the subset of $T_\Gamma(x)$ consisting of all trees with exactly one occurrence of the variable $x$. $P_\Gamma$ becomes a monoid with operation the substitution at $x$: for $\tau, \pi \in P_\Gamma$, $\tau \cdot \pi = \tau[\pi/x]$. This monoid is *free* over the set of trees of the form

$$f(t_1, \ldots, t_{i-1}, x, t_{i+1}, \ldots, t_k), f \in \Gamma_k, k \geq 1, t_j \in T_\Gamma \ (j \neq i)$$

and acts, again by substitution at $x$, on the set $T_\Gamma$:

$$P_\Gamma \times T_\Gamma \to T_\Gamma, \quad (\tau, t) \mapsto \tau \cdot t = \tau[t/x].$$

The classical machine model consuming trees is the deterministic bottom up $\Gamma$-tree automaton. Such a system is a structure $\mathcal{M} = (Q, \mu, F)$ where $Q$ is the finite set of states, $F \subseteq Q$ is the final state set and $\mu = (\mu_f : Q^k \to Q)_{f \in \Gamma_k, k \geq 0}$ is the table of moves of $\mathcal{M}$. The reachability map $\mu_\mathcal{M} : T_\Gamma \to Q$ is inductively defined by

$$\mu_\mathcal{M}(f(t_1, \ldots, t_k)) = \mu_f(\mu_\mathcal{M}(t_1), \ldots, \mu_\mathcal{M}(t_k)), \quad f \in \Gamma_k, t_i \in T_\Gamma, k \geq 0$$

and the behavior of $\mathcal{M}$ is the tree language

$$|\mathcal{M}| = \{t \mid t \in T_\Gamma, \mu_\mathcal{M}(t) \in F\} = \mu_\mathcal{M}^{-1}(F).$$

Tree languages obtained in this way are called *recognizable*. The automaton $\mathcal{M}$ is said to be *reachable* whenever $\mu_\mathcal{M}$ is a surjective function. Given a tree automaton $\mathcal{M} = (Q, \mu, F)$ the monoid $P_\Gamma$ acts on each state set $Q$

$$P_\Gamma \times Q \to Q, \quad (\tau, q) \mapsto \tau \cdot q$$

as follows

- $x \cdot q = q$, $q \in Q$
- if $\tau$ is of the form $f(t_1, \ldots, t_{i-1}, x, t_{i+1}, \ldots, t_k)$ then

$$\tau \cdot q = \mu_f(\mu_\mathcal{M}(t_1), \ldots, \mu_\mathcal{M}(t_{i-1}), q, \mu_\mathcal{M}(t_{i+1}), \ldots, \mu_\mathcal{M}(t_k))$$

- if $\tau = \tau_1 \cdot \tau_2$, with $\tau_1 \neq x \neq \tau_2$, then

$$\tau \cdot q = \tau_1 \cdot (\tau_2 \cdot q), \quad q \in Q.$$

The reachability map respects the above action.

**Proposition 1.** *It holds that*

$$\mu_{\mathcal{M}}(\tau \cdot t) = \tau \cdot \mu_{\mathcal{M}}(t) \text{ for every } \tau \in P_\Gamma \text{ and } t \in T_\Gamma.$$

We are going to characterize recognizability in algebraic terms. The *right* and *left derivatives* of a tree language $L \subseteq T_\Gamma$ at $t \in T_\Gamma$ and $\tau \in P_\Gamma$ are given by

$$Lt^{-1} = \{\tau \mid \tau \in P_\Gamma, \tau \cdot t \in L\}, \quad \tau^{-1}L = \{t \mid t \in T_\Gamma, \tau \cdot t \in L\}$$

respectively. The equivalence relation $\sim_L$ on $T_\Gamma$

$$t \sim_L t' \text{ if } Lt^{-1} = Lt'^{-1}$$

is well known to be a congruence, i.e.,

$$t_1 \sim_L t'_1, \ldots, t_k \sim_L t'_k \text{ and } f \in \Gamma_k \quad \text{imply} \quad f(t_1, \ldots, t_k) \sim_L f(t'_1, \ldots, t'_k).$$

The next result is folklore.

**Proposition 2.** *The folowing conditions are equivalent for a language $L \subseteq T_\Gamma$*

**i)** *$L$ is recognizable*
**ii)** *$card\{Lt^{-1} \mid t \in T_\Gamma\} < \infty$*
**iii)** *$card\{\tau^{-1}L \mid \tau \in P_\Gamma\} < \infty$*
**iv)** *The syntactic congruence $\sim_L$ has finite index (i.e., a finite number of classes).*

A device which is equipowerful to tree automata is the *regular tree grammar*. Such a grammar is a triple $G = (\Gamma, X_n, \mathcal{R})$ where $\Gamma$, $X_n$ are the input ranked alphabet and the set of variables respectively, whereas $\mathcal{R}$ is a finite set of rules $x_i \to t$, $t \in T_\Gamma(X_n)$. For $s, s' \in T_\Gamma(X_n)$, we write $s \underset{G}{\Rightarrow} s'$ if there exist $\tau \in P_\Gamma$ and a rule $x_i \to t \in \mathcal{R}$ such that $s = \tau \cdot x_i$ and $s' = \tau \cdot t$. We set

$$L(G, x_i) = \{t \mid t \in T_\Gamma, x_i \overset{*}{\underset{G}{\Rightarrow}} t\}$$

where $\overset{*}{\underset{G}{\Rightarrow}}$ denotes as usual the reflexive and transitive closure of $\underset{G}{\Rightarrow}$.

**Proposition 3 (cf. [3,4,5]).** *A language $L \subseteq T_\Gamma$ is recognizable if and only if it is generated by a regular tree grammar $G$: $L = L(G, x_1)$.*

## 3 Syntactic Complexity of Tree Languages

Syntactic complexity is a tool to study the syntax of a tree language. It counts the number of distinct syntactic classes of trees with a fixed yield length. Formally the *syntactic complexity* of a tree language $L \subseteq T_\Gamma$ is the function

$$SC_L : \mathbb{N} \to \mathbb{N}, \quad SC_L(n) = card\{\bar{t} \mid t \in T_\Gamma, |y(t)| = n\}, \ \ n \in \mathbb{N}$$

where $\bar{t}$ stands for the $\sim_L$-class of t and the function yield, $y : T_\Gamma \to \Gamma_0^*$, is inductively defined by

$$y(c) = c, \ \ (c \in \Gamma_0), \quad y(f(t_1, \ldots, t_k)) = y(t_1) \cdots y(t_k), \ \ (f \in \Gamma_k, t_i \in T_\Gamma).$$

Alternatively we have

$$SC_L(n) = card\{Lt^{-1} \mid t \in T_\Gamma, |y(t)| = n\}, \quad n \in \mathbb{N}.$$

We say that a language $L \subseteq T_\Gamma$ has bounded, polynomial or exponential syntactic complexity if the explicit formula defining the function $SC_L$ is upper bounded by a constant, polynomial or exponential function respectively.

First let us point out that augmenting the basis alphabet $\Gamma$ the syntactic complexity remains unchanged. Indeed, if $\Gamma \subseteq \Gamma'$ and $L \subseteq T_\Gamma \subseteq T_{\Gamma'}$ then the syntactic complexity of $L$ computed with respect to $\Gamma$ and $\Gamma'$ differ at most by 1 since, for all $t, t' \in T_{\Gamma'} \setminus T_\Gamma$, we have that $t \sim_L t'$. Thus $SC_L$ does not depend on $\Gamma$.

According to Proposition 2 every recognizable tree language has bounded syntactic complexity $SC_L(n) \leq k$ for a fixed $k$ and all $n \in \mathbb{N}$. However this fact does not characterize tree language recognizability as is confirmed by the next example.

*Example 1.* Take the alphabet $\Gamma = \{f, \alpha\}$ with $rank(f) = 2$, $rank(\alpha) = 0$ and consider the tree languages $L_{bal}$ of all balanced trees and $L_{fib}$ of all Fibonacci trees

$$L_{bal} = \{t_k \mid t_0 = \alpha,\ t_{k+1} = f(t_k, t_k),\ k \geq 1\},$$
$$L_{fib} = \{s_k \mid s_0 = s_1 = a,\ s_{k+2} = f(s_{k+1}, s_k),\ k \geq 0\},$$

respectively. Observe that $|y(t_k)| = 2^k$ while $|y(s_k)| = f_k$, the $k$-th Fibonacci number. The trees

$$\tau_k = f(t_k, x), \quad \pi_k = f(s_{k+1}, x),$$

have the properties

$$\tau_k \cdot t_k \in L_{bal}, \text{ but } \tau_k \cdot t \notin L_{bal} \text{ for } t \neq t_k,$$

and

$$\pi_k \cdot s_k \in L_{fib}, \text{ but } \pi_k \cdot s \notin L_{fib} \text{ for } s \neq s_k,$$

respectively. Therefore the derivatives $L_{bal}t_k^{-1}$ are pairwise distinct and so are the derivatives $L_{fib}s_k^{-1}$ respectively. It turns out that

$$card\{L_{bal}t^{-1} \mid t \in T_\Gamma\} = \infty = card\{L_{fib}s^{-1} \mid s \in T_\Gamma\}$$

and so both the languages $L_{bal}$ and $L_{fib}$ are not recognizable. Moreover, it holds

$$SC_{L_{bal}}(n) = 2, \text{ if } n = 2^k$$
$$= 1, \text{ otherwise}$$

and similarly,

$$SC_{L_{fib}}(n) = 2, \text{ if } n = f_k$$
$$= 1, \text{ otherwise.}$$

Thus, although $L_{bal}$, $L_{fib}$ are not recognizable, they have bounded syntactic complexity.

Consequently,

**Proposition 4.** *The class BSC of tree languages with bounded syntactic complexity properly contains the class REC of recognizable tree languages.*

Our notion of complexity permits to classify the non recognizable tree languages in a non trivial way as it is presented below.

**Proposition 5.** *Given the ranked alphabet* $\Gamma = \{f_1, \ldots, f_k, \alpha\}$, $rank(f_i) = 2$, $1 \leq i \leq k$, $rank(\alpha) = 0$, *the Dyck tree language of order* $k$

$$D_k = \{t \mid t \in T_\Gamma, |t|_{f_1} = \cdots = |t|_{f_k}\}$$

*has polynomial syntactic complexity of degree* $k-1$, *namely*

$$SC_{D_k}(n) = \frac{1}{(k-1)!}\, n(n+1)\cdots(n+k-2).$$

*Proof.* For $t, t' \in T_\Gamma$ we have

$$D_k t^{-1} = D_k t'^{-1} \text{ if and only if } |t|_{f_i} = |t'|_{f_i} \text{ for } i = 1, \ldots, k.$$

On the other hand the number of binary symbols occurring in a tree $t \in T_\Gamma$ with yield length $n$ is just $n-1$. Therefore the different ways to share the symbols $f_1, \ldots, f_k$ in the nodes of $t$ is equal with the number of $k$-tuples of natural numbers $(x_1, \ldots, x_k)$ verifying the equation

$$x_1 + \cdots + x_k = n - 1$$

which, as it is well known from Combinatorics, is equal with

$$\binom{n-1+k-1}{k-1} = \binom{n+k-2}{k-1} = \frac{1}{(k-1)!}\, n(n+1)\cdots(n+k-2).$$

Hence the proposed formula.

A tree language $L \subseteq T_\Gamma$ such that for every $n$

$$card\{Lt^{-1} \mid t \in T_\Gamma, |y(t)| = n\} = card\{t \mid t \in T_\Gamma, |y(t)| = n\}$$

will be called *syntactically hard*. Of course such a language $L$ has the highest possible syntactic complexity, i.e.,

$$SC_L(n) = card\{t \mid t \in T_\Gamma, |y(t)| = n\}.$$

In the case that $\Gamma = \{f, a\}$, with $rank(f) = 2$, $rank(\alpha) = 0$ the above number is well known from Combinatorics and is the $n-1$-th Catalan number $C_{n-1}$, where

$$C_n = \frac{1}{n+1}\binom{2n}{n} \simeq \frac{4^n}{n^{3/2}\sqrt{\pi}}.$$

**Proposition 6.** *The diagonal language*

$$L_d = \{f(t,t) \mid t \in T_\Gamma\}, \quad \Gamma = \{f, \alpha\},$$

*is syntactically hard*

$$SC_{L_d}(n+1) = \frac{1}{n+1}\binom{2n}{n}.$$

*Proof.* Actually, we shall show that the right derivatives $L_d t^{-1}$, $t \in T_\Gamma$, are pairwise distinct. First observe that

$$L_d t^{-1} = \{f(s,\tau) \mid s \in T_\Gamma,\, \tau \in P_\Gamma,\, s = \tau{\cdot}t\} \cup \{f(\tau,s) \mid s \in T_\Gamma,\, \tau \in P_\Gamma,\, s = \tau{\cdot}t\}.$$

Now, if $L_d t^{-1} \cap L_d t'^{-1} \neq \emptyset$, then

$$f(s,\tau) = f(s',\tau'), \quad s = \tau \cdot t,\ s' = \tau' \cdot t',$$

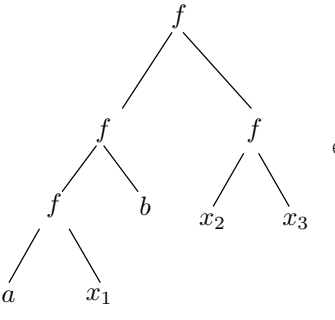
or

$$f(\tau,s) = f(\tau',s'), \quad s = \tau \cdot t,\ s' = \tau' \cdot t'.$$

Hence $s = s'$, $\tau = \tau'$ and $t = t'$.

## 4 Refined Syntactic Complexity

As we have seen the growth rate of the function $SC_L$ introduced in the previous section gives no information that allows us to compare recognizable tree languages with respect to their complexity. Our intention in the present section is to provide an efficient complexity measure for recognizable tree languages. Let us denote by $P_\Gamma^{(n)}$ the subset of $T_\Gamma(X_n)$ formed by all trees where $x_1, \ldots, x_n$ occur in the yield of the tree (in this order from left to right) exactly once. For instance the tree

$\tau =$  $\in P_\Gamma^{(3)}$.



For every $n \geq 1$ there is a junction function

$$P_\Gamma^{(n)} \times T_\Gamma^n \to T_\Gamma, \quad (\tau, t_1, \ldots, t_n) \mapsto \tau[t_1, \ldots, t_n].$$

With respect to $L \subseteq T_\Gamma$, two dual notions of derivatives can be defined:

$$\tau^{-1}L = \{(t_1, \ldots, t_n) \mid \tau[t_1, \ldots, t_n] \in L\},$$

$$L(t_1, \ldots, t_n)^{-1} = \{\tau \mid \tau \in P_\Gamma^{(n)},\ \tau[t_1, \ldots, t_n] \in L\},$$

for all $\tau \in P_\Gamma^{(n)}$ and $t_1, \ldots, t_n \in T_\Gamma$.

**Theorem 1.** *For $L \subseteq T_\Gamma$, the following conditions are equivalent*

**i)** *$L$ is recognizable*
**ii)** *for every $n \geq 1$, $card\{\tau^{-1}L \mid \tau \in P_\Gamma^{(n)}\} < \infty$*
**iii)** *for every $n \geq 1$, $card\{L(t_1, \ldots, t_n)^{-1} \mid t_1, \ldots, t_n \in T_\Gamma\} < \infty$.*

*Proof. iii)* $\Rightarrow$ *ii)*. Assume that $L(t_{11}, \ldots, t_{1n})^{-1}, \ldots, L(t_{k1}, \ldots, t_{kn})^{-1}$ are the distinct right derivatives of $L$. Then the function

$$\phi : \{\tau^{-1}L \mid \tau \in P_\Gamma^{(n)}\} \to \{0,1\}^k, \quad \phi(\tau^{-1}L) = (\varepsilon_1, \ldots, \varepsilon_k)$$

with $\varepsilon_i = 1$ iff $\tau[t_{i1}, \ldots, t_{in}] \in L$ is well defined and moreover it is injective since

$$\phi(\tau^{-1}L) = \phi(\tau'^{-1}L) \text{ implies } \tau^{-1}L = \tau'^{-1}L.$$

The hypothesis $\phi(\tau^{-1}L) = \phi(\tau'^{-1}L)$ is equivalent to

$$\tau[t_{i1}, \ldots, t_{in}] \in L \text{ iff } \tau'[t_{i1}, \ldots, t_{in}] \in L \tag{1}$$

for all $i = 1, 2, \ldots, k$. We have

$$\begin{aligned}
(s_1, \ldots, s_n) \in \tau^{-1}L &\Leftrightarrow \tau[s_1, \ldots, s_n] \in L \\
&\Leftrightarrow \tau \in L(s_1, \ldots, s_n)^{-1} = L(t_{i1}, \ldots, t_{in})^{-1}, \quad \text{for some } i \\
&\Leftrightarrow \tau[t_{i1}, \ldots, t_{in}] \in L \quad \text{(by 1 above)} \\
&\Leftrightarrow \tau'[t_{i1}, \ldots, t_{in}] \in L \\
&\Leftrightarrow \tau' \in L(t_{i1}, \ldots, t_{in})^{-1} \\
&\Leftrightarrow (s_1, \ldots, s_n) \in \tau'^{-1}L
\end{aligned}$$

that is $\tau^{-1}L = \tau'^{-1}L$ as wanted. From the injectivity of $\phi$ we get

$$card\{\tau^{-1}L \mid \tau \in P_\Gamma^{(n)}\} < \infty.$$

The implication *ii)* $\Rightarrow$ *iii)* can be proved in a similar way.

The fact that *ii)* $\Rightarrow$ *i)* follows from Proposition 2 since $P_\Gamma = P_\Gamma^{(1)}$.

*i)* $\Rightarrow$ *ii)*. Consider a tree automaton $\mathcal{M} = (Q, \mu, F)$ with behavior $L$ and let $\mu_\mathcal{M} : T_\Gamma \to Q$ be its reachability map. We shall demonstrate that for every $t_1, \ldots, t_n \in T_\Gamma$ there exist $\bar{t}_1, \ldots, \bar{t}_n \in T_\Gamma$ with *height* less or equal to *cardQ* such that

$$\mu_\mathcal{M}(\tau[t_1, \ldots, t_n]) = \mu_\mathcal{M}(\tau[\bar{t}_1, \ldots, \bar{t}_n]), \quad \text{for all } \tau \in P_\Gamma^{(n)}.$$

Indeed let us choose $\bar{t}_i$ with the property

$$\mu_\mathcal{M}(\bar{t}_i) = \mu_\mathcal{M}(t_i), \quad height(\bar{t}_i) \leq cardQ$$

for all $i = 1, \ldots, n$. Then we get

$$\begin{aligned}
\mu_\mathcal{M}(\tau[t_1, \ldots, t_n]) &= \mu_\mathcal{M}(\tau[x, t_2, \ldots, t_n] \cdot t_1) && \text{(by Prop. 1)} \\
&= \tau[x, t_2, \ldots, t_n] \cdot \mu_\mathcal{M}(t_1) \\
&= \tau[x, t_2, \ldots, t_n] \cdot \mu_\mathcal{M}(\bar{t}_1) && \text{(by Prop. 1)} \\
&= \mu_\mathcal{M}(\tau[x, t_2, \ldots, t_n] \cdot \bar{t}_1) \\
&= \mu_\mathcal{M}(\tau[\bar{t}_1, \ldots, t_n]) = \cdots = \mu_\mathcal{M}(\tau[\bar{t}_1, \ldots, \bar{t}_n]).
\end{aligned}$$

Now it holds that

$$L(t_1, \ldots, t_n)^{-1} = L(\bar{t}_1, \ldots, \bar{t}_n)^{-1}.$$

In fact

$$\begin{aligned} \tau \in L(t_1, \ldots, t_n)^{-1} &\Leftrightarrow \tau[t_1, \ldots, t_n] \in L = \mu_{\mathcal{M}}^{-1}(F) \\ &\Leftrightarrow \mu_{\mathcal{M}}(\tau[t_1, \ldots, t_n]) \in F \\ &\Leftrightarrow \mu_{\mathcal{M}}(\tau[\bar{t}_1, \ldots, \bar{t}_n]) \in F \\ &\Leftrightarrow \tau[\bar{t}_1, \ldots, \bar{t}_n] \in \mu_{\mathcal{M}}^{-1}(F) = L \\ &\Leftrightarrow \tau \in L(\bar{t}_1, \ldots, \bar{t}_n)^{-1} \end{aligned}$$

as wanted. It follows that

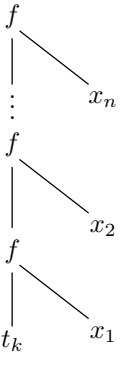
$$card\{L(t_1, \ldots, t_n)^{-1} \mid t_1, \ldots t_n \in T_\Gamma\} \leq (card Q)^n < \infty$$

and the proof is completed.

The *refined syntactic complexity* of a recognizable tree language $L \subseteq T_\Gamma$ is the function $RSC_L : \mathbb{N} \to \mathbb{N}$ sending every natural number $n$ to the number of the distinct left derivatives $\tau^{-1}L$ when $\tau$ ranges over $P_\Gamma^{(n)}$, i.e.,

$$RSC_L(n) = card\{\tau^{-1}L \mid \tau \in P_\Gamma^{(n)}\}.$$

*Example 2.* Return to the non-recognizable language $L_{bal}$ and let us choose the trees

$\tau_{k,n} =$  $\in P_\Gamma^{(n)}$.

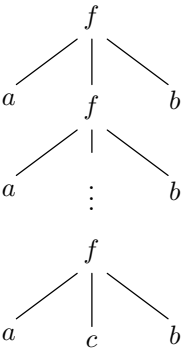


For $s_0, \ldots, s_{n-1} \in T_\Gamma$, we have $\tau_{k,n}[s_0, \ldots, s_{n-1}] \in L_{bal}$ iff $s_i = t_{k+i}$ for $0 \leq i < n$. In other words, there are infinitely many distinct left derivatives $\tau^{-1}L_{bal}$, $\tau \in P_\Gamma^{(n)}$, i.e., $RSC_{L_{bal}}(n) = \infty$ for all $n$.

Similar observations can be made for $L_{fib}$.

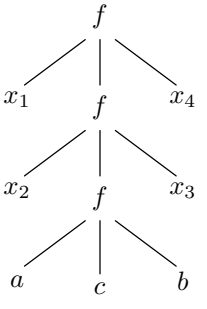
In the sequel we display two recognizable languages having linear and exponential syntactic complexity respectively.

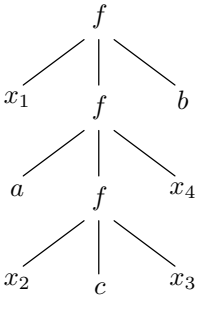
*Example 3.* Consider the recognizable tree language $L$ consisting of all trees of the form

$t_m =$ 

($m$ occurrences of $f$, $m \geq 1$).

For $0 \leq k \leq n \leq 2m$ let us denote by $d_k$ a strictly increasing function from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, 2m\}$ such that $d_k(i) \leq m$ for all $i = 1, 2, \ldots, k$ and $d_k(i) \geq m + 1$ for $i = k + 1, k + 2, \ldots, n$. We introduce the tree $\tau(d_k)$ obtained from $t_m$ above by distributing via $d_k$ the variables $x_1, \ldots, x_k$ on the left nodes labelled by $a$ and the remaining variables $x_{k+1}, \ldots x_n$ on the right nodes labelled by $b$. For instance

$\tau(d_2) =$ 

and $\tau(d'_2) =$ 


with $d_2, d'_2 : \{1, 2, 3, 4\} \to \{1, 2, 3, 4, 5, 6\}$ given by $d_2(1) = 1$, $d_2(2) = 2$, $d_2(3) = 5$, $d_2(4) = 6$ and $d'_2(1) = 1$, $d'_2(2) = 3$, $d'_2(3) = 4$, $d'_2(4) = 5$ respectively. It is not hard to see that, if $d_k, d'_k$ are two distributions as defined previously, then we have

$$\tau(d_k)^{-1}L = \tau(d'_k)^{-1}L$$

therefore there are exactly $n + 1$ distinct left derivatives of the above form, namely,

$$\tau(d_0)^{-1}L, \tau(d_1)^{-1}L, \ldots, \tau(d_n)^{-1}L.$$

Next for $1 \leq k \leq n \leq 2m+1$ let us denote by $\delta_k$ a strictly increasing function from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, 2m + 1\}$ with the property $\delta_k(k) = m + 1$ and $\delta_k(i) \leq m$ $(i \leq k - 1)$, $\delta_k(i) \geq m + 2$ $(i \geq k + 1)$. Also we denote by $\tau(\delta_k)$ the tree obtained from $t_m$ above by distributing via $\delta_k$ the variables $x_1, \ldots x_{k-1}$ on the left nodes labelled by $a$, the variables $x_{k+1}, \ldots x_n$ on the right nodes labelled by $b$ whereas the variable $x_k$ replaces the node labelled by $c$. As above we can verify that there are exactly $n$ distinct left derivatives

$$\tau(\delta_1)^{-1}L, \tau(\delta_2)^{-1}L, \ldots, \tau(\delta_n)^{-1}L.$$

Of course if $\tau \in P_\Gamma^{(n)}$ is neither of the form $\tau(d)$ nor $\tau(\delta)$ then $\tau^{-1}L = \emptyset$. We conclude that there are in total $n+1+n+1 = 2(n+1)$ distinct left derivatives of $L$ at level $n$, i.e., $RSC_L(n) = 2(n+1)$ and the language $L$ has linear syntactic complexity.

*Example 4.* Consider the regular tree grammar

$$G: \quad y_1 \to f(y_1, y_2), \quad y_2 \to g(y_1, y_2), \quad y_1 \to a, \quad y_2 \to b,$$

and the tree language $L(G, y_1)$ generated by $G$ starting from the variable $y_1$. A tree $t$ belongs to $L(G, y_1)$ if and only if the left (resp. right) child of any node of $t$ is labelled either by $f$ or $a$ (resp. $g$ or $b$). From any $t \in L(G, y_1)$ and any strictly increasing function $d : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, |y(t)|\}$ we derive the tree $\tau(t, d) \in P_\Gamma^{(n)}$ by replacing the letter located at the $d(i)$-th position of $y(t)$ (from left to right) with the variable $x_i$ $(1 \leq i \leq n)$. It is not hard to see that two such trees $\tau(t, d)$ and $\tau(t', d')$ define the same left derivative,

$$\tau(t, d)^{-1}L(G, y_1) = \tau(t', d')^{-1}L(G, y_1)$$

if and only if for every $i \in \{1, 2, \ldots, n\}$ the $d(i)$-th and $d'(i)$-th letters in $y(t)$, $y(t')$, respectively, are equal. It turns out that the distinct left derivatives of $L(G, y_1)$ correspond to the possible ways of substituting $x_1$ by $a$ or $b$, $x_2$ by $a$ or $b$, etc. Taking into account the empty left derivative we finally obtain

$$RSC_{L(G,y_1)}(n) = 2^n + 1$$

that is $L(G, y_1)$ has exponential syntactic complexity.

## 5 Constructive Complexity of a Tree Automaton

Here we display a way to measure how complicated the structure of a tree automaton is. First we need some additional notation. Given a tree automaton $\mathcal{M} = (Q, \mu, F)$, for every $t \in T_\Gamma(X_n)$ and every $q_1, \ldots, q_n \in Q$, the element $t[q_1, \ldots, q_n] \in Q$ is inductively defined as follows

- for $t = x_i$, $x_i[q_1, \ldots, q_n] = q_i$, $1 \leq i \leq n$;
- for $t = c \in \Gamma_0$, $c[q_1, \ldots, q_n] = \mu_c$;
- for $t = f(t_1, \ldots, t_k)$, $f \in \Gamma_k$, $t_i \in T_\Gamma(X_n)$

$$f(t_1, \ldots, t_k)[q_1, \ldots, q_n] = \mu_f(t_1[q_1, \ldots, q_n], \ldots, t_k[q_1, \ldots, q_n]).$$

The *constructive complexity* of the automaton $\mathcal{M} = (Q, \mu, F)$ is the function $CC_\mathcal{M} : \mathbb{N} \to \mathbb{N}$ defined by the formula

$$CC_\mathcal{M}(n) = card\{\tau^{-1}F \mid \tau \in P_\Gamma^{(n)}\}$$

where

$$\tau^{-1}F = \{(q_1, \ldots, q_n) \mid \tau[q_1, \ldots, q_n] \in F\}.$$

Since for all $n$ we have $\tau^{-1}F \subseteq Q^n$, we get that

$$CC_{\mathcal{M}}(n) \leq 2^{(cardQ)^n}$$

and so $CC_{\mathcal{M}}$ is everywhere defined.

*Example 5.* Let $\Gamma$ be a finite ranked alphabet and consider the automaton $\mathcal{M} = (\mathbb{Z}_m, \mu, F = \{0\})$ where $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$ is the additive group of integers $mod\, m$. The moves $\mu_f : \mathbb{Z}_m^k \to \mathbb{Z}_m$ are given by

$$\mu_c = 1, \ \ (c \in \Gamma_0), \qquad \mu_f(\alpha_1, \ldots, \alpha_k) = 1 + \alpha_1 + \cdots + \alpha_k, \ \ (f \in \Gamma_k, k \geq 1)$$

where at the right hand side the designated addition is the $mod\, m$ addition. The reachability map $\mu_{\mathcal{M}} : T_\Gamma \to \mathbb{Z}_m$ sends every tree $t$ to its $mod\, m$ size, i.e.,

$$\mu_{\mathcal{M}}(t) = |t| (mod\, m)$$

and the behavior of $\mathcal{M}$ consists of all trees whose size is divisible by $m$. For $\tau, \tau' \in P_\Gamma^{(n)}$, we have

$$\tau^{-1}F = \tau'^{-1}F \text{ if and only if } |\tau| \equiv |\tau'| (mod\, m).$$

Consequently, there are exactly $m$ distinct classes $\tau^{-1}F$, that is $CC_{\mathcal{M}}(n) = m$ for all $n$ and thus $\mathcal{M}$ has constant constructive complexity.

A naturally arising question concerns the comparison of the complexities $CC_{\mathcal{M}}$ and $RSC_{|\mathcal{M}|}$. Recall that a *simulation* of $\mathcal{M} = (Q, \mu, F)$ to $\mathcal{M}' = (Q', \mu', F')$ is a surjective function $h : Q \to Q'$ respecting the moves

$$h(\mu_f(q_1, \ldots, q_n)) = \mu'_f(h(q_1), \ldots, h(q_k)), \qquad f \in \Gamma_k, \ q_i \in Q,$$

and moreover $h^{-1}(F') = F$. An induction argument on the complexity of the tree $\tau \in P_\Gamma^{(n)}$ shows that

$$h(\tau[q_1, \ldots, q_n]) = \tau[h(q_1), \ldots, h(q_n)], \qquad q_1, \ldots, q_n \in Q. \tag{2}$$

**Proposition 7.** *Let $\mathcal{M}, \mathcal{M}'$ be reachable tree automata. If there is a simulation $h : \mathcal{M} \to \mathcal{M}'$ then both $\mathcal{M}$ and $\mathcal{M}'$ have the same constructive complexity*

$$CC_{\mathcal{M}} = CC_{\mathcal{M}'}.$$

*Proof.* We have to show that

$$CC_{\mathcal{M}}(n) = CC_{\mathcal{M}'}(n), \quad \text{for all } n$$

or that

$$card\{\tau^{-1}F \mid \tau \in P_\Gamma^{(n)}\} = card\{\tau^{-1}F' \mid \tau \in P_\Gamma^{(n)}\}, \quad \text{for all } n.$$

The last fact will follow if we show that the assignment $\tau^{-1}F \mapsto \tau^{-1}F'$ is a well defined injection, which is expressed by the following logical equivalence

$$\tau^{-1}F = \tau'^{-1}F \Leftrightarrow \tau^{-1}F' = \tau'^{-1}F'.$$

Assume that $\tau^{-1}F = \tau'^{-1}F$, then

$$\begin{aligned}(q'_1,\ldots,q'_n) \in \tau^{-1}F' &\Leftrightarrow (h(q_1),\ldots,h(q_n)) \in \tau^{-1}F', \qquad q'_i = h(q_i),\ 1 \le i \le n\\ &\Leftrightarrow \tau[h(q_1),\ldots,h(q_n)] \in F' \qquad \text{(by Eq. (2))}\\ &\Leftrightarrow h(\tau[q_1,\ldots,q_n]) \in F' \qquad (\text{by } F = h^{-1}(F'))\\ &\Leftrightarrow \tau[q_1,\ldots,q_n] \in F\\ &\Leftrightarrow (q_1,\ldots,q_n) \in \tau^{-1}F = \tau'^{-1}F\\ &\Leftrightarrow \tau'[q_1,\ldots,q_n] \in F\\ &\Leftrightarrow h(\tau'[q_1,\ldots,q_n]) \in F'\\ &\Leftrightarrow (q'_1,\ldots,q'_n) \in \tau'^{-1}F'\end{aligned}$$

and thus $\tau^{-1}F' = \tau'^{-1}F'$. The implication

$$\tau^{-1}F' = \tau'^{-1}F' \Rightarrow \tau^{-1}F = \tau'^{-1}F$$

is proved analogously.

The minimal automaton associated with a tree language $L \subseteq T_\Gamma$ is

$$\mathcal{M}_L = (Q_L, \mu_L, F_L)$$

where

- $Q_L = \{Lt^{-1} \mid t \in T_\Gamma\}, \quad F_L = \{Lt^{-1} \mid t \in L\}$;
- $(\mu_L)_f : Q_L^k \to Q_L, \quad (\mu_L)_f(Lt_1^{-1},\ldots,Lt_k^{-1}) = Lf(t_1,\ldots,t_k)^{-1}, \quad f \in \Gamma_k.$

Clearly $\mathcal{M}_L$ is a reachable automaton with behavior $L$ and for every reachable automaton $\mathcal{M} = (Q, \mu, F)$ with behavior $L$, there is a (unique) simulation $h : \mathcal{M} \to \mathcal{M}_L$ defined by

$$h(q) = Lt^{-1}, \quad \mu_{\mathcal{M}}(t) = q, \quad q \in Q$$

therefore, by virtue of Proposition 7, we get

$$CC_{\mathcal{M}} = CC_{\mathcal{M}_L}.$$

On the other hand

**Proposition 8.** *If $\mathcal{M}$ is a reachable automaton with behavior $L$, then*

$$CC_{\mathcal{M}} = RSC_L.$$

*Proof.* Analogous to that of Proposition 7.

Taking into account the previous discussion we conclude that

**Proposition 9.** *For every recognizable tree language $L \subseteq T_\Gamma$ it holds*

$$RSC_L(n) \le 2^{(CardQ_L)^n}, \quad \textit{for all } n.$$

## Acknowledgement

We thank the referees for their fruitful suggestions and remarks.

## References

1. Bozapalidis, S.: Picture languages: Recognizability, Syntax and Context Freeness (submitted)
2. Bozapalidis, S., Kalampakas, A.: Recognizability of graph and pattern languages. Acta Informatica 42, 553–581 (2006)
3. Engelfriet, J.: Tree Automata and Tree Grammars, DAIMNI FN-10, University of Aarhus (1975)
4. Gécseg, F., Steinby, M.: Tree Automata. Akademiai Kiado, Budapest (1984)
5. Gécseg, F., Steinby, M.: Tree Languages. Handbook of Formal Languages 3, 1–68 (1997)
6. Kalampakas, A.: The syntactic complexity of eulerian graphs. In: Bozapalidis, S., Rahonis, G. (eds.) CAI 2007. LNCS, vol. 4728, pp. 208–217. Springer, Heidelberg (2007)