Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 307 (2003) 103 – 116

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Growth-sensitivity of context-free languages

Tullio Ceccherini-Silberstein[a,*,1], Wolfgang Woess[b]

[a] *Dipartimento di Ingegneria, Università del Sannio, Corso Garibaldi 107, Benevento I-82100, Italy*
[b] *Institut für Mathematik, Technische Universität Graz, Steyrergasse 30, Graz A-8010, Austria*

Abstract

A language L over a finite alphabet $\mathbf{\Sigma}$ is called growth-sensitive if forbidding any set of subwords F yields a sub-language L^F whose exponential growth rate is smaller than that of L. It is shown that every (essentially) ergodic non-linear context-free language of convergent type is growth-sensitive. "Ergodic" means that the dependency di-graph of the generating context-free grammar is strongly connected, and "essentially ergodic" means that there is only one non-regular strong component in that graph. The methods combine (1) an algorithm for constructing from a given grammar one that generates the associated 2-block language and (2) a generating function technique regarding systems of algebraic equations. Furthermore, the algorithm of (1) preserves unambiguity as well as the number of non-regular strong components of the dependency di-graph.
© 2003 Elsevier B.V. All rights reserved.

Keywords: Context-free grammar; Ambiguity; Ergodicity; Dependency di-graph; Higher block languages; Growth; Perron–Frobenius eigenvalue

---

1. Introduction

Let L be a language over the alphabet $\mathbf{\Sigma}$, that is, a subset of the free monoid $\mathbf{\Sigma}^*$ of all finite words over $\mathbf{\Sigma}$. We write ε for the empty word and $\mathbf{\Sigma}^+ = \mathbf{\Sigma}^* \setminus \{\varepsilon\}$. For a word $w \in \mathbf{\Sigma}^*$, its length (number of letters) is denoted by $|w|$. The *growth* of L is the number

$$\gamma(L) = \limsup_{n\to\infty} |\{w \in L\colon |w| = n\}|^{1/n}.$$

---

[*] Corresponding author.

E-mail addresses: tceccher@mat.uniroma1.it (T. Ceccherini-Silberstein), woess@weyl.math.tu-graz.ac.at (W. Woess).

[1] Partially supported by TU Graz and by the Swiss National Science Foundation.

doi:10.1016/S0304-3975(03)00095-1

If L is infinite (the interesting case) then $1 \leqslant \gamma(L) \leqslant |\mathbf{\Sigma}|$. The number $1/\gamma(L)$ is the radius of convergence of the *growth series* of L,

$$f_L(z) = \sum_{n=0}^{\infty} |\{w \in L\colon |w| = n\}| z^n, \quad z \in \mathbb{C}. \tag{1}$$

Definition 1. A language L over $\mathbf{\Sigma}$ is *growth-sensitive* if

$$\gamma(L^F) < \gamma(L)$$

for any non-empty $F \subset \mathbf{\Sigma}^*$ consisting of subwords of elements of L, where

$$L^F = \{w \in L\colon \text{ no } v \in F \text{ is a subword of } w\}.$$

The general question addressed in this note is the following: under which conditions is a language L growth-sensitive? Note that, in principle, the question is of interest only when L has exponential growth, that is $\gamma(L) > 1$. Indeed, if $\gamma(L) = 1$ then either $\gamma(L^F) = 1$ or $\gamma(L^F) = 0$, in which case L^F is finite. Also note that without specific assumptions on L, one cannot expect growth-sensitivity: for example, if $\mathbf{\Sigma} = \{a, b, c, d\}$ and $L = \{a, b\}^* \cup \{c, d\}^*$, then $\gamma(L) = \gamma(L^{\{a\}}) = 2$.

Our principal result is the part B of the following theorem.

Theorem 2. (A) *Ergodic*, *unambiguous regular languages are growth-sensitive*.

(B) *Ergodic*, *unambiguous*, *non-linear context-free languages are growth-sensitive*.

In this theorem, we intend that the respective language is generated by a grammar that has all the indicated properties. In particular, *ergodic* means for a context-free grammar that its *dependency-digraph* is strongly connected, see Section 2 below where the setup will be described carefully.

Part (A) of Theorem 2 is well-known in somewhat different setups and terminologies (see, e.g., [3]), it has several analogues in the context of Symbolic Dynamics [13, Corollary 4.4.9; 1, Theorem 2.14] and Asymptotic Group Theory [6]; see also the respective remarks in [3, Section 5], which is the basis for this extended abstract. A basic example where part (B) of Theorem 2 applies is the *Dyck language*, see Section 5.

Our strategy for proving that every L in a given class $\mathscr{L}$ of languages is growth-sensitive is the following.

Step 1: Consider the set $(\mathbf{\Sigma}^2)^*$ of all words over $\mathbf{\Sigma}^2$. Its letters are of the form (ab), where $a, b \in \mathbf{\Sigma}$. Define $\phi\colon \mathbf{\Sigma}^* \to (\mathbf{\Sigma}^2)^*$ by $\phi(w) = \varepsilon$ if $|w| \leqslant 1$ and

$$\phi(a_1 \cdots a_n) = (a_1a_2)(a_2a_3) \cdots (a_{n-2}a_{n-1})(a_{n-1}a_n) \quad \text{if } n \geqslant 2.$$

For any language $L \subset \mathbf{\Sigma}^*$, consider the associated 2-*block-language* $\phi(L)$ over the alphabet

$$\mathbf{\Sigma}_{(2)} = \mathbf{\Sigma}_{(2)}(L) = \{(ab)\colon a, b \in \mathbf{\Sigma},\ ab \text{ is a subword of } w \text{ for some } w \in L\}. \tag{2}$$

Then $\gamma(L) = \gamma(\phi(L))$. Step 1 is the following: prove that $L \in \mathscr{L}$ implies $\phi(L) \in \mathscr{L}$.

Step 2: Show that each $L \in \mathscr{L}$ is growth-sensitive to forbidding one (or more) elements of its alphabet $\mathbf{\Sigma}$.

Then each $L \in \mathscr{L}$ will be growth-sensitive to forbidding any $F \subset \mathbf{\Sigma}^*$. Indeed, it is enough to prove this when $F = \{v_1\}$ consists of a single word v_1. If $m = |v_1|$, then after $m-1$ iterations, $v_m = \phi^{(m-1)}(v_1)$ is a *letter* in the alphabet of $\phi^{(m-1)}(L)$. Therefore, Steps 1 and 2 imply

$$\gamma(L) = \gamma(\phi^{(m-1)}(L)) < \gamma((\phi^{(m-1)}(L))^{\{v_m\}}) = \gamma(L^{\{v_1\}}).$$

Below (Section 3) we shall present an algorithm for passing from a context-free grammar that generates L to a new grammar that generates $\phi(L)$. While this algorithm preserves unambiguity (as is easily seen), it does not preserve ergodicity. Thus, we are led to an extended definition of what we call an *essentially ergodic* context-free grammar, see Section 2. This involves a careful analysis of the *strong components* of the dependency di-graph: we introduce the notion of a *regular* strong component, and "essentially ergodic" means in principle that there is precisely one non-regular component. A careful analysis of our algorithm then shows that it preserves the number of essential components (Section 3, Theorem 8).

We now add another notion.

Definition 3. We say that a language L is of *convergent type*, if $f_L(1/\gamma(L)) < \infty$, and of *divergent type*, otherwise.

Every (strictly) ergodic, unambiguous context-free language that is non-linear must be of convergent type [3, (3.3)]. The following extension of Theorem 2(B) is the main result of this work.

Theorem 4. *Every essentially ergodic, unambiguous context-free language of convergent type is growth-sensitive.*

Having already outlined how we achieve Step 1 of our proof-strategy in Sections 2 and 3, we now explain the structure of the remaining sections of the present note. In Section 4, we state a general theorem on the radii of convergence of generating functions that satisfy a system of algebraic equations. Then we explain how this applies to an essentially ergodic context-free language L as well as to L^F, where $F \subset \mathbf{\Sigma}$, hereby completing Step 2. In Section 5 we present a class of examples that we call the *restricted Dyck languages.*

Some additional results, other examples as well as an explanation of the interplay of the results presented here with geometric group theory can be found in the long version [3] of this extended abstract.

We decided to publish this extended abstract in a theoretical computer science environment because we would like to circulate our results (namely Theorems 2(B) and 4) among computer scientists. Also, we believe that our method of analyzing a context-free grammar via the strongly connected components of its dependency di-graph (focusing, in particular, on the regular and essential components) could be also useful in other settings.

2. Analyzing a context-free grammar by its dependency-digraph

In order to set up our notation, we start by reviewing some basic notions.

A *context-free grammar* is a quadruple $\mathscr{C} = (\mathbf{V}, \mathbf{\Sigma}, \mathbf{P}, S)$, where $\mathbf{V}$ is a finite set of *variables*, disjoint from the finite alphabet $\mathbf{\Sigma}$ (the *terminal symbols*), the variable S is the *start symbol*, and $\mathbf{P} \subset \mathbf{V} \times (\mathbf{V} \cup \mathbf{\Sigma})^*$ is a finite set of *production rules.* We write $T \vdash u$ or $(T \vdash u) \in \mathbf{P}$ if $(T, u) \in \mathbf{P}$. For $v, w \in (\mathbf{V} \cup \mathbf{\Sigma})^*$, we write $v \Rightarrow w$ if $v = v_1 T v_2$ and $w = v_1 u v_2$, where $T \vdash u$, $v_1 \in (\mathbf{V} \cup \mathbf{\Sigma})^*$ and $v_2 \in \mathbf{\Sigma}^*$. A *rightmost derivation* is a sequence $v = w_0, w_1, \ldots, w_k = w \in (\mathbf{V} \cup \mathbf{\Sigma})^*$ such that $w_{i-1} \Rightarrow w_i$; we then write $v \overset{*}{\Rightarrow} w$. For $T \in \mathbf{V}$, we consider the language $L_T = \{w \in \mathbf{\Sigma}^* : T \overset{*}{\Rightarrow} w\}$. The *language generated by* $\mathscr{C}$ is $L(\mathscr{C}) = L_S$.

A *context-free language* is a language generated by a context-free grammar.

Recall that a grammar and the language generated by it are called *linear*, if every production rule in $\mathbf{P}$ is of the form $T \vdash v_1 U v_2$ or $T \vdash v$, where $v, v_1, v_2 \in \mathbf{\Sigma}^*$ and $T, U \in \mathbf{V}$. If furthermore in this situation one always has $v_2 = \varepsilon$, then grammar and language are called *right linear.* Analogously, it is called *left linear*, if instead one always has $v_1 = \varepsilon$. In both cases, language and grammar are also called *regular*.

If $\mathscr{C}$ is a general context free grammar, then for any variable $T \in \mathbf{V}$, the *ambiguity degree* $d_T(w)$ of a word $w \in \mathbf{\Sigma}^*$ is the number of all different rightmost derivations $T \overset{*}{\Rightarrow} w$. Thus $d_T(w) > 0$ if and only if $w \in L_T$. We shall assume that $d_T(w) < \infty$ always. The grammar is called *unambiguous*, if $d_S(w) = 1$ for all $w \in L$. A context-free language is called unambiguous if it is generated by some unambiguous grammar.

We shall always assume to have a *reduced* grammar $\mathscr{C}$, that is, each variable is used in some rightmost derivation of a word in $L(\mathscr{C})$ and in particular, $L_T \neq \emptyset$ for each variable T.

The *dependency di-graph* $\mathscr{D} = \mathscr{D}(\mathscr{C})$ of a context-free grammar $\mathscr{C} = (\mathbf{V}, \mathbf{\Sigma}, \mathbf{P}, S)$ is an oriented graph with vertex set $\mathbf{V}$, with an edge from T to U (notation $T \to U$) if in $\mathbf{P}$ there is a production $T \vdash u$ with u containing U (compare e.g. with [9]). We write $T \overset{*}{\to} U$ if in $\mathscr{D}$ there is an oriented path of length $\geqslant 0$ from T to U.

Consider the equivalence relation on $\mathbf{V}$ where $T \sim U$ if $T \overset{*}{\to} U$ and $U \overset{*}{\to} T$. The equivalence classes, denoted $\mathbf{V}_j$, $j = 0, \ldots, N$ (with $S \in \mathbf{V}_0$), are called the *strong components* of $\mathscr{D}(\mathscr{C})$. The strong components are partially ordered: $\mathbf{V}_j \preccurlyeq \mathbf{V}_k$ if there is an oriented path from $T \in \mathbf{V}_j$ to U in $\mathbf{V}_k$ (independent of the choice of representatives).

Definition 5. A context-free grammar $\mathscr{C}$ is called *ergodic* if the dependency di-graph $\mathscr{D}(\mathscr{C})$ is strongly connected, i.e., it consists of a single strong component.

If $\mathscr{C}$ is linear then we require in addition that every terminal word $w \in \mathbf{\Sigma}^*$ occurs in a non-terminal sentential form (see [3, Section 5] for details).

A context-free (resp. linear/regular) language L is ergodic, if it is generated by an ergodic, reduced context-free (resp. linear/right linear) grammar. If this grammar is also unambiguous, we say that L is an ergodic, unambiguous language of the respective type.

We shall say that the start symbol S of $\mathscr{C}$ is *isolated*, if it does not occur in the right-hand side of any production rule. For any language $L \subset \mathbf{\Sigma}^*$, we define its *subword closure* as $SUB(L) = \{v \in \mathbf{\Sigma}^* : v \text{ is a subword of } w \text{ for some } w \in L\}$.

Definition 6. Let $\mathscr{C}$ be a reduced context-free grammar with $|L(\mathscr{C})| = \infty$.

A strong component $\mathbf{V}_j$ of $\mathscr{D}(\mathscr{C})$ is called *left*, resp. *right linear* (for both in short *regular*), resp. *linear*, if the following holds for every production $T \vdash w$ with $T \in \mathbf{V}_j$: the word w contains at most one element of $\mathbf{V}_j$ which, if present, must be in the leftmost (resp. rightmost, resp. any) position of w. In this case, every variable in $\mathbf{V}_j$ is also called regular, resp. linear. We write $\mathbf{V}_{\text{ess}}$ for the set of all *essential*, i.e., non-regular variables.

The grammar $\mathscr{C}$ is called *essentially ergodic* if

(i) the set $\mathbf{V}_{\text{ess}}$ forms a single strong component and

(ii) for each $w \in L(\mathscr{C})$, there is $T \in \mathbf{V}_{\text{ess}}$ such that $w \in SUB(L_T)$.

This definition is useful only when L is a context-free, non-regular language:

Lemma 7. *If all variables are regular, then $L(\mathscr{C})$ is a regular language.*

Proof. We associate with each strong component $\mathbf{V}_j$ a grammar $\mathscr{C}_j = (\mathbf{V}_j, \mathbf{\Sigma}_j, \mathbf{P}_j, S_j)$. Here, $\mathbf{P}_j$ is the set of productions of $\mathscr{C}$ whose left-hand side is in $\mathbf{V}_j$, and writing $\mathbf{W}_j$ for the set of all $T \in \mathbf{V} \setminus \mathbf{V}_j$ that occur in a right-hand side of some $\mathbf{P}_j$, the alphabet is $\mathbf{\Sigma}_j = \mathbf{\Sigma} \cup \mathbf{W}_j$. The start symbol S_j is any element of $\mathbf{V}_j$. Then, by assumption, each grammar $\mathscr{C}_j$ is either left or right linear.

We now use induction with respect to $\preccurlyeq$. If $\mathbf{V}_j$ is maximal with respect to this partial order, then $\mathbf{W}_j = \emptyset$, whence L_T, the language generated by $T \in \mathbf{V}_j$ with respect to $\mathscr{C}$, coincides with $L(\mathscr{C}_j)$ when we chose $S_j = T$. Thus L_T is rational.

Now suppose that given $\mathbf{V}_j$, we have already that L_U (with respect to $\mathscr{C}$) is rational for all $U \in \mathbf{W}_j$. Then the Substitution Theorem for regular languages, see [7, Section 3.4], implies that L_T is regular for every $T \in \mathbf{V}_j$. □

3. The 2-block language: ergodicity and unambiguity

Theorem 8. *Let $\mathscr{C}$ be a context-free grammar that generates the language L. Then we can construct a context-free grammar $\mathscr{C}_{(2)}$ which generates $L_{(2)} = \phi(L) \setminus \{\varepsilon\}$ and has the following properties*:

- *The graphs $\mathscr{D}(\mathscr{C})$ and $\mathscr{D}(\mathscr{C}_{(2)})$ have the same number of non-regular strong components.*
- *If $\mathscr{C}$ is right linear/left linear/linear/unambiguous/essentially ergodic then so is $\mathscr{C}_{(2)}$.*

Proof. Note that it is no restriction to work with $L_{(2)} = \phi(L) \setminus \{\varepsilon\}$ instead of $\phi(L)$.

In order to outline our approach for the general case, we first present the easy proof for a right linear, ε-free grammar $\mathscr{C} = (\mathbf{V}, \mathbf{\Sigma}, \mathbf{P}, S)$. Thus, the productions are of the form

$$T \vdash a_1 a_2 \cdots a_n, \quad T \vdash a_1 a_2 \cdots a_n U$$

with $T, U \in \mathbf{V}$, $a_i \in \mathbf{\Sigma}$ and $n \in \mathbb{N}$. Consider the new grammar $\mathscr{C}_{(2)} = (\mathbf{V}_{(2)}, \mathbf{\Sigma}_{(2)}, \mathbf{P}_{(2)}, [S])$ over the alphabet $\mathbf{\Sigma}_{(2)}$ as in (2) where $\mathbf{V}_{(2)} = \{[S], [aU]$: there is a production in $\mathbf{P}$ of the form $T \vdash a_1 \cdots a_{n-1} a U\}$, and the new productions are

$$[S] \vdash \phi(w) \mid \phi(w)[a_n T] \quad \text{if } S \vdash w \mid wT \quad \text{in } \mathbf{P}$$

and

$$[aT] \vdash (aa_1)\phi(w) \mid (aa_1)\phi(w)[a_n U], \quad \text{if } T \vdash w \mid wU \quad \text{in } \mathbf{P},$$

with $w = a_1 a_2 \cdots a_n \in \mathbf{\Sigma}^*$ and $T, U \in \mathbf{V}$. Then it is obvious that $L(\mathscr{C}_{(2)}) = L_{(2)}$ (see below for the details in the general case) and it is easy to check that if $\mathscr{C}$ is unambiguous, then so is $\mathscr{C}_{(2)}$. The new start symbol $[S]$ is isolated, but if $\mathscr{C}$ is ergodic, then all the other variables of $\mathscr{C}_{(2)}$ form a single strong component.

Now let $\mathscr{C}$ be a general context-free grammar generating the language L. To obtain $\mathscr{C}_{(2)}$ in a similar manner, we need to apply the above construction to a grammar that is standardized suitably. Recall that two grammars are called *equivalent* if they generate the same language. We have already said in the introduction that we always assume our grammars to be reduced (there are no superfluous variables).

(1) Transform the (reduced) grammar $\mathscr{C}$ into a grammar $\mathscr{C}'$ which is ε-free (there is no rule of the form $T \vdash \varepsilon$) and generates $L \setminus \{\varepsilon\}$. There is a simple algorithm for passing from $\mathscr{C}$ to $\mathscr{C}'$ that generates $L \setminus \{\varepsilon\}$; see e.g. [7, Section 4.3]. A small modification yields an algorithm that decreases the ambiguity degrees and preserves being reduced and the number of non-regular strong components, see [3].

(2) Eliminate all *chain rules*, i.e., productions of the form $T \vdash U$, where $T, U \in \mathbf{V}$. Again, there is a simple algorithm that transforms a reduced grammar $\mathscr{C}'$ into an equivalent reduced grammar $\mathscr{C}''$ without chain rules, see e.g. [7, Section 4.3] or [10, Corollary 5.3]. It is easy to check that it decreases the ambiguity degrees and preserves ε-freeness and the number of non-regular strong components as well as (essential) ergodicity.

(3) Transform the grammar $\mathscr{C}''$ into an equivalent grammar $\mathscr{C}'''$ in what we call *binary form* (BF), where each right-hand side of a production rule is contained in the set $\mathbf{\Sigma} \cup \mathbf{\Sigma}^2 \cup \mathbf{\Sigma V} \cup \mathbf{V \Sigma} \cup \mathbf{V}^2$. Passing from any reduced, chain-rule-free and ε-free grammar to an equivalent one in BF is simple and similar to passing to Chomsky normal form. We do not use the latter, since we want to preserve right, resp. left linearity of single production rules. Again, the algorithm preserves being reduced, ambiguity degrees, the number of non-regular components and (essential) ergodicity, see [3, Proposition 1].

(4) Use a slight variation of the algorithm described in [10, Theroem 5.9] to pass from a grammar in BF to an equivalent grammar $\bar{\mathscr{C}} = (\bar{\mathbf{V}}, \mathbf{\Sigma}, \bar{\mathbf{P}}, \bar{S})$ in *operator normal form* (ONF). Following [10], this algorithm is such that the start symbol $\bar{S}$ of $\bar{\mathscr{C}}$ is isolated and the right-hand side of every production is in

$$\mathbf{\Sigma} \cup \mathbf{\Sigma}\bar{\mathbf{V}} \cup \bar{\mathbf{V}}\mathbf{\Sigma} \cup \bar{\mathbf{V}}\mathbf{\Sigma}\bar{\mathbf{V}}.$$

The ambiguity degrees with respect to $\mathscr{C}'''$ and with respect to $\bar{\mathscr{C}}$ are the same. It is not completely immediate that this algorithm preserves essential ergodicity, but it does, see [3, Proposition 2]. The proof of the latter proposition in [3] also shows that the number of non-regular components is preserved.

(5) We can now finally explain the construction of a grammar $\mathscr{C}_{(2)}$ producing $L_{(2)} = \phi(L) \backslash \{\varepsilon\}$. By the above we can assume that $L \backslash \{\varepsilon\}$ is generated by a context-free (reduced) grammar $\mathscr{C} = (\mathbf{V}, \mathbf{\Sigma}, \mathbf{P}, S)$ in ONF with isolated start symbol, and whose production rules have their right-hand side in $\mathbf{\Sigma} \cup \mathbf{\Sigma V} \cup \mathbf{V \Sigma} \cup \mathbf{V \Sigma V}$.

A *sentential form* is any element w of $(\mathbf{\Sigma} \cup \mathbf{V})^*$ such that there is a rightmost derivation $T \overset{*}{\Rightarrow} w$ for some $T \in \mathbf{V}$. Since $\mathscr{C}$ is in ONF, a sentential form cannot have any subword in $\mathbf{V}^2$, and looks as follows:

$$w = T_1 v_1 T_2 v_2 \cdots T_k v_k T_{k+1}, \tag{3}$$

where $v_i \in \mathbf{\Sigma}^+$, $T_i \in \mathbf{V}$ and possibly T_1 and/or T_{k+1} may be missing. Since $\mathscr{C}$ is reduced, every v_i is in $SUB(L)$. Let a_i and b_i be the first and last letters of v_i, respectively.

We now transform each sentential form w as in (3) into a new expression $\Phi(w)$ by using ϕ and inserting brackets as follows:

$$\Phi(w) = [T_1 a_1]\phi(v_1)[b_1 T_2 a_1]\phi(v_2) \cdots [b_{k-1} T_k a_k]\phi(v_k)[b_k T_{k+1}],$$

where $[T_1 a_1]$ and/or $[b_k T_{k+1}]$ will be missing when T_1 and/or T_{k+1} are missing in w.

We now exhibit the new grammar $\mathscr{C}_{(2)} = (\mathbf{V}_{(2)}, \mathbf{\Sigma}_{(2)}, \mathbf{P}_{(2)}, [S])$ for $L_{(2)}$ with the new start symbol $[S]$. The set of variables $\mathbf{V}_{(2)}$ consists of $[S]$ and all expressions $[Ta]$, $[bT]$ and $[bTa]$ that appear in some $\Phi(w)$, where w is a sentential form of $\mathscr{C}$ (it is easy to write down a "greedy" algorithm for finding all of them). The next list displays the rules in $\mathbf{P}$ followed by the corresponding new rules in $\mathbf{P}_{(2)}$.

$$\begin{aligned}
&\text{If } S \vdash bT\text{: } [S] \vdash [bT],\\
&\text{if } S \vdash Ta\text{: } [S] \vdash [Ta],\\
&\text{if } S \vdash TbU\text{: } [S] \vdash [Tb][bU],\\
&\text{if } T \vdash c\text{: } [Ta] \vdash (ca),\ [bT] \vdash (bc),\ [bTa] \vdash (bc)(ca),\\
&\text{if } T \vdash cU\text{: } [Ta] \vdash [cUa],\ [bT] \vdash (bc)[cU],\ [bTa] \vdash (bc)[cUa],\\
&\text{if } T \vdash Uc\text{: } [Ta] \vdash [Uc](ca),\ [bT] \vdash [bUc],\ [bTa] \vdash [bUc](ca),\\
&\text{if } T \vdash UcV\text{: } [Ta] \vdash [Uc][cVa],\ [bT] \vdash [bUc][cV],\ [bTa] \vdash [bUc][cVa].
\end{aligned} \tag{4}$$

Here, $T, U, V \in \mathbf{V} \backslash \{S\}$ and $a, b, c \in \mathbf{\Sigma}$ have to be such that the occurring expressions in brackets belong to $\mathbf{V}_{(2)}$.

It is clear that from $\Phi(w)$ one can reconstruct w. That is, the mapping Φ is one-to-one. Also, the restriction of Φ to L coincides with ϕ. Thus, by the construction of

$\mathscr{C}_{(2)}$, for any sequence of sentential forms $w_1, w_2, \ldots, w_n$ with respect to $\mathscr{C}$, we have

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n \quad \text{in } \mathscr{C} \quad \text{if and only if}$$

$$[S] \Rightarrow \Phi(w_1) \Rightarrow \Phi(w_2) \Rightarrow \cdots \Rightarrow \Phi(w_n) \quad \text{in } \mathscr{C}_{(2)}.$$

Consequently, $\mathscr{C}_{(2)}$ generates $\phi(L) \setminus \{\varepsilon\} = L_{(2)}$, and the ambiguity degrees are preserved, that is, $d_{[S]}(\phi(w)) = d_S(w)$ for every $w \in L$ with $|w| \geqslant 2$.

We note that $\mathscr{C}_{(2)}$ has chain rules, which can be eliminated by applying the transformation of Step (2) but are "harmless" anyway (they cannot be concatenated into an infinite loop). Thus, we continue to work with $\mathscr{C}_{(2)}$. The remaining difficulties are to show that the number of non-regular components as well as essential ergodicity are preserved.

For $T \in \mathbf{V}$, we write $desc(T)$ for the set of all variables of the form $[bT]$, $[Ta]$, $[bTa]$ in $\mathbf{V}_{(2)}$, and for $\mathbf{W} \subset \mathbf{V}$, let $desc(\mathbf{W}) = \bigcup_{T \in \mathbf{W}} desc(T)$. When $T, U \in \mathbf{V}$ belong to different strong components of $\mathscr{D}(\mathscr{C})$, then elements of $desc(T)$ and $desc(U)$ clearly cannot lie in the same strong component of $\mathscr{D}(\mathscr{C}_{(2)})$.

Now consider a non-regular strong component $\mathbf{V}_j \subset \mathbf{V}$. We say that $[bTa]$ is an *interior* variable of $desc(\mathbf{V}_j)$ if $T \in \mathbf{V}_j$ and the string bTa occurs in a sentential form of $\mathscr{C}$ that derives from some element of $\mathbf{V}_j$. Then the proof of [3, Proposition 3] shows the following: the interior variables of $desc(\mathbf{V}_j)$ form a non-regular strong component of $\mathscr{D}(\mathscr{C}_{(2)})$, and all other variables of $desc(\mathbf{V}_j)$ are regular in the sense of Definition 6 above. Now, it is immediate to see that for every regular variable $T \in \mathbf{V}$, every element of $desc(T)$ is regular with respect to $\mathscr{C}_{(2)}$. Thus, $\mathscr{D}(\mathscr{C})$ and $\mathscr{D}(\mathscr{C}_{(2)})$ have the same number of non-regular strong components.

In [3, Proposition 3] we also showed that when there is only one non-regular strong component in $\mathscr{D}(\mathscr{C})$, then point (ii) of essential ergodicity is also preserved when passing to $\mathscr{C}_{(2)}$. □

4. Algebraic equations associated with context-free grammars

We now give an outline of Step 2 of our proof-strategy. We first present a general result regarding systems of algebraic equations based on [11], see [3] and, for previous variants, [12,5,8;17, Section 19.B] or [15].

Let $f_i(z) = \sum_{n \geqslant 0} f_{i,n} z^n$, $i = 1, \ldots, v$, be the generating functions of non-negative sequences, and let $r_i \leqslant \infty$ be the radius of convergence of $f_i(z)$. Each r_i has to be a singularity of $f_i(z)$. We suppose that $f_i(0) = 0$ and that $r = \min_i r_i > 0$. This is the number which we want to study. We assume that the $f_i(z)$ satisfy a system of equations

$$f_i(z) = \mathscr{Q}_i(z, f_1(z), \ldots, f_v(z)), \quad i = 1, \ldots, v, \tag{5}$$

where

$$\mathscr{Q}_i(z, y_1, \ldots, y_v) = \sum_{\mathbf{n}} a_{i,\mathbf{n}}(z) \mathbf{y}^{\mathbf{n}}, \quad z \in \mathbb{C}, \; i = 1, \ldots, v$$

are polynomials in the variables $y_1, \ldots, y_v$ ($\mathbf{y} = (y_1, \ldots, y_v)$, $\mathbf{n} = (n_1, \ldots, n_v) \in \mathbb{N}_0^v$, and $\mathbf{y}^{\mathbf{n}} = y_1^{n_1} \cdots y_v^{n_v}$). We further assume that the coefficient functions $a_{i,\mathbf{n}}(z)$ are not all constant and are also expressed as power series around $z=0$ with non-negative coefficients and radii of convergence $R_{i,\mathbf{n}}$ such that $R=\min_i R_i>0$, where $R_i=\min_{\mathbf{n}} R_{i,\mathbf{n}}$.

The *dependency di-graph* $\mathscr{D}$ of our system (5) of equations has vertex set $\{1,\dots,v\}$, and there is an oriented edge from i to j (notation $i\to j$), if y_j appears in a non-zero term of $\mathscr{Q}_i(z,y_1,\dots,y_v)$. The following is obvious.

Lemma 9. *If $i\to j$, then $r_i\leqslant\min\{r_j,R_i\}$. Thus $r_i=r_{[i]}$ depends only on the strong component $[i]$ of i in $\mathscr{D}$, and either $f_j(r_{[i]})<\infty$ for all $j\in[i]$ or $f_j(r_{[i]})=\infty$ for all $j\in[i]$. In particular, if $\mathscr{D}$ is strongly connected, then $r_i=r\leqslant R$ for all i.*

The Jacobian matrix of system (5),

$$\mathfrak{J}(z)=\left(\frac{\partial\mathscr{Q}_i}{\partial y_j}(z,f_1(z),\dots,f_v(z))\right)_{i,j=1}^{v},\quad 0\leqslant z\leqslant r$$

is non-negative with entries that are increasing in z. By our assumptions, $\mathfrak{J}(0)$ is the zero matrix, and for $0<z<r$, we have $\mathfrak{J}(z)_{i,j}>0$ if and only if $i\to j$ in $\mathscr{D}$. By the theory of non-negative matrices [16], there is a positive eigenvalue $\lambda(z)$ of $\mathfrak{J}(z)$ with maximal absolute value. If $\mathscr{D}$ is strongly connected and $z>0$ then $\lambda(z)$ has algebraic and geometric multiplicity 1 and strictly positive left and right eigenvectors. We have $\lambda(0)=0$, and $\lambda(z)$ increase with z. More generally, given a strong component $[i]$, consider the restriction $\mathfrak{J}_{[i]}(z)$ of $\mathfrak{J}(z)$ to $[i]$, defined for $0\leqslant z<r_{[i]}$, and write $\lambda_{[i]}(z)$ for its Perron–Frobenius eigenvalue.

The basic result is the following, proved in [3, Section 3].

Proposition 10. *Suppose that the generating functions $f_i(z)$, $i=1,\dots,v$, satisfy a system of polynomial equations of form (5), let $r=\min_i r_i$ be their minimal radius of convergence, R the minimal radius of convergence of the coefficient functions of the $\mathscr{Q}_i$, and $\mathfrak{J}(z)$ the Jacobian matrix of the system (as above). Then $r<R$ if and only if there are $z\in(0,R)$ and i such that $\lambda_{[i]}(z)=1$.*

If this is the case, then $\mathfrak{J}(r)$ is finite and there is an $i\in\{1,\dots,v\}$ such that $r=\min\{z>0:\lambda_{[i]}(z)=1\}$.

Now let L be generated by an essentially ergodic, unambiguous context-free grammar $\mathscr{C}=(\mathbf{V},\mathbf{\Sigma},\mathbf{P},S)$ that can be assumed, without loss of generality, to be reduced, ε-free and without chain rules. For $T\in\mathbf{V}$, consider the power series $f_T(z)=f_{L_T}(z)$ according to (1) and the definition of L_T. If r_T denotes its radius of convergence, then $\gamma(L)=1/r_S$.

Besides the complex variable z, we introduce complex variables y_T, $T\in\mathbf{V}$. We define $\pi(a)=z$ for every $a\in\mathbf{\Sigma}$ and $\pi(T)=y_T$ for every $T\in\mathbf{V}$, and for $u=u_1\cdots u_k\in(\mathbf{\Sigma}\cup\mathbf{V})^*$, let $\pi(u)=\pi(u_1)\cdots\pi(u_k)$, a product of commuting complex variables. With $T\in\mathbf{V}$ we associate the polynomial

$$\mathscr{P}_T(z;y_U,U\in\mathbf{V})=\sum_{T\vdash u}\pi(u)\tag{6}$$

in the complex variables z and y_U, $U\in\mathbf{V}$. A famous theorem of Chomsky and Schützenberger [4] implies that the functions $f_T(z)$ satisfy the system of equations

$$f_T(z)=\mathscr{P}_T(z;f_U(z),U\in\mathbf{V}),\quad T\in\mathbf{V}. \tag{7}$$

This system has form (5) with $R=\infty$, since all coefficient functions are polynomials in z. The dependency di-graph of this system is $\mathscr{D}(\mathscr{C})$. By Lemma 9, $r=r_S=\min\{r_T : T\in\mathbf{V}\}$, since $S\xrightarrow{*}T$ for all T, and more generally, $r_T\leqslant r_U$ if $T\xrightarrow{*}U$. For variables in the same strong component, the radii of convergence coincide.

The following lemma does not require essential ergodicity, but if the latter holds and L is of convergent type, it will tell us which strong component is "responsible" for the growth of L. Recall (Definition 6) the concept of a linear strong component.

Lemma 11. *Suppose that $L(\mathscr{C})$ is infinite.*

(a) *If $\mathscr{C}$ is of convergent type then it has non-linear variables, and there is a non-linear strong component $\mathbf{V}_j$ such that $r_T=r$ and $f_T(r)<\infty$ for every $T\in\mathbf{V}_j$.*

(b) *On the other hand, if all variables are linear, then $f_T(z)$ is a rational function for each $T\in\mathbf{V}$.*

Proof (Outline). Statement (b) is the commuting-variables-analogue of Lemma 7. Note that the difference between "right linear", "left linear" and "linear" disappears in the commutative case.

For (a), note that r must be a singularity of $f_S(z)$, and $f_S(r)<\infty$. Now, if $\mathbf{V}_j$ is a linear strong component, then system (7) restricted to $\mathbf{V}_j$ is linear. Therefore, if $T\in\mathbf{V}_j$ then $f_T(z)$ is a rational function of z and the functions $f_U(z)$, where $T\xrightarrow{*}U$ and $U\notin\mathbf{V}_j$. Singularities of rational functions can only be poles. Therefore, the singularity r must come from a non-linear component. □

Now let $F\subset\mathbf{\Sigma}$ be a set of forbidden *letters.* Then L^F is generated by the grammar $\mathscr{C}^F=(\mathbf{V},\mathbf{\Sigma}\backslash F,\mathbf{P}^F,S)$, where $\mathbf{P}^F$ is the set of all productions in $\mathbf{P}$ whose right-hand sides contain no element of F. Of course, $\mathscr{C}^F$ is not necessarily reduced and essentially ergodic. However, it is important to note that $\mathscr{C}^F$ is unambiguous when $\mathscr{C}$ is unambiguous. Step 2 is accomplished by the following.

Proposition 12. *Suppose that L is generated by the essentially ergodic, reduced, unambiguous context-free grammar $\mathscr{C}$ without chain and ε-rules. If $\mathscr{C}$ is of convergent type, then for any non-empty $F\subset\mathbf{\Sigma}$, $\gamma(L^F)<\gamma(L)$.*

Proof (Slightly condensed). We know from Lemma 11 that $1/\gamma(L)=r=r_T$ for all $T\in\mathbf{V}_{\text{ess}}$. Hence we restrict system (7) to the variables of $\mathbf{V}_{\text{ess}}$. We number $\mathbf{V}_{\text{ess}}=\{T_1,T_2,\ldots,T_v\}$ and write $y_i=y_{T_i}$ and $f_i(z)=f_{T_i}(z)$. For each $i\in\{1,\ldots,v\}$, we define a polynomial $\mathscr{P}_i(z,y_1,\ldots,y_v)$ in the y_i by substituting in $\mathscr{P}_{T_i}$ for each appearing y_U with $U\notin\mathbf{V}_{\text{ess}}$ the corresponding function $f_U(z)$. The latter $U\in\mathbf{V}$ must be such that $T_i\rightarrow U$ in $\mathscr{D}(\mathscr{C})$. By Lemma 11, $f_U(z)$ is rational.

We have obtained a system that is precisely of form (5), its dependency di-graph is strongly connected, and the coefficient functions are generating functions of non-negative sequences that are either polynomials or rational functions. Therefore, either $R=\infty$ or else R is a pole of one of the coefficient functions. Since L is infinite, $r<\infty$, and $f_i(r)<\infty$ for all i by Lemma 11. Thus, Proposition 10 applies, and $r=\min\{z>0:\ \lambda(z)=1\}$.

Next, consider the grammar $\mathscr{C}^F$, where $F\subset\mathbf{\Sigma}$. We eliminate from $\mathbf{V}$ all variables that cannot be reached from S in the dependency di-graph of $\mathscr{C}^F$, thus obtaining a set of variables $\mathbf{V}^F\subset\mathbf{V}$ and the corresponding, suitably numbered subset $\mathbf{V}^F_{\mathrm{ess}}=\{T_1,\dots,T_{v'}\}\subset\mathbf{V}_{\mathrm{ess}}$. Also, we eliminate from $\mathbf{P}^F$ the production rules that contain some $U\in\mathbf{V}\backslash\mathbf{V}^F$. For simplicity, write again $\mathbf{P}^F$ for this new set of production rules, and $\mathscr{C}^F=(\mathbf{V}^F,\mathbf{\Sigma},\mathbf{P}^F,S)$. For the associated polynomials, Jacobian matrix, Perron–Frobenius eigenvalue, etc., we use the superscript F. Thus $1/\gamma(L^F)=r^F=r^F_S$. We have $r^F\geqslant r_S=r$ and want to show that $r^F>r$ strictly.

Case 1: $\mathscr{C}^F$ is of divergent type. Then $r_F>r$ since otherwise $f^F_S(r)\leqslant f_S(r)<\infty$.

Case 2: $\mathscr{C}^F$ is of convergent type. Then $\mathscr{D}(\mathscr{C}^F)$ must have non-regular (in fact non-linear) strong components by Lemma 11. Since $\mathscr{D}(\mathscr{C}^F)$ is obtained from $\mathscr{D}(\mathscr{C})$ by deleting some edges and vertices, each strong component of $\mathscr{D}(\mathscr{C}^F)$ is—as a set of variables—contained in some strong component of $\mathscr{D}(\mathscr{C})$. In particular, every non-regular strong component of $\mathscr{D}(\mathscr{C}^F)$ must be contained in $\mathbf{V}_{\mathrm{ess}}$. Lemma 11 now implies that

$$r^F=\min\{r^F_i:\ i=1,\dots v'\},\quad \text{where}\quad r^F_i=r^F_{T_i}.$$

Every $U\in\mathbf{V}^F\backslash\mathbf{V}_{\mathrm{ess}}$ is a regular variable of $\mathscr{C}^F$, and if it occurs in the right-hand side of some $T\vdash u$ in $\mathbf{P}^F$, where $T\in\mathbf{V}_{\mathrm{ess}}$, then Lemma 11 shows that $f^F_U(z)$ is rational.

For $i=1,\dots,v'$, we can now construct the polynomials $\mathscr{P}^F_i(z,y_1,\dots,y_{v'})$ in the variables $y_1,\dots,y_{v'}$ in the same way as we did above for $\mathscr{P}_i(z,y_1,\dots,y_v)$. Their coefficient functions are rational functions by what we just said. Since L^F is of convergent type and denoting by R^F the minimum among the radii of convergence of these coefficient functions (which are poles or $=\infty$), we have $r^F<R^F$.

Thus, we can apply Proposition 10. The dependency-digraph $\mathscr{D}^F$ of the $\mathscr{P}^F_i$, $i=1,\dots,v'$, is a subgraph of $\mathscr{D}(\mathscr{C}^F)$. There is a strong component $[k]:=\mathbf{V}^F_k$ of $\mathscr{D}^F$ such that $r^F=\min\{z>0:\ \lambda^F_{[k]}(z)=1\}$. Extend the matrix $\mathfrak{J}^F_{[k]}(z)$ to a matrix over $\{1,\dots,v\}$ by setting all elements outside of $[k]\times[k]$ equal to 0. We also write $\mathfrak{J}^F_{[k]}(z)$ for this extended matrix.

For $i\in\{1,\dots,v'\}$, consider the generating functions $f^F_i(z)$ and the $f_i(z)$. Then clearly $r^F_i\geqslant r$. Condition (ii) in the definition of essential ergodicity implies that $f^F_i(z)<f_i(z)$ strictly for $0<z\leqslant r$, as L_{T_i} contains words having a letter in F. This yields that for all $z\in(0,r]$, $i\in\{1,\dots,v'\}$,

$$\mathscr{P}^F_i(z,f^F_1(z),\dots,f^F_{v'}(z))<\mathscr{P}_i(z,f_1(z),\dots,f_v(z)).$$

Again, since L^F is of convergent type, there must be some rule of the form $T\vdash u$ in $\mathbf{P}^F$ such that $T\in\mathbf{V}_{\mathrm{ess}}\cap\mathbf{V}^F$ and u contains at least two (not necessarily distinct) elements of $\mathbf{V}_{\text{ess}} \cap \mathbf{V}^F$. Indeed, otherwise the system

$$f_i^F(z) = \mathscr{P}_i^F(z, f_1^F(z), \ldots, f_{v'}^F(z)), \quad i = 1, \ldots, v',$$

would be linear, its solutions rational functions in z, and their singularities would be poles. Putting all these facts together, we get that

$$\mathfrak{J}_{[k]}^F(z) \leqslant \mathfrak{J}(z) \quad \text{and} \quad \mathfrak{J}_{[k]}^F(z) \neq \mathfrak{J}(z) \quad \text{for all } z \in (0, r].$$

But then [16, Theorem 1.6] implies that $\lambda_{[k]}^F(z) < \lambda(z) \leqslant \lambda(r) = 1$ for all $z \in (0, r]$. This yields that $r^F > r$, thus concluding the proof. □

5. Context-free languages associated with finite state automata

We now present a class of examples.

A regular language is one that is generated by a right linear grammar, or equivalently, accepted by a finite state automaton. The latter is a di-graph $\mathscr{D}$ with vertex set (set of *states*) $\mathbf{V}$, where each oriented edge carries a label from $\boldsymbol{\Sigma}$. There are one specified *initial state* S and a set $\mathbf{F} \subset \mathbf{V}$ of *final states.* The language $L(\mathscr{D})$ *accepted by* $\mathscr{D}$ consists of all words over $\boldsymbol{\Sigma}$ that are obtained by reading the successive labels along a path in $\mathscr{D}$ that starts in S and ends at a vertex of $\mathbf{F}$ (by a path we mean a sequence of oriented edges where the endpoint of one is the initial point of the next one; repetitions are permitted as well as paths with length 0). We admit multiple edges, but edges with the same initial and end points must have different labels. We remark that it is well known that every regular language is accepted by a *deterministic* finite state automaton, i.e., one where edges with the same initial vertex must have different labels.

Now consider a finite state automaton $\mathscr{D}$, not yet necessarily deterministic. We can construct the (rooted, labelled, oriented) *covering tree* $\mathscr{T} = \mathscr{T}_{\mathscr{D}}$. Its vertex set is the set of all (oriented) paths in $\mathscr{D}$ starting at S, including the empty path $\mathbf{o}$ which is the root of $\mathscr{T}$. If $\mathbf{x}, \mathbf{y}$ are two paths, then by definition there is an oriented edge in $\mathscr{T}$ from $\mathbf{x}$ to $\mathbf{y}$ when the path $\mathbf{y}$ extends $\mathbf{x}$ by one, final edge of $\mathscr{D}$. The label of the latter edge also labels the edge $\mathbf{x} \to \mathbf{y}$. Also, we label each vertex $\mathbf{x}$ of $\mathscr{T}$ with the endvertex $T(\mathbf{x})$ of the path $\mathbf{x}$ in $\mathscr{D}$, and $T(\mathbf{o}) = S$. Then $\mathscr{T}$ has *finitely many cone types.* Indeed, if $\mathbf{x}$, $\mathbf{x}'$ are vertices of $\mathscr{T}$ with $T(\mathbf{x}) = T(\mathbf{x}')$ then there is a natural isomorphism between the subtrees $\mathscr{T}_{\mathbf{x}}$ and $\mathscr{T}_{\mathbf{x}'}$ of $\mathscr{T}$ that are rooted at $\mathbf{x}$ and $\mathbf{x}'$, respectively. This isomorphism preserves all labels.

We now augment the alphabet $\boldsymbol{\Sigma}$ by a disjoint copy $\bar{\boldsymbol{\Sigma}} = \{\bar{a} : a \in \boldsymbol{\Sigma}\}$. We shall consider non-oriented paths in $\mathscr{T}$. If $\mathbf{x} \to \mathbf{y}$ is an edge of T with label $a \in \boldsymbol{\Sigma}$, then we shall read this label when walking along the edge in positive direction (away from the root); in the opposed direction we shall read the label $\bar{a}$. We call *restricted Dyck language* associated with $\mathscr{D}$ the language $L(\mathscr{T}_{\mathscr{D}})$ over the alphabet $\boldsymbol{\Sigma} \cup \bar{\boldsymbol{\Sigma}}$ consisting of all words that can be obtained by reading the successive labels along some closed path that starts and ends at the root.

This class of languages extends the classical *Dyck language* which corresponds to the automaton $\mathscr{D}$ where $\mathbf{V}=\{S\}$, and for each $a\in\mathbf{\Sigma}$ there is a loop with label a at S.

Proposition 13. *For any finite state automaton* $\mathscr{D}$, *the language* $L=L(\mathscr{T}_{\mathscr{D}})$ *is generated by the context-free grammar* $\mathscr{C}=\mathscr{C}_{\mathscr{D}}$ *with set of variables* $\mathbf{V}$ (*the vertex set of* $\mathscr{D}$), *start symbol* S (*the initial vertex of* $\mathscr{D}$) *and the production rules*

$$T\vdash\varepsilon \quad \text{and} \quad T\vdash aU\bar{a}T,$$

if in $\mathscr{D}$ *there is an edge from* T *to* U *with label* a.

In particular, the dependency di-graph of $\mathscr{C}$ *is* $\mathscr{D}$, *the grammar* $\mathscr{C}$ *is ergodic if and only if* $\mathscr{D}$ *is strongly connected, and* $\mathscr{C}$ *is unambiguous if and only if the automaton* $\mathscr{D}$ *is deterministic.*

Proof. For each $T\in\mathbf{V}$, consider the subtrees $\mathscr{T}_{\mathbf{x}}$ of $\mathscr{T}$ that are rooted at some $\mathbf{x}$ with $T(\mathbf{x})=T$. Clearly, $L(\mathscr{T}_{\mathbf{x}})$ is the same for all those $\mathbf{x}$, and $L=L(\mathscr{T}_{\mathbf{o}})$. As a variable of our grammar, T will be such that $L_T=L(\mathscr{T}_{\mathbf{x}})$, and for $w\in(\mathbf{\Sigma}\cup\bar{\mathbf{\Sigma}})^*$, the ambiguity degree $d_T(w)$ will be the number of paths in $\mathscr{T}_{\mathbf{x}}$ that start and end in $\mathbf{x}$ and are labelled by w.

In addition to $L(\mathscr{T}_{\mathbf{x}})$, we also introduce the language $L^{(0)}(\mathscr{T}_{\mathbf{x}})$ consisting of the labels of all paths as above with the restriction that we only admit non-trivial paths that return to the root $\mathbf{x}$ just once, at the end. With this language, we associate the auxiliary variable $T^{(0)}$, that will be eliminated later on. Every $w\in L^{(0)}(\mathscr{T}_{\mathbf{x}})$ has a decomposition of the form $w=av\bar{a}$, where a is the label of an edge $\mathbf{x}\to\mathbf{y}$ of $\mathscr{T}$ and $v\in L(\mathscr{T}_{\mathbf{x}})$. Let $U=T(\mathbf{y})$. Then corresponding to this edge we find the rule $T^{(0)}\vdash aU\bar{a}$. Furthermore, we have that

$$L(\mathscr{T}_{\mathbf{x}})=L^{(0)}(\mathscr{T}_{\mathbf{x}})^*,$$

the sub-monoid of $(\mathbf{\Sigma}\cup\bar{\mathbf{\Sigma}})^*$ generated by $L^{(0)}(\mathscr{T}_{\mathbf{x}})$. It is well known that this corresponds to the two rules $T\vdash\varepsilon$ and $T\vdash T^{(0)}T$. If in the latter of these two we replace $T^{(0)}$ with all possible right-hand sides of the auxiliary rules $T^{(0)}\vdash aU\bar{a}$ then we find the proposed grammar.

The statements about ergodicity and ambiguity are now obvious. □

In particular, if $\mathscr{D}$ is strongly connected, then Theorem 2(B) applies, and the corresponding restricted Dyck language is growth-sensitive.

Remark 14. (A) In [3, Section 5], we also prove an extended variant of the above Theorems 2 and 3, which take ambiguities into account. This is done by assigning integer *weights* (multiplicities) to the production rules.

(B) The above class of examples can be considered as a special case of the *context-free graphs* of [14]. We intend to pursue the study of (essential) ergodicity of the associated grammars in future work.

(C) An extended list of references, in particular including growth of groups and languages, can be found in [3].

References

[1] T. Ceccherini-Silberstein, F. Fiorenzi, F. Scarabotti, The Garden of Eden theorem for cellular automata and for symbolic dynamical systems, in: V.A. Kaimanovich, K. Schmidt, W. Woess (Eds.), Proceedings of the Workshop “Random Walks and Geometry” held at Erwin Schroedinger Institute for Mathematical Physics, Vienna, Austria, 18.06-13.07 2001, De Gruyter, Berlin, in press.

[2] T. Ceccherini-Silberstein, A. Machì, F. Scarabotti, On the entropy of regular languages, Theoret. Comput. Sci., this volume (2003).

[3] T. Ceccherini-Silberstein, W. Woess, Growth and ergodicity of context-free languages, Trans. Amer. Math. Soc. 354 (2002) 4597–4625.

[4] N. Chomsky, P.M. Schützenberger, The algebraic theory of context-free languages, in: P. Braffort, D. Hirschberg (Eds.), Computer Programming and Formal Systems, North-Holland, Amsterdam, 1963, pp. 118–161.

[5] M. Drmota, Systems of functional equations, Random Struct. Algebra 10 (1997) 103–124.

[6] R.I. Grigorchuk, P. de la Harpe, On problems related to growth, entropy, and spectrum in group theory, J. Dynam. Control Systems 3 (1997) 51–89.

[7] M. Harrison, Introduction to Formal Language Theory, Addison-Wesley Publishing Co., Reading, MA, 1978.

[8] S. Hersonsky, J. Hubbard, Groups of automorphisms of trees and their limit sets, Ergodic Theory Dynam. Systems 17 (1997) 869–884.

[9] W. Kuich, On the entropy of context-free languages, Inform. and Control 16 (1970) 173–200.

[10] W. Kuich, Semirings and formal power series: their relevance to formal languages and automata, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 1, Springer, Berlin, 1997, pp. 609–677.

[11] St.P. Lalley, Finite range random walk on free groups and homogeneous trees, Ann. Probab. 21 (1993) 2087–2130.

[12] St.P. Lalley, Random walks on regular languages and algebraic systems of generating functions, Algebraic methods in statistics and probability (Notre Dame, IN, 2000), Contemp. Math., Vol. 287, Amer. Math. Soc., Providence, RI, 2001, pp. 201–230.

[13] D. Lind, B. Marcus, An Introduction to Symbolic Dynamics and Coding, Cambridge University Press, Cambridge, 1995.

[14] D.E. Muller, P.E. Schupp, The theory of ends, pushdown automata, and second-order logic, Theoret. Comput. Sci. 37 (1985) 51–75.

[15] T. Nagnibeda, W. Woess, Random walks on trees with finitely many cone types, J. Theoret. Probab. 15 (2002) 383–422.

[16] E. Seneta, Non-negative Matrices and Markov Chains, 2nd Edition, Springer, New York, 1981.

[17] W. Woess, Random Walks on Infinite Graphs and Groups, Cambridge University Press, Cambridge, 2000.