ON DIFFERENTIATION FUNCTIONS, STRUCTURE FUNCTIONS, AND RELATED LANGUAGES OF CONTEXT-FREE GRAMMARS

JÜRGEN DASSOW[1], VICTOR MITRANA[2,3], GHEORGHE PĂUN[4] AND RALF STIEBE[1]

Abstract. We introduce the notion of a differentiation function of a context-free grammar which gives the number of terminal words that can be derived in a certain number of steps. A grammar is called narrow (or k-narrow) iff its differentiation function is bounded by a constant (by k). We present the basic properties of differentiation functions, especially we relate them to structure function of context-free languages and narrow grammars to slender languages. We discuss the decidability of the equivalence of grammars with respect to the differentiation function and structure function and prove the decidability of the k-narrowness of context-free grammars. Furthermore, we introduce languages representing the graph of the differentiation and structure function and relate these languages to those of the Chomsky hierarchy.

Mathematics Subject Classification. 68Q45, 68Q50.

Introduction

In biology, differentiation means the evolution of a variety of organisms which are modifications of the species from which they originate. The differentiation function gives the numerical size of the variety at a certain moment.

Lindenmayer systems form a description of the development of (lower) organisms on the basis of formal language theory. The differentiation function of a Lindenmayer system introduced in [2] corresponds to the above biological term and gives the number of words (representing organisms) which can be obtained in a certain number of derivation steps (representing steps of the development, as division of cells or changes of the state of cells etc.) from an axiom (representing the basic organism).

Analogously, the differentiation function of an evolutionary grammar – which describes an evolutionary process on the basis of formal language theory – gives the number of words (representing DNA sequences) which can be obtained in a certain number of derivation steps (representing mutations) from a set of axioms [3].

Surprisingly, the corresponding notion for "classical" Chomsky grammars – the function giving the number of (terminal) words derivable in a certain number of derivation steps – has not been studied yet and is the subject of this paper. Besides its analogy to the well motivated functions in formal language approaches to biological processes, there is a further reason to study the differentiation function of context-free grammars. This function is nearly related to the structure function of context-free languages which gives the number of words of a certain length in the language. The latter concept is well investigated (*e.g.* see [1, 10–12, 17]).

We call a grammar narrow iff its differentiation function is bounded by a constant. This concept is the counterpart of semi-discreet or slender languages, where the structure function is bounded by a constant. Slender languages were intensively studied in the last years (see [8, 9, 13–16]). In particular, context-free slender languages are finite unions of sets of the form $\{uv^n wx^n y \mid n \geq 0\}$.

The paper is organized as follows. In Section 1, we present the formal definitions and give some examples. Section 2 contains the basic results on the differentiation function, such as upper bounds, closure properties and relations to the structure function. In Section 3, we discuss some languages over a two letter alphabet which are derived from the differentiation function and the structure function and determine their places with respect to the Chomsky hierarchy. In Section 4, we present the undecidability of the equivalence problem for differentiation (or structure) functions and the decidability of k-narrowness for context-free grammars.

1. Definitions and examples

Throughout the paper we assume that the reader is familiar with the basic concepts of formal language theory. For the notions of context-free matrix and valence grammars and languages, we refer to [4].

For a language L, we define its *structure function* s_L by

$$s_L : \mathbb{N} \to \mathbb{N} \text{ and } s_L(n) = card\left(\{w \mid w \in L, |w| = n\}\right).$$

A language L is said to be *k-slender*, for $k \geq 1$, if $s_L(n) \leq k$ for all $n \geq 0$. A language is *slender* if it is k-slender for some $k \geq 1$.

For a grammar $G = (N, T, P, S)$ (with the set N of nonterminals, the set T of terminals, the set P of productions and the axiom S) and $n \geq 0$, we set

$$L_n(G) = \left\{w \mid w \in T^*, S \xRightarrow[G]{n} w\right\},$$

where $x \xRightarrow[G]{n} y$ means that x derives y in exactly n derivation steps, and define the differentiation function d_G of G by

$$d_G : \mathbb{N} \to \mathbb{N} \text{ and } d_G(n) = card(L_n(G)).$$

For $k \geq 1$, a grammar G is said to be *k-narrow*, if $d_G(n) \leq k$ for all $n \geq 1$. A grammar is *narrow* if it is k-narrow for some $k \geq 1$.

We define the *structural language* associated with a language L by

$$\sigma(L) = \left\{a^n b^{s_L(n)} \mid n \geq 0\right\},$$

and with a grammar G we associate the *derivational language*

$$\delta(G) = \left\{a^n b^{d_G(n)} \mid n \geq 1\right\}.$$

The languages $\sigma(L)$ and $\delta(G)$ characterize the graphs of the mappings s_L, d_G, respectively. Because these languages are bounded by $a^* b^*$, their context-freeness implies their linearity (see [6]).

Example 1.1. For the grammar $G_1 = (\{S\}, \{a\}, \{S \to aS, S \to a\}, S)$ and $n \geq 1$, we obtain

$$L_n(G_1) = \{a^n\} \quad \text{and} \quad d_{G_1}(n) = 1.$$

Thus G_1 is 1-narrow. Moreover,

$$\delta(G_1) = \{a^n b \mid n \geq 1\}.$$

Example 1.2. Let $G_2 = (\{S\}, \{a\}, \{S \to S, S \to aS, S \to a\}, S)$. Then

$$L_n(G_2) = \left\{a, a^2, \dots a^n\right\} \quad \text{and} \quad d_{G_2}(n) = n$$

for $n \geq 1$ and

$$\delta(G_2) = \{a^n b^n \mid n \geq 1\}.$$

Example 1.3. For the grammar $H_1 = (\{S, B\}, \{a, b\}, \{S \to aS, S \to bB, S \to b, B \to bB, B \to b\}, S)$ we get

$$L_n(H_1) = \{a^i b^{n-i} : 0 \leq i < n\} \quad \text{and} \quad d_{G_2}(n) = n \text{ for } n \geq 1,$$

and $\delta(H_1) = \sigma(L(H_1)) = \{a^n b^n \mid n \geq 1\}$.

By a product construction, one can find, for any $k \geq 2$, a regular grammar H_k with terminal alphabet $\{a, b\}^k$, where $L_n(H_k)$ contains all words whose i-th projections are in $L_n(H_1)$, and thus $d_{H_k}(n) = s_{L(H_k)}(n) = n^k$.

2. Remarks on the differentiation function

The structure function of a language over an alphabet X has the trivial upper bound $f(n) = card(X)^n$, which is sharp for the language X^*. A similar result holds for the differentiation function.

Theorem 2.1.

    (i) *For any grammar G, there is a constant c such that $d_G(n) \leq c^n$ for $n \geq 1$.*

    (ii) *For any natural number $c \geq 1$, there is a regular grammar G such that $d_G(n) = c^n$ for $n \geq 1$.*

Proof.

(i) Let $G = (N, T, P, S)$ be a grammar, $card(T) = k$, and let r be the greatest length of a right side of a rule in P. A word derived in n steps has at most length $r \cdot n$. Hence, the number of terminal words generated in n steps is bounded by $k^{r \cdot n + 1}$. Choosing $c = k^{r+1}$, the assertion follows.

(ii) Let $T = \{a_1, a_2, \ldots, a_c\}$ be a set consisting of c elements. Then the regular grammar

$$G = (\{S\}, T, \{S \to a_i S \mid 1 \leq i \leq c\} \cup \{S \to a_i \mid 1 \leq i \leq c\}, S)$$

satisfies $L_n(G) = T^n$ for $n \geq 1$, and therefore $d_G(n) = c^n$ holds. $\qquad\square$

Let us now turn to closure properties.

Theorem 2.2. *Let f and g be differentiation functions of context-free grammars and $k \in \mathbb{N}_+$. Then the functions*

    (i) *$f + g$ defined by $(f + g)(n) = f(n) + g(n)$, for $n \geq 0$;*

    (ii) *$k \cdot f$ defined by $(k \cdot f)(n) = k \cdot f(n)$, for $n \geq 0$;*

    (iii) *f_* defined by $f_*(n) = \sum_{i=1}^{n} f(i)$, for $n \geq 0$;*

    (iv) *$f_{[k]}$ defined by $f_{[k]}(n) = f(\lfloor \frac{n}{k} \rfloor)$, for $n \geq 0$;*

    (v) *$f_{[1/k]}$ defined by $f_{[1/k]}(n) = f(k \cdot n)$, for $n \geq 0$,*

are differentiation functions of context-free grammars, too.

Proof.

(i) Let $f = d_G$ and $g = d_H$, for two context-free grammars $G = (N_G, T_G, P_G, S_G)$ and $H = (N_H, T_H, P_H, S_H)$. Without loss of generality, we can assume that

$N_G \cap N_H = \emptyset$ and $T_G \cap T_H = \emptyset$. Consider the context-free grammar G' with the nonterminal $S \notin N_G \cup N_H$ and the rules $P_G \cup P_H \cup P$, where

$$P = \{S \to \alpha \mid S_G \to \alpha \in P_G \text{ or } S_H \to \alpha \in P_H\}.$$

Clearly, $d_{G'}(k) = d_G(k) + d_H(k) = f(k) + g(k)$, for $k \geq 1$.

(ii) Follows by repeated application of the previous construction.

(iii) Let $f = d_G$, for a context-free grammar $G = (N, T, P, S)$. Set $G' = (N \cup \{S'\}, T \cup \{a\}, P', S')$ with $S', a \notin N \cup T$ and

$$P' = P \cup \{S' \to \alpha \mid S \to \alpha \in P\} \cup \{S' \to aS'\}.$$

Then, we have $L_n(G') = \bigcup_{i=1}^{n} a^{n-i} L_i(G)$ and $d_n(G') = \sum_{i=1}^{n} d_i(G)$, as $a \notin T$.

(iv) Consider a context-free grammar $G = (N, T, P, S_0)$. One can easily construct a grammar $G' = (N', T, P', S_0)$, where one derivation step in G is simulated by k derivation steps in G. Hence, $f_{G'}(k \cdot n) = f(n)$ and $f_{G'}(k \cdot n + i) = 0$, for $n \geq 0$, $1 \leq i < k$. Finally, a grammar H is obtained from G' by adding the nonterminal symbols $\{S_1, \ldots, S_{k-1}\}$ and the set of rules

$$\{S_{i-1} \to S_i : 1 \leq i < k\} \cup \{S_i \to \alpha : S_0 \to \alpha \in P'\}.$$

Obviously, $f_H(m) = \sum_{i=0}^{k-1} f_{G'}(m - i)$, i.e., $f_H(k \cdot n + i) = f_{G'}(k \cdot n)$, for $0 \leq i < k$.

(v) Let $G = (N, T, P, S)$ be a context-free grammar. We construct a grammar H whose nonterminal alphabet is $N \times \{-k, \ldots, k\}$ with start symbol $(S, 0)$.

Consider a derivation $A \overset{\ell}{\underset{G}{\Longrightarrow}} \alpha$ in G, whose tree has a height between $(k+1)$ and $2k$. Let $\ell = (k \cdot m + i)$, $0 \leq i \leq k$, Clearly, $\ell \geq k + 1$.

We replace this derivation with the derivations,

$$(A, j) \overset{m}{\underset{H}{\Longrightarrow}} \alpha' \quad \text{and} \quad (A, -j) \overset{m+1}{\underset{H}{\Longrightarrow}} \alpha'',$$

where the nonterminals in α are replaced with nonterminals whose second components add up to $j - i$ in α' and to $k - i - j$ in α''.

It is easily shown by induction that $(A, j) \overset{n}{\underset{H}{\Longrightarrow}} w \in T^*$ holds iff $A \overset{kn+j}{\underset{G}{\Longrightarrow}} w$ holds, for some derivation whose tree has height at least $k+1$. In particular, $(S, 0) \overset{n}{\underset{H}{\Longrightarrow}} w$ holds iff $S \overset{kn}{\underset{G}{\Longrightarrow}} w$. Finally, one introduces derivations $(S, 0) \overset{n}{\underset{H}{\Longrightarrow}} w \in T^*$, for any derivation $S \overset{kn}{\underset{G}{\Longrightarrow}} w$. □

Note that the proofs of parts (iv) and (v) imply a "linear slow-down/speed-up theorem" for context-free grammars. For any context-free grammar G, we can construct context-free grammars G' and G'', generating in $k \cdot n$ steps (in n/k steps) the same terminal words as G generates in n steps.

Now we give relations between the structure function and the differentiation function.

Theorem 2.3. *For any (unambiguous) context-free language L, there is a context-free (unambiguous) grammar G such that $d_G = s_L$.*

Proof. Let G be a context-free grammar in Greibach normal form with $L(G) = L$. If L is unambiguous, then we can assume that G is an unambiguous grammar in Greibach normal form. The normal form in which exactly one terminal is produced in each derivation step. Hence, any word w with length n is derived in exactly n steps. Therefore,

$$\{w \mid w \in L(G), |w| = n\} = \{w \mid S \stackrel{n}{\Longrightarrow} w, w \in T^*\}, \text{ for any } n \geq 1,$$

from which the statement follows. $\square$

Theorem 2.4. *For any unambiguous context-free grammar G, there is an unambiguous context-free language L such that $s_L = d_G$.*

Proof. Let $G = (N, T, P, S)$ be an unambiguous context-free grammar. Let $P = \{p_1, p_2, \ldots, p_n\}$. We define the morphism $h : (N \cup T)^* \to N^*$ by $h(a) = \lambda$, for $a \in T$, and $h(A) = A$, for $A \in N$, and set

$$G' = (N, \{[i] \mid 1 \leq i \leq n\}, \{A \to [i]h(\alpha) \mid p_i = A \to \alpha\}, S).$$

Obviously, G' is unambiguous.

It is easily shown by induction on n that there is a leftmost derivation

$$S \stackrel{n}{\underset{G'}{\Longrightarrow}} \quad [i_1] \cdots [i_n]\beta, \beta \in N^*,$$

iff there is a leftmost derivation in G, using the sequence of rules $p_{i_1}, \ldots, p_{i_n}$ and generating some word $\alpha \in h^{-1}(\beta)$. In particular, $[i_1] \cdots [i_n] \in L(G')$ holds iff the leftmost derivation using the sequence of rules $p_{i_1}, \ldots, p_{i_n}$ is possible in G and terminating. As G is unambiguous, the number of words of length n in $L(G')$ is equal to the number of words derivable in G in n steps. $\square$

Essentially, we have shown that, for unambiguous grammars, the sets of structure functions and of differentiation functions coincide. We do not know whether an analogous result holds in the general context-free case. At least, we can show:

Theorem 2.5.

  (i) *For any context-free grammar G, there is a context-free additive valence language L such that $s_L = d_G$.*
  (ii) *For any context-sensitive grammar G, there is a context-sensitive language L such that $s_L = d_G$.*
  (iii) *For any regular grammar G, there is a linear language L such that $s_L = d_G$.*

Proof.

(i) Let $G = (N, T, P, S)$ and let r be the maximum over the lengths of the right hand sides in P. We construct the additive valence grammar $G' = (N', T', P', S')$ with $N' = N \cup \{S', X\}$, $T' = T \cup \{x\}$, and

$$P' = \{(S' \to SX, -1)\} \cup \{(A \to \alpha, r + 1 - |\alpha|) \mid A \to \alpha \in P\} \cup$$
$$\{(X \to xX, -1), (X \to \lambda, 0)\}.$$

It is easily seen that $(S, -1) \overset{n}{\underset{G'}{\Longrightarrow}} (\alpha, z)$ iff $S \overset{n}{\underset{G}{\Longrightarrow}} \alpha$ and $z + |\alpha| = r \cdot n$. Hence, a word $w' = wx^z$, $w \in T^*$, can be generated by G' iff $|w'| = r \cdot n$ and $S \overset{n}{\underset{G}{\Longrightarrow}} w$. Finally, as the family of context-free additive valence languages is closed under inverse morphisms, one can construct an additive valence grammar H such that $s_{L(H)}(n) = s_{L(G')}(r \cdot n)$.

(ii) Essentially the same construction can be used to find a context-sensitive additive valence language L such that $d_G = s_L$. As context-sensitive grammars with and without valences are equivalent, the claim holds.

(iii) If G is regular, construct the linear grammar $G' = (N, T', P', S)$ with $T' = T \cup \{x\}$, and

$$P' = \{A \to wBx^{r+1-|\alpha|} \mid A \to wB \in P, A, B \in N, w \in T^*\} \cup$$
$$\{A \to wx^{r-|\alpha|} \mid A \to w \in P, A \in N, w \in T^*\}.$$

Again, a word $w' = wx^z$, $w \in T^*$ can be generated by G' iff $|w'| = r \cdot n + 1$ and $S \overset{n}{\underset{G}{\Longrightarrow}} w$. We can construct a linear grammar H such that $s_{L(H)}(n) = s_{L(G')}(r \cdot n)$ using the effective closure of the family of linear languages under inverse homomorphisms. $\square$

Theorem 2.6. *If a context-free grammar G is narrow, then $L(G)$ is slender.*

By Example 1.2, the converse implication does not hold.

Proof. Let $G = (N, T, P, S)$ be a context-free grammar such that $L(G) \subseteq V^*$ is not slender. We have to show that G is not narrow, *i.e.*, for any c, there is an $n \in \mathbb{N}$ such that $d_G(n) \geq c$.

We call a nonterminal $A \in N$ *recurring* iff $A \overset{*}{\Longrightarrow} uAv$, for some $u, v \in T^*$, $|uv| > 0$. For a recurring A, let

$$q_A = \min\{|uv| : A \overset{*}{\Longrightarrow} uAv, u, v \in T^*, |uv| > 0\},$$
$$s_A = \min\{n : A \overset{n}{\Longrightarrow} uAv, \text{ for some } u, v \in T^* \text{ with } |uv| = q_A\},$$

and let $s = \max\{s_A \mid A \in N, A \text{ is recurring}\}$. There is a constant p such that any derivation of a word $w \in L(G)$ with $|w| \geq p$ uses a recurring symbol. We can formulate the following *length iteration lemma:*

> For any $w \in T^*$ such that $S \xrightarrow{n} w$ and $|w| \geq p$, there is a recurring A such that G can generate a terminal word of length $|w| + iq_A$ in $n + is_A$ steps, for any $i \geq 0$.

Moreover, let r be the number of recurring nonterminals. For a word $w \in L(G)$, let $|w|_G$ be the length of the shortest derivation of w in G, and call $|w|_G$ the G-length of w.

The non-slenderness of $L(G)$ implies the existence of a number $n \geq p$ such that $s_{L(G)}(n) \geq rsc^2$. Consider the set $S_0 := L(G) \cap V^n$. If S_0 contains c words with the same G-length m, then $d_G(m) \geq c$ and we are done. Otherwise, we obtain by several applications of the pigeonhole-principle the existence of:

- $S_1 \subseteq S_0$ with rsc words of pairwise different G-lengths;
- $S_2 \subseteq S_1$ with sc words whose shortest derivations use a common recurrent symbol, say A;
- $S_3 \subseteq S_2$ with c words whose G-lengths are congruent modulo s_A.

Suppose that $S_3 = \{w_1, \ldots, w_c\}$ with $|w_i|_G < |w_j|_G$, for all $1 \leq i < j \leq c$. Set $t_i = (|w_c|_G - |w_i|_G)/s_A$, for $1 \leq i \leq c$. The length iteration lemma guarantees the existence of words $w_i' \in L(G)$, $1 \leq i \leq c$, with $|w_i'| = |w_i| + t_i \cdot s_A$ such that w_i' can be generated in $|w_i|_G + t_i \cdot r = |w_c|_G$ steps. Since $|w_i'| > |w_j'|$, for $1 \leq i < j \leq c$, the words w_i' are pairwise different, and we obtain $d_G(|w_c|_G) \geq c$. $\qquad\square$

3. Structural and derivational languages

We first consider the case that the structural or derivational languages are regular.

Theorem 3.1.

    (i) *If $\sigma(L)$ is a regular language, then L is slender.*
    (ii) *If $\delta(G)$ is a regular language, then G is narrow.*

Proof. Let $L \subseteq V^*$ be a language such that $\sigma(L)$ is regular, and let p be the constant from the regular pumping lemma for $\sigma(L)$. Then, each word $a^n b^m \in \sigma(L)$ satisfies $m < p$, since otherwise $a^n b^{m+p}$ would belong to $\sigma(L)$.

The correctness of the second claim is shown analogously. $\qquad\square$

Note that the type of the language L or of the grammar G in the previous proof plays no role.

For context-free languages and grammars, the converse of Theorem 3.1 does also hold.

Theorem 3.2.

    (i) *If L is a slender context-free matrix language, then the language $\sigma(L)$ is regular.*
    (ii) *If G is a narrow context-free grammar, then the language $\delta(G)$ is regular.*

Proof. (i) Let $L \subseteq V^*$ be a context-free matrix language. According to [18], the language $S_i(L) = \{a^n : s_L(n) = j\}$ is regular, for any $j \in \mathbb{N}$. If, additionally, L is k-slender

$$\sigma(L) = \bigcup_{j=0}^{k} S_j(L)b^j$$

is a union of regular languages and therefore regular.

(ii) Follows by Theorem 2.5 and the first part.   □

A generalization of Theorem 3.2 does not hold for arbitrary languages For $L = \{a^{2^n} \mid n \geq 1\}$, the language $\sigma(L) = \{a^{2^n}b \mid n \geq 1\} \cup \{a^n \mid n \neq 2^i, i \geq 1\}$ is not context-free, although L is a 1-slender language.

Combining the last two theorems, we get a characterization of slender languages in the case of context-free matrix languages.

Corollary 3.3. *A context-free matrix language is slender if, and only if its structural language is regular.*

We now investigate the relation of structural and derivational languages to well-known language families.

Theorem 3.4.
  (i) *For any grammar G, $\delta(G)$ is context-sensitive.*
  (ii) *For any context-free grammar G, $\delta(G)$ is deterministic context-sensitive.*
  (iii) *For a (deterministic) context-sensitive language L, $\sigma(L)$ is (deterministic) context-sensitive.*

Proof.
(i) Let $G = (N, T, P, S)$ be a grammar and let r be the greatest length of a right side in P. Then any word derived in n steps has at most length $n \cdot r$. First, we construct a Turing machine $\mathcal{A}$ accepting

$$\delta'(G) := \{a^n b^m \mid n \geq 0, m \leq d_G(n)\}.$$

$\mathcal{A}$ works as follows. It guesses a word $w \in T^*$ of length at most $n \cdot r$. Then it performs a guessed derivation of length n in G. If the word obtained by this derivation is not w, it rejects. Otherwise it chooses a new word w which is lexicographically greater than the old one. If this process can be repeated m times, the input $a^n b^m$ is accepted. The workspace needed by $\mathcal{A}$ is $O(n)$.

Finally, it is a standard construction to find an LBA accepting $\delta(G)$.

(ii) If G is context-free, we can modify the above nondeterministic algorithm as follows. Instead of guessing words and derivations, we test for all words $w \in T^*$ of length up to rn if there is a left-derivation with n steps generating w. The words w are tested in lexicographic order, and for each w, the left-derivations are tested in lexicographic order, too. We need additionally space $O(n)$ to store the currently performed left-derivation.

(iii) A construction similar to that in part (i) of the proof works.   □

We now prove that the structural language of a regular language and the derivational language of a regular grammar are not necessarily matrix languages.

Theorem 3.5.

    (i) *There is a regular language L for which the language $\sigma(L)$ cannot be generated by a matrix grammar.*

    (ii) *There is an unambiguous regular grammar G such that $\delta(G)$ cannot be generated by a matrix grammar.*

Proof. By Theorem 2.1 (ii) and its proof, there is an unambiguous regular grammar G such that $d_G(n) = s_G(n) = 2^n = 2^n$, *i.e.*, $\delta(G) = \{a^n b^{2^n} \mid 1 \leq n\}$ and $\sigma(L(G)) = \delta(G) \cup \{\lambda\}$.

With the morphism $h : \{a, b\}^* \to \{b\}^*$, $h(a) = \lambda, h(b) = b$, we get the languages $h(\delta(G)) = \{b^{2^n} \mid n \geq 1\}$ and $h(\sigma(L(G))) = \{b^{2^n} \mid n \geq 0\}$. By [5], these languages are not matrix languages, and neither are $\sigma(L)$ and $\delta(G)$. $\qquad\square$

4. DECISION PROBLEMS

We first prove the undecidability of the equality of the differentiation and structure functions of two context-free grammars.

Theorem 4.1. *For context-free grammars G and G', it is undecidable whether or not $d_G = d_{G'}$ or $s_{L(G)} = s_{L(G')}$ holds.*

Proof. For the structure function, the result follows immediately from the undecidability of the universe problem:

$$\text{Is } L(G) = T^*, \text{ where } T \text{ is the terminal alphabet of } G?$$

The result for the differentiation function follows by Theorem 2.3. $\qquad\square$

Note that the universe problem is undecidable for very specific subclasses of the family of context-free languages (see, *e.g.*, [7]). Thus Theorem 4.1 can be refined. In contrast to the general case, we have decidability if we restrict to slender context-free (matrix) languages.

Theorem 4.2. *For slender context-free (matrix) languages L_1 and L_2, it is decidable whether or not $s_{L_1} = s_{L_2}$ holds.*

Proof. Obviously, $s_{L_1} = s_{L_2}$ holds iff $\sigma(L_1) = \sigma(L_2)$. Since the equivalence of regular grammars is decidable and the determination of $\sigma(L)$ according to the proof of Theorem 3.2 is constructive (see [18]), the statement follows. $\qquad\square$

Finally, we discuss the decision question whether or not a given context-free grammar is narrow. While the general problem remains open, we are able to show decidability if the bound for narrowness is given or the grammar is regular.

Theorem 4.3.

    (i) *Given a context-free grammar G and a constant k, it is decidable whether G is k-narrow.*

    (ii) *Given a regular grammar G, it is decidable whether G is narrow.*

Proof. In the proof of Theorem 2.5, we have shown how to construct a context-free additive valence grammar (in the regular case, a linear grammar) H such that $s_{L(H)} = d_G$. Obviously, G is k-narrow iff $L(H)$ is k-slender. As k-slenderness of languages generated by context-free additive valence grammars is decidable [18], the first assertion is true. The second claim follows by Theorem 2.5 (iii) and the decidability of the slenderness problem for context-free languages [16].   □

REFERENCES

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

[2] J. Dassow, Eine neue Funktion für Lindenmayer-Systeme. *EIK* **12** (1976) 515–521.

[3] J. Dassow, Numerical parameters of evolutionary grammars, in *Jewels are forever*, edited by J. Karhumäki, H. Maurer, Gh. Păun, G. Rozenberg. Springer-Verlag, Berlin (1999) 171–181.

[4] J. Dassow and Gh. Păun, *Regulated Rewriting in Formal Language Theory*. Akademie-Verlag, Berlin and Springer-Verlag, Berlin (1989).

[5] D. Hauschildt and M. Jantzen, Petri nets algorithms in the theory of matrix grammars. *Acta Inform.* **31** (1994) 719–728.

[6] S. Ginsburg, *The Mathematical Theory of Context-Free Languages*. McGraw Hill Book Comp., New York (1966).

[7] O. Ibarra, Restricted one-counter machines with undecidable universe problems. *Math. Syst. Theory* **13** (1979) 181–186.

[8] L. Ilie, On a conjecture about slender context-free languages. *Theor. Comput. Sci.* **132** (1994) 427–434.

[9] L. Ilie, On lengths of words in context-free languages. *Theor. Comput. Sci.* **242** (2000) 327–359.

[10] R. Incitti, The growth function of context-free languages. *Theor. Comput. Sci.* **255** (2001) 601–605.

[11] T. Katayama, M. Okamoto and H. Enomoto, Characterization of the structure-generating functions of regular sets and the D0L systems. *Inform. Control* **36** (1978) 85–101.

[12] W. Kuich and R.K. Shyamasundar, The structure generating function of some families of languages. *Inform. Control* **32** (1976) 85–92.

[13] M. Kunze, H.J. Shyr and G. Thierrin, h-bounded and semidiscrete languages. *Inform. Control* **51** (1981) 147–187.

[14] M. Latteux and G. Thierrin, Semidiscrete context-free languages. *Internat. J. Comput. Math.* **14** (1983) 3–18.

[15] Gh. Păun and A. Salomaa, Thin and slender languages. *Discrete Appl. Math.* **61** (1995) 257–270.

[16] D. Raz, Length considerations in context-free languages. *Theor. Comput. Sci.* **183** (1997) 21–32.

[17] A. Salomaa and M. Soittola, *Automata-Theoretic Aspects of Formal Power Series*. Springer-Verlag (1978).

[18] R. Stiebe, Slender matrix languages, in *Developments in Language Theory*, edited by G. Rozenberg, W. Thomas. World Scientific, Singapore (2000) 375–385.