To appear in LATA 2016.
The final publication will be available at Springer.

Ogden's Lemma, Multiple Context-Free Grammars, and the Control Language Hierarchy

Makoto Kanazawa*

National Institute of Informatics and SOKENDAI, 2–1–2 Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430, Japan

Abstract. I present a simple example of a multiple context-free language for which a very weak variant of generalized Ogden's lemma fails. This language is generated by a non-branching (and hence well-nested) 3-MCFG as well as by a (non-well-nested) binary-branching 2-MCFG; it follows that neither the class of well-nested 3-MCFLs nor the class of 2-MCFLs is included in Weir's control language hierarchy, for which Palis and Shende proved an Ogden-like iteration theorem. I then give a simple sufficient condition for an MCFG to satisfy a natural analogue of Ogden's lemma, and show that the corresponding class of languages is a substitution-closed full AFL which includes Weir's control language hierarchy. My variant of generalized Ogden's lemma is incomparable in strength to Palis and Shende's variant and is arguably a more natural generalization of Ogden's original lemma.

Keywords: grammars, Ogden's lemma, multiple context-free grammars, control languages

1 Introduction

A *multiple context-free grammar* [12] is a context-free grammar on tuples of strings (of varying length). An analogue of the pumping lemma, which asserts the existence of a certain number of substrings that can be simultaneously iterated, has been established for *well-nested* MCFGs and (non-well-nested) MCFGs of dimension 2 [6]. So far, it has been unknown whether an analogue of Ogden's [10] strengthening of the pumping lemma holds of these classes. This paper negatively answers the question for both classes, and moreover proves a generalized Ogden's lemma for the class of MCFGs satisfying a certain simple property. The class of languages generated by the grammars in this class includes Weir's [13] control language hierarchy, the only non-trivial subclass of MCFLs for which an Ogden-style iteration theorem has been proved so far [11].

2 Preliminaries

The set of natural numbers is denoted $\mathbb{N}$. If i and j are natural numbers, we write $[i, j]$ for the set $\{ n \in \mathbb{N} \mid i \leq n \leq j \}$. We write $|w|$ for the length of a string w

* This work was supported by JSPS KAKENHI Grant Number 25330020.

and $|S|$ for the cardinality of a set S; the context should make it clear which is intended. If u, v, w are strings, we write $(u[v]w)$ for the subinterval $[|u| + 1, |uv|]$ of $[1, |uvw|]$. If w is a string, w^R denotes the reversal of w.

2.1 Multiple Context-Free Grammars

A *multiple context-free grammar* (MCFG) [12] is a quadruple $G = (N, \Sigma, P, S)$, where N is a finite set of *nonterminals*, each with a fixed *dimension* ≥ 1, Σ is a finite alphabet of *terminals*, P is a set of *rules*, and S is the distinguished *initial nonterminal* of dimension 1. We write $N^{(q)}$ for the set of nonterminals in N of dimension q. A nonterminal in $N^{(q)}$ is interpreted as a q-ary predicate over Σ^*. A rule is stated with the help of *variables* interpreted as ranging over Σ^*. Let $\mathcal{X}$ be a denumerable set of variables. We use boldface lower-case letters as elements of $\mathcal{X}$. A rule is a *definite clause* (in the sense of logic programming) constructed with *atoms* of the form $A(\alpha_1, \dots, \alpha_q)$, with $A \in N^{(q)}$ and $\alpha_1, \dots, \alpha_q$ *patterns*, i.e., strings over $\Sigma \cup \mathcal{X}$. An MCFG rule is of the form

$$A(\alpha_1, \dots, \alpha_q) \leftarrow B_1(\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,q_1}), \dots, B_n(\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,q_n}),$$

where $n \geq 0$, $A, B_1, \dots, B_n$ are nonterminals of dimensions $q, q_1, \dots, q_n$, respectively, the $\boldsymbol{x}_{i,j}$ are pairwise distinct variables, and each α_i is a string over $\Sigma \cup \{\, \boldsymbol{x}_{i,j} \mid i \in [1, n], j \in [1, q_i] \,\}$, such that $(\alpha_1, \dots, \alpha_q)$ contains at most one occurrence of each $\boldsymbol{x}_{i,j}$. An MCFG is an *m-MCFG* if the dimensions of its nonterminals do not exceed m; it is *r-ary branching* if each rule has no more than r occurrences of nonterminals in its *body* (i.e., the part that follows the symbol $\leftarrow$). We call a unary branching grammar *non-branching*.[1]

An atom $A(\alpha_1, \dots, \alpha_q)$ is *ground* if $\alpha_1, \dots, \alpha_q \in \Sigma^*$. A *ground instance* of a rule is the result of substituting a string over Σ for each variable in the rule. Given an MCFG $G = (N, \Sigma, P, S)$, a ground atom $A(w_1, \dots, w_q)$ *directly follows* from a sequence of ground atoms $B_1(v_{1,1}, \dots, v_{1,q_1}), \dots, B_n(v_{n,1}, \dots, v_{n,q_n})$ if $A(w_1, \dots, w_q) \leftarrow B_1(v_{1,1}, \dots, v_{1,q_1}), \dots, B_n(v_{n,1}, \dots, v_{n,q_n})$ is a ground instance of some rule in P. A ground atom $A(w_1, \dots, w_q)$ is *derivable*, written $\vdash_G A(w_1, \dots, w_q)$, if it directly follows from some sequence of derivable ground atoms. In particular, if $A(w_1, \dots, w_q) \leftarrow$ is a rule in P, we have $\vdash_G A(w_1, \dots, w_q)$.

A derivable ground atom is naturally associated with a *derivation tree*, each of whose nodes is labeled by a derivable ground atom, which directly follows from the sequence of ground atoms labeling its children. The language generated by G is defined as $L(G) = \{\, w \in \Sigma^* \mid \vdash_G S(w) \,\}$, or equivalently, $L(G) = \{\, w \in \Sigma^* \mid G \text{ has a derivation tree for } S(w) \,\}$. The class of languages generated by m-MCFGs is denoted m-MCFL, and the class of languages generated by r-ary branching m-MCFGs is denoted m-MCFL(r).

Example 1. Consider the following 2-MCFG:

$$\begin{array}{rlrl} S(\boldsymbol{x}_1 \# \boldsymbol{x}_2) \leftarrow D(\boldsymbol{x}_1, \boldsymbol{x}_2) & & D(\boldsymbol{x}_1 \boldsymbol{y}_1, \boldsymbol{y}_2 \boldsymbol{x}_2) \leftarrow E(\boldsymbol{x}_1, \boldsymbol{x}_2), D(\boldsymbol{y}_1, \boldsymbol{y}_2) \\ D(\varepsilon, \varepsilon) \leftarrow & & E(a \boldsymbol{x}_1 \bar{a}, \bar{a} \boldsymbol{x}_2 a) \leftarrow D(\boldsymbol{x}_1, \boldsymbol{x}_2) \end{array}$$

[1] Non-branching MCFGs have been called *linear* in [1].

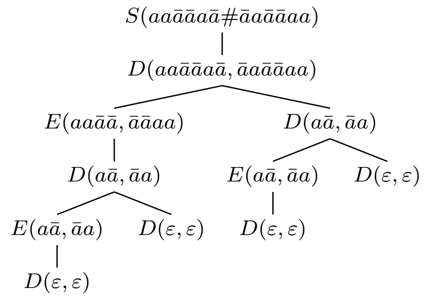

Fig. 1. A derivation tree of a 2-MCFG.

Here, S is the initial nonterminal and D and E are both nonterminals of dimension 2. This grammar is binary branching and generates the language $\{ w\#w^R \mid w \in D_1^* \}$, where D_1^* is the (one-sided) *Dyck language* over the alphabet $\{a, \bar{a}\}$. Figure 1 shows the derivation tree for $aa\bar{a}\bar{a}a\bar{a}\#\bar{a}a\bar{a}\bar{a}aa$.

It is also useful to define the notion of a derivation of an atom $A(\alpha_1, \dots, \alpha_q)$ from an assumption $C(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r)$, where $\boldsymbol{x}_1, \dots, \boldsymbol{x}_r$ are pairwise distinct variables. An atom $A(\alpha_1, \dots, \alpha_q)$ is *derivable from an assumption* $C(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r)$, written $C(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r) \vdash_G A(\alpha_1, \dots, \alpha_q)$, if either

1. $A = C$ and $(\alpha_1, \dots, \alpha_q) = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_r)$, or
2. there are some atom $B_i(\beta_1, \dots, \beta_{q_i})$ and ground atoms $B_j(v_{j,1}, \dots, v_{j,q_j})$ for each $j \in [1, i-1] \cup [i+1, n]$ such that $C(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r) \vdash_G B_i(\beta_1, \dots, \beta_{q_i})$, $\vdash_G B_j(v_{j,1}, \dots, v_{j,q_j})$, and

$$\begin{aligned} A(\alpha_1, \dots, \alpha_q) \leftarrow B_1(v_{1,1}, \dots, v_{1,q_1}), \dots, B_{i-1}(v_{i-1,1}, \dots, v_{i-1,q_{i-1}}), \\ B_i(\beta_1, \dots, \beta_{q_i}), B_{i+1}(v_{i+1,1}, \dots, v_{i+1,q_{i+1}}), \dots, B_n(v_{n,1}, \dots, v_{n,q_n}) \end{aligned}$$

   is an instance of some rule in P.

Let us write $[v_1/\boldsymbol{x}_1, \dots, v_r/\boldsymbol{x}_r]$ for the simultaneous substitution of strings $v_1, \dots, v_r$ for variables $\boldsymbol{x}_1, \dots, \boldsymbol{x}_r$. Evidently, when we have $\vdash_G B(v_1, \dots, v_r)$ and $B(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r) \vdash_G A(\alpha_1, \dots, \alpha_q)$, the two derivations can be combined into one witnessing $\vdash_G A(\alpha_1, \dots, \alpha_q)[v_1/\boldsymbol{x}_1, \dots, v_r/\boldsymbol{x}_r]$. The following lemma says that when $B(v_1, \dots, v_r)$ is derived in the course of a derivation of $A(w_1, \dots, w_q)$, the derivation can be decomposed into one for $B(v_1, \dots, v_r)$ and a derivation from an assumption $B(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r)$:

Lemma 2. *Let τ be a derivation tree of an MCFG G for some ground atom $A(w_1, \dots, w_q)$, and let $B(v_1, \dots, v_r)$ be the label of some node of τ. Then there is an atom $A(\alpha_1, \dots, \alpha_q)$ such that $B(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r) \vdash_G A(\alpha_1, \dots, \alpha_q)$ and $(w_1, \dots, w_q) = (\alpha_1, \dots, \alpha_q)[v_1/\boldsymbol{x}_1, \dots, v_r/\boldsymbol{x}_r]$.*

Example 3. Consider the derivation tree in Figure 1 and the node ν labeled by $E(aa\bar{a}\bar{a}, \bar{a}\bar{a}aa)$. Let τ be the subtree of this derivation tree consisting of ν

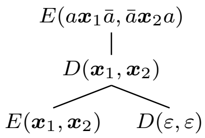

Fig. 2. A derivation of $E(a\boldsymbol{x}_1\bar{a}, \bar{a}\boldsymbol{x}_2 a)$ from assumption $E(\boldsymbol{x}_1, \boldsymbol{x}_2)$.

and the nodes that lie below it. Consider the node ν_1 labeled by $E(a\bar{a}, \bar{a}a)$ in τ. The rules used in the portion of τ that remains after removing the nodes below ν_1 determine a derivation tree for $E(\boldsymbol{x}_1, \boldsymbol{x}_2) \vdash_G E(a\boldsymbol{x}_1\bar{a}, \bar{a}\boldsymbol{x}_2 a)$, depicted in Figure 2. Note that substituting $a\bar{a}, \bar{a}a$ for $\boldsymbol{x}_1, \boldsymbol{x}_2$ in $E(a\boldsymbol{x}_1\bar{a}, \bar{a}\boldsymbol{x}_2 a)$ gives back $E(aa\bar{a}\bar{a}, \bar{a}\bar{a}aa)$.

An MCFG rule $A(\alpha_1, \dots, \alpha_q) \leftarrow B_1(\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,q_1}), \dots, B_n(\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,q_n})$ is said to be

- *non-deleting* if all variables $\boldsymbol{x}_{i,j}$ in its body occur in $(\alpha_1, \dots, \alpha_q)$;
- *non-permuting* if for each $i \in [1, n]$, the variables $\boldsymbol{x}_{i,1}, \dots, \boldsymbol{x}_{i,q_i}$ occur in $(\alpha_1, \dots, \alpha_q)$ in this order;
- *well-nested* if it is non-deleting and non-permuting and there are no $i, j \in [1, n], k \in [1, q_i - 1], l \in [1, q_l - 1]$ such that $\boldsymbol{x}_{i,k}, \boldsymbol{x}_{j,l}, \boldsymbol{x}_{i,k+1}, \boldsymbol{x}_{j,l+1}$ occur in $(\alpha_1, \dots, \alpha_q)$ in this order.

Every m-MCFG(r) has an equivalent m-MCFG(r) whose rules are all non-deleting and non-permuting, and henceforth we will always assume that these conditions are satisfied. An MCFG whose rules are all well-nested is a *well-nested MCFG* [6]. The 2-MCFG in Example 1 is well-nested. It is known that there is no well-nested MCFG for the language $\{\, w\#w \mid w \in D_1^* \,\}$ [9], although it is easy to write a non-well-nested 2-MCFG for this language.

Every (non-deleting and non-permuting) non-branching MCFG is by definition well-nested. The class $\bigcup_m m$-MCFL(1) coincides with the class of output languages of *deterministic two-way finite-state transducers* (see [1]).

2.2 The Control Language Hierarchy

Weir's [13] *control language hierarchy* is defined in terms of the notion of a *labeled distinguished grammar*, which is a 5-tuple $G = (N, \Sigma, P, S, f)$, where $\overline{G} = (N, \Sigma, P, S)$ is an ordinary context-free grammar and $f \colon P \to \mathbb{N}$ is a function such that if $\pi \in P$ is a context-free production with n occurrences of nonterminals on its right-hand side, then $f(\pi) \in [0, n]$. We view P as a finite alphabet, and use a language $C \in P^*$ to restrict the derivations of G. The pair (G, C) is a *control grammar*. For each nonterminal $A \in N$, define $R_{(G,C)}(A) \subseteq \Sigma^* \times P^*$ inductively as follows: for each production $\pi = A \to w_0 B_1 w_1 \dots B_n w_n$ in P,

- if $f(\pi) = 0$ and $(\{v_j\} \times C) \cap R_{(G,C)}(B_j) \neq \varnothing$ for each $j \in [1, n]$, then $(w_0 v_1 w_1 \dots v_n w_n, \pi) \in R_{(G,C)}(A)$;

– if $f(\pi) = i \in [1, n]$, $(v_i, z) \in R_{(G,C)}(B_i)$, and $(\{v_j\} \times C) \cap R_{(G,C)}(B_j) \neq \varnothing$ for each $j \in [1, i-1] \cup [i+1, n]$, then $(w_0 v_1 w_1 \dots v_n w_n, \pi z) \in R_{(G,C)}(A)$.

The language of the control grammar (G, C) is $L(G, C) = \{\, w \in \Sigma^* \mid (\{w\} \times C) \cap R_{(G,C)}(S) \neq \varnothing \,\}$.

The first level of the control language hierarchy is $\mathcal{C}_1 = \mathrm{CFL}$, the family of context-free languages, and for $k \geq 1$,

$$\mathcal{C}_{k+1} = \{\, L(G, C) \mid (G, C) \text{ is a control grammar and } C \in \mathcal{C}_k \,\}.$$

The second level $\mathcal{C}_2$ is known to coincide with the family of languages generated by well-nested 2-MCFGs, or equivalently, the family of *tree-adjoining languages* [13].

Example 4. Let $G = (N, \Sigma, P, S, f)$ be a labeled distinguished grammar consisting of the following productions:

$$\pi_1 \colon S \to aS\bar{a}S, \qquad \pi_2 \colon S \to bS\bar{b}S, \qquad \pi_3 \colon S \to \varepsilon,$$

where $f(\pi_1) = 1, f(\pi_2) = 1, f(\pi_3) = 0$. Let $C = \{\, \pi_1^n \pi_2^n \pi_3 \mid n \in \mathbb{N} \,\}$. Then $L(G, C) = D_2^* \cap (\{\, a^n b^n \mid n \in \mathbb{N} \,\}\{\bar{a}, \bar{b}\}^*)^*$, where D_2^* is the Dyck language over $\{a, \bar{a}, b, \bar{b}\}$. Since C is a context-free language, this language belongs to $\mathcal{C}_2$.

Palis and Shende [11] proved the following Ogden-like theorem for $\mathcal{C}_k$:

Theorem 5 (Palis and Shende). *If $L \in \mathcal{C}_k$, then there is a number p such that for all $z \in L$ and $D \subseteq [1, |z|]$, if $|D| \geq p$, there are $u_1, \dots, u_{2^k+1}, v_1, \dots, v_{2^k} \in \Sigma^*$ that satisfy the following conditions:*

(i) $z = u_1 v_1 u_2 v_2 \dots u_{2^k} v_{2^k} u_{2^k+1}$.
(ii) *for some* $j \in [1, 2^k]$,

$$\begin{aligned} D \cap (u_1 v_1 \dots [u_j] v_j u_{j+1} v_{j+1} \dots u_{2^k} v_{2^k} u_{2^k+1}) &\neq \varnothing, \\ D \cap (u_1 v_1 \dots u_j [v_j] u_{j+1} v_{j+1} \dots u_{2^k} v_{2^k} u_{2^k+1}) &\neq \varnothing, \\ D \cap (u_1 v_1 \dots u_j v_j [u_{j+1}] v_{j+1} \dots u_{2^k} v_{2^k} u_{2^k+1}) &\neq \varnothing. \end{aligned}$$

(iii) $|D \cap (u_1 v_1 \dots u_{2^{k-1}} [v_{2^{k-1}} u_{2^{k-1}+1} v_{2^{k-1}+1}] \dots u_{2^k} v_{2^k} u_{2^k+1})| \leq p$.
(iv) $u_1 v_1^n u_2 v_2^n \dots u_{2^k} v_{2^k}^n u_{2^k+1} \in L$ *for all* $n \in \mathbb{N}$.

Kanazawa and Salvati [8] proved the inclusion $\mathcal{C}_k \subseteq 2^{k-1}$-MCFL, while using Theorem 5 to show that the language $\mathrm{RESP}_{2^{k-1}}$ belongs to 2^{k-1}-MCFL $- \mathcal{C}_k$ for $k \geq 2$, where $\mathrm{RESP}_l = \{\, a_1^m a_2^m b_1^n b_2^n \dots a_{2l-1}^m a_{2l}^m b_{2l-1}^n b_{2l}^n \mid m, n \in \mathbb{N} \,\}$.

3 The Failure of Ogden's Lemma for Well-Nested MCFGs and 2-MCFGs

Let G be an MCFG, and consider a derivation tree τ for an element z of $L(G)$. When a node of τ and one of its descendants are labeled by ground atoms

$B(w_1, \dots, w_r)$ and $B(v_1, \dots, v_r)$ sharing the same nonterminal B, the portion of τ consisting of the nodes that are neither above the first node nor below the second node determines a derivation tree σ witnessing $B(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r) \vdash_G B(\beta_1, \dots, \beta_r)$ (called a *pump* in [6]), where $(\beta_1, \dots, \beta_r)[v_1/\boldsymbol{x}_1, \dots, v_r/\boldsymbol{x}_r] = (w_1, \dots, w_r)$. This was illustrated by Example 3. When each $\boldsymbol{x}_i$ occurs in β_i, i.e., $\beta_i = v_{2i-1}\boldsymbol{x}_i v_{2i}$ for some $v_{2i-1}, v_{2i} \in \Sigma^*$ (in which case σ is an *even pump* [6]), iterating σ gives a derivation tree for $B(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r) \vdash_G B(v_1^n \boldsymbol{x}_1 v_2^n, \dots, v_{2r-1}^n \boldsymbol{x}_r v_{2r}^n)$. Combining this with the rest of τ gives a derivation tree for $z(n) = u_1 v_1^n u_2 v_2^n \dots u_{2r} v_{2r}^n u_{2r+1} \in L(G)$ for every $n \in \mathbb{N}$, where $z(1) = z$. When some $\boldsymbol{x}_i$ occurs in β_j with $j \neq i$ (σ is an *uneven pump*), however, the result of iterating σ exhibits a complicated pattern that is not easy to describe.

A language L is said to be *k-iterative* if all but finitely many elements of L can be written in the form $u_1 v_1 u_2 v_2 \dots u_k v_k u_{k+1}$ so that $v_1 \dots v_k \neq \varepsilon$ and $u_1 v_1^n u_2 v_2^n \dots u_k v_k^n u_{k+1} \in L$ for all $n \in \mathbb{N}$. A language that is either finite or includes an infinite k-iterative subset is said to be *weakly k-iterative*. (These terms are from [4,3].) The possibility of an uneven pump explains the difficulty of establishing $2m$-iterativity of an m-MCFL. In 1991, Seki et al. [12] proved that every m-MCFL is weakly $2m$-iterative, but whether every m-MCFL is $2m$-iterative remained an open question for a long time, until Kanazawa et al. [7] negatively settled it in 2014 by exhibiting a (non-well-nested) 3-MCFL that is not k-iterative for any k. Earlier, Kanazawa [6] had shown that the language of a well-nested m-MCFG is always $2m$-iterative, and moreover that a 2-MCFL is always 4-iterative. The proof of this last pair of results was much more indirect than the proof of the pumping lemma for the context-free languages, and did not suggest a way of strengthening them to an Ogden-style theorem. Below, we show that there is indeed no reasonable way of doing so.

Let us say that a language L has the *weak Ogden property* if there is a natural number p such that for every $z \in L$ and $D \subseteq [1, |z|]$ with $|D| \geq p$, there are strings $u_1, \dots, u_{k+1}, v_1, \dots, v_k$ ($k \geq 1$) satisfying the following conditions:

1. $z = u_1 v_1 \dots u_k v_k u_{k+1}$,
2. $D \cap (u_1 v_1 \dots u_i [v_i] \dots u_k v_k u_{k+1}) \neq \varnothing$ for some $i \in [1, k]$, and
3. $u_1 v_1^n \dots u_k v_k^n u_{k+1} \in L$ for all $n \geq 0$.

The elements of D are referred to as *distinguished positions* in z.

Theorem 6. *There is an $L \in$ 3-MCFL(1) $\cap$ 2-MCFL(2) that does not satisfy the weak Ogden property.*

Proof. Let L be the set of all strings over the alphabet $\{a, b, \$\}$ that are of the form

$$a^{i_1} b^{i_0} \$ a^{i_2} b^{i_1} \$ a^{i_3} b^{i_2} \$ \dots \$ a^{i_n} b^{i_{n-1}} \tag{$\dagger$}$$

for some $n \geq 3$ and $i_0, \dots, i_n \geq 0$. This language is generated by the non-branching 3-MCFG (left) as well as by the binary branching 2-MCFG (right) in Figure 3. Now suppose L has the weak Ogden property, and let p be the number satisfying the required conditions. Let

$$z = a \$ a^2 b \$ a^3 b^2 \$ \dots \$ a^{p+1} b^p,$$

$$\begin{array}{rl} A(\varepsilon) & \leftarrow \\ A(b\boldsymbol{x}_1) & \leftarrow A(\boldsymbol{x}_1) \\ B(\boldsymbol{x}_1, \varepsilon) & \leftarrow A(\boldsymbol{x}_1) \\ B(a\boldsymbol{x}_1, b\boldsymbol{x}_2) & \leftarrow B(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ C(\boldsymbol{x}_1, \boldsymbol{x}_2, \varepsilon) & \leftarrow B(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ C(\boldsymbol{x}_1, a\boldsymbol{x}_2, b\boldsymbol{x}_3) & \leftarrow C(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) \\ C(\boldsymbol{x}_1 \$ \boldsymbol{x}_2, \boldsymbol{x}_3, \varepsilon) & \leftarrow C(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) \\ D(\boldsymbol{x}_1 \$ \boldsymbol{x}_2, \boldsymbol{x}_3) & \leftarrow C(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) \\ D(\boldsymbol{x}_1, a\boldsymbol{x}_2) & \leftarrow D(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ S(\boldsymbol{x}_1 \$ \boldsymbol{x}_2) & \leftarrow D(\boldsymbol{x}_1, \boldsymbol{x}_2) \end{array} \qquad \begin{array}{rl} A(\varepsilon) & \leftarrow \\ A(b\boldsymbol{x}_1) & \leftarrow A(\boldsymbol{x}_1) \\ B(\boldsymbol{x}_1, \varepsilon) & \leftarrow A(\boldsymbol{x}_1) \\ B(a\boldsymbol{x}_1, b\boldsymbol{x}_2) & \leftarrow B(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ C(\varepsilon, \varepsilon) & \leftarrow \\ C(a\boldsymbol{x}_1, b\boldsymbol{x}_2) & \leftarrow C(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ D(\boldsymbol{x}_1 \$ \boldsymbol{y}_1 \boldsymbol{x}_2, \boldsymbol{y}_2) & \leftarrow B(\boldsymbol{x}_1, \boldsymbol{x}_2), C(\boldsymbol{y}_1, \boldsymbol{y}_2) \\ D(\boldsymbol{x}_1 \$ \boldsymbol{y}_1 \boldsymbol{x}_2, \boldsymbol{y}_2) & \leftarrow D(\boldsymbol{x}_1, \boldsymbol{x}_2), C(\boldsymbol{y}_1, \boldsymbol{y}_2) \\ E(\boldsymbol{x}_1, \boldsymbol{x}_2) & \leftarrow D(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ E(\boldsymbol{x}_1, a\boldsymbol{x}_2) & \leftarrow E(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ S(\boldsymbol{x}_1 \$ \boldsymbol{x}_2) & \leftarrow E(\boldsymbol{x}_1, \boldsymbol{x}_2) \end{array}$$

Fig. 3. Two grammars generating the same language.

and let D consist of the positions in z occupied by \$. Note that $|D| = p$. By the weak Ogden property, there must be strings $u_1, \dots, u_{k+1}, v_1, \dots, v_k$ ($k \geq 1$) such that $z = u_1 v_1 \dots u_k v_k u_{k+1}$, at least one of $v_1, \dots, v_k$ contains an occurrence of \$, and $u_1 v_1^n \dots u_k v_k^n u_{k+1} \in L$ for all n. Without loss of generality, we may assume that $v_1, \dots, v_k$ are all nonempty strings. Let us write $z(n)$ for $u_1 v_1^n \dots u_k v_k^n u_{k+1}$. First note that none of of $v_1, \dots, v_k$ can start in a and end in b, since otherwise $z(2)$ would contain ba as a factor and not be of the form (†). Let i be the greatest number such that v_i contains an occurrence of \$. Since none of $v_{i+1}, \dots, v_k$ contains an occurrence of \$, it is easy to see that $v_{i+1}, \dots, v_k$ are all in $a^+ \cup b^+$. We consider two cases, depending on the number of occurrences of \$ in v_i. Each case leads to a contradiction.

Case 1. v_i contains just one occurrence of \$. Then $v_i = x\$y$, where x is a suffix of $a^{j+1}b^j$ and y is a prefix of $a^{j+2}b^{j+1}$ for some $j \in [0, p-1]$. Note that $z(3)$ contains $\$yx\$yx\$$ as a factor. Since $z(3)$ is of the form (†), this means that $yx = a^l b^l$ for some $l \geq 0$.

Case 1.1 $l \leq j+1$. Then y must be a prefix of a^{j+1} and since x is a suffix of $a^{j+1}b^j$, it follows that $l \leq j$. Since $yu_{i+1}v_{i+1} \dots u_k v_k u_{k+1}$ has $a^{j+2}b^{j+1}$ as a prefix and $v_{i+1}, \dots, v_k \in a^+ \cup b^+$, $\$yx\$yu_{i+1}v_{i+1}^2 \dots u_k v_k^2 u_{k+1}$ has $\$a^l b^l \$ a^q b^r$ as a prefix for some $q \geq j+2$ and $r \geq j+1$. The string $\$a^l b^l \$ a^q b^r$ is a factor of $z(2)$ and since $z(2)$ is of the form (†), we must have $l \geq r$, but this contradicts $l \leq j$.

Case 1.2. $l \geq j+2$ In this case x must be a suffix of b^j and y must have $a^{j+2}b^2$ as a prefix, so $l = j+2$. Note that

$$\$yx\$yu_{i+1}v_{i+1}^2 \dots u_k v_k^2 u_{k+1} = \$a^l b^l \$yu_{i+1}v_{i+1}^2 \dots u_k v_k^2 u_{k+1}$$

is a suffix of $z(2)$, so either $yu_{i+1}v_{i+1}^2 \dots u_k v_k^2 u_{k+1}$ equals $a^q b^l$ or has $a^q b^l \$$ as a prefix for some q. Since $l = j+2$ and $yu_{i+1}v_{i+1} \dots u_k v_k u_{k+1}$ either equals $a^{j+2}b^{j+1}$ or has $a^{j+2}b^{j+1}\$$ as a prefix, it follows that there is some $h > i$ such that $v_h = b$ and $v_{i+1}, \dots, v_{h-1}$ are all in a^+. But then $z(3)$ will contain

$$\$yx\$yu_{i+1}v_{i+1}^3 \dots u_k v_k^3 u_{k+1},$$

which must have

$$\$a^{j+2}b^{j+2}\$a^{q'}b^{j+3}$$

as a prefix for some q', contradicting the fact that $z(3)$ is of the form (†).

Case 2. v_i contains at least two occurrences of \$. Then we can write

$$v_i = x\$a^{l+1}b^l\$ \dots \$a^{m+1}b^m\$y,$$

where $1 \leq l \leq m \leq p-1$, x is a suffix of $a^l b^{l-1}$, and y is a prefix of $a^{m+2}b^{m+1}$. Since

$$\$a^{m+1}b^m\$yx\$a^{l+1}b^l\$$$

is a factor of $z(2)$, we must have

$$yx = a^l b^{m+1}.$$

Since y is a prefix of $a^{m+2}b^{m+1}$ and $l < m+2$, y must be a prefix of a^l. It follows that x has b^{m+1} as a suffix. But then b^{m+1} must be a suffix of $a^l b^{l-1}$, contradicting the fact that $l-1 < m+1$. □

Since Theorem 5 above implies that every language in Weir's control language hierarchy satisfies the weak Ogden property, we obtain the following corollary:[2]

Corollary 7. *There is a language in* 3-MCFL(1) ∩ 2-MCFL(2) *that lies outside of Weir's control language hierarchy.*

Previously, Kanazawa et al. [7] showed that Weir's control language hiearchy does not include 3-MCFL(2), but left open the question of whether the former includes the languages of well-nested MCFGs. The above corollary settles this question in the negative.

4 A Generalized Ogden's Lemma for a Subclass of the MCFGs

An easy way of ensuring that an m-MCFG G satisfies a generalized Ogden's lemma is to demand that whenever $B(\boldsymbol{x}_1, \dots, \boldsymbol{x}_r) \vdash_G B(\beta_1, \dots, \beta_r)$, each $\boldsymbol{x}_i$ occurs in β_i. This is a rather strict requirement, however, and the resulting class of grammars does not seem to cover even the second level $\mathcal{C}_2$ of the control

[2] The language L in the proof of Theorem 6 was inspired by Lemma 5.4 of Greibach [5], where a much more complicated language was used to show that the range of a deterministic two-way finite-state transducer need not be *strongly iterative*. One can see that the language Greibach used is an 8-MCFL(1). In her proof, Greibach essentially relied on a stronger requirement imposed by her notion of strong iterativity, namely that in the factorization $z = u_1 v_1 \dots u_k v_k u_{k+1}$, there must be some i such that u_i and u_{i+1} contain at least one distinguished position and v_i contains at least *two* distinguished positions. Strong iterativity is not implied by the condition in Theorem 5, so Greibach's lemma fell short of providing an example of a language in $\bigcup_m m$-MCFL(1) that does not belong to Weir's hierarchy.

language hierarchy. In this section, we show that a weaker condition implies a natural analogue of Ogden's [10] condition; we prove in the next section that the result covers the entire control language hierarchy.

Let us say that a derivation of $B(\beta_1, \dots, \beta_r)$ from assumption $A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$ is *non-decreasing* if it cannot be broken down into two derivations witnessing $A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \vdash_G C(\gamma_1, \dots, \gamma_s)$ and $C(\boldsymbol{z}_1, \dots, \boldsymbol{z}_s) \vdash_G B(\beta'_1, \dots, \beta'_r)$ such that $s < q$. (If $q > r$, there can be no non-decreasing derivation witnessing $A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \vdash_G B(\beta_1, \dots, \beta_r)$.) An m-MCFG $G = (N, \Sigma, P, S)$ is *proper* if for each $A \in N^{(q)}$, whenever $A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \vdash_G A(\alpha_1, \dots, \alpha_q)$ with a non-decreasing derivation, each $\boldsymbol{x}_i$ occurs in α_i. It is easy to see that properness is a decidable property of an MCFG.

Theorem 8. *Let L be the language of a proper m-MCFG. There is a natural number p such that for every $z \in L$ and $D \subseteq [1, |z|]$ with $|D| \geq p$, there are strings $u_1, \dots, u_{2m+1}, v_1, \dots, v_{2m}$ satisfying the following conditions:*

1. $z = u_1 v_1 \dots u_{2m} v_{2m} u_{2m+1}$,
2. *for some* $j \in [1, 2m]$,

$$D \cap (u_1 v_1 \dots [u_j] v_j u_{j+1} v_{j+1} \dots u_{2m} v_{2m} u_{2m+1}) \neq \varnothing,$$
$$D \cap (u_1 v_1 \dots u_j [v_j] u_{j+1} v_{j+1} \dots u_{2m} v_{2m} u_{2m+1}) \neq \varnothing,$$
$$D \cap (u_1 v_1 \dots u_j v_j [u_{j+1}] v_{j+1} \dots u_{2m} v_{2m} u_{2m+1}) \neq \varnothing,$$

3. $|D \cap \bigcup_{i=1}^{m} (u_1 v_1 \dots u_{2i-1} [v_{2i-1} u_{2i} v_{2i}] \dots u_{2m} v_{2m} u_{2m+1})| \leq p$,
4. $u_1 v_1^n u_2 v_2^n \dots u_{2m} v_{2m}^n u_{2m+1} \in L$ *for all* $n \in \mathbb{N}$.

The case $m = 1$ of Theorem 8 exactly matches the condition in Ogden's [10] original lemma (as does the case $k = 1$ of Theorem 5).

Proof. Let $G = (N, \Sigma, P, S)$ be a proper m-MCFG. For a rule $A(\alpha_1, \dots, \alpha_q) \leftarrow B_1(\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,q_1}), \dots, B_n(\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,q_n})$, let its *weight* be the number of occurrences of terminal symbols in $\alpha_1, \dots, \alpha_q$ plus n, and let d be the maximal weight of a rule in P.

Let $z \in L$, $D \subseteq [1, |z|]$, and τ be a derivation tree for z. We refer to elements of D as *distinguished positions*. Note that it makes sense to ask whether a particular symbol occurrence in the atom $A(w_1, \dots, w_q)$ labeling a node ν of τ is in a distinguished position or not. This is because by Lemma 2, there are strings $z_1, \dots, z_{q+1}$ such that ν determines a derivation witnessing $A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \vdash_G S(z_1 \boldsymbol{x}_1 z_2 \boldsymbol{x}_2 \dots z_q \boldsymbol{x}_q z_{q+1})$, which tells us where in z each argument of $A(w_1, \dots, w_q)$ ends up. Henceforth, when the ground atom labeling a node ν contains a symbol occurrence in a distinguished position, we simply say that ν contains a distinguished position. We call a node ν a *B-node* (cf. [10]) if at least one of its children contains a distinguished position and ν contains more distinguished positions than any of its children. The *B-height* of a node ν is defined as the maximal B-height h of its children if ν is not a B-node, and $h + 1$ if ν is a B-node. (When ν has no children, its B-height is 0.) It is easy to see that a node of B-height h can contain no more than d^{h+1} distinguished positions.

Our goal is to find an h such that, when $|D| \geq d^{h+1}$, we can locate four nodes $\mu_1, \mu_2, \mu_3, \mu_4$, all of B-height $\leq h$, on the same path of τ that together decompose τ into five derivations witnessing

$$
\begin{aligned}
A(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q) &\vdash_G S(z_1 \boldsymbol{x}_1 z_2 \boldsymbol{x}_2 \ldots z_q \boldsymbol{x}_q z_{q+1}), && (1)\\
B(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q) &\vdash_G A(y_1 \boldsymbol{x}_1 y_2, \ldots, y_{2q-1} \boldsymbol{x}_q y_{2q}), && (2)\\
B(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q) &\vdash_G B(v_1 \boldsymbol{x}_1 v_2, \ldots, v_{2q-1} \boldsymbol{x}_q v_{2q}), && (3)\\
C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q) &\vdash_G B(x_1 \boldsymbol{x}_1 x_2, \ldots, x_{2q-1} \boldsymbol{x}_q x_{2q}), && (4)\\
&\vdash_G C(w_1, \ldots, w_q), && (5)
\end{aligned}
$$

where for some $j \in [1, 2q]$, each of x_j, v_j, y_j contains at least one distinguished position. Since $y_1 v_1 x_1 w_1 x_2 v_2 y_2, \ldots, y_{2q-1} v_{2q-1} x_{2q-1} w_q x_{2q} v_{2q} y_{2q}$ together can contain no more than d^{h+1} distinguished positions, this establishes the theorem, with $p = d^{h+1}$ and $u_1 = z_1 y_1, u_2 = x_1 w_1 x_2, u_3 = y_2 z_2 y_3$, etc.

We let $h = \sum_{q=1}^{m} h(q)$, where $h(0) = 0$ and $h(q) = (2q \cdot (|N| + 1) + 1) \cdot (h(q - 1) + 1)$ for $q \in [1, m]$. By the "dimension" of a node, we mean the dimension of the nonterminal in the label of that node. Assume $|D| \geq d^{h+1}$. Then the root of τ has B-height $\geq h$, and τ must have a path that contains a node of each B-height $\leq h$. For each $i = 0, \ldots, h$, from among the nodes of B-height i on that path, pick a node ν_i of the lowest dimension.

By a *q-stretch*, we mean a contiguous subsequence of $\nu_0, \nu_1, \ldots, \nu_h$ consisting entirely of nodes of dimension $\geq q$. We claim that some q-stretch contains more than $2q \cdot (|N|+1)+1$ nodes of dimension q. For, suppose not. Then we can show by induction on q that $\nu_0, \nu_1, \ldots, \nu_h$ contains no more than $h(q)$ nodes of dimension q, which contradicts $h = \sum_{q=1}^{m} h(q)$. Since the entire sequence $\nu_0, \nu_1, \ldots, \nu_h$ is a 1-stretch, the sequence contains at most $2 \cdot (|N|+1)+1 = h(1)$ nodes of dimension 1. If the sequence contains at most $h(q-1)$ nodes of dimension $q-1$, then there are at most $h(q-1)+1$ maximal q-stretches, so the number of nodes of dimension q in the sequence cannot exceed $(2q \cdot (|N| + 1) + 1) \cdot (h(q - 1) + 1) = h(q)$.

So we have a q-stretch that contains nodes $\nu_{i_0}, \ldots, \nu_{i_k}$ of dimension q for some $q \in [1, m]$, where $k = 2q \cdot (|N| + 1) + 1$. Let A_n be the nonterminal in the label of ν_{i_n}. By the definition of a q-stretch and the way the original sequence $\nu_0, \ldots, \nu_h$ is defined, the nodes of τ that are neither below $\nu_{i_{n-1}}$ nor above ν_{i_n} determine a non-decreasing derivation witnessing $A_{n-1}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q) \vdash_G A_n(x_{n,1} \boldsymbol{x}_1 x_{n,2}, \ldots, x_{n,2q-1} \boldsymbol{x}_q x_{n,2q})$ for some strings $x_{n,1}, \ldots, x_{n,2q}$. Since there must be a B-node lying above $\nu_{i_{n-1}}$ and below or at ν_{i_n}, at least one of $x_{n,1}, \ldots, x_{n,2q}$ must contain a distinguished position. By the pigeon-hole principle, there is a $j \in [1, 2q]$ such that $\{\, n \in [1, k] \mid x_{n,j}$ contains a distinguished position $\}$ has at least $|N| + 2$ elements. This means that we can pick three elements n_1, n_2, n_3 from this set so that $n_1 < n_2 < n_3$ and $A_{n_1} = A_{n_2}$. Letting $\mu_1 = \nu_{i_0}, \mu_1 = \nu_{i_{n_1}}, \mu_2 = \nu_{i_{n_2}}, \mu_3 = \nu_{i_{n_3}}$, we see that (2), (3), (4) hold with $C = A_{i_0}, B = A_{i_{n_1}} = A_{i_{n_2}}, A = A_{i_{n_3}}$ and x_j, v_j, y_j all containing a distinguished position, as desired. □

Let us write m-MCFL$_{\text{prop}}$ for the family of languages generated by proper m-MCFGs. Using standard techniques (cf. Theorem 3.9 of [12]), we can easily show that for each $m \geq 1$, m-MCFL$_{\text{prop}}$ is a substitution-closed full abstract family of languages.

5 Relation to the Control Language Hierarchy

Kanazawa and Salvati [8] showed $\mathcal{C}_k \subseteq 2^{k-1}$-MCFL for each k through a tree grammar generating the derivation trees of a level k control grammar (G, C). In fact, detour through tree languages is not necessary—a level k control language can be obtained from a level $k-1$ control language by certain string language operations. It is easy to see that the family $\bigcup_m m$-MCFL$_{\text{prop}}$ is closed under those operations.

Let us sketch the idea using Example 4. We start by applying a *homomorphic replication* [2,5] $\langle (1, R), h_1, h_2 \rangle$ to the control set $C = \{\, \pi_1^n \pi_2^n \pi_3 \mid n \in \mathbb{N} \,\}$, obtaining

$$\langle (1, R), h_1, h_2 \rangle (C) = \{\, h_1(w) h_2(w^R) \mid w \in C \,\}, \tag{6}$$

where $h_1(\pi_1) = a, h_1(\pi_2) = b, h_1(\pi_3) = \varepsilon, h_2(\pi_1) = \bar{a}S, h_2(\pi_2)\bar{b}S, h_2(\pi_3) = \varepsilon$. For instance, $\pi_1^2 \pi_2^2 \pi_3$ is mapped to $aabb\bar{b}S\bar{b}S\bar{a}S\bar{a}S$. Iterating the substitution $S \leftarrow \langle (1, R), h_1, h_2 \rangle (C)$ on the resulting language and then throwing away strings that contain S gives the language of the control grammar of this example.

In general, if π is a production $A \to w_0 B_1 w_1 \dots B_n w_n$ of a labeled distinguished grammar $G = (N, \Sigma, P, S, f)$ and $f(\pi) = i \in [1, n]$, then we let $h_1(\pi) = w_0 B_1 w_1 \dots B_{i-1} w_{i-1}$ and $h_2(\pi) = w_i B_{i+1} w_{i+1} \dots B_n w_n$. In case $f(\pi) = 0$, $h_1(\pi)$ is the entire right-hand side of π and $h_2(\pi) = \varepsilon$. The control set C is first intersected with a local set so as to ensure consistency of nonterminals in adjacent productions, and then partitioned into sets C_A indexed by nonterminals, with C_A holding only those strings whose first symbol is a production that has A on its left-hand side. Let $L_A = \langle (1, R), h_1, h_2 \rangle (C_A)$ for each $A \in N$. The final operation is iterating simultaneous substitution $A \leftarrow L_A$ and throwing away strings containing nonterminals:

$$L_0 = L_S, \qquad L_{n+1} = L_n[A \leftarrow L_A]_{A \in N}, \qquad L = \bigcup_n L_n \cap \Sigma^*. \tag{7}$$

This last step may be thought of as the fixed point computation of a "context-free grammar" with an infinite set of rules $\{\, A \to \alpha \mid A \in N, \alpha \in L_A \,\}$.

Lemma 9. *If $L \in m$-MCFL$_{\text{prop}}$ and h_1, h_2 are homomorphisms, then the language $\langle (1, R), h_1, h_2 \rangle (L)$ defined by* (6) *belongs to* $2m$-MCFL$_{\text{prop}}$.

Example 1 in Section 2.1 illustrates Lemma 9 with $m = 1$, $L = D_1^*$, and h_1, h_2 both equal to the identity function.

The proof of the next lemma is similar to that of closure under substitution.

Lemma 10. *If $L_A \subseteq (N \cup \Sigma)^*$ belongs to m-MCFL$_{\text{prop}}$ for each $A \in N$, then the language L defined by* (7) *also belongs to m-MCFL$_{\text{prop}}$.*

Theorem 11. *For each $k \geq 1$, $\mathcal{C}_k \subsetneq 2^{k-1}$-MCFL$_{\text{prop}}$.*

Again, the language $\mathrm{RESP}_{2^{k-1}}$ separates 2^{k-1}-$\mathrm{MCFL}_{\mathrm{prop}}$ from $\mathcal{C}_k$. For $k = 2$, $\{ w\#w \mid w \in D_1^* \}$ also witnesses the separation. I currently do not see how to settle the question of whether the inclusion of $\bigcup_k \mathcal{C}_k$ in $\bigcup_m m$-$\mathrm{MCFL}_{\mathrm{prop}}$ is strict.

6 Conclusion

Theorem 5 and Theorem 8 with $m = 2^{k-1}$ both apply to languages in $\mathcal{C}_k$, but place incomparable requirements on the factorization $z = u_1 v_1 \dots u_{2^k} v_{2^k} u_{2^k+1}$. Theorem 8 does not require $v_{2^{k-1}} u_{2^{k-1}} v_{2^{k-1}+1}$ to contain $\leq p$ distinguished positions. On the other hand, it does not seem easy to derive additional restrictions on $v_{2i-1} u_{2i} v_{2i}$ from Palis and Shende's [11] proof. From the point of view of MCFGs, the conditions in Theorem 8 are very natural: the substrings that are simultaneously iterated should contain only a small number of distinguished positions.

References

1. Engelfriet, J.: Context-free graph grammars. In: Handbook of Formal Languages, Volume 3: Beyond Words, pp. 125–213. Springer, Berlin (1997)
2. Ginsburg, S., Spanier, E.H.: AFL with the semilinear property. Journal of Computer and System Sciences 5(4), 365–396 (1971)
3. Greibach, S.A.: Hierarchy theorems for two-way finite state transducers. Acta Informatica 11, 89–101 (1978)
4. Greibach, S.A.: One-way finite visit automata. Theoretical Computer Science 6, 175–221 (1978)
5. Greibach, S.A.: The strong independence of substitution and homomorphic replication. R.A.I.R.O. Informatique théorique 12(3), 213–234 (1978)
6. Kanazawa, M.: The pumping lemma for well-nested multiple context-free languages. In: Diekert, V., Nowotka, D. (eds.) Developments in Language Theory: 13th International Conference, DLT 2009. Lecture Notes in Computer Science, vol. 5583, pp. 312–325. Springer, Berlin (2009)
7. Kanazawa, M., Kobele, G.M., Michaelis, J., Salvati, S., Yoshinaka, R.: The failure of the strong pumping lemma for multiple context-free languages. Theory of Computing Systems 55(1), 250–278 (2014)
8. Kanazawa, M., Salvati, S.: Generating control languages with abstract categorial grammars. In: Preliminary Proceedings of FG-2007: The 12th Conference on Formal Grammar (2007)
9. Kanazawa, M., Salvati, S.: The copying power of well-nested multiple context-free grammars. In: Dediu, A.H., Fernau, H., Martín-Vide, C. (eds.) Language and Automata Theory and Applications, Fourth International Conference, LATA 2010. Lecture Notes in Computer Science, vol. 6031, pp. 344–355. Springer, Berlin (2010)
10. Ogden, W.: A helpful result for proving inherent ambiguity. Mathematical Systems Theory 2(3), 191–194 (1968)
11. Palis, M.A., Shende, S.M.: Pumping lemmas for the control language hierarchy. Mathematical Systems Theory 28(3), 199–213 (1995)
12. Seki, H., Matsumura, T., Fujii, M., Kasami, T.: On multiple context-free grammars. Theoretical Computer Science 88(2), 191–229 (1991)
13. Weir, D.J.: A geometric hierarchy beyond context-free languages. Theoretical Computer Science 104(2), 235–261 (1992)