

On the Generative Power of Multiple Context-Free Grammars and Macro Grammars

Hiroyuki SEKI^{†a)}, Member and Yuki KATO^{††}, Nonmember

SUMMARY Several grammars of which generative power is between context-free grammar and context-sensitive grammar were proposed. Among them are macro grammar and tree adjoining grammar. Multiple context-free grammar is also a natural extension of context-free grammars, and is known to be stronger in its generative power than tree adjoining grammar and yet to be recognizable in polynomial time. In this paper, the generative power of several subclasses of variable-linear macro grammars and that of multiple context-free grammars are compared in details.

key words: multiple context-free grammar, macro grammar, context-free tree grammar, generative power, linearity

1. Introduction

Several grammars of which generative power is between context-free grammar (cfg) and context-sensitive grammar (csg) were proposed. Among them, multiple context-free grammar (mcfg) [10], [24] is a natural extension of cfg. A nonterminal symbol of an mcfg derives tuples of strings while a nonterminal symbol of a cfg derives strings. Mcfg inherits good properties of cfg. The recognition (or membership) problem for mcfg is solvable in polynomial time of the length of an input string. The class of languages generated by mcfgs is a full AFL. There are a few formalisms of which generative power is the same as mcfg. String-based linear context-free rewriting system [27] is essentially the same formalism as mcfg. Later, Weir [28] showed that the generative power of mcfg is equal to that of finite-copying tree transducer [5]. Rambow and Satta showed that the generative powers of mcfg and local unordered scattered context grammar are the same [18], [19]. Tree adjoining grammar (tag) [8], [9] generates a proper subclass of the class of languages generated by mcfgs.

Context-free tree grammar [21], [22] (cftg) is another extension of cfg obtained by introducing arguments into nonterminal symbols. While derivation of cftg is defined over trees (or terms) rather than strings, the class of yield languages generated by cftgs is known to be the same as the class of languages generated by macro grammars [6] and that of indexed grammars [2]. Whether the membership problem for macro grammar is solvable in polynomial time

of the size of an input string is not known though there exists a macro grammar that generates an NP-complete language. Also, it is an open problem whether the generative power of cftg is properly stronger than that of mcfg (or vice versa).

In this paper, the generative power of several subclasses of variable-linear macro grammars and that of multiple context-free grammars are compared in details. A macro grammar (mg) is variable-linear if each variable in the left-hand side of a rule appears at most once in its right-hand side. An mg is double-linear if it is variable-linear and the number of nonterminal symbols in the right-hand side of a rule is at most one. We show that

$$\begin{aligned} 1\text{-MCFL}(m) = L^2\text{-ML}(m-1) \subset VL\text{-ML}(m-1) \\ \subseteq 2\text{-MCFL}(m) \quad (m \geq 1) \end{aligned}$$

where $r\text{-MCFL}(m)$ is the class of languages generated by mcfgs with dimension at most m and rank at most r , $VL\text{-ML}(m)$ and $L^2\text{-ML}(m)$ are the classes of languages generated by variable-linear mgs and double-linear mgs with arity at most m , respectively. It is also shown that the rightmost inclusion is proper when $m \geq 2$. Inclusion relations are also shown in Fig. 2 in Sect. 4, where * indicates the new results obtained in this paper. Detailed proofs can be found in [23], some of which are also given in appendix of this paper.

The main purpose of this paper is to formally compare the generative power of grammars between cfg and csg, but these grammars have some real-world applications. Grammars considered in this paper have been mainly applied to the description of natural language syntax in computational linguistics and the syntax-directed translation in compiler construction. Recently, these grammars have been paid much attention in bioinformatics. For example, secondary structure of biological sequences such as RNA and protein is modeled by these grammars so that secondary structure prediction can be realized by parsing in stochastic extension of these grammars [3], [4]. Early studies applied parsing methods of stochastic cfg to structure prediction [4]. However, it has been pointed out that secondary structure contains substructures that cannot be represented by cfg. To solve this problem, later studies used grammars of which generative power is greater than cfg. Among them, a subclass of tag was applied to RNA secondary structure prediction [17], [26] and to RNA secondary structure alignment [25]. For the former problem, Rivas and Eddy used RNA pseudoknot grammar [20], and Kato, et al. used a subclass of mcfgs [13]. These grammars can be naturally considered as subclasses

Manuscript received April 2, 2007.

Manuscript revised June 28, 2007.

[†]The author is with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.

^{††}The author is with Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji-shi, 611-0011 Japan.

a) E-mail: seki@is.naist.jp

DOI: 10.1093/ietisy/e91-d.2.209

of mcfgs, and Kato, et al. [12] clarified the relation between the generative power of these grammars.

Meanwhile, Abe and Mamitsuka used a subclass of cftgs for protein secondary structure prediction [1]. They defined a *ranked node rewriting grammar (rnrg)* and a *linear rnrg* as a variable-linear cftg and a double-linear cftg, respectively, that satisfy nonerasing and nonpermuting conditions in this paper[†]. Let $\text{RNRL}(m)$ and $\text{L-RNRL}(m)$ be the classes of yield languages generated by rnrgs and l-rnrgs with nonterminals of which arity is at most m , respectively. They claimed that $\text{L-RNRL}(m) \subset \text{RNRL}(m)$, and $\text{RNRL}(m)$ and $\text{L-RNRL}(m)$ can be parsed in $O(n^{3(m+1)})$ and $O(n^{2(m+1)})$, respectively, but formal proofs were not provided. They also presented a parsing algorithm for stochastic version of l-rnrg(1) and reported the experimental results of applying the algorithm to protein structure prediction. By Lemmas 7 and 8, $\text{RNRL}(m) = \text{VL-ML}(m)$ and $\text{L-RNRL}(m) = \text{L}^2\text{-ML}(m)$. Thus, the above claims for RNRL and L-RNRL can be obtained as corollaries of Theorem 20 and Corollaries 11 and 18. These corollaries are based on the property of an mcfg G that for a given $w \in T^*$, whether $w \in L(G)$ can be decided in $O(n^e)$ time where $n = |w|$ and e is a constant depending only on G (Proposition 2). This property was proved in a constructive way in [11], [24] by providing a so-called CYK-style recognition algorithm scheme for mcfg. Also, the proofs for comparing the generative power of mcfg and mg in this paper are constructive; we provide a translation from variable-linear mg to 2-mcfg and a bidirectional transformation between double-linear mg and 1-mcfg. Hence, once a secondary structure model of specific biological sequences is described as an mg (or equivalently, rnrg), we only have to translate the mg to an equivalent mcfg, and use the CYK algorithm for mcfg; it is not needed to implement a recognition algorithm from scratch.

As a final remark, closure properties of the class of *tree* languages generated by vl-mgs were extensively studied in [14].

2. Preliminaries

2.1 Multiple Context-Free Grammar

We will use standard notions and notations on strings and languages. Let ε denote the empty string. Let Γ be a finite alphabet. For a string $\alpha \in \Gamma^*$ and a symbol $a \in \Gamma$, let $|\alpha|$ denote the number of symbols appearing in α , called the length of α , and let $|\alpha|_a$ denote the number of a 's appearing in α . Let \subseteq and \subset denote the set inclusion relation and the proper set inclusion relation, respectively. A *multiple context-free grammar (mcfg)* is a 5-tuple $G = (N, T, F, P, S)$ where N is a finite set of nonterminals, T a finite set of terminals, F a finite set of *mcfg-functions* defined below and $S \in N$ the start symbol. For each $A \in N$, a positive integer denoted as $\dim(A)$ is given and A derives $\dim(A)$ -tuples of terminal strings. For the start symbol S , $\dim(S) = 1$. We say that f is an *mcfg-function* if a nonnegative integer k and positive

integers d_i ($0 \leq i \leq k$) are given and f is a total function from $(T^*)^{d_1} \times \dots \times (T^*)^{d_k}$ to $(T^*)^{d_0}$ satisfying the following condition (F):

- (F) Let $\bar{x}_i = (x_{i1}, \dots, x_{id_i})$ denote the i th argument of f for $1 \leq i \leq k$. The h th component of function value for $1 \leq h \leq d_0$, denoted by $f^{[h]}$, is defined as

$$f^{[h]}[\bar{x}_1, \dots, \bar{x}_k] = \beta_{h0} z_{h1} \beta_{h1} z_{h2} \dots z_{hv_h} \beta_{hv_h} \quad (*)$$

where v_h is a nonnegative integer, $\beta_{hl} \in T^*$ ($0 \leq l \leq v_h$) and $z_{hl} \in \{x_{ij} \mid 1 \leq i \leq k, 1 \leq j \leq d_i\}$ ($1 \leq l \leq v_h$). The total number of occurrences of x_{ij} in the right-hand sides of (*) from $h = 1$ through d_0 is at most one. For example, $f[(x_{11}, x_{12}), (x_{21}, x_{22})] = (x_{11}x_{21}, x_{12}x_{22})$.

Each rule in P has the form of $A_0 \rightarrow f[A_1, \dots, A_k]$ where $A_i \in N$ ($0 \leq i \leq k$) and $f : (T^*)^{\dim(A_1)} \times \dots \times (T^*)^{\dim(A_k)} \rightarrow (T^*)^{\dim(A_0)} \in F$. If $k \geq 1$, the rule is called a *nonterminating rule*, and if $k = 0$, it is called a *terminating rule*. A terminating rule $A_0 \rightarrow f[\]$ with $f^{[h]}[\] = \beta_h$ ($1 \leq h \leq \dim(A_0)$) is simply written as $A_0 \rightarrow (\beta_1, \dots, \beta_{\dim(A_0)})$.

Example 1. Let m be an arbitrary positive integer.

- (1) Let $G_1 = (N_1, T_1, F_1, P_1, S)$ be an mcfg where $N_1 = \{S, A\}$, $T_1 = \{a, b\}$ and $P_1 = \{S \rightarrow J[A], A \rightarrow f_a[A] \mid f_b[A] \mid (\varepsilon, \varepsilon)\}$ where $\dim(S) = 1$, $\dim(A) = 2$, $J[(x_1, x_2)] = x_1x_2$ and $f_a[(x_1, x_2)] = (\alpha x_1, \alpha x_2)$ with $\alpha = a, b$.
- (2) Let $G_2^{(m)} = (N_2^{(m)}, T_2^{(m)}, F_2^{(m)}, P_2^{(m)}, S)$ be an mcfg where $N_2^{(m)} = \{S, A^{(m)}\}$, $T_2^{(m)} = \{a_i \mid 1 \leq i \leq 2m\}$ and $P_2^{(m)} = \{S \rightarrow J_m[A^{(m)}], A^{(m)} \rightarrow g_m[A^{(m)}] \mid (\varepsilon, \dots, \varepsilon)\}$ where $\dim(S) = 1$, $\dim(A^{(m)}) = m$, $J_m[(x_1, \dots, x_m)] = x_1 \dots x_m$ and $g_m[(x_1, \dots, x_m)] = (a_1x_1a_2, \dots, a_{2m-1}x_m a_{2m})$.
- (3) Let $G_3^{(m)} = (N_3^{(m)}, T_3^{(m)}, F_3^{(m)}, P_3^{(m)}, S)$ be an mcfg where $P_3^{(m)} = \{S \rightarrow J_m^2[A^{(m)}, A^{(m)}], A^{(m)} \rightarrow g[A^{(m)}] \mid (\varepsilon, \dots, \varepsilon)\}$ where $J_m^2[(x_1, \dots, x_m), (y_1, \dots, y_m)] = x_1 \dots x_m y_1 \dots y_m$ and g_m is defined as in (2). \square

For a function f defined by (*) in condition (F) and tuples of terminal strings $\alpha_i = (\alpha_{i1}, \dots, \alpha_{id_i}) \in (T^*)^{d_i}$ ($1 \leq i \leq k$), let $f[\alpha_1, \dots, \alpha_k]$ denote the tuple of terminal strings obtained from the right-hand sides of (*) by substituting α_{ij} ($1 \leq i \leq k, 1 \leq j \leq \dim(A_i)$) into x_{ij} . For example, $f_a[(bba, ab)] = (abba, aab)$ in Example 1. We recursively define the relation $\stackrel{*}{\Rightarrow}$ by the following (L1) and (L2):

- (L1) If $A \rightarrow \alpha \in P$ ($\alpha \in (T^*)^{\dim(A)}$), we write $A \stackrel{*}{\Rightarrow} \alpha$.
- (L2) If $A \rightarrow f[A_1, \dots, A_k] \in P$ and $A_i \stackrel{*}{\Rightarrow} \alpha_i$ ($1 \leq i \leq k$), we write $A \stackrel{*}{\Rightarrow} f[\alpha_1, \dots, \alpha_k]$.

Let $G = (N, T, F, P, S)$ be an mcfg. For $A \in N$, the set generated from A in G is defined as $L_G(A) = \{w \in (T^*)^{\dim(A)} \mid A \stackrel{*}{\Rightarrow} w\}$ and the language generated by G is defined as

[†]Unfortunately, the word *rank* is used for different meaning in mcfg and cftg. In this paper, the word *arity* will be used instead of *rank* for cftg and mg.

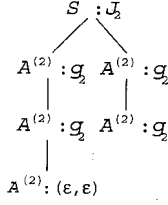


Fig. 1 A derivation tree.

$L(G) = L_G(S)$. A language L is a *multiple context-free language (mcf)* if there exists an mcfg G such that $L = L(G)$. The class of all mcfgs and the class of all mcfls are denoted by MCFG and MCFL, respectively. The same notational convention will be used for other classes of grammars and languages. In parallel with the relation \Rightarrow^* , we define derivation trees:

- (D1) If $A \rightarrow \alpha \in P$ ($\alpha \in (T^*)^{\dim(A)}$), then a derivation tree of α is the tree with a single node labeled $A : \alpha$.
- (D2) If $A \rightarrow f[A_1, \dots, A_k] \in P$, $A_i \xRightarrow{*} \alpha_i$ ($1 \leq i \leq k$) and t_1, \dots, t_k are derivation trees of $\alpha_1, \dots, \alpha_k$, then a derivation tree of $f[\alpha_1, \dots, \alpha_k]$ is the tree with the root labeled $A : f$ that has t_1, \dots, t_k as (immediate) subtrees from left to right.

- Example 1** (continued). (1) By (L1), $A \xRightarrow{*}_{G_1} (\varepsilon, \varepsilon)$ since $A \rightarrow (\varepsilon, \varepsilon) \in P$. Since $f_a[(\varepsilon, \varepsilon)] = (a, a)$ and $f_b[(a, a)] = (ba, ba)$, we have $A \xRightarrow{*}_{G_1} (a, a)$ and $A \xRightarrow{*}_{G_1} (ba, ba)$ by (L2). Also by $S \rightarrow J[A]$, $S \xRightarrow{*}_{G_1} J[(ba, ba)] = baba$. In fact, $L_{G_1}(A) = \{(w, w) \mid w \in \{a, b\}^*\}$ and $L(G_1) = \{ww \mid w \in \{a, b\}^*\}$.
- (2) Likewise, $A^{(m)} \xRightarrow{*}_{G_2^{(m)}} (\varepsilon, \dots, \varepsilon)$ by (L1), $A^{(m)} \xRightarrow{*}_{G_2^{(m)}} f[(\varepsilon, \dots, \varepsilon)] = (a_1 a_2, \dots, a_{2m-1} a_{2m})$ by (L2), etc. This tells us that $L(G_2^{(m)}) = \{a_1^n \cdots a_{2m}^n \mid n \geq 0\}$. This language is called $L_1^{(m)}$ in the rest of the paper.
- (3) Since $L_{G_3^{(m)}}(A^{(m)}) = L_{G_2^{(m)}}(A^{(m)})$, $L(G_3^{(m)}) = \{\alpha\beta \mid \alpha, \beta \in L_1^{(m)}\}$. This language is called $L_2^{(m)}$. \square

Figure 1 shows a derivation tree of $a_1 a_2 a_3 a_4 a_1 a_2 a_3 a_4$ in $G_3^{(2)}$ of Example 1.

To introduce subclasses of MCFG, we define a few notations. For an mcf-function $f : (T^*)^{d_1} \times \dots \times (T^*)^{d_k} \rightarrow (T^*)^{d_0}$, we define $\dim(f)$, $\text{rank}(f)$ and $\text{deg}(f)$ called the *dimension*, *rank* and *degree* of f , respectively, as in Table 1. For an mcfg $G = (N, T, F, P, S)$, let $\dim(G)$, $\text{rank}(G)$ and $\text{deg}(G)$ denote the maximum of $\dim(f)$, $\text{rank}(f)$ and $\text{deg}(f)$ among all $f \in F$, respectively. For example, $\dim(G_3) = m$, $\text{rank}(G_3) = 2$ and $\text{deg}(G_3) = 2m + 1$. By definition, $\text{deg}(G) \leq \dim(G)(\text{rank}(G) + 1)$.

An r -mcf $g(m)$ is an mcfg G with $\dim(G) \leq m$ and $\text{rank}(G) \leq r$. Likewise, an mcf $g(m)$ is an mcfg G with $\dim(G) \leq m$ and an r -mcf g is an mcfg G with $\text{rank}(G) \leq r$. Thus, G_1 is a 1-mcf $g(2)$ and G_3 is a 2-mcf $g(m)$.

For grammars G_1 and G_2 , we say that G_1 is weakly equivalent to G_2 if $L(G_1) = L(G_2)$.

Table 1 Parameters of mcf g .

notation	definition	examples	
		f_α	J_m^2
$\dim(f)$	$\max\{d_i \mid 0 \leq i \leq k\}$	2	m
$\text{rank}(f)$	k	1	2
$\text{deg}(f)$	$\sum_{j=0}^k d_j$	4	$2m + 1$

Lemma 1 (Normal form mcf g [10], [24]). For a given r -mcf $g(m)$ G , we can construct an r -mcf $g(m)$ $G' = (N', T', F', P', S')$ that is weakly-equivalent to G and satisfies the following conditions:

- (N1) (nonerasing) For any $f \in F'$, every variable appears exactly once in the right-hand side of $(*)$ for some h ($1 \leq h \leq d_0$) in definition (F).
- (N2) For any $A \in N'$ ($A \neq S'$), $A \xRightarrow{*}_{G'} (\alpha_1, \dots, \alpha_{\dim(A)})$ implies $\alpha_i \neq \varepsilon$ ($1 \leq i \leq \dim(A)$).
- (N3) If $A \rightarrow \varepsilon$, then $A = S'$ and S' does not appear in the right-hand side of any rule in P' . \square

Recognition (or membership) problem for mcf g can be solved in polynomial time:

Proposition 2 ([11], [24]). Let G be an mcf g with $\text{deg}(G) = e$. For a given $w \in T^*$, whether $w \in L(G)$ or not can be decided in $O(n^e)$ time where $n = |w|$. \square

2.2 Macro Grammar

Let $\Sigma = \bigcup_{k \geq 0} \Sigma_k$ be a family of indexed alphabets where $\Sigma_k \cap \Sigma_{k'} = \emptyset$ for $k \neq k'$. For $f \in \Sigma_k$, we write $a(f) = k$, called the *arity* of f . For a countable set X of variables, let $T_\Sigma(X)$ and $T_\Sigma^+(X)$ denote the sets of *terms* and *sequence-terms* (or *s-terms*) generated by Σ and X respectively, defined as the smallest sets satisfying the following conditions:

- (1) $\Sigma_0 \cup X \subseteq T_\Sigma(X)$.
- (2) $f(t_1, \dots, t_n) \in T_\Sigma(X)$ if $t_i \in T_\Sigma^+(X)$ ($1 \leq i \leq n$) and $f \in \Sigma_n$.
- (3) $t_1 \cdots t_l \in T_\Sigma^+(X)$ if $t_i \in T_\Sigma(X)$ ($1 \leq i \leq l$) and $l \geq 1$.

For $n \geq 0$, let $X_n = \{x_1, x_2, \dots, x_n\} \subseteq X$. For an s-term t , its subterm t_1 and an s-term t_2 , let $t[t_1 \leftarrow t_2]$ denote the s-term obtained from t by replacing one of the occurrences of the subterm t_1 with t_2 . A *substitution* $\theta : X \rightarrow T_\Sigma^+(X)$ is a mapping that is an identity except for a finite subset of domain X , and θ is uniquely extended to a mapping $\theta : T_\Sigma^+(X) \rightarrow T_\Sigma^+(X)$. We write $t\theta$ to denote the result of applying a substitution θ to an s-term t . We sometimes write an s-term t containing different variables x_1, \dots, x_n as $t[x_1, \dots, x_n]$. For a substitution θ such that $\theta(x_i) = t_i$ ($1 \leq i \leq n$) and $\theta(x) = x$ otherwise, we write $t[t_1, \dots, t_n]$ to denote $t[x_1, \dots, x_n]\theta$.

A *macro grammar* (abbreviated as *mg*) is a 4-tuple $G = (N, T, P, S)$ where $N = \bigcup_{k \geq 0} N_k$ is an indexed, finite alphabet of nonterminals, T is a (nonindexed) finite set of terminals, and $S \in N_0$ is the start symbol. Let $\Sigma_0 = N_0 \cup T$ and $\Sigma_k = N_k$ ($k \geq 1$). P is a finite set of (production) rules of which

shapes are:

$$S \rightarrow \varepsilon, \text{ or}$$

$$A(x_1, \dots, x_k) \rightarrow t_A \text{ where } A \in N_k \text{ and } t_A \in T_{\Sigma}^+(X_k).$$

If $S \rightarrow \varepsilon \in P$, S does not appear in the right-hand side of any rule. For an s-term t , its subterm t' and a rule $A(x_1, \dots, x_k) \rightarrow t_A$, if $t' = A(t_1, \dots, t_k)$, then we write $t \Rightarrow_G t[t' \leftarrow t_A\theta]$ where θ is the substitution defined by $\theta(x_i) = t_i$ ($1 \leq i \leq k$) and $\theta(x) = x$ otherwise. Let \Rightarrow_G^* and \Rightarrow_G^+ be the reflexive-transitive closure and the transitive closure of \Rightarrow_G , respectively. We will omit the subscript G if G is clear from the context.

Let us define $L(G) = \{t \in T^* \mid S \xRightarrow{*}_G t\}$, called the language generated by G . A language L is a *macro language* (abbreviated as *ml*) if there exists an mg G such that $L = L(G)$.

For an mg $G = (N, T, P, S)$, if $N_k = \emptyset$ for every $k > m$, G is called an $\text{mg}(m)$. Let $A(x_1, \dots, x_k) \rightarrow t_A \in P$ be a rule. If each variable appears in t_A at most once, the rule is called *variable-linear* (*v-linear*). If there exists at most one nonterminal in t_A , the rule is called *nonterminal-linear* (*n-linear*). If the rule is v-linear and n-linear, the rule is *double-linear*. For example, $A(x) \rightarrow B(x)C$ is v-linear and not n-linear. $A(x) \rightarrow B(xx)$ is not v-linear but n-linear. $A(x) \rightarrow B(axb)$ is double-linear. An mg G is a vl-mg, nl-mg and l^2 -mg if every rule of G is v-linear, n-linear and double-linear, respectively. We define vl-mg(m), nl-mg(m) and l^2 -mg(m) in a similar way.

Example 2. Let m be an arbitrary positive integer.

- (1) Let $G_4^{(m)} = (N_4^{(m)}, T_4^{(m)}, P_4^{(m)}, S)$ be an mg where $N_4^{(m)} = \{S, A^{(m)}\}$ with $a(S) = 0$ and $a(A^{(m)}) = m - 1$, $T_4^{(m)} = \{a_i \mid 1 \leq i \leq 2m\}$ and $P_4^{(m)} = \{S \rightarrow a_1 A^{(m)}(a_2 a_3, \dots, a_{2m-2} a_{2m-1}) a_{2m} \mid \varepsilon, A^{(m)}(x_1, \dots, x_{m-1}) \rightarrow a_1 A^{(m)}(a_2 x_1 a_3, \dots, a_{2m-2} x_{m-1} a_{2m-1}) a_{2m} \mid x_1 \cdots x_{m-1}\}$. $G_4^{(m)}$ is an l^2 -mg($m - 1$). $S \Rightarrow_{G_4} \varepsilon$ and

$$\begin{aligned} S &\Rightarrow_{G_4} a_1 A^{(m)}(a_2 a_3, \dots, a_{2m-2} a_{2m-1}) a_{2m} \\ &\Rightarrow_{G_4} a_1^2 A^{(m)}(a_2^2 a_3^2, \dots, a_{2m-2}^2 a_{2m-1}^2) a_{2m}^2 \\ &\xRightarrow{*}_{G_4} a_1^i a_2^i \cdots a_{2m}^i \quad (i \geq 0). \end{aligned}$$

Thus $L(G_4^{(m)}) = L_1^{(m)}$ (Example 1 (2)).

- (2) Let $G_5^{(m)} = (N_5^{(m)}, T_4^{(m)}, P_5^{(m)}, S_5)$ be an mg where $N_5^{(m)} = \{S_5, S, A^{(m)}\}$ with $a(S_5) = 0$, and $P_5^{(m)} = \{S_5 \rightarrow S S\} \cup P_4^{(m)}$. $G_5^{(m)}$ is a vl-mg($m - 1$) but not n-linear. $L(G_5^{(m)}) = L_2^{(m)}$ (Example 1 (3)).
- (3) Let $G_6 = (N_6, T_6, P_6, S)$ be an mg where $N_6 = \{S, A\}$ with $a(S) = 0$ and $a(A) = 1$, $T_6 = \{a\}$ and $P_6 = \{S \rightarrow A(a), A(x) \rightarrow A(xx) \mid x\}$. G_6 is an nl-mg(1) but not v-linear. $S \Rightarrow_{G_6} A(a) \Rightarrow_{G_6} A(a^2) \Rightarrow_{G_6} A(a^4) \Rightarrow_{G_6} \cdots$. $L(G_6) = \{a^{2^n} \mid n \geq 0\}$, which is not an mcfl [10], [24]. \square

2.3 Known Results

The following hierarchy theorem on ranks was shown by

Rambow and Satta [18], [19].

Proposition 3. For $m \geq 2$, $r \geq 1$ except for $m = 2$ and $r = 2$,

$$r\text{-MCFL}(m) \subset (r + 1)\text{-MCFL}(m).$$

$$2\text{-MCFL}(2) = 3\text{-MCFL}(2).$$

For $r \geq 2$, $1\text{-MCFL}(1) \subset r\text{-MCFL}(1) = (r + 1)\text{-MCFL}(1)$. \square

By definition, it is easy to see $\text{LCFL} = 1\text{-MCFL}(1)$ and $\text{CFL} = r\text{-MCFL}(1)$ ($r \geq 2$) where CFL and LCFL are the classes of context-free languages (cfls) and linear cfls, respectively. Another hierarchy theorem on dimension was shown in [10], [24].

Proposition 4. For $m \geq 1$,

$$\text{MCFL}(m) \subset \text{MCFL}(m + 1),$$

$$L_1^{(m+1)} \in 1\text{-MCFL}(m + 1) \setminus \text{MCFL}(m). \quad \square$$

A tradeoff between dimension and rank was also investigated in [18], [19].

Proposition 5. For $m \geq 1$, $r \geq 3$ and $1 \leq k \leq r - 2$,

$$r\text{-MCFL}(m) \subseteq (r - k)\text{-MCFL}((k + 1)m). \quad \square$$

As a corollary, $r\text{-MCFL}(m) \subseteq 2\text{-MCFL}((r - 1)m)$ ($m \geq 1$, $r \geq 3$). Thus, $\text{MCFL} = 2\text{-MCFL}$. Also, we have $\text{LCFL} = L^2\text{-ML}(0) = \text{NL-ML}(0)$ and $\text{CFL} = \text{VL-ML}(0) = \text{ML}(0)$ by definition and $\text{TAL} = \text{VL-ML}(1)$ by [1], [7].

Finally, we present a few closure properties.

Proposition 6. $r\text{-MCFL}(m)$ ($m \geq 1$, $r \geq 2$), $\text{MCFL}(m)$ ($m \geq 1$) and $r\text{-MCFL}$ ($r \geq 2$) are all substitution closed full AFLs[†]. $1\text{-MCFL}(m)$ ($m \geq 1$) is not closed under concatenation. \square

The first claim of the above proposition was shown by [18], [19], the second by [24], the third by [5] (as a closure property of tree transducers), and the last claim by [15] (as a closure property of EDTOL systems).

3. Normal Forms for Macro Grammars

In this section, we will discuss simplification of macro grammars. First, we will show that an arbitrary mg(m) G can be transformed into a weakly equivalent mg G' such that every variable in the left-hand side of a rule also occurs in its right-hand side (nonerasing) (Lemma 7). Next, it is shown that any nonerasing vl-mg(m) G can be transformed into a weakly equivalent mg(m) G' such that variables appear in ascending order of suffixes x_1, x_2, \dots in the right-hand side of each rule (nonpermuting) (Lemma 8). Lastly, it is shown

[†]A full AFL (abstract family of languages) is a class of languages closed under homomorphism, inverse homomorphism, intersection with regular languages, union, concatenation and Kleene closure.

that any vl-mg can be transformed into a normal form like Chomsky normal form for cfg. These simplifications help us establish the weak equivalence between subclasses of MCFL and ML in the following sections.

Lemma 7. Let G be an $mg(m)$ (rsp. vl- $mg(m)$, nl- $mg(m)$, l^2 - $mg(m)$). We can construct an $mg(m)$ (rsp. vl- $mg(m)$, nl- $mg(m)$, l^2 - $mg(m)$) G' that is weakly-equivalent to G and satisfies the following condition:

(Nonerasing) For each rule $A(x_1, \dots, x_{a(A)}) \rightarrow t_A$ of G' , each variable x_i ($1 \leq i \leq a(A)$) appears at least once in t_A .

(Proof is given in appendix.) \square

Lemma 8. Let G be a nonerasing vl- $mg(m)$ (rsp. l^2 - $mg(m)$). We can construct a nonerasing vl- $mg(m)$ (rsp. l^2 - $mg(m)$) that is weakly-equivalent to G and satisfies the following condition:

(Nonpermuting) For each rule $A(x_1, \dots, x_{a(A)}) \rightarrow t_A$ of G' , $x_1, \dots, x_{a(A)}$ appear in this order from left to right in t_A .

Proof. For example, assume that G has a rule $A(x_1, x_2, x_3) \rightarrow B(x_2, C(x_3, x_1))$, violating the condition. We eliminate this rule and add $A^\pi(x_1, x_2, x_3) \rightarrow B(x_1, C(x_2, x_3))$ instead where π is the permutation defined by $\pi(1) = 2, \pi(2) = 3$ and $\pi(3) = 1$. Also, for each rule containing A in its right-hand side, say, $D(x_1, x_2, x_3) \rightarrow E(A(t_1, t_2, t_3))$, eliminate the rule and add $D(x_1, x_2, x_3) \rightarrow E(A^\pi(t_{\pi(1)}, t_{\pi(2)}, t_{\pi(3)}))$. This may require further elimination and addition of rules. In general, we systematically eliminate and add rules as follows.

Let $G = (N, T, P, S)$ be a nonerasing vl- $mg(m)$. For each rule $A(x_1, \dots, x_{a(A)}) \rightarrow t_A \in P$, eliminate this rule and add rules as follows. For each subterm $t = B(t_1, \dots, t_{a(B)})$ of t_A , choose an arbitrary permutation π_t on $\{1, 2, \dots, a(B)\}$. Let t'_A be the s-term obtained from t_A by replacing each t with $B^{\pi_t}(t_{\pi_t(1)}, \dots, t_{\pi_t(a(B))})$ in the topdown way. Assume that the left to right listing of variables in t'_A is $x_{\pi(1)}, \dots, x_{\pi(a(A))}$. Then add the rule $A^\pi(x_1, \dots, x_{a(A)}) \rightarrow t''_A$ where $t''_A = t'_A[x_{\pi^{-1}(1)}, \dots, x_{\pi^{-1}(a(A))}]$. That is, t''_A is obtained from t'_A by replacing $x_{\pi(1)}, \dots, x_{\pi(a(A))}$ with $x_1, \dots, x_{a(A)}$. Add the above rules for every combination of permutations π_t for subterms t of t_A . Delete every rule of which right-hand side contains a nonterminal that does not appear as the left-hand side of any of the rules constructed above, and let P' be the set of the remaining rules. Also let $N' = \{A^\pi \mid \exists A(x_1, \dots, x_{a(A)}) \rightarrow t''_A \in P'\}$.

Let $G' = (N', T, P', S)$ be the resulting mg . Obviously, G' is a vl- $mg(m)$ if G is a vl- $mg(m)$, and G' is an l^2 - $mg(m)$ if so is G . $L(G) = L(G')$ can be shown by proving that $A(x_1, \dots, x_{a(A)}) \xrightarrow{*}_G t \in (T \cup X)^*$ if and only if $A^\pi(x_{\pi(1)}, \dots, x_{\pi(a(A))}) \xrightarrow{*}_{G'} t$ where $A \in N, \pi$ is a permutation such that $A^\pi \in N'$. The latter can be shown by the double induction; the induction on the length of the derivations and the structural induction on subterms in the right-hand side of the applied rule for one step derivation. \square

Lemma 9. Let G be an $mg(m)$. We can construct an $mg(m)$ G' such that $L(G') = L(G)$ and every rule of G' has one of the following shapes:

$$\begin{aligned} A &\rightarrow a \quad (a \in T), \\ A(x) &\rightarrow x \quad (x \in X), \\ A(x_1, \dots, x_{a(A)}) &\rightarrow B(y_1, \dots, y_{a(B)}), \\ A(x_1, \dots, x_{a(A)}) &\rightarrow B(y_1, \dots, y_{a(B)}) \cdot C(z_1, \dots, z_{a(C)}), \\ A(x_1, \dots, x_{a(A)}) &\rightarrow B(y_1, \dots, y_{i-1}, C(z_1, \dots, z_{a(C)}), \\ &\quad y_i, \dots, y_{a(B)-1}), \\ \{y_1, \dots, y_{a(B)}, z_1, \dots, z_{a(C)}\} &\subseteq \{x_1, \dots, x_{a(A)}\}. \end{aligned}$$

This construction preserves nonerasing, nonpermuting and v-linear properties but does not always preserve n-linearity. G' is called a *normal form* mg if G' is nonerasing and nonpermuting. (Proof is given in appendix.) \square

4. Variable-Linear Macro Grammars

In this section, we first show that $VL\text{-}ML(m) \subseteq 2\text{-}MCFL(m+1)$ for every $m \geq 0$ (Lemma 10). The idea is as follows. Let G be a vl- $mg(m)$ that satisfies the nonerasing and nonpermuting conditions. For a derivation

$$A(x_1, \dots, x_{a(A)}) \xrightarrow{*}_G \alpha_0 x_1 \alpha_1 \cdots x_{a(A)} \alpha_{a(A)}, \quad (1)$$

variables $x_1, \dots, x_{a(A)}$ are regarded as gaps in the derived string to be filled in. We would like to construct a 2-mcflg($m+1$) G' weakly equivalent to G . To do so, we introduce a nonterminal A with $\dim(A) = a(A) + 1$ and construct rules of G' so that

$$A \xrightarrow{*}_{G'} (\alpha_0, \dots, \alpha_{a(A)}). \quad (2)$$

That is, variables $x_1, \dots, x_{a(A)}$ in (1) correspond to the gaps between the components of the tuple derived in (2). In this correspondence, v-linearity and the nonpermuting condition are essential.

Next we prove a pumping lemma for VL-ML(m) (Lemma 12). By using the lemma, it is shown in Lemma 13 that the inclusion $VL\text{-}ML(m) \subseteq 2\text{-}MCFL(m+1)$ is proper for every $m \geq 1$ (see the right half of Fig. 2). Note that if

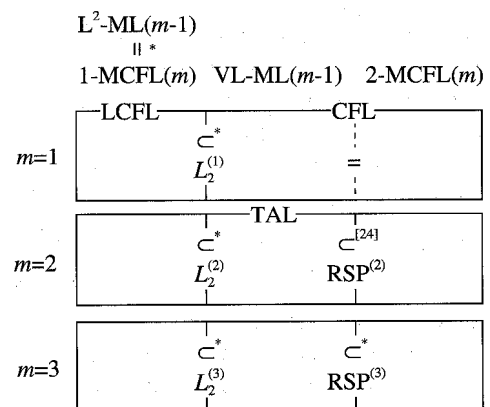


Fig. 2 Inclusion relations between classes of languages (*: new results).

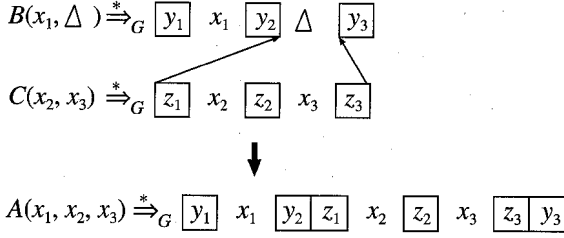


Fig. 3 A derivation in mg in case (5) of the proof of Lemma 10.

$m = 0$, VL-ML(0) = 2-MCFL(1) = CFL.

Finally, we discuss closure properties of VL-ML(m).

Lemma 10. Let $m \geq 0$. For a given vl-mg(m) G , we can construct a 2-mcfg($m + 1$) G' such that $L(G') = L(G)$.

Proof. Let G_0 be a given vl-mg(m). We can assume that G_0 is nonerasing by Lemma 8 and let $G = (N, T, P, S)$ be the nonerasing vl-mg(m) obtained from G_0 by Lemma 9. We construct a 2-mcfg($m + 1$) $G' = (N, T, F, P', S)$ where $\dim(A) = a(A) + 1$ ($A \in N$). F and P' are defined as follows:

- (1) If $A \rightarrow a \in P$, add $A \rightarrow a$ to P' .
- (2) If $A(x) \rightarrow x \in P$, add $A \rightarrow (\varepsilon, \varepsilon)$ to P' .
- (3) If $A(x_1, \dots, x_{a(A)}) \rightarrow B(x_1, \dots, x_{a(B)}) \in P$, add $A \rightarrow id[B]$ to P' and id to F where id is an identity function $id[(x_1, \dots, x_{\dim(A)})] = (x_1, \dots, x_{\dim(A)})$. Note that since G is nonerasing, $a(A) = a(B)$.
- (4) If $A(x_1, \dots, x_{a(A)}) \rightarrow B(x_1, \dots, x_{a(B)}) \cdot C(x_{a(B)+1}, \dots, x_{a(B)+a(C)}) \in P$ where $a(A) = a(B) + a(C)$, add $A \rightarrow f[B, C]$ to P' and f to F where $f[(y_1, \dots, y_{\dim(B)}), (z_1, \dots, z_{\dim(C)})] = (y_1, \dots, y_{\dim(B)}z_1, \dots, z_{\dim(C)})$.
- (5) If $A(x_1, \dots, x_{a(A)}) \rightarrow B(x_1, \dots, x_i, C(x_{i+1}, \dots, x_{i+a(C)}), x_{i+a(C)+1}, \dots, x_{a(B)+a(C)-1}) \in P$ where $a(A) = a(B) + a(C) - 1$, add $A \rightarrow g[B, C]$ to P' and g to F where $g[(y_1, \dots, y_{\dim(B)}), (z_1, \dots, z_{\dim(C)})] = (y_1, \dots, y_{i+1}z_1, z_2, \dots, z_{\dim(C)-1}, z_{\dim(C)}y_{i+2}, y_{i+3}, \dots, y_{\dim(B)})$. Exception: If $a(C) = 0$, $g[(y_1, \dots, y_{\dim(B)}), z_1] = (y_1, \dots, y_{i+1}z_1y_{i+2}, \dots, y_{\dim(B)})$.

As an example of (5), suppose $A(x_1, x_2, x_3) \rightarrow B(x_1, C(x_2, x_3)) \in P$. We construct an mcfg rule $A \rightarrow g[B, C]$ and a function g where $g[(y_1, y_2, y_3), (z_1, z_2, z_3)] = (y_1, y_2z_1, z_2, z_3y_3)$. Correspondence between the mg rule and function g is depicted in Fig. 3.

We can show that $A(x_1, \dots, x_{a(A)}) \xrightarrow{*}_G \alpha_0 x_1 \dots x_{a(A)}$ if and only if $A \xrightarrow{*}_{G'} (\alpha_0, \dots, \alpha_{a(A)})$ for every $A \in N$ and $\alpha_i \in T^*$ ($0 \leq i \leq a(A)$). \square

As noted in Sect. 2.1, the degree of an r -mcfg(m) is not greater than $(r + 1)m$. Thus, we obtain the following corollary from Proposition 2 and Lemma 10.

Corollary 11. Let G be a vl-mg(m). For a given $w \in T^*$, whether $w \in L(G)$ or not can be decided in $O(n^{3(m+1)})$ time where $n = |w|$. \square

To establish the proper inclusion of VL-ML(m) in 2-MCFL($m + 1$), we use the following language:

$$\text{RSP}^{(m)} = \{a_1^i a_2^j b_1^j b_2^j \dots a_{2m-1}^i a_{2m}^j b_{2m-1}^j b_{2m}^j \mid i, j \geq 0\}.$$

It is shown in [24] that $\text{RSP}^{(2)} \in 2\text{-MCFL}(2) \setminus \text{TAL}$. Since $\text{TAL} = \text{VL-ML}(1)$, $\text{RSP}^{(2)} \in 2\text{-MCFL}(2) \setminus \text{VL-ML}(1)$. Here we show that $\text{RSP}^{(m)} \in 2\text{-MCFL}(m) \setminus \text{VL-ML}(m - 1)$ for every $m \geq 2$. First, we prove a pumping lemma for VL-ML($m - 1$).

Lemma 12 (Pumping lemma for VL-ML($m - 1$)). Let L be a vl-ml($m - 1$) ($m \geq 2$). Assume that, for a given $n \geq 0$ there exists α in L such that $|\alpha|_a \geq n$ for every $a \in T$. Then, there exists a constant $M \geq 0$ depending only on L , such that for any $n \geq 0$ there exists z in L satisfying the following conditions (1) and (2):

- (1) For each $a \in T$, $|z|_a \geq n$ and
- (2) z can be written as $z = u_1 v_1 w_1 s_1 u_2 v_2 w_2 s_2 u_3 \dots u_m v_m w_m s_m u_{m+1}$ where $\sum_{j=1}^m |v_j s_j| \geq 1$ and $\sum_{j=2}^m |u_j| \leq M$, and for any $i \geq 0$, $z_i = u_1 v_1^i w_1 s_1^i u_2 v_2^i w_2 s_2^i u_3 \dots u_m v_m^i w_m s_m^i u_{m+1} \in L$.

Proof sketch. (See appendix for detailed proof.) Let G_0 be an mg($m - 1$) and let $G = (N, T, F, P, S)$ be a 2-mcfg(m) constructed from G_0 in the proof of Lemma 10 such that $L(G) = L(G_0)$. Without loss of generality, we assume that G satisfies conditions (N1) through (N3) in Lemma 1. Let n be a nonnegative integer. By the assumption, there exists α in L satisfying that (1) $|\alpha|_a \geq n$ for each $a \in T$ and (2) $|\alpha| \geq 2^{|N|+1}$. Let t_0 be a derivation tree of α . There exists a path p from the root r to a leaf in t_0 such that the number of the nodes on p which has two children is at least $\log_2 |\alpha| = |N| + 1$ by the assumption $|\alpha| \geq 2^{|N|+1}$. Therefore, there exist distinct nodes v and v' on p with a same label (say, $A \in N$) which have two children. The proof is similar to that of Lemma 4.14 of [24]. If we repeat the path between v' and v sufficiently large number of times, we obtain a derivation tree t of a string z that can be divided as claimed in the lemma. The crucial point is that when we determine such a path p , for each node v on the path, we can select a child of v so that $\sum_{j=2}^m |u_j|$ is not greater than some constant depending only on L . \square

Assumption " $\forall n \geq 0, \exists \alpha \in L, \forall a \in T : |\alpha|_a \geq n$ " and claim (1) of Lemma 12 are needed for the proof of Lemma 13. Lemma 19 contains the same assumption and claim for the proof of Theorem 20.

Lemma 13. For $m \geq 2$, $\text{RSP}^{(m)} \in 2\text{-MCFL}(m) \setminus \text{VL-ML}(m - 1)$.

Proof. ($\text{RSP}^{(m)} \in 2\text{-MCFL}(m)$) The following 2-mcfg(m) G generates $\text{RSP}^{(m)}$:

- $S \rightarrow g[A, B]$ where

$$g[(x_1, \dots, x_m), (y_1, \dots, y_m)] = x_1 y_1 \dots x_m y_m.$$

- $A \rightarrow f_A[A] \mid (\varepsilon, \dots, \varepsilon)$ where

$$f_A[(x_1, \dots, x_m)] = (a_1 x_1 a_2, \dots, a_{2m-1} x_m a_{2m}).$$

- $B \rightarrow f_B[B] \mid (\varepsilon, \dots, \varepsilon)$ where

$$f_B[(x_1, \dots, x_m)] = (b_1 x_1 b_2, \dots, b_{2m-1} x_m b_{2m}).$$

(RSP^(m) \notin VL-ML($m - 1$)) Assume that RSP^(m) \in VL-ML($m - 1$). Then, RSP^(m) satisfies the condition of Lemma 12. For the constant M in the lemma, let z be

$$z = a_1^q a_2^q b_1^r b_2^r \cdots a_{2m-1}^q a_{2m}^q b_{2m-1}^r b_{2m}^r \\ (q, r > M/(2m - 2)).$$

Divide z as

$$z = u_1 v_1 w_1 s_1 u_2 v_2 w_2 s_2 u_3 \cdots u_m v_m w_m s_m u_{m+1}.$$

The condition $\sum_{j=1}^m |v_j s_j| \geq 1$ and $u_1 v_1^i w_1 s_1^i u_2 v_2^i w_2 s_2^i u_3 \cdots u_m v_m^i w_m s_m^i u_{m+1} \in \text{RSP}^{(m)}$ for all $i \geq 0$ holds only if

- (1) $v_1 = a_1^j, s_1 = a_2^j, \dots, v_m = a_{2m-1}^j, s_m = a_{2m}^j$
($1 \leq j \leq q$), or
- (2) $v_1 = b_1^j, s_1 = b_2^j, \dots, v_m = b_{2m-1}^j, s_m = b_{2m}^j$
($1 \leq j \leq r$).

However, neither (a) nor (b) satisfies $\sum_{j=2}^m |u_j| \leq M$. \square

Theorem 14. For each $m \geq 0$, VL-ML(m) \subseteq 2-MCFL($m + 1$). When $m \geq 1$, the inclusion is proper. When $m = 0$, VL-ML(0) = 2-MCFL(1) = CFL. \square

Next, we present closure properties of VL-ML(m).

Theorem 15. For $m \geq 0$, VL-ML(m) is a substitution closed full AFL. VL-ML(m) is not closed under intersection.

Proof. For the first claim, it suffices to show that VL-ML(m) contains all regular languages and is closed under intersection with regular languages and substitution by Theorem 3.3 of [16]. First, VL-ML(0) = CFL and hence every VL-ML(m) contains all regular languages.

(intersection with regular languages) Let $G = (N, T, P, S)$ be a normal form vl-mg(m) and $M = (Q, T, \delta, q_I, Q_F)$ be a deterministic finite automaton where $Q, T, \delta : Q \times T \rightarrow Q$, $q_I \in Q$ and $Q_F \subseteq Q$ are a finite set of states, a finite set of input symbols, a state transition function, an initial state and a set of final states, respectively. Let $L(M)$ denote the language accepted by M . We construct a vl-mg(m) $G' = (N', T, P', S')$ that generates $L(G) \cap L(M)$ as follows:

- (1) $N' = \{A[q_{00}, q_{01}; \dots; q_{a(A)0}, q_{a(A)1}] \mid A \in N \text{ and } q_{ij} \in Q (0 \leq i \leq d(A), j = 0, 1)\}$.
- (2) P' consists of the following rules:
 - (a) $S_0 \rightarrow S[q_I, q_F]$ for $q_F \in Q_F$.
 - (b) $A[q_{00}, q_{01}] \rightarrow a$ if $A \rightarrow a$ and $\delta(q_{00}, a) = q_{01}$.
 - (c) $A[q_{00}, q_{01}; q_{10}, q_{11}] \rightarrow a$ if $A(x) \rightarrow x$ and $q_{ij} \in Q (0 \leq i, j \leq 1)$.

The other rules are bothersome, hence we present them by examples. See Fig. 4.

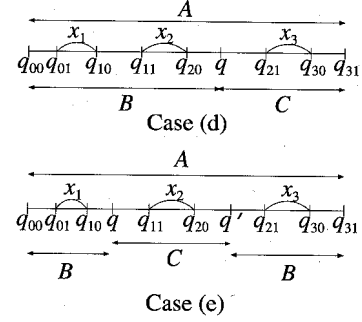


Fig. 4 Construction of vl-mg rules in the proof of Theorem 15.

- (d) $A[q_{00}, q_{01}; q_{10}, q_{11}; q_{20}, q_{21}; q_{30}, q_{31}](x_1, x_2, x_3) \rightarrow B[q_{00}, q_{01}; q_{10}, q_{11}; q_{20}, q](x_1, x_2) \cdot C[q, q_{21}; q_{30}, q_{31}](x_3)$ if $A(x_1, x_2, x_3) \rightarrow B(x_1, x_2) \cdot C(x_3) \in P$ and $q_{ij}, q \in Q (0 \leq i \leq 3, j = 0, 1)$.
- (e) $A[q_{00}, q_{01}; q_{10}, q_{11}; q_{20}, q_{21}; q_{30}, q_{31}](x_1, x_2, x_3) \rightarrow B[q_{00}, q_{01}; q_{10}, q, q', q_{21}; q_{30}, q_{31}](x_1, C[q, q_{11}; q_{20}, q'](x_2), x_3)$ if $A(x_1, x_2, x_3) \rightarrow B(x_1, C(x_2), x_3) \in P$ and $q_{ij}, q, q' \in Q (0 \leq i \leq 3, j = 0, 1)$.

We can show by induction on the length of the derivations and transitions that $A[q_{00}, q_{01}; \dots; q_{a(A)0}, q_{a(A)1}](x_1, \dots, x_{a(A)}) \xrightarrow{*G} \alpha_0 x_1 \cdots x_{a(A)} \alpha_{a(A)}$ if and only if $A(x_1, \dots, x_{a(A)}) \xrightarrow{*G} \alpha_0 x_1 \cdots x_{a(A)} \alpha_{a(A)}$ and $\delta(q_{i0}, \alpha_i) = q_{i1} (0 \leq i \leq a(A))$.

(substitution) Let $G = (N, T, P, S)$, $G_a = (N_a, T_a, P_a, S_a)$ ($a \in T$) be vl-mg(m)s where any two of N and N_a ($a \in T$) share no nonterminal. Let $G' = (N \cup \bigcup_{a \in T} N_a, \bigcup_{a \in T} T_a, P' \cup \bigcup_{a \in T} P_a, S)$ where $P' = \{A(x_1, \dots, x_{a(A)}) \rightarrow t'_A \mid A(x_1, \dots, x_{a(A)}) \rightarrow t_A \in P \text{ and } t'_A \text{ is obtained from } t_A \text{ by replacing } a \in T \text{ with } S_a\}$. It is easy to see that G' is a vl-mg(m) such that $L(G') = s(L(G))$ where s is the substitution defined by $s(a) = L(G_a)$ for $a \in T$.

(intersection) Let $L = \{a_1^{n_1} a_2^{n_2} \cdots a_{2m+3}^{n_{2m+3}} \mid n_1, n_2 \geq 0\}$ and $L' = \{a_1^{n_1} \cdots a_{2m+2}^{n_{2m+2}} a_{2m+3}^{n_{2m+3}} \mid n_1, n_2 \geq 0\}$. We can easily give vl-mg(m)s G and G' such that $L(G) = L$ and $L(G') = L'$ by observing $L = \{a_1^n \mid n \geq 0\} \cdot h(L_1^{(m+1)})$, $L' = L_1^{(m+1)} \cdot \{a_{2m+3}^n \mid n \geq 0\}$ and $L_1^{(m+1)} \in \text{VL-ML}(m)$ where h is the homomorphism such that $h(a_i) = a_{i+1} (1 \leq i \leq 2m + 2)$. On the other hand, $L \cap L' = \{a_1^n \cdots a_{2m+3}^n \mid n \geq 0\} \notin \text{VL-ML}(m) \subseteq 2\text{-MCFL}(m + 1)$ [10], [24]. \square

5. Double-Linear Macro Grammars

In this section, we show that $L^2\text{-ML}(m) = 1\text{-MCFL}(m + 1) \subseteq \text{VL-ML}(m)$ for every $m \geq 0$ (see the left half of Fig. 2). First, we prove $L^2\text{-ML}(m) = 1\text{-MCFL}(m + 1)$ in Theorem 17. Unfortunately, the construction in the proof of Lemma 9 does not preserve n-linearity since new nonterminals may be introduced in the right-hand side of a rule during the construction. Hence, we directly translate a given $L^2\text{-mg}(m)$ into a weakly equivalent $1\text{-mcf}(m + 1)$. For the other direction, we will introduce a nonpermuting condition for 1-mcf (Lemma 16). Next, we provide a pumping lemma for $L^2\text{-ML}(m)$ (Lemma 19), which implies the proper inclusion

$L^2\text{-ML}(m) \subset \text{VL-ML}(m)$ (Theorem 20).

Lemma 16. Let $G = (N, T, F, P, S)$ be a nonerasing 1-mcfg(m). We can construct a 1-mcfg(m) that is weakly equivalent to G and satisfies the following condition:

(Nonpermuting) Let $f : (T^*)^{d_1} \rightarrow (T^*)^{d_0}$ be an arbitrary function in F defined by (see (F) in Sect. 2.1):

$$f[(x_1, \dots, x_{d_1})] = (\alpha_1, \alpha_2, \dots, \alpha_{d_0}).$$

Variables x_1, \dots, x_{d_1} appear in this order from left to right in $\alpha_1 \alpha_2 \dots \alpha_{d_0}$.

Proof. Similar to the proof of Lemma 8. \square

Theorem 17. For each $m \geq 0$, $L^2\text{-ML}(m) = 1\text{-MCFL}(m+1)$.

Proof. ($L^2\text{-ML}(m) \subseteq 1\text{-MCFL}(m+1)$) Let $G = (N, T, P, S)$ be an arbitrary $l^2\text{-mg}(m)$. Without loss of generality, we assume that G is nonerasing and nonpermuting by Lemmas 7 and 8. From G , we construct a 1-mcfg($m+1$) $G' = (N, T, F, P', S)$ as follows. Let $A(x_1, \dots, x_{a(A)}) \rightarrow t_A$ be an arbitrary rule of G . Since G is n -linear, t_A can be written as $t_A = \alpha B(\beta_1, \dots, \beta_{a(B)}) \gamma$ where $\alpha, \beta_i (1 \leq i \leq a(B)), \gamma \in (T \cup X)^*$. Let $\Omega_1, \dots, \Omega_{a(B)+1}$ be new symbols. Since G is nonerasing and nonpermuting, there exist strings $\delta_j \in (T \cup \{\Omega_1, \dots, \Omega_{a(B)+1}\})^*$ for $0 \leq j \leq a(A)$ such that

$$\alpha \Omega_1 \beta_1 \Omega_2 \dots \beta_{a(B)} \Omega_{a(B)+1} \gamma = \delta_0 x_1 \delta_1 x_2 \dots x_{a(A)} \delta_{a(A)}.$$

We add the following function and rule to F and P' , respectively:

$$A \rightarrow f[B]$$

$$f[(x_1, \dots, x_{a(B)+1})] = (\delta'_0, \dots, \delta'_{a(A)})$$

where for $0 \leq j \leq a(A)$, δ'_j is obtained from δ_j by replacing Ω_i with x_i ($1 \leq i \leq a(B)+1$). For example, if there exists a rule $A(x_1, x_2, x_3, x_4) \rightarrow aB(bx_1c, x_2x_3d)x_4e$ in P , add the following f to F and add $A \rightarrow f[B]$ to P'

$$f[(x_1, x_2, x_3)] = (ax_1b, cx_2, \varepsilon, dx_3, e).$$

By induction on the length of the derivations, we can show that $A(x_1, \dots, x_{a(A)}) \xrightarrow{*}_G \alpha_0 x_1 \dots x_{a(A)} \alpha_{a(A)}$ if and only if $A \xrightarrow{*}_{G'} (\alpha_0, \dots, \alpha_{a(A)})$ where $\alpha_i \in T^*$ ($0 \leq i \leq a(A)$). ($1\text{-MCFL}(m+1) \subseteq L^2\text{-ML}(m)$) Let $G = (N, T, F, P, S)$ be a nonerasing and nonpermuting 1-mcfg($m+1$) due to Lemmas 1 and 16. We construct an $l^2\text{-mg}(m)$ $G' = (N, T, P', S)$ in a similar way to the above proof. Let $A \rightarrow f[B] \in P$ be an arbitrary rule where

$$f[(x_1, \dots, x_{\dim(B)})] = (\alpha_1, \dots, \alpha_{\dim(B)}).$$

Let $\Delta_1, \dots, \Delta_{\dim(A)-1}$ be new symbols. Since G is nonerasing and nonpermuting, there exist $\xi_j \in (T \cup \{\Delta_1, \dots, \Delta_{\dim(A)-1}\})^*$ for $0 \leq j \leq \dim(B)$ such that

$$\alpha_1 \Delta_1 \alpha_2 \dots \Delta_{\dim(A)-1} \alpha_{\dim(A)}$$

$$= \xi_0 x_1 \xi_1 \dots x_{\dim(B)} \xi_{\dim(B)}.$$

Add the following rule to P' :

$$A(x_1, \dots, x_{\dim(A)-1})$$

$$= \xi'_0 B(\xi'_1, \dots, \xi'_{\dim(B)-1}) \xi'_{\dim(B)}$$

where for $0 \leq j \leq \dim(B)$, ξ'_j is obtained from ξ_j by replacing Δ_i with x_i ($1 \leq i \leq \dim(A)-1$). The formal proof is similar to that of $L^2\text{-ML}(m) \subseteq 1\text{-MCFL}(m+1)$. \square

Similar to the case of $\text{vl-mg}(m)$, we obtain the following corollary from Proposition 2 and Theorem 17.

Corollary 18. Let G be an $l^2\text{-mg}(m)$. For a given $w \in T^*$, whether $w \in L(G)$ or not can be decided in $O(n^{2(m+1)})$ time where $n = |w|$. \square

Next, we present a pumping lemma for $1\text{-MCFL}(m) = L^2\text{-ML}(m-1)$.

Lemma 19 (Pumping lemma for $1\text{-MCFL}(m)$). Let L be a 1-mcfg(m). Assume that, for a given $n \geq 0$ there exists α in L such that $|\alpha|_a \geq n$ for every $a \in T$. Then, there exists a constant $M \geq 0$ depending only on L , such that for any $n \geq 0$ there exists z in L satisfying the following conditions (1) and (2):

- (1) For each $a \in T$, $|z|_a \geq n$ and
- (2) z can be written as $z = u_1 v_1 w_1 s_1 u_2 v_2 w_2 s_2 u_3 \dots u_m v_m w_m s_m u_{m+1}$ where $\sum_{j=1}^m |v_j s_j| \geq 1$ and $\sum_{j=1}^m |u_j v_j s_j| + |u_{m+1}| \leq M$, and for any $i \geq 0$,

$$z_i = u_1 v_1^i w_1 s_1^i u_2 v_2^i w_2 s_2^i u_3 \dots u_m v_m^i w_m s_m^i u_{m+1} \in L.$$

(Proof is given in appendix.) Choose a path for pumping as close as possible to the root as is in a proof of the pumping lemma for LCFL. \square

Theorem 20. For $m \geq 1$, $L_2^{(m)} \in \text{VL-ML}(m-1) \setminus L^2\text{-ML}(m-1)$.

Proof. Remember that G_5 in Example 2(2) is a $\text{vl-mg}(m-1)$ such that $L(G_5) = L_2^{(m)}$. Next, we show that $L_2^{(m)} \notin L^2\text{-ML}(m-1)$. Suppose $L_2^{(m)} \in L^2\text{-ML}(m-1) = 1\text{-MCFL}(m)$. Then $L_2^{(m)}$ satisfies the condition of Lemma 19. Let M be the constant of the lemma and let z be $z = a_1^q \dots a_{2m}^q b_1^r \dots b_{2m}^r$ ($q, r > M/2m$). Divide z as $z = u_1 v_1 w_1 s_1 u_2 v_2 w_2 s_2 u_3 \dots u_m v_m w_m s_m u_{m+1}$. The condition $\sum_{j=1}^m |v_j s_j| \geq 1$ and $z_i = u_1 v_1^i w_1 s_1^i u_2 v_2^i w_2 s_2^i u_3 \dots u_m v_m^i w_m s_m^i u_{m+1} \in L$ for all $i \geq 0$ holds only if

- (1) $v_1 = a_1^j, s_1 = a_1^j, \dots, v_m = a_{2m-1}^j, s_m = a_{2m}^j$ ($1 \leq j \leq q$), or
- (2) $v_1 = b_1^j, s_1 = b_1^j, \dots, v_m = b_{2m-1}^j, s_m = b_{2m}^j$ ($1 \leq j \leq r$).

However, neither (a) nor (b) satisfies $\sum_{j=1}^m |u_j v_j s_j| + |u_{m+1}| \leq M$. \square

6. Conclusion

In this paper, we compared the generative power of multiple context-free grammars with that of variable-linear macro grammars. Equivalent transformations between mcfg and vl-mg were established by taking the correspondence between arguments of nonterminals in vl-mg and *gaps* between components of tuples derived in mcfg. In Theorem 6.3 of [5], a different transformation was used so that we can take the correspondence between arguments of nonterminals in vl-mg and components of tuples derived in mcfg. For example, an mg $S \rightarrow A(ab, cd), A(x_1, x_2) \rightarrow A(ax_1b, cx_2d) \mid x_1x_2$ is transformed into an mcfg $S \rightarrow c[A], A \rightarrow g[A] \mid (ab, cd), c[(x_1, x_2)] = x_1x_2, g[(x_1, x_2)] = (ax_1b, cx_2d)$ and vice versa. A derivation in an mg G corresponds to the reverse of a derivation in mcfg G' constructed from G by this transformation. From this reason, the transformation in [5] can work only for nl-mg or 1-mcfg.

Acknowledgments

The authors would like to heartily express their thanks to late Dr. Tadao Kasami for his guiding them to formal language theory and its application to bioinformatics with his deep knowledge, keen insight and warm heart. They also thank the reviewers for their carefully reading the paper and giving useful comments.

References

- [1] N. Abe and H. Mamitsuka, "Predicting protein secondary structure using stochastic tree grammars," *Mach. Learn.*, vol.29, pp.275–301, 1997.
- [2] A.V. Aho, "Indexed grammars – An extension of the context-free grammars," *J. ACM*, vol.15, pp.647–671, 1968.
- [3] A. Condon, "Problems on RNA secondary structure prediction and design," *ICALP2003, LNCS 2719*, pp.22–32, 2003.
- [4] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis*, Cambridge University Press, 1998.
- [5] J. Engelfriet and G. Rozenberg, "Tree transducers, L systems, and two-way machines," *J. Comput. Syst. Sci.*, vol.20, pp.150–202, 1980.
- [6] M.J. Fischer, *Grammars with Macro-like Productions*, Ph.D. Thesis, Harvard University, 1968, also in 9th IEEE Sympo. on Switching and Automata Theory, 1968.
- [7] A. Fujiyoshi and T. Kasai, "Spinal-formed context-free tree grammars," *Theor. Comput. Syst.*, vol.33, no.1, pp.59–83, 2000.
- [8] A.K. Joshi, L. Levy, and M. Takahashi, "Tree adjunct grammars," *J. Comput. Syst. Sci.*, vol.10, no.1, pp.136–163, 1975.
- [9] A.K. Joshi and Y. Schabes, "Tree adjoining grammars," in *Handbook of Formal Languages*, vol.3 (Beyond Words), eds. G. Rozenberg and A. Salomaa, pp.69–123, Springer, 1997.
- [10] T. Kasami, H. Seki, and M. Fujii, "Generalized context-free grammar and multiple context-free grammar," *IEICE Trans. Inf. & Syst. (Japanese Edition)*, vol.J71-D, no.5, pp.758–765, May 1988.
- [11] T. Kasami, H. Seki, and M. Fujii, "On the membership problem for head languages and multiple context-free languages," *IEICE Trans. Inf. & Syst. (Japanese Edition)*, vol.J71-D, no.6, pp.935–941, June 1988.
- [12] Y. Kato, H. Seki, and T. Kasami, "On the generative power of grammars for RNA secondary structure," *IEICE Trans. Inf. & Syst.*, vol.E88-D, no.1, pp.53–64, Jan. 2005.
- [13] Y. Kato, H. Seki, and T. Kasami, "RNA pseudoknotted structure prediction using stochastic multiple context-free grammar," *IPSP Trans. Bioinformatics*, vol.47, SIG17(TBIO1), pp.12–21, 2006.
- [14] S. Kepser and U. Mönnich, "Closure properties of linear context-free tree languages with an application to optimality theory," *Theor. Comput. Sci.*, vol.354, pp.82–97, 2006.
- [15] M. Latteux, "Substitutions dans les EDTOL-systèmes ultralinéaires," *Information & Control*, vol.42, pp.194–260, 1979.
- [16] A. Mateescu and A. Salomaa, "Aspects of classical language theory," in *Handbook of Formal Languages*, vol.1 (Word, Language, Grammar), eds. G. Rozenberg and A. Salomaa, pp.175–251, Springer, 1997.
- [17] H. Matsui, K. Sato, and Y. Sakakibara, "Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures," *Bioinformatics*, vol.21, no.11, pp.2611–2617, 2005.
- [18] O. Rambow and G. Satta, "A two-dimensional hierarchy for parallel rewriting systems," *IRCS Report 94-02*, Institute for Research in Cognitive Science, University of Pennsylvania, 1994.
- [19] O. Rambow and G. Satta, "Independent parallelism in finite copying parallel rewriting systems," *Theor. Comput. Sci.*, vol.223, pp.87–120, 1999.
- [20] E. Rivas and S. Eddy, "The language of RNA: A formal grammar that includes pseudoknots," *Bioinformatics*, vol.16, no.4, pp.334–340, 2000.
- [21] W.C. Rounds, "Mappings and grammars on trees," *Mathematical Systems Theory*, vol.4, no.3, pp.257–287, 1970.
- [22] W.C. Rounds, "Tree-oriented proofs of some theorems on context-free and indexed languages," *2nd Annual ACM Symposium on Theory of Computing (STOC)*, pp.109–116, 1970.
- [23] H. Seki and Y. Kato, "On the generative power of multiple context-free grammars and macro grammars," *Information Science Technical Report, NAIST-IS-TR2006007*, Nara Institute of Science and Technology, 2006.
- [24] H. Seki, T. Matsumura, M. Fujii, and T. Kasami, "On multiple context-free grammars," *Theor. Comput. Sci.*, vol.88, pp.191–229, 1991.
- [25] S. Seki and S. Kobayashi, "A grammatical approach to the alignment of structure-annotated strings," *IEICE Trans. Inf. & Syst.*, vol.E88-D, no.12, pp.2727–2737, Dec. 2005.
- [26] Y. Uemura, A. Hasegawa, S. Kobayashi, and T. Yokomori, "Tree adjoining grammars for RNA structure prediction," *Theor. Comput. Sci.*, vol.210, pp.277–303, 1999.
- [27] D.J. Weir, *Characterizing mildly context-sensitive grammar formalisms*, Ph.D. Thesis, Department of Computer and Information Science, University of Pennsylvania, 1988.
- [28] D.J. Weir, "Linear context-free rewriting systems and deterministic tree-walking transducers," *30th Meeting of the Assoc. for Computational Linguistics (ACL92)*, pp.136–143, 1992.

Appendix: Supplementary Proofs

A.1 Proof of Lemma 7

Let $A(x_1, \dots, x_{a(A)}) \rightarrow t_A$ be a rule that does not satisfy the condition. Let $\Psi \subseteq \{1, 2, \dots, a(A)\}$ be such that $i \in \Psi$ if and only if x_i does not appear in t_A . Let $i_1, \dots, i_{a(A)-|\Psi|}$ be the listing in the ascending order of variable suffixes of the set $\{1, 2, \dots, a(A)\} \setminus \Psi$. Add a new nonterminal A_Ψ with $a(A_\Psi) = a(A) - |\Psi|$ and add a rule

$$A_\Psi(x_1, \dots, x_{a(A)-|\Psi|}) \rightarrow t'_A$$

where t'_A is obtained from t_A by replacing $x_{i_1}, \dots, x_{i_{a(A)-|\Psi|}}$

with $x_1, \dots, x_{a(A)-|\Psi|}$. For each rule $B(x_1, \dots, x_{a(B)}) \rightarrow t_B$ where t_B contains A , add a rule

$$B(x_1, \dots, x_{a(B)}) \rightarrow t'_B$$

where t'_B is obtained from t_B by replacing each subterm $A(t_1, \dots, t_{a(A)})$ with $A_\Psi(t_{i_1}, \dots, t_{i_{a(A)-|\Psi|}})$. Remove the rule $A(x_1, \dots, x_{a(A)}) \rightarrow t_A$.

Repeat the above procedure until every rule satisfies the nonerasing condition. The procedure always halts since the cardinality of Ψ is bounded by $a(A)$. Also, we only add nonterminals A_Ψ with arity less than that of A . Hence, if G is an $mg(m)$ (rsp. vl- $mg(m)$, nl- $mg(m)$), so is G' .

A.2 Proof of Lemma 9

A.2.1 Preliminaries

Before proceeding with the proof, we need a technical definition on the shape of s-terms.

- S-terms t_1, \dots, t_l are *balanced* if either
 - $t_j \in X$ for $1 \leq j \leq l$, or
 - $t_j = B_{j1}(t_{j11}, \dots, t_{j1a(B_{j1})}) \cdots B_{js_j}(t_{js_j1}, \dots, t_{js_ja(B_{js_j})})$ where $B_{jk} \in N$ and $t_{jkl}, \dots, t_{jka(B_{jk})}$ are balanced for $1 \leq j \leq l$ and $1 \leq k \leq s_j$.
- An s-term t is in *weak normal form* if either
 - $t \in X \cup T$,
 - $t = t_1 \cdots t_l$ where $l \geq 2$ and t_1, \dots, t_l are balanced and not variables, or
 - $t = A(t_1, \dots, t_{a(A)})$ ($A \in N$) where $t_1, \dots, t_{a(A)}$ are balanced.

For example, $a \in T$ and $A(x, y)$ and $B(C(z, x), D(A(x)E(y)))$ are all in weak normal form. Neither xy nor $A(B(x), y)$ nor $A(x, a, y)$ is in weak normal form.

A.2.2 Transformation

For a given $mg G$, construct an $mg G'$ from G by the following procedure.

(Step 1) Each subterm t of the right-hand side of a rule in G is transformed into an s-term in weak normal form as follows: Each rule $A(x_1, \dots, x_{a(A)}) \rightarrow t_A$ with $t_A \notin X \cup T$ is replaced with the rule $A(x_1, \dots, x_{a(A)}) \rightarrow \text{Trans}(t_A)$ and the auxiliary rules generated by $\text{Trans}(t_A)$.

$\text{Trans}(t)$:

- $t = x \in X$: $\text{Trans}(x) = I(x)$ where I is a new nonterminal. Add $I(x) \rightarrow x$ as an auxiliary rule.
- $t = a \in T$: $\text{Trans}(a) = [a]$ where $[a]$ is a new nonterminal. Add $[a] \rightarrow a$ as an auxiliary rule.
- $t = t_1 \cdots t_l$ ($l \geq 2$): $\text{Trans}(t) = \text{Trans}(t_1) \cdots \text{Trans}(t_l)$.
- $t = A(x_{i_1}, \dots, x_{i_{a(A)}})$: $\text{Trans}(t) = t$.
- $t = A(t_1, \dots, t_{a(A)})$ with at least one t_i not a variable: $\text{Trans}(t) = A(\text{Trans}(t_1), \dots, \text{Trans}(t_{a(A)}))$.

(Step 2) Since the right-hand side of each rule is in weak normal form by (Step 1), if a rule has a shape that is not permitted by a normal form mg , then the rule can be written as $A(x_1, \dots, x_{a(A)}) \rightarrow B_1(t_{11}, \dots, t_{1a(B_1)}) \cdots B_s(t_{s1}, \dots, t_{sa(B_s)})$ ($s \geq 2$). This rule is replaced with the following rule:

$$A(x_1, \dots, x_{a(A)}) \rightarrow \alpha_1 \cdots \alpha_s \quad (\text{A.1})$$

where for $1 \leq i \leq s$,

- (a) $\alpha_i = B_i(t_{i1}, \dots, t_{ia(B_i)})$ if all t_{ij} ($1 \leq j \leq a(B_i)$) are variables,
- (b) $\alpha_i = B'_i(x_{i1}, \dots, x_{ili})$ otherwise, where $x_{i1}, \dots, x_{ili} \in \{x_1, \dots, x_{a(A)}\}$ are the variables appearing in $t_{i1}, \dots, t_{ia(B_i)}$, arranged without duplication in the order of their (first) occurrences from left to right, and B'_i is a new nonterminal with $a(B'_i) = l_i$.

In case (b) above, the following rules are also added. Since each t_{ij} ($1 \leq j \leq a(B_i)$) is in weak normal form by (Step 1), t_{ij} has the shape of $C_{j1}(s_{j11}, \dots, s_{j1a(C_{j1})}) \cdots C_{jn_j}(s_{jn_j1}, \dots, s_{jn_ja(C_{jn_j})})$ where s_{jkl} is an s-term for $1 \leq k \leq n_j$ and $1 \leq l \leq a(C_{jk})$. Let $y_{j1}, \dots, y_{jm_j} \in \{x_1, \dots, x_{a(A)}\}$ be the variables appearing in t_{ij} , arranged without duplication from left to right. Let θ be the substitution that renames x_{ij} by x_j ($1 \leq j \leq a(B'_i)$).

$$B'_i(x_1, \dots, x_{a(B'_i)}) \rightarrow B_i(C'_{j1}(y_{j11}, \dots, y_{j1m_{j1}}), \dots, C'_{jn_j}(y_{jn_j1}, \dots, y_{jn_jm_{jn_j}}))\theta, \quad (\text{A.2})$$

$$C'_j(x_1, \dots, x_{a(C'_j)}) \rightarrow C_{j1}(s_{j11}, \dots, s_{j1a(C_{j1})}) \cdots C_{jn_j}(s_{jn_j1}, \dots, s_{jn_ja(C_{jn_j})})\theta_j \quad (1 \leq j \leq a(B_i)) \quad (\text{A.3})$$

where C'_j is a new nonterminal with $a(C'_j) = m_j$ and θ_j is the substitution that renames y_{jk} by x_k ($1 \leq k \leq a(C'_j)$) for $1 \leq j \leq a(B'_i)$. Apply (Step 2) to the rule (A.3) repeatedly until all the rules have the shape of (A.1) or (A.2).

(Step 3) This step is similar to the construction of Chomsky normal form for cfg. Each rule $A(x_1, \dots, x_{a(A)}) \rightarrow \bar{B}_1(x_{11}, \dots, x_{1a(\bar{B}_1)}) \cdots \bar{B}_s(x_{s1}, \dots, x_{sa(\bar{B}_s)})$ ($s \geq 3$) obtained as (A.1) in (Step 2) is replaced with the following rules:

$$\begin{aligned} & A(x_1, \dots, x_{a(A)}) \\ & \rightarrow \bar{B}_1(x_{11}, \dots, x_{1a(\bar{B}_1)}) \cdot D_1(y_{11}, \dots, y_{1n_1}), \\ & \quad D_1(y_{11}, \dots, y_{1n_1}) \\ & \rightarrow \bar{B}_2(x_{21}, \dots, x_{2a(\bar{B}_2)}) \cdot D_2(y_{21}, \dots, y_{2n_2}), \\ & \quad \dots \\ & \quad D_{s-2}(y_{(s-2)1}, \dots, y_{(s-2)n_{s-2}}) \\ & \rightarrow \bar{B}_{s-1}(x_{(s-1)1}, \dots, x_{(s-1)a(\bar{B}_{s-1})}) \cdot \bar{B}_s(x_{s1}, \dots, x_{sa(\bar{B}_s)}) \end{aligned}$$

where y_{j1}, \dots, y_{jn_j} are the variables appearing in $\bar{B}_{j+1}(x_{(j+1)1}, \dots, x_{(j+1)a(\bar{B}_{j+1})})$, $\dots, \bar{B}_s(x_{s1}, \dots, x_{sa(\bar{B}_s)})$ arranged without

duplication from left to right, and D_j is a new nonterminal with $a(D_j) = n_j$ for $1 \leq j \leq s-2$.

(Step 4) Each rule $A(x_1, \dots, x_{a(A)}) \rightarrow B(C_1(x_{11}, \dots, x_{1a(C_1)}), \dots, C_{a(B)}(x_{a(B)1}, \dots, x_{a(B)a(C_{a(B)})}))$ obtained as (A·2) in (Step 2) is replaced with the following rules:

$$\begin{aligned} & A(x_1, \dots, x_{a(A)}) \\ \rightarrow & E_1(C_1(x_{11}, \dots, x_{1a(C_1)}), y_{11}, \dots, y_{1m_1}), \\ & E_1(x'_1, y_{11}, \dots, y_{1m_1}) \\ \rightarrow & E_2(x'_1, C_2(x_{21}, \dots, x_{2a(C_2)}), y_{21}, \dots, y_{2m_2}), \\ & \dots \\ & E_{a(B)-1}(x'_1, \dots, x'_{a(B)-1}, y_{(a(B)-1)1}, \dots, \\ & y_{(a(B)-1)m_{a(B)-1}}) \\ \rightarrow & B(x'_1, \dots, x'_{a(B)-1}, C_{a(B)}(x_{a(B)1}, \dots, x_{a(B)a(C_{a(B)})})) \end{aligned}$$

where y_{j1}, \dots, y_{jm_j} are the variables appearing in $C_{j+1}(x_{(j+1)1}, \dots, x_{(j+1)a(C_{j+1})}), \dots, C_{a(B)}(x_{a(B)1}, \dots, x_{a(B)a(C_{a(B)})})$ arranged without duplication from left to right, x'_j is a new variable not in $\{x_1, \dots, x_{a(A)}\}$, and E_j is a new nonterminal with $a(E_j) = j + m_j$ for $1 \leq j \leq a(B) - 1$.

In (Step 3) and (Step 4), rename variables so that the variables in the left-hand side of the constructed rules are $x_1, x_2, \dots, x_{a(H)}$ where H is the nonterminal in the left-hand side.

A.2.3 Example

We will transform the following rule to a normal form:

$$A(x_1, x_2, x_3) \rightarrow B(C(x_3, F(x_1))aH(x_3), C(x_1, d)) \cdot x_2.$$

(Step 1)

$$\begin{aligned} & \text{Trans}(B(C(x_3, F(x_1))aH(x_3), C(x_1, d)) \cdot x_2) \\ = & \text{Trans}(B(C(x_3, F(x_1))aH(x_3), C(x_1, d))) \cdot \text{Trans}(x_2) \\ = & B(\text{Trans}(C(x_3, F(x_1))aH(x_3)), \text{Trans}(C(x_1, d))) \cdot I(x_2) \\ = & B(C(I(x_3), F(x_1))[a]H(x_3), C(I(x_1), [d])) \cdot I(x_2). \end{aligned}$$

Thus, we obtain:

$$A(x_1, x_2, x_3) \rightarrow B(C(I(x_3), F(x_1))[a]H(x_3), C(I(x_1), [d])) \cdot I(x_2), \quad (\text{A} \cdot 4)$$

$$I(x) \rightarrow x, \quad (\text{A} \cdot 5)$$

$$[a] \rightarrow a, \quad (\text{A} \cdot 6)$$

$$[d] \rightarrow d. \quad (\text{A} \cdot 7)$$

(Step 2) Rule (A·4) is transformed into the following rules:

$$A(x_1, x_2, x_3) \rightarrow B'(x_3, x_1)I(x_2), \quad (\text{A} \cdot 8)$$

$$B'(x_1, x_2) \rightarrow B(C'_1(x_1, x_2), C'_2(x_2)), \quad (\text{A} \cdot 9)$$

$$C'_1(x_1, x_2) \rightarrow C(I(x_1), F(x_2))[a]H_3(x_1), \quad (\text{A} \cdot 10)$$

$$C'_2(x_1) \rightarrow C(I(x_1), [d]). \quad (\text{A} \cdot 11)$$

Rule (A·10) is further transformed as:

$$C'_1(x_1, x_2) \rightarrow C'(x_1, x_2)[a]H_3(x_1), \quad (\text{A} \cdot 12)$$

$$C'(x_1, x_2) \rightarrow C(C''_1(x_1), C''_2(x_2)), \quad (\text{A} \cdot 13)$$

$$C''_1(x_1) \rightarrow I(x_1), \quad (\text{A} \cdot 14)$$

$$C''_2(x_1) \rightarrow F(x_1). \quad (\text{A} \cdot 15)$$

(Step 3) Since the right-hand side of rule (A·12) is an s -term consisting of three terms, this rule is transformed as:

$$C'_1(x_1, x_2) \rightarrow C'(x_1, x_2)D_1(x_1),$$

$$D_1(x_1) \rightarrow [a]H_3(x_1).$$

(Step 4) Rules (A·9), (A·11) and (A·13) are transformed as follows. The constructed rules are the ones before renaming.

$$B'(x_1, x_2) \rightarrow E_1(C'_1(x_1, x_2), x_2),$$

$$E_1(x'_1, x_2) \rightarrow B(x'_1, C'_2(x_2)),$$

$$C'_2(x_1) \rightarrow E'_1(I(x_1)),$$

$$E'_1(x'_1) \rightarrow C(x'_1, [d]),$$

$$C'(x_1, x_2) \rightarrow E''_1(C''_1(x_1), x_2),$$

$$E''_1(x'_1, x_2) \rightarrow C(x'_1, C''_2(x_2)).$$

The obtained rules are (A·5)–(A·8), (A·14), (A·15), and all the rules constructed in (Step 3) and (Step 4).

A.3 Proof of Lemma 12

Let t be a derivation tree of G . Let v and v' be internal nodes in t labeled with A and A' , respectively, where v' is an ancestor of v or v itself. A function $g_{v,v'} : (T^*)^{\dim(A)} \rightarrow (T^*)^{\dim(A')}$ is defined as follows. Let $\bar{y} = (y_1, y_2, \dots, y_{\dim(A)})$ be a variable over $(T^*)^{\dim(A)}$:

$$(1) \quad g_{v,v'}(\bar{y}) = \bar{y}$$

(2) Assume that $v \neq v'$. Let v_1, v_2, \dots, v_w (labeled with A_1, A_2, \dots, A_w , respectively) be the children of v' , and v_i ($1 \leq i \leq w$) be the child of v' on the path from v' to v in t . Let $A' \rightarrow f[A_1, A_2, \dots, A_w]$ be the rule applied at v' in t , and s_j be the constant string derived from v_j ($j \neq i$) in t . Then

$$g_{v,v'}(\bar{y}) = (s_1, \dots, s_{i-1}, g_{v,v_i}(\bar{y}), s_{i+1}, \dots, s_w).$$

From the definition, for any $\bar{a} \in L_G(A)$,

$$g_{v,v'}(\bar{a}) \in L_G(A'). \quad (\text{A} \cdot 16)$$

Since G satisfies the nonerasing condition (N1) of Lemma 1, each variable y_i is contained in one and only one component of $g_{v,v'}(\bar{y})$. Let us denote the sum of string lengths of components of $g_{v,v'}(\bar{y})$ by $|g_{v,v'}(\bar{y})|$. Since G satisfies the conditions (N1), (N2) and (N3) of Lemma 1, if there exists a node which is not v and has two or more children on the path from v' to v in t , the following inequality holds:

$$|g_{v,v'}(\bar{y})| > \dim(A). \quad (\text{A} \cdot 17)$$

Let n be a nonnegative integer. Choose a string $\alpha \in L$ and its derivation tree t_0 as described in the proof sketch of

Lemma 12 in Sect. 4. Assume that v is a descendant of v' . Let $k = \dim(A)$. Let us denote $g_{v,v'}$ by g for simplicity, and the function obtained by composing g i times by g^i . Note that g^i is not a value obtained by concatenating the value of g i times. For a function g , let us denote the j th component of g by g_j .

Let $K = \{1, 2, \dots, k\}$. We define a function μ from K to K such that if a variable y_n ($n \in K$) is contained in g_j , then $\mu(n) = j$. Let \bar{J} be the maximal nonempty subset K' of K which satisfies the condition: if we regard μ as a function from K' to K (by restricting the domain), μ is a permutation over K' . This subset \bar{J} (called the *kernel*) can always be found.

From the definition of \bar{J} and the fact that the number of components of g is k , for each variable y_n ($n \notin \bar{J}$), y_n is moved to one of the components in the kernel by composing g at most $(k-1)$ times. Therefore, if we let $J^i = \{j \mid \text{the } j\text{th component of } g^i \text{ is a constant string}\}$, then $J^i = J^{k-1}$ holds for each i ($i \geq k$). Let $\nu = \mu^{k-1}$. Since ν is also a permutation over the kernel \bar{J} , there exists some integer p such that the permutation obtained by composing ν p times is the identity permutation. Let us denote $g^{p(k-1)}$ by \bar{g} for simplicity. $\bar{g}_j(\bar{y})$ is a constant string of the form $\gamma_j \in T^+$ if $j \notin \bar{J}$ and $\gamma_{j_1} \gamma_j \gamma_{j_2}$ if $j \in \bar{J}$, where γ_{j_1} and γ_{j_2} are strings over $T \cup \{y_j \mid j \notin \bar{J}\}$. Hence, for any $j \in \bar{J}$, $\bar{g}_j^2(\bar{y}) = \gamma'_{j_1} \bar{g}_j(\bar{y}) \gamma'_{j_2}$, where γ'_{j_1} and γ'_{j_2} are the strings over T obtained from γ_{j_1} and γ_{j_2} , respectively, by substituting γ_i for y_i ($i \notin \bar{J}$). For any positive integer i ,

(1) if $j \in \bar{J}$, then

$$\bar{g}_j^i(\bar{y}) = (\gamma'_{j_1})^{i-1} \bar{g}_j(\bar{y}) (\gamma'_{j_2})^{i-1}; \quad (\text{A} \cdot 18)$$

(2) otherwise,

$$\bar{g}_j^i(\bar{y}) = \gamma_j. \quad (\text{A} \cdot 19)$$

Since $|g(\bar{y})| > k$ from (A·17) and $|\bar{g}^{i+1}(\bar{y})| > |\bar{g}^i(\bar{y})|$,

$$\sum_{j \in \bar{J}} |\gamma'_{j_1} \gamma'_{j_2}| > 0. \quad (\text{A} \cdot 20)$$

On the other hand, from the condition (N1),

$$g_{v',r}(\bar{y}) = u_1 y_{h_1} u_2 y_{h_2} \cdots u_k y_{h_k} u_{k+1}, \quad (\text{A} \cdot 21)$$

where r is the root of t_0 , $u_h \in T^*$ ($1 \leq h \leq k+1$) and (h_1, h_2, \dots, h_k) is a permutation of $(1, 2, \dots, k)$. Let $\bar{\beta} \in L_G(A)$ be the string derived from v in t_0 . Then, from (A·16), $\bar{g}^i(\bar{\beta}) \in L_G(A)$, $i \geq 0$. Again from (A·16),

$$g_{v',r}(\bar{g}^i(\bar{\beta})) \in L_G(A), \quad i \geq 0. \quad (\text{A} \cdot 22)$$

The iteration property of the lemma holds by (A·18) through (A·22) letting $z_i = g_{v',r}(\bar{g}^i(\bar{\beta}))$ for $i \geq 0$. Note that $k = \dim(A) \leq m$.

In what follows, we evaluate the length of $\sum_{j=2}^k |u_j|$. Let $v_1 v_2 \cdots v_s$ be the path from r to v' ($v_1 = r$ and $v_s = v'$). Note that by the translation from a given mcfg to a normal form mcfg given in the proof of Lemma 1 (Lemma 2.2 of [24]), every function f in F is either

- a constant function $f[\] = a \in T$, or
- a function obtained from a function defined in (3),(4) or (5) in the proof of Lemma 10 by deleting some (possibly zero, but not all) variables in the definition of f and deleting the resulting components that are the empty strings.

For $g_{v_{h+1},v_h}(\bar{y}) = (\beta_1, \dots, \beta_w)$, let $\downarrow g_{v_{h+1},v_h}(\bar{y}) = \beta_1 \cdots \beta_w$. By the above property, for each h ($1 \leq h < s$), $\downarrow g_{v_{h+1},v_h}(\bar{y})$ has either of the the following forms:

$$\downarrow g_{v_{h+1},v_h}(\bar{y}) = y_1 y_2 \cdots y_i \gamma_0 y_{i+1} \cdots y_k, \quad (\text{A} \cdot 23)$$

or

$$= \gamma_1 y_1 \cdots y_k \gamma_2 \quad (\text{A} \cdot 24)$$

$$(\gamma_0, \gamma_1, \gamma_2 \in T^*).$$

Therefore, the length of $u_2 u_3 \cdots u_k$ is the sum of $|\gamma_0|$ in (A·23) for each h ($1 \leq h < s$). On the other hand, $|\gamma_0|$ is positive only if

- the function appearing in the right-hand side of the applied rule at v_h is (obtained by the translation in Lemma 1 from) a function constructed in (5) in the proof of Lemma 10, and
- v_{h+1} is the first (left) child of v_h .

Let such v_h 's be $v_{i_1}, v_{i_2}, \dots, v_{i_d}$ in the order from r to v' , and let $l(v)$ denote the sum of the lengths of the components of the strings derived from the second (right) child of v ; then

$$\sum_{j=2}^k |u_j| = \sum_{j=1}^d l(v_{i_j}).$$

In order to make $\sum_{j=2}^k |u_j|$ not greater than some constant depending only on L , we choose a path v_1, v_2, \dots from the root r to v' in such a way that if the function appearing in the right-hand side of the applied rule at v_h is constructed in (5) of Lemma 10, we let the next node be the second child of v_h (if possible) in the following way. Let k denote $|N|$.

Let p be a path from the root r to a leaf in t_0 such that the number of the nodes on p which have two children is at least $k+1$ and p satisfies the following conditions (such a path always exists in t_0):

Let v_h be a node on p which has two children, and u_1 and u_2 be the first and the second children of v_h , respectively. Let j denote the number of the nodes which are in the sequence of nodes from r to v_h and have two children. If there exists a path from u_2 to a leaf such that the number of the nodes on the path which have two children is $k+1-j$ or more, then the next node v_{h+1} to v_h on p is u_2 , and u_1 otherwise.

By the definition of p mentioned above, $l(v_{i_j}) \leq 2^{k-j}$. If we choose a pair v, v' of nodes having identical labels which have two children in such a way that v' is nearest to the root r among such pairs, then $d \leq k$ holds. Therefore,

$$\sum_{j=2}^k |u_j| = \sum_{j=1}^d l(v_{i_j}) \leq \sum_{j=1}^k 2^{k-j} = 2^k - 1.$$

By the definition of α and z , $v_a(z) \geq v_a(\alpha) \geq n$ for each n . Let M be $2^k - 2$. This completes the proof.

A.4 Proof of Lemma 19

Let $G = (N, T, F, P, S)$ be a 1-mcfg(m) that satisfies conditions (N1), (N2) and (N3) in Lemma 1. Let $\text{nt}(f)$ be the number of terminals appearing in the right-hand side of the definition of $f \in F$. For example, if $f[(x_1, x_2)] = (ax_1b, cx_2d)$, $\text{nt}(f) = 4$. Let $\text{nt}(G) = \max\{\text{nt}(f) \mid f \in F\}$. By the assumption, there exists $\alpha \in L$ such that (1) $|\alpha|_a \geq n$ and (2) $|\alpha| \geq 2^{\text{nt}(G)|N|+1}$. Let t_0 be a derivation tree of α . Since G is a 1-mcfg(m), t_0 is just a path. We call a node v *productive* if $\text{nt}(f) \geq 1$ for the rule $A \rightarrow f[A_1]$ applied at v . Thus, there exist two distinct productive nodes v and v' with the same label (say A) in t_0 . The proof proceeds in a similar way to that of Lemma 12. Let t be the derivation tree for $z_1 = g_{v',r}(\bar{g}(\beta))$ in the proof of Lemma 12. If we let v be the node as close as possible to the root r , the number of nodes above v' (not including v) in t is at most $|N|$. Thus $|u_1u_2 \cdots u_{\dim(A)+1}| \leq \text{nt}(G)|N|$. Since the number of nodes between v' and v (including v' and not including v) in t_0 is also at most $|N|$ and t is obtained from t_0 by repeating $p(\dim(A) - 1)$ times the path between v' and v (see the proof of Lemma 12), the number of nodes between v' and v in t is at most $|N|p(\dim(A) - 1)$. Hence, $\sum_{j=1}^{\dim(A)} |v_j s_j| = \sum_{j \in J} |\gamma'_{j1} \gamma'_{j2}| + \sum_{j \notin J} |\gamma_j| \leq \text{nt}(G)|N|p(\dim(A) - 1)$. Summarizing, if we let $M = \text{nt}(G)|N|(p(\dim(A) - 1) + 1)$, $\sum_{j=1}^{\dim(A)} |u_j v_j s_j| + |u_{\dim(A)+1}| \leq M$ as desired.



Hiroyuki Seki received the Ph.D. degree in information and computer sciences from Osaka University in 1987. He was with Osaka University as an Assistant Professor in 1990–1992 and an Associate Professor in 1992–1994. In 1994, he joined the faculty of Nara Institute of Science and Technology, where he has been a Professor since 1996. His current research interests include formal language theory and formal approach to software development.



Yuki Kato received the Ph.D. degree in information science from Nara Institute of Science and Technology in 2007. He is with Kyoto University as a Research Fellow of the Japan Society for the Promotion of Science from April 2007. His current research interests include algorithms, formal language theory and their application to biological sequence analysis.