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

Hiroyuki Seki and Yuki Kato

**Abstract**

Several grammars of which generative power is between context-free grammar and context-sensitive grammar were proposed. Among them, multiple context-free grammar and variable-linear macro grammar are known to be recognizable in polynomial time. In this paper, the generative power of some subclasses of variable-linear macro grammar and that of multiple context-free grammar are compared in details. We also mention an application of parsing methods for these grammars to structure prediction of biological sequences.

## 1 Introduction

Several grammars of which generative power is between context-free grammar (cfg) and context-sensitive grammar were proposed. Among them, multiple context-free grammar (mcfg) [10, 23] 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 [26] is essentially the same formalism as mcfg. Later, Weir[27] 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 yield (the string obtained by concatenating the leaf labels) of cftg corresponds to macro grammar [6]. Also, the class of yield languages generated by cftgs is known to be the same as the class of indexed languages [2]. Whether the membership problem for macro grammar is solvable in polynomial time 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

$$\text{1-MCFL}(m+1) = \text{L}^2\text{-ML}(m) \subset \text{VL-ML}(m) \subseteq \text{2-MCFL}(m+1) \quad (m \geq 0)$$

where $r$-MCFL($m$) is the class of languages generated by mcfgs with dimension at most $m$ and rank at most $r$, VL-ML($m$) and L$^2$-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 $m \geq 1$.

# 2 Preliminaries

## 2.1 Multiple Context-Free Grammar

We will use standard notions and notations on strings and languages. Let $\varepsilon$ denote the empty string. Let $\Sigma$ be an alphabet. For a string $\alpha \in \Sigma^*$ and a symbol $a \in \Sigma$, let $|\alpha|$ denote the number of symbols appearing in $\alpha$, called the length of $\alpha$, and let $|\alpha|_a$ denote the number of $a$'s appearing in $\alpha$. 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 functions, $P$ a finite set of (production) rules 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 $S$, $\dim(S) = 1$. For each $f \in F$, positive integers $d_i$ $(0 \leq i \leq k)$ are given and $f$ is a total function from $(T^*)^{d_1} \times \cdots \times (T^*)^{d_k}$ to $(T^*)^{d_0}$ satisfying the following condition (F):

**(F)** Let $\overline{x_i} = (x_{i1}, \ldots, x_{id_i})$ 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

$$f^{[h]}[\overline{x_1}, \ldots, \overline{x_k}] = \beta_{h0} z_{h1} \beta_{h1} z_{h2} \cdots z_{hv_h} \beta_{hv_h} \qquad (*)$$

where $\beta_{hl} \in T^*$ $(0 \le l \le v_h)$ and $z_{hl} \in \{x_{ij} \mid 1 \le i \le k,\ 1 \le j \le d_i\}$ $(1 \le l \le v_h)$. 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 $A_0 \to f[A_1, \ldots, A_k]$ where $A_i \in N$ $(0 \le i \le k)$ and $f : (T^*)^{\dim(A_1)} \times \cdots \times (T^*)^{\dim(A_k)} \to (T^*)^{\dim(A_0)} \in F$. If $k \ge 1$, the rule is called a *nonterminating rule*, and if $k = 0$, it is called a *terminating rule*. A terminating rule $A_0 \to f[\,]$ with $f^{[h]}[\,] = \beta_h$ $(1 \le h \le \dim(A_0))$ is simply written as $A_0 \to (\beta_1, \ldots, \beta_{\dim(A_0)})$.

**Example 1.** (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 $P_1 = \{S \to J[A],\ A \to f_a[A] \mid f_b[A] \mid (\varepsilon, \varepsilon)\}$ where $\dim(S) = 1,\ \dim(A) = 2,\ J[(x_1, x_2)] = x_1x_2$ and $f_\alpha[(x_1, x_2)] = (\alpha x_1, \alpha x_2)$ with $\alpha = a, b$.

(2) Let $G_2 = (N_2, T_2, F_2, P_2, S)$ be an mcfg where $N_2 = \{S, A\}$, $T_2 = \{a_i \mid 1 \le i \le 2m\}$ and $P_2 = \{S \to J_m[A],\ A \to g[A] \mid (\varepsilon, \ldots, \varepsilon)\}$ where $\dim(S) = 1,\ \dim(A) = m,\ J_m[(x_1, \ldots, x_m)] = x_1 \cdots x_m$ and $g[(x_1, \ldots, x_m)] = (a_1x_1a_2, \ldots, a_{2m-1}x_ma_{2m})$.

(3) Let $G_3 = (N_2, T_2, F_3, P_3, S)$ be an mcfg where $P_3 = \{S \to J_m^2[A, A],\ A \to g[A] \mid (\varepsilon, \ldots, \varepsilon)\}$ where $J_m^2[(x_1, \ldots, x_m), (y_1, \ldots, y_m)] = x_1 \cdots x_m y_1 \cdots y_m$. □

For a function $f$ defined by $(*)$ in condition (F) and tuples of terminal strings $\alpha_i = (\alpha_{i1}, \ldots, \alpha_{id_i}) \in (T^*)^{d_i}$ $(1 \le i \le k)$, let $f[\alpha_1, \ldots, \alpha_k]$ denote the tuple of terminal strings obtained from the right-hand sides of $(*)$ by substituting $\alpha_{ij}$ $(1 \le i \le k,\ 1 \le j \le \dim(A_i))$ 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 $A \to \alpha \in P$ $(\alpha \in (T^*)^{\dim(A)})$, then we write $A \stackrel{*}{\Rightarrow} \alpha$.

**(L2)** If $A \to f[A_1, \ldots, A_k] \in P$ and $A_i \stackrel{*}{\Rightarrow} \alpha_i$ $(1 \le i \le k)$, then we write $A \stackrel{*}{\Rightarrow} f[\alpha_1, \ldots, \alpha_k]$.

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 $L(G) = L_G(S)$. A language $L$ is a *multiple context-free language* (*mcfl*) 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 $\stackrel{*}{\Rightarrow}$, we define derivation trees:

**(D1)** If $A \to \alpha \in P$ $(\alpha \in (T^*)^{\dim(A)})$, then a derivation tree of $\alpha$ is the tree with a single node labeled $A : \alpha$.

**(D2)** If $A \to f[A_1, \ldots, A_k] \in P$, $A_i \overset{*}{\Rightarrow} \alpha_i$ $(1 \leq i \leq k)$ and $t_1, \ldots, t_k$ are derivation trees of $\alpha_1, \ldots, \alpha_k$, then a derivation tree of $f[\alpha_1, \ldots, \alpha_k]$ is the tree with the root labeled $A : f$ that has $t_1, \ldots, t_k$ as (immediate) subtrees from left to right.

**Example 1** (continued). (1) By (L1), $A \overset{*}{\Rightarrow}_{G_1} (\varepsilon, \varepsilon)$ since $A \to (\varepsilon, \varepsilon) \in P$. Since $f_a[(\varepsilon, \varepsilon)] = (a, a)$ and $f_b[(a, a)] = (ba, ba)$, we have $A \overset{*}{\Rightarrow}_{G_1} (a, a)$ and $A \overset{*}{\Rightarrow}_{G_1} (ba, ba)$ by (L2). Also by $S \to J[A]$, $S \overset{*}{\Rightarrow}_{G_1} J[(ba, ba)] = baba$. 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, $A \overset{*}{\Rightarrow}_{G_2} (\varepsilon, \ldots, \varepsilon)$ by (L1), $A \overset{*}{\Rightarrow}_{G_2} f[(\varepsilon, \ldots, \varepsilon)] = (a_1 a_2, \ldots, a_{2m-1} a_{2m})$ by (L2), etc. This tells us that $L(G_2) = \{a_1^n \cdots a_{2m}^n \mid n \geq 0\}$. This language is called $L_1^{(m)}$ in the rest of the paper.

(3) Since $L_{G_3}(A) = L_{G_2}(A)$, $L(G_3) = \{\alpha\beta \mid \alpha, \beta \in L_1^{(m)}\}$. This language is called $L_2^{(m)}$. □

To introduce subclasses of MCFG, we define a few terminologies. Let $G = (N, T, F, P, S)$ be an arbitrary mcfg. For a function $f : (T^*)^{d_1} \times \cdots \times (T^*)^{d_k} \to (T^*)^{d_0}$, let $\dim(f) = \max\{d_i \mid 0 \leq i \leq k\}$, $\text{rank}(f) = k$ and $\deg(f) = \Sigma_{j=0}^{k} d_j$, which are called the *dimension*, *rank* and *degree* of $f$, respectively. $\dim(G)$, $\text{rank}(G)$ and $\deg(G)$ are defined as the maximum of $\dim(f)$, $\text{rank}(f)$ and $\deg(f)$ among all $f \in F$, respectively. By definition, $\deg(G) \leq \dim(G)(\text{rank}(G) + 1)$. With these parameters, we define subclasses of MCFG. An mcfg $G$ with $\dim(G) \leq m$ and $\text{rank}(G) \leq r$ is called an $r$-mcfg($m$). Likewise, an mcfg $G$ with $\dim(G) \leq m$ is called an mcfg($m$) and an mcfg $G$ with $\text{rank}(G) \leq r$ is called an $r$-mcfg.

**Lemma 1** (Normal form mcfg [10, 23]). For a given $r$-mcfg($m$) $G$, we can construct an $r$-mcfg($m$) $G' = (N', T', F', P', S')$ such that $L(G') = L(G)$ and $G'$ 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 $A \in N'$ $(A \neq S')$, $A \overset{*}{\Rightarrow}_{G'} (\alpha_1, \ldots, \alpha_{\dim(A)})$ implies $\alpha_i \neq \varepsilon$ $(1 \leq i \leq \dim(A))$.

(N3) If $A \to \varepsilon$, then $A = S'$ and $S'$ does not appear in the right-hand side of any rule in $P'$. □

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

**Proposition 2** ([11, 23]). Let $G$ be an mcfg with $\deg(G) = e$. For a given $w \in T^*$, whether $w \in L(G)$ or not can be decided in $O(n^e)$ time where $n = |w|$. □

## 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 $\Sigma$ and $X$ respectively, defined as the smallest sets satisfying the following conditions:

(1) $\Sigma_0 \cup X \subseteq T_\Sigma(X)$.

(2) $f(t_1, \ldots, t_n) \subseteq T_\Sigma(X)$ if $t_i \in T_\Sigma^+(X)$ ($1 \leq i \leq n$) and $f \in \Sigma_n$.

(3) $t_1 \cdots t_l \subseteq T_\Sigma^+(X)$ if $t_i \in T_\Sigma(X)$ ($1 \leq i \leq l$) and $l \geq 1$.

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

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. $P$ is a finite set of (production) rules of which shapes are:

$$
\begin{aligned}
S &\to \varepsilon \text{ or} \\
A(x_1, \ldots, x_k) &\to t_A \text{ where } A \in N_k \text{ and } t_A \in T_\Sigma^+(X_k).
\end{aligned}
$$

If $S \to \varepsilon \in P$, $S$ does not appear in the right-hand side of any rule. For an s-term $t$, its subterm $t_1$ and a rule $A(x_1, \ldots, x_k) \to t_A$, if $t_1 = A(t_1, \ldots, t_k)$, then we write $t \Rightarrow_G t[t_1 \leftarrow t_A\theta]$ where $\theta$ is the substitution defined by $\theta(x_i) = t_i$ ($1 \leq i \leq k$) and $\theta(x) = x$ otherwise. Let $\overset{*}{\Rightarrow}_G$ and $\overset{+}{\Rightarrow}_G$ 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 $L(G) = \{t \in T^* \mid S \overset{*}{\Rightarrow}_G t\}$, 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 mg($m$). Let $A(x_1, \ldots, x_k) \to t_A \in P$ 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*. 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.** (1) Let $G_4 = (N_4, T_4, P_4, S)$ be an mg where $N_4 = \{S, A\}$ with $a(S) = 0$ and $a(A) = m - 1$, $T_4 = \{a_i \mid 1 \leq i \leq 2m\}$ and
$P_4 = \{S \to a_1 A(a_2 a_3, \ldots, a_{2m-2} a_{2m-1}) a_{2m} \mid \varepsilon,$
$A(x_1, \ldots, x_{m-1}) \to a_1 A(a_2 x_1 a_3, \ldots, a_{2m-2} x_{m-1} a_{2m-1}) a_{2m} \mid x_1 \cdots x_{m-1}\}$. $G_4$ is an l$^2$-mg($m-1$). $S \Rightarrow_{G_4} \varepsilon$ and

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

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

(2) Let $G_5 = (N_5, T_4, P_5, S_5)$ be an mg where $N_5 = \{S_5, S, A\}$ with $a(S_5) = 0$, and $P_5 = \{S_5 \to SS\} \cup P_4$. $G_5$ is a vl-mg($m-1$) but not n-linear. $L(G_5) = L_2^{(m)}$.

(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 $P_6 = \{S \to A(a), A(x) \to A(xx) \mid x\}$. $G_6$ is an nl-mg(1) but not v-linear. $S \Rightarrow_{G_6} A(a) \Rightarrow_{G_6} A(a^2) \Rightarrow_{G_6} A(a^4) \Rightarrow_{G_6} \ldots$. $L(G_6) = \{a^{2^n} \mid n \geq 0\}$, which is not an mcfl [10, 23]. □

## 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 $r = 2$,

$$\begin{aligned} r\text{-MCFL}(m) &\subset (r+1)\text{-MCFL}(m). \\ 2\text{-MCFL}(2) &= 3\text{-MCFL}(2). \end{aligned}$$

For $r \geq 2$, 1-MCFL$(1) \subset r$-MCFL$(1) = (r+1)$-MCFL$(1)$. □

By definition, it is easy to see LCFL = 1-MCFL(1) and CFL = $r$-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, 23].

**Proposition 4.** For $m \geq 1$,

$$\begin{aligned} \text{MCFL}(m) &\subset \text{MCFL}(m+1). \\ L_1^{(m+1)} &\in 1\text{-MCFL}(m+1) \setminus \text{MCFL}(m). \quad \square \end{aligned}$$

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

Finally, we present a few closure properties.

**Proposition 6.** $r$-MCFL($m$) ($m \geq 1$, $r \geq 2$), MCFL($m$) ($m \geq 1$) and $r$-MCFL ($r \geq 2$) are all substitution closed full AFLs. 1-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 [23], the third by [5] (as a closure property of tree transducers), and the last claim by [15] (as a closure property of EDT0L systems).

# 3 Normal Forms for Macro Grammars

In this section, we will discuss simplification of macro grammars. For grammars $G_1$ and $G_2$, we say that $G_1$ is weakly equivalent to $G_2$ if $L(G_1) = L(G_2)$. First, we will show that an arbitrary mg($m$) $G$ can be tranformed to 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 to a weakly equivalent mg($m$) $G'$ such that variables appear in ascending order of suffixes $x_1, x_2, \ldots$ in the right-hand side of each rule (nonpermuting) (Lemma 8). Lastly, it is shown 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($m$)). We can construct an mg($m$) (rsp. vl-mg($m$), nl-mg($m$)) $G'$ such that $L(G') = L(G)$ and $G'$ satisfies the following condition: (Nonerasing) For each rule $A(x_1, \ldots, x_{a(A)}) \to t_A$ of $G'$ each variable $x_i$ ($1 \leq i \leq a(A)$) appears at least once in $t_A$.

*Proof.* Let $A(x_1, \ldots, x_{a(A)}) \to t_A$ be a rule that does not satisfy the condition. Let $\Psi \subseteq \{1, 2, \ldots, a(A)\}$ be such that $i \in \Psi$ if and only if $x_i$ does not appear in $t_A$. Let $i_1, \ldots, i_{a(A)-|\Psi|}$

be the listing in the ascending order of variable suffixes of the set $\{1, 2, \dots, a(A)\}\backslash\Psi$. Add a new nonterminal $A_\Psi$ with $a(A_\Psi) = a(A) - |\Psi|$ and add a rule

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

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 $B(x_1, \dots, x_{a(B)}) \to t_B$ where $t_B$ contains $A$, add a rule

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

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

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

**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($m$)) such that $L(G') = L(G)$ and $G'$ satisfies the following condition: (Nonpermuting) For each rule $A(x_1, \dots, x_{a(A)}) \to t_A$ 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 $A(x_1, x_2, x_3) \to B(x_2, C(x_3, x_1))$, violating the condition. We eliminate this rule and add $A^\pi(x_1, x_2, x_3) \to B(x_1, C(x_2, x_3))$ instead where $\pi$ 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, $D(x_1, x_2, x_3) \to E(A(t_1, t_2, t_3))$, eliminate the rule and add $D(x_1, x_2, x_3) \to E(A^\pi(t_{\pi(1)}, t_{\pi(2)}, t_{\pi(3)}))$. 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 $A(x_1, \dots, x_{a(A)}) \to t_A \in P$, 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 $\pi_t$ 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 $A^\pi(x_1, \dots, x_{a(A)}) \to t''_A$ 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 $\pi_t$ for subterms $t$ of $t_A$.

Let $G'$ 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 $A(x_1, \dots, x_{a(A)}) \overset{*}{\Rightarrow}_G t \in (T \cup X)^*$ if and only if $A^\pi(x_{\pi(1)}, \dots, x_{\pi(a(A))}) \overset{*}{\Rightarrow}_{G'} t$. 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. $\square$

Before proceeding the main result on normal forms for mgs, we need a technical definition on the shape of s-terms.

- An s-term $t$ is in *b-normal form* if either
  - $t \in X \cup T$,
  - $t = t_1 \cdots t_l$ where $l \geq 2$ and $t_1, \ldots, t_l$ are balanced and not variables, or
  - $t = A(t_1, \ldots, t_{a(A)})$ ($A \in N$) where $t_1, \ldots, t_{a(A)}$ are balanced.
- S-terms $t_1, \ldots, t_k$ are *balanced* if either
  - $t_j \in X$ ($1 \leq j \leq k$) or
  - $t_j = B_{j1}(t_{j11}, \ldots, t_{j1a(B_{j1})}) \cdots B_{js_j}(t_{js_j1}, \ldots, t_{js_ja(B_{js_j})})$ where $B_{jk} \in N$ and $t_{jk1}, \ldots, t_{jka(B_{jk})}$ are balanced for each $k$ ($1 \leq k \leq s_j$).

For example, $a(\in T)$ and $A(x, y)$ and $B(C(z, x), D(x)C(y))$ are all in b-normal form. Neither $xy$ or $A(B(x), y)$ or $A(x, a, y)$ is in b-normal form.

**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:

$$\begin{array}{rcl} A & \to & a \ \ (a \in T) \\ A(x) & \to & x \ \ (x \in X) \\ A(x_1, \ldots, x_{a(A)}) & \to & B(y_1, \ldots, y_{a(B)}) \\ A(x_1, \ldots, x_{a(A)}) & \to & B(y_1, \ldots, y_{a(B)}) \cdot C(z_1, \ldots, z_{a(C)}) \\ A(x_1, \ldots, x_{a(A)}) & \to & B(y_1, \ldots, C(z_1, \ldots, z_{a(C)}), \ldots, y_{a(B)}) \\ & & \{y_1, \ldots, y_{a(B)}, z_1, \ldots, z_{a(C)}\} \subseteq \{x_1, \ldots, x_{a(A)}\} \end{array}$$

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.* 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 of $G$ is transformed into an s-term in b-normal form in the topdown way as follows: Each rule $A(x_1, \ldots, x_{a(A)}) \to t_A$ with $t_A \notin X \cup T$ and $t_A \neq B(y_1, \ldots, y_{a(B)})$ is replaced with the rule $A(x_1, \ldots, x_{a(A)}) \to$ Trans($t_A$) and the rules generated by Trans($t_A$) as side effects.
Trans($t$) :

- $t = x \in X$: Trans($x$) $= I(x)$ where $I$ is a new nonterminal. Add $I(x) \to x$ as a side effect.

- $t = a \in T$: Trans$(a) = [a]$ where $[a]$ is a new nonterminal. Add $[a] \to a$ as a side effect.
- $t = t_1 \cdots t_l$ $(l \geq 2)$: Trans$(t) =$ Trans$(t_1) \cdots$ Trans$(t_l)$.
- $t = A(x_{i_1}, \ldots, x_{i_{a(A)}})$: Trans$(t) = t$.
- $t = A(t_1, \ldots, t_{a(A)})$ with at least one $t_i$ not a variable:
  Trans$(t) = A($Trans$(t_1), \ldots,$ Trans$(t_{a(A)}))$.

(Step 2) Each rule $A(x_1, \ldots, x_{a(A)}) \to B_1(t_{11}, \ldots, t_{1a(B_1)}) \cdots B_s(t_{s1}, \ldots, t_{sa(B_s)})$ $(s \geq 2)$ is replaced with the following rule:

$$A(x_1, \ldots, x_{a(A)}) \to \alpha_1 \cdots \alpha_s \text{ where} \tag{3.1}$$

(a) $\alpha_i = B_i(t_{i1}, \ldots, t_{ia(B_i)})$ if all $t_{ij}$ $(1 \leq j \leq a(B_i))$ are variables,

(b) $\alpha_i = B_i'(x_{i1}, \ldots, x_{ia(B_i')})$ otherwise. $B_i'$ is a new nonterminal and $x_{i1}, \ldots, x_{ia(B_i')}$ are variables appearing in $t_{i1}, \ldots, t_{ia(B_i)}$, arranged without duplication in the order of their (first) occurrences from left to right.

In case (b) above, the following rules are also added. Since each $t_{ij}$ $(1 \leq j \leq d(B_i))$ is in b-normal form by (Step 1), $t_{ij}$ has the shape of $C_{j1}(\ldots) \cdots C_{jn_j}(\ldots)$. Let $y_{j1}, \ldots, y_{ja(C_j')}$ be the variables appearing in $t_{ij}$, arranged without duplication from left to right. Let $\theta$ be the substitution that renames $x_{ij}$ by $x_j$ $(1 \leq j \leq a(B_i'))$.

$$B_i'(x_1, \ldots, x_{a(B_i')}) \to B_i(C_1(y_{11}, \ldots), \ldots, C_{a(B_i)}(y_{a(B_i)1}, \ldots))\theta \tag{3.2}$$

$$C_j'(x_1, \ldots, x_{a(C_j')}) \to C_{j1}'(\ldots) \cdots C_{jn_j}'(\ldots)\theta_j \ (1 \leq j \leq a(B_i)) \tag{3.3}$$

where $\theta_j$ is the substitution that renames $y_{jk}$ by $x_k$ $(1 \leq k \leq a(C_j'))$. Note that the rules (3.1) and (3.2) are already in b-normal form. Apply (Step 2) to the rule (3.3) repeatedly until all the rules are in b-normal form.

(Step 3) Each rule $A(\ldots) \to B_1(\ldots) \cdots B_s(\ldots)$ $(s \geq 3)$ obtained as (3.1) in (Step 2) is replaced with the following rules. This step is similar to the construction of Chomsky normal form for cfg.

$$\begin{array}{rcl} A(\ldots) & \to & B_1(\ldots) \cdot D_1(\ldots) \\ D_1(\ldots) & \to & B_2(\ldots) \cdot D_2(\ldots) \\ & \cdots & \\ D_{s-2}(\ldots) & \to & B_{s-1}(\ldots) \cdot B_s(\ldots) \end{array}$$

(Step 4) Each rule $A(\ldots) \to B(C_1(\ldots), \ldots, C_{a(B)}(\ldots))$ obtained as (3.2) in (Step 2) is replaced with the following rules.

$$\begin{array}{rcl} A(\ldots) & \to & E_1(C_1(\ldots), y_2, \ldots) \\ E_1(\ldots) & \to & E_2(y_1, C_2(\ldots), \ldots) \\ & \cdots & \\ E_{a(B)-1}(\ldots) & \to & B(y_1, \ldots, C_{a(B)}(\ldots)) \end{array}$$

$\square$

# 4 Variable-Linear Macro Grammars

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

$$A(x_1, \ldots, x_{a(A)}) \stackrel{*}{\Rightarrow}_G \alpha_0 x_1 \alpha_1 \cdots x_{a(A)} \alpha_{a(A)}, \tag{4.1}$$

variables $x_1, \ldots, x_{a(A)}$ are regarded as gaps in the derived string to be filled in. We would like to construct a 2-mcfg($m+1$) $G'$ 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

$$A \stackrel{*}{\Rightarrow}_{G'} (\alpha_0, \ldots, \alpha_{a(A)}). \tag{4.2}$$

That is, gaps $x_1, \ldots, x_{a(A)}$ in (4.1) corresponds to the gaps between the components of the tuple derived in (4.2). In this correspondence, v-linearity and the nonpermuting condition are essential.

Next we prove a pumping lemma for VL-ML($m$). By using the lemma, it is shown that the inclusion VL-ML($m$) $\subsetneq$ 2-MCFL($m+1$) is proper for every $m \geq 1$. Note that if $m = 0$, VL-ML(0) = 2-MCFL(1) = CFL.

**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($m$) and $G = (N, T, P, S)$ be the 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 $A \to a \in P$, add $A \to a$ to $P'$.

(2) If $A(x) \to x \in P$, add $A \to (\varepsilon, \varepsilon)$ to $P'$.

(3) If $A(x_1, \ldots, x_{a(A)}) \to B(x_1, \ldots, x_{a(B)}) \in P$, add $A \to id[B]$ to $P'$ and $id$ to $F$ where $id$ is an identity function $id[(x_1, \ldots, x_{\dim(A)})] = (x_1, \ldots, x_{\dim(A)})$. Note that since $G'$ is nonerasing, $a(A) = a(B)$.

(4) If $A(x_1, \ldots, x_{a(A)}) \to B(x_1, \ldots, x_{a(B)}) \cdot C(x_{a(B)+1}, \ldots, x_{a(B)+a(C)}) \in P$ where $a(A) = a(B) + a(C)$, add $A \to f[B, C]$ to $P'$ and add $f$ to $F$ where
$f[(y_1, \ldots, y_{\dim(B)}), (z_1, \ldots, z_{\dim(C)})] = (y_1, \ldots, y_{\dim(B)} z_1, \ldots, z_{\dim(C)})]$.

(5) If $A(x_1, \ldots, x_{a(A)}) \to B(x_1, \ldots, x_i, C(x_{i+1}, \ldots, x_{i+a(C)}), x_{i+a(C)+1}, \ldots, x_{a(B)+a(C)-1})$ $\in P$ where $a(A) = a(B) + a(C) - 1$, add $A \to g[B, C]$ to $P'$ and $g$ to $F$ where
$g[(y_1, \ldots, y_{\dim(B)}), (z_1, \ldots, z_{\dim(C)})]$
$= (y_1, \ldots, y_{i+1} z_1, z_2, \ldots, z_{\dim(C)-1}, z_{\dim(C)} y_{i+2}, y_{i+3}, \ldots, y_{\dim(B)})$.
Exception: If $a(C) = 0$, $g[\cdots] = (y_1, \ldots, y_{i+1} z_1 y_{i+2}, \ldots, y_{\dim(B)})$.

We can show that $A(x_1, \ldots, x_{a(A)}) \stackrel{*}{\Rightarrow}_G \alpha_0 x_1 \cdots x_{a(A)} \alpha_{a(A)}$ if and only if $A \stackrel{*}{\Rightarrow}_{G'} (\alpha_0, \ldots, \alpha_{a(A)})$. □

As noted in Sec 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 $n = |w|$. □

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

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

It is shown in [23] that RSP$^{(2)}$ $\in$ 2-MCFL(2) $\setminus$ TAL. Since TAL = VL-ML(1), RSP$^{(2)}$ $\in$ 2-MCFL(2) $\setminus$ VL-ML(1). Here we show that RSP$^{(m)}$ $\in$ 2-MCFL($m$) $\setminus$ 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 $\alpha$ 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 $z = u_1 v_1 w_1 s_1 u_2 v_2 w_2 s_2 u_3 \cdots$
$u_m v_m w_m s_m u_{m+1}$ where $\sum_{j=1}^{m} |v_j s_j| \geq 1$ and $\sum_{j=2}^{m} |u_j| \leq M$, and for any $i \geq 0$,
$z_i = u_1 v_1^i w_1 s_1^i u_2 v_2^i w_2 s_2^i u_3 \cdots u_m v_m^i w_m s_m^i u_{m+1} \in L$.

*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 $t$ 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 $g_{v,v'}$ from $(T^*)^{\dim(A)}$ to $(T^*)^{\dim(A')}$ is defined as follows. Let $\bar{y} = (y_1, y_2, \ldots, y_{\dim(A)})$ be a variable over $(T^*)^{\dim(A)}$:

(1) $g_{v,v}[\bar{y}] = \bar{y}$

(2) Assume that $v \neq v'$. Let $v_1, v_2, \ldots, v_w$ (labeled with $A_1, A_2, \ldots, A_w$, 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 $A' \to f[A_1, A_2, \ldots, A_w]$ be the rule applied at $v'$ in $t$, and $s_j$ be the string derived from $v_j$ $(j \neq i)$ in $t$. Then

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

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

$$g_{v,v'}[\bar{\alpha}] \in L_G(A'). \tag{4.3}$$

Each component of $g_{v,v'}(\bar{y})$ can be represented by the concatenation of some variables in $\{y_1, y_2, \ldots, y_{\dim(A)}\}$ and constant strings. 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:

$$|g_{v,v'}(\bar{y})| > \dim(A). \tag{4.4}$$

Let $n$ be a nonnegative integer. By the assumption, there exists $\alpha$ in $L$ satisfying that (1) $|\alpha|_a \geq n$ for each $a \in T$ and (2) $|\alpha| \geq 2^{|N|+1}$. Let $t$ be a derivation tree of $\alpha$. There exists a path $p$ from the root $r$ to a leaf in $t$ 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 [23]. We will construct a path in $t$ in such a way that $\sum_{j=2}^{m} |u_j|$ is not greater than some constant depending only on $L$.

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 compositing $g$ $i$ 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 $K = \{1, 2, \ldots, k\}$. We define a function $\mu$ 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 $\mu$ as a function from $K'$ to $K$ (by restricting the domain), $\mu$ 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 compositing $g$ 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 $\nu$ is also a permutation over the kernel $\bar{J}$, there exists some integer $p$ such that the permutation obtained by compositing $\nu$ $p$ 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 $\gamma_{j1} y_j \gamma_{j2}$ if $j \in \bar{J}$, where $\gamma_{j1}$ and $\gamma_{j2}$ are strings over $T \cup \{y_j \mid j \notin \bar{J}\}$. Hence, for any $j \in \bar{J}$, $\bar{g}_j^2(\bar{y}) = \gamma'_{j1} \bar{g}_j(\bar{y}) \gamma'_{j2}$, where $\gamma'_{j1}$ and $\gamma'_{j2}$ are the strings over $T$ obtained from $\gamma_{j1}$ and $\gamma_{j2}$, respectively, by substituting $\gamma_i$ for $y_i$ $(i \notin \bar{J})$. For any positive integer $i$,

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

$$\bar{g}_j^i(\bar{y}) = (\gamma'_{j1})^{i-1} \bar{g}_j(\bar{y}) (\gamma'_{j2})^{i-1}; \tag{4.5}$$

(2) otherwise,

$$\bar{g}_j^i(\bar{y}) = \gamma_j. \tag{4.6}$$

Since $|g(\bar{y})| > k$ from (4.4) and $|\bar{g}^{i+1}(\bar{y})| > |\bar{g}^i(\bar{y})|$,

$$\sum_{j \in \bar{J}} |\gamma'_{j1} \gamma'_{j2}| > 0. \tag{4.7}$$

On the other hand, from the condition (f3),

$$g_{v',r}(\bar{y}) = u_0 y_{h_1} u_1 y_{h_2} \cdots u_{k-1} y_{h_k} u_k, \tag{4.8}$$

where $r$ is the root of $t$, $u_h \in T^*$ $(0 \leq h \leq k)$ and $(h_1, h_2, \ldots, h_k)$ is a permutation of $(1, 2, \ldots, k)$. Let $\bar{\beta} \in L_G(A)$ be the string derived from $v$ in $t$. Then, from (4.3), $\bar{g}^i(\bar{\beta}) \in L_G(A)$, $i \geq 0$. Again from (4.3),

$$g_{v',r}(\bar{g}^i(\bar{\beta})) \in L_G(A), \; i \geq 0. \tag{4.9}$$

The iteration property of the lemma holds by (4.5) through (4.9) letting $z_i = g_{v',r}(\bar{g}^i(\bar{\beta}))$ for $i \geq 0$.

In what follows, we evaluate the length of $u_2$. 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 [23]), 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 $g_{v,v'}[\bar{y}] = (\beta_1, \ldots, \beta_w)$, let $\downarrow g_{v,v'}[\bar{y}] = \beta_1 \cdots \beta_w$. By this property, for each $h$ $(1 \leq h \leq m)$, $\downarrow g_{v_{h+1},v_h}$ has either of the the following forms:

$$\begin{aligned} \downarrow g_{v_{h+1},v_h}[y] &= y_1 y_2 \cdots y_i \gamma_0 y_{i+1} \cdots y_{\dim(A)}, \text{ or} && (4.10) \\ &= \gamma_1 y_1 \cdots y_{\dim(A)} \gamma_2 && (4.11) \\ &\qquad (\gamma_0, \gamma_1, \gamma_2 \in T^*) \end{aligned}$$

Therefore, the length of $u_2 u_3 \cdots u_m$ is the sum of $|\gamma_0|$ in (4.10) for each $h$ $(1 \leq h \leq s)$. 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 $v_{i_1}, v_{i_2}, \ldots, v_{i_d}$ 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

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

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

Let $p$ be a path from the root $r$ to a leaf in $t$ 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 $t$):

> Let $v$ be a node on $p$ which has two children, and $v_1$ and $v_2$ be the first and the second children of $v$, respectively. Let $j$ denote the number of the nodes which are in the sequence of nodes from $r$ to $v$ and have two children. If there exists a path from $v_2$ 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 to $v$ on $p$ is $v_2$, and $v_1$ 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,

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

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

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

*Proof.* (RSP$^{(m)}$ $\in$ 2-MCFL($m$)) A 2-mcfg($m$) $G$ generates RSP$^{(m)}$ where the functions and the rules of $G$ are:

- $S \rightarrow g[A, B]$ where $g[(x_1, \ldots, x_m), (y_1, \ldots, y_m)] = x_1 y_1 \cdots x_m y_m$.
- $A \rightarrow f_A[A]$ where $f_A[(x_1, \ldots, x_m)] = (a_1 x_1 a_2, \ldots, a_{2m-1} x_m a_{2m})$.
- $B \rightarrow f_B[B]$ where $f_B[(x_1, \ldots, x_m)] = (b_1 x_1 b_2, \ldots, b_{2m-1} x_m b_{2m})$.

(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

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

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 $u_1 v_1^i w_1 s_1^i u_2 v_2^i w_2 s_2^i u_3 \cdots u_m v_m^i w_m s_m^i u_{m+1} \in$ RSP$^{(m)}$ for all $i \geq 0$ holds only if

(1) $v_1 = a_1^j$, $s_1 = a_2^j$, $\ldots$, $v_m = a_{2m-1}^j$, $s_m = a_{2m}^j$ $(1 \leq j \leq q)$, or

(2) $v_1 = b_1^j$, $s_1 = b_2^j$, $\ldots$, $v_m = b_{2m-1}^j$, $s_m = b_{2m}^j$ $(1 \leq j \leq r)$.

However, neither of (a) and (b) satisfies $\sum_{j=2}^{m} |u_j| \leq M$. □

**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. □

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

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

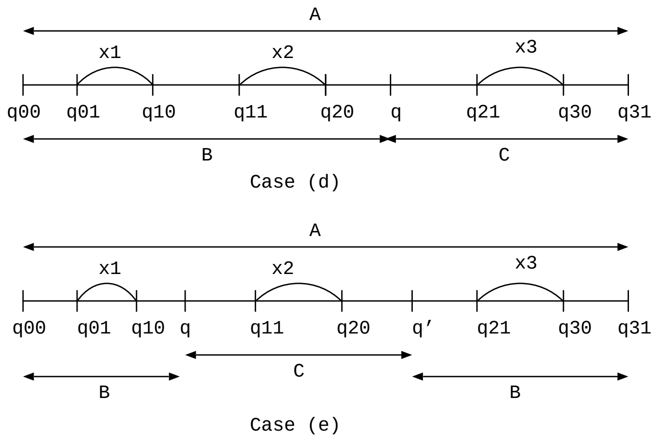


Figure 1: Construction of vl-mg rules.

*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 $Q$, $T$, $\delta : Q \times T \to Q$, $q_I \in Q$ 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) $N' = \{A[q_{00}, q_{01}; \ldots ; q_{a(A)0}, q_{a(A)1}] \mid A \in N$ and $q_{ij} \in Q$ $(0 \le i \le d(A), j = 0, 1)\}$.

(2) $P'$ consists of the following rules.

(a) $S_0 \to S[q_I, q_F]$ for $q_F \in Q_F$.

(b) $A[q_{00}, q_{01}] \to a$ if $A \to a$ and $\delta(q_{00}, a) = q_{01}$.

(c) $A[q_{00}, q_{01}; q_{10}, q_{11}] \to a$ if $A(x) \to x$ and $q_{ij} \in Q$ $(0 \le i, j \le 1)$.
The other rules are bothersome, hence we present them by examples. See Figure 1.

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

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

We can show by induction on the length of the derivations and transitions that $A[q_{00}, q_{01}; \dots; q_{a(A)0}, q_{a(A)1}](x_1, \dots, x_{a(A)}) \stackrel{*}{\Rightarrow}_{G'} \alpha_0 x_1 \dots x_{a(A)} \alpha_{a(A)}$ if and only if $A(x_1, \dots, x_{a(A)}) \stackrel{*}{\Rightarrow}_G \alpha_0 x_1 \dots x_{a(A)} \alpha_{a(A)}$ and $\delta(q_{i0}, \alpha_i) = q_{i1}$ ($0 \le i \le a(A)$).
(substitution) Let $G = (N, T, P, S)$, $G_a = (N_a, T_a, P_a, S_a)$ ($a \in T$) be vl-mg($m$) 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 $P' = \{A(x_1, \dots, x_{a(A)}) \to t'_A \mid A(x_1, \dots, x_{a(A)}) \to t_A \in P$ and $t'_A$ is obtained from $t_A$ by replacing $a \in T$ 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 $L = \{a_1^{n_1} a_2^{n_2} \cdots a_{2m+3}^{n_2} \mid n_1, n_2 \ge 0\}$ and $L' = \{a_1^{n_1} \cdots a_{2m+2}^{n_1} a_{2m+3}^{n_2} \mid n_1, n_2 \ge 0\}$. We can easily give vl-mg($m$) $G$ and $G'$ such that $L(G) = L$ and $L(G') = L'$ by observing $L = \{a_1^n \mid n \ge 0\} \cdot s(L_1^{(m+1)})$, $L' = L_1^{(m+1)} \cdot \{a_{2m+3}^n \mid n \ge 0\}$ and $L_1^{(m+1)} \in$ VL-ML($m$) where $s$ is the homomorphism such that $s(a_i) = a_{i+1}$ ($1 \le i \le 2m+2$). On the other hand, $L \cap L' = \{a_1^n \cdots a_{2m+3}^n \mid n \ge 0\} \notin$ VL-ML($m$) $\subseteq$ 2-MCFL($m+1$) [10, 23]. □

# 5 Double-Linear Macro Grammars

In this section, we show that L$^2$-ML($m$) = 1-MCFL($m+1$) $\subset$ VL-ML($m$) for every $m \ge 0$. First, we prove L$^2$-ML($m$) = 1-MCFL($m+1$). 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$^2$-mg($m$) into a weakly equivalent 1-mcfg($m+1$). For the other direction, we will introduce a nonpermuting condition for 1-mcfg. Next, we provide a pumping lemma for L$^2$-ML($m$), which implies the proper inclusion L$^2$-ML($m$) $\subset$ VL-ML($m$).

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

$$f^h[(x_1, \dots, x_{d_1})] = \alpha_h \ \ (1 \le h \le d_0).$$

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

*Proof.* Similar to the proof of Lemma 8. □

**Theorem 17.** For each $m \ge 0$, L$^2$-ML($m$) = 1-MCFL($m+1$).

*Proof.* (L$^2$-ML($m$) $\subseteq$ 1-MCFL($m+1$)) Let $G = (N, T, P, S)$ be an arbitrary l$^2$-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

$A(x_1, \ldots, x_{a(A)}) \to t_A$ be an arbitrary rule of $G$. Since $G$ is n-linear, $t_A$ can be written as $t_A = \alpha B(\beta_1, \ldots, \beta_{a(B)})\gamma$ where $\alpha, \beta_i (1 \leq i \leq a(B)), \gamma \in (T \cup X)^*$.

Let $\Omega_0, \ldots, \Omega_{a(B)}$ be new symbols. Since $G$ is nonerasing and nonpermuting, there exist strings $\delta_j \in (T \cup \{\Omega_0, \ldots, \Omega_{a(B)}\})^*$ such that

$$\alpha\Omega_0\beta_1\Omega_1 \cdots \beta_{a(B)}\Omega_{a(B)}\gamma = \delta_0 x_1 \delta_1 x_2 \cdots x_{a(A)}\delta_{a(A)}.$$

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

$$\begin{array}{rcl} A & \to & f[B] \\ f[(x_1, \ldots, x_{a(B)+1})] & = & (\delta_0\theta, \ldots, \delta_{a(A)}\theta) \end{array}$$

where $\theta(\Omega_i) = x_{i+1}$ $(0 \leq i \leq a(B))$. For example, if there exists a rule

$$A(x_1, x_2, x_3, x_4) \to aB(bx_1c, x_2x_3d)x_4e$$

in $P$, add the following $f$ to $F$ and add $A \to f[B]$ 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 $A(x_1, \ldots, x_{a(A)}) \stackrel{*}{\Rightarrow}_G \alpha_0 x_1 \cdots x_{a(A)}\alpha_{a(A)}$ if and only if $A \stackrel{*}{\Rightarrow}_{G'} (\alpha_0, \ldots, \alpha_{a(A)})$ where $\alpha_i \in T^*$ $(0 \leq i \leq a(A))$.
(1-MCFL$(m+1) \subseteq$ L$^2$-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$^2$-mg$(m)$ $G' = (N, T, P', S)$ in a similar way to the above proof. Let $A \to f[B] \in P$ be an arbitrary rule where

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

Since $G$ is nonerasing and nonpermuting, there exist $\delta_j$ $(0 \leq j \leq \dim(B))$ such that

$$\alpha_1\alpha_2 \cdots \alpha_{\dim(A)} = \delta_0 x_1 \delta_1 \cdots x_{\dim(B)}\delta_{\dim(B)}.$$

Add the following rule to $P'$.

$$A(x_1, \ldots, x_{\dim(A)-1}) = \delta_0 B(\delta_1, \ldots, \delta_{\dim(B)-1})\delta_{\dim(B)}.$$

The formal proof is similar to that of L$^2$-ML$(m) \subseteq$ 1-MCFL$(m+1)$. □

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

**Corollary 18.** Let $G$ be an l$^2$-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 $n = |w|$. □

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

**Lemma 19** (Pumping lemma for 1-MCFL($m$)**).** Let $L$ be a 1-mcfl($m$). Assume that, for a given $n \geq 0$ there exists $\alpha$ 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 $z = u_1 v_1 w_1 s_1 u_2 v_2 w_2 s_2 u_3 \cdots$ $u_m v_m w_m s_m u_{m+1}$ 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 $i \geq 0$, $z_i = u_1 v_1^i w_1 s_1^i u_2 v_2^i w_2 s_2^i u_3 \cdot u_m v_m^i w_m s_m^i u_{m+1} \in L$.

*Proof Sketch.* Let $G = (N, T, F, P, S)$ be a 1-mcfg($m$) that satisfies conditions (N1), (N2) and (N3) in Lemma 1. Let nt($f$) be the number of terminals appearing in the right-hand side of the definition of $f \in F$. For example, if $f[(x_1, x_2)] = (ax_1 b, cx_2 d)$, nt$(f) = 4$. Let 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) $|\alpha| \geq 2^{nt(G)|N|+1}$. Let $t_0$ be a derivation tree of $z_0$. Since $G$ is a 1-mcfg($m$), $t_0$ is just a path. We call a node $v$ *productive* if nt$(f) \geq 1$ for the rule $A \to f[A_1]$ 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. We can find another $z \in L$ and its derivation tree $t$ obtained from $t_0$ by repeating the path between $v'$ and $v$ sufficiently large number of times. For $t$,we can write

$$g_{v',r}(\bar{y}) = u_1 y_{h_1} u_1 y_{h_2} \cdots u_{k-1} y_{h_{\dim(A)}} u_{\dim(A)+1} \tag{5.1}$$

for the root $r$ of $t$. Let $g_j$ denote the $j$th component of $g_{v,v'}$. For each $j$ ($1 \leq j \leq \dim(A)$),

$$\begin{aligned} g_j^i(\bar{y}) &= (\gamma_{j1})^{i-1} g_j(\bar{y}) (\gamma_{j2})^{i-1} \text{ or} \\ &= \gamma_{j1} \quad (\gamma_{j1}, \gamma_{j2} \in T^*). \end{aligned}$$

For the latter case, let $\gamma_{j2} = \varepsilon$ for simplicity. If we let $v$ be the node as close as the root $r$, the number of nodes above $v'$ (not including $v$) in $t$ is at most $|N|$. Thus $|u_1 u_2 \cdots u_{\dim(A)+1}| \leq$ 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, $\sum_{j=1}^{\dim(A)} |v_j s_j| = \sum_{j=1}^{\dim(A)} |\gamma_{j1}\gamma_{j2}| \leq$ nt$(G)|N|p(\dim(A)-1)$. Summarizing, if we let $M =$ 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. □

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

*Proof.* Remember that $G_5$ in Example 2 (2) is a vl-mg($m-1$) such that $L(G_5) = L_2^{(m)}$. Next, we show that $L_2^{(m)} \notin$ L$^2$-ML($m-1$). Suppose $L_2^{(m)} \in$ L$^2$-ML($m-1$) = 1-MCFL($m$). Then $L_2^{(m)}$ satisfies the condition of Lemma 19. Let $M$ be the constant of the lemma and let $z$ be $z = a_1^q \cdots a_{2m}^q a_1^r \cdots a_{2m}^r$ ($q, r > M/2m$). Divide $z$ as

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

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

(1) $v_1 = a_1^j$, $s_1 = a_2^j, \ldots, v_m = a_{2m-1}^j$, $s_m = a_{2m}^j$ $(1 \leq j \leq q)$, or

(2) $v_1 = b_1^j$, $s_1 = b_2^j, \ldots, v_m = b_{2m-1}^j$, $s_m = b_{2m}^j$ $(1 \leq j \leq r)$.

However, neither of (a) and (b) satisfies $\sum_{j=1}^{m} |u_j v_j s_j| + |u_{m+1}| \leq M$. □

# 6 Conclusion

In this paper, we have compared the generative power of multiple context-free grammars with that of variable-linear macro grammars.

These grammars 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 are payed 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 [4, 3]. 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 use grammars of which generative power is greater than cfg. Among them, a subclass of tag is applied to RNA secondary structure prediction [25, 17] and to RNA secondary structure alignment [24]. For the former problem, Rivas and Eddy use RNA pseudoknot grammar [20], and Kato, et al. use a subclass of mcfg [13]. These grammars can be naturally considered as subclasses of mcfgs, and Kato, et al. [12] clarify the relation among the generative power of these grammars.

Meanwhile, Abe and Mamitsuka use a subclass of cftg for protein secondary structure prediction [1]. They define 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 non-permuting conditions in this paper [1]. Let RNRL($m$) and L-RNRL($m$) be the classes of yield

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

languages generated by rnrgs and l-rnrgs with nonterminals of which arity is at most $m$, respectively. They claim that L-RNRL($m$) $\subset$ RNRL($m$), and RNRL($m$) and L-RNRL($m$) can be parsed in $O(n^{3(m+1)})$ and $O(n^{2(m+1)})$, respectively, but formal proofs are not provided. They also present a parsing algorithm for stochastic version of l-rnrg(1) and report the experimental results of applying the algorithm to protein structure prediction. By Lemmas 7 and 8, RNRL($m$) $=$ VL-ML($m$) and L-RNRL($m$) $=$ L$^2$-ML($m$). Thus, the above claims for RNRL and L-RNRL can be obtained as corollaries of Theorem 20 and Corollaries 11 and 18.

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

# References


[1] N. Abe and H. Mamitsuka, "Predicting protein secondary structure using stochastic tree grammars," Machine Learning, 29, pp.275–301, 1997.

[2] A. V. Aho, "Indexed grammars – an extension of the context-free grammars," JACM, 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. Computer & System Sciences, 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," Theory of Computing Systems, 33(1), pp.59–83, 2000.

[8] A. K. Joshi, L. Levy, and M. Takahashi, "Tree adjunct grammars," J. Computer & System Sciences, 10(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., J71-D(5), pp.758–765, 1988 (in Japanese).

[11] T. Kasami, H. Seki, and M. Fujii, “On the membership problem for head languages and multiple context-free languages,” IEICE Trans., J71-D(6), pp.935–941, 1988 (in Japanese).

[12] Y. Kato, H. Seki, and T. Kasami, “On the generative power of grammars for RNA secondary structure,” IEICE Transactions on Information and Systems, E88-D(1), pp.53–64, 2005.

[13] Y. Kato, H. Seki, and T. Kasami, “RNA pseudoknotted structure prediction using stochastic multiple context-free grammar,” IPSJ Transactions on Bioinformatics, 47, SIG 1(BIO 1), 2006, to appear.

[14] S. Kepser and U. Mönnich, “Closure properties of linear context-free tree languages with an application to optimality theory,” Theoretical Computer Science, 354, pp.82–97, 2006.

[15] M. Latteux, “Substitutions dans les EDT0L-systèmes ultralinéaires,” Information & Control, 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, 21(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,” Theoretical Computer Science, 223, pp.87–120, 1999.

[20] E. Rivas and S. Eddy, “The language of RNA: A formal grammar that includes pseudoknots,” Bioinformatics, 16(4), pp.334–340, 2000.

[21] W. C. Rounds, “Mappings and grammars on trees,” Mathematical Systems Theory, 4(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, T. Matsumura, M. Fujii, and T. Kasami, “On multiple context-free grammars,” Theoretical Computer Science, 88, pp.191–229, 1991.

[24] S. Seki and S. Kobayashi, “A grammatical approach to the alignment of structure-annotated strings,” IEICE Transactions on Information and Systems, E88-D(12), pp.2727–2737, 2005.

[25] Y. Uemura, A. Hasegawa, S. Kobayashi, and T. Yokomori, “Tree adjoining grammars for RNA structure prediction,” Theoretical Computer Science, 210, pp.277–303, 1999.

[26] D. J. Weir, “Characterizing mildly context-sensitive grammar formalisms,” Ph.D. Thesis, Department of Computer and Information Science, University of Pennsylvania, 1988.

[27] D. J. Weir, “Linear context-free rewriting systems and deterministic tree-walk transducers,” 30th Meeting of the Assoc. for Computational Linguistics (ACL92), 1992.