The Copying Power of Well-Nested Multiple Context-Free Grammars

Makoto Kanazawa[1] and Sylvain Salvati[2,⋆]

[1] National Institute of Informatics, Tokyo, Japan
[2] INRIA Bordeaux – Sud-Ouest, Talence, France

Abstract. We prove a copying theorem for well-nested multiple context-free languages: if $L = \{ w\#w \mid w \in L_0 \}$ has a well-nested m-MCFG, then L has a 'non-branching' well-nested m-MCFG. This can be used to give simple examples of multiple context-free languages that are not generated by any well-nested MCFGs.

1 Introduction

For a long time, the formalism of *multiple context-free grammars* [18], together with many others equivalent to it, has been regarded as a reasonable formalization of Joshi's [9] notion of *mildly context-sensitive grammars.* Elsewhere [10], we have made a case that a smaller class of grammars, consisting of MCFGs whose rules are *well-nested*, might actually provide a better formal approximation to Joshi's informal concept. Well-nested MCFGs are equivalent to *non-duplicating macro grammars* [5] and to *coupled-context-free grammars* [7]. Kanazawa [11] proves the pumping lemma for well-nested multiple context-free languages. The well-nestedness constraint has also been a focus of attention recently in the area of *dependency grammars* (e.g., [12]).

Seki and Kato [17] present a series of languages that are generated by MCFGs of dimension m, but not by any well-nested MCFGs of the same dimension. These examples illustrate the limiting effect that the well-nestedness constraint has on the class of generated languages at each level m of the infinite hierarchy of m-multiple context-free languages ($m \geq 1$).

An interesting fact is that the examples of Seki and Kato [17] all belong to the class of well-nested MCFLs at some higher level of the hierarchy, so they do not serve to separate the whole class of MCFLs from the whole class of well-nested MCFLs. In fact, to our knowledge, the only example that has appeared in the literature of an MCFL which is not a well-nested MCFL is the language discussed by Michaelis [13], originally due to Staudacher [20]. Staudacher uses Hayashi's [6] theorem to show that this language is not an indexed language, while Michaelis gives a (non-well-nested) 3-MCFG generating it. Since well-nested MCFLs are all indexed languages, it follows that this language is an MCFL which is not

⋆ We are grateful to Uwe Mönnich for pointing us to Engelfriet and Skyum's [4] paper in connection with the question of what languages are in $\mathrm{MCFL} - \mathrm{MCFL}_{\mathrm{wn}}$.

a well-nested MCFL. As it happens, the definition of this language is rather complex, and Staudacher's proof is not easy to understand.

As a matter of fact, what we would like to call the "triple copying theorem" for *OI* (the class of OI macro languages), due to Engelfriet and Skyum [4], can be used to give a simple example of a language that separates MCFL (the class of MCFLs) from $\mathrm{MCFL_{wn}}$ (the class of well-nested MCFLs). This theorem says that $L = \{\, w\#w\#w \mid w \in L_0 \,\} \in \mathrm{OI}$ implies $L_0 \in \mathrm{EDT0L}$.[3] (Here and henceforth, L_0 is a language over some alphabet Σ and # is a symbol not in Σ.) Since OI is the same as the class of indexed languages [5] and includes the class of well-nested MCFLs, and $L = \{\, w\#w\#w \mid w \in L_0 \,\} \in$ 3-MCFL for all $L_0 \in \mathrm{CFL}$, this theorem implies that $L \in \text{3-MCFL} - \mathrm{MCFL_{wn}}$ whenever $L_0 \in \mathrm{CFL} - \mathrm{EDT0L}$. Examples of such L_0 are D_2^*, the one-sided Dyck language over two pairs of parentheses [2, 3] and D_1^*, the one-sided Dyck language over a single pair of parentheses [15]. A question that immediately arises is the status of the "double copying theorem" for OI: when is $L = \{\, w\#w \mid w \in L_0 \,\}$ in OI? We do not yet have an answer to this open question. In this paper, we prove a double copying theorem for well-nested multiple context-free languages, which implies, among other things, that $L = \{\, w\#w \mid w \in L_0 \,\} \in \text{2-MCFL} - \mathrm{MCFL_{wn}}$ for all $L_0 \in \mathrm{CFL} - \mathrm{EDT0L}$. Unlike Staudacher's [20] proof, our proof of this result does not depend on a pumping argument but instead makes use of simple combinatorial properties of strings.

In addition to shedding light on the difference between the class of MCFLs and the class of well-nested MCFLs, the double copying theorem for well-nested MCFLs also highlights a general question underlying Joshi's notion of mild context-sensitivity: what are the limitations found in the kind of cross-serial dependency exhibited in natural language? For, if $\mathcal{L}$ is a family of languages closed under rational transductions, $\{\, w\#w \mid w \in L_0 \,\} \in \mathcal{L}$ implies $\{\, w\,h(w) \mid w \in L_0 \,\} \in \mathcal{L}$ for any homomorphism h, and languages of the latter form, together with some restriction on L_0, may serve as a model of natural language constructions exhibiting cross-serial dependency. This may offer a more fruitful approach than concentrating on languages like $\{\, w\,h(w) \mid w \in \Sigma^* \,\}$, which has been a common practice in the mathematical study of natural language syntax.

2 The Double Copying Theorem for Context-Free Languages

Let us first look at the double copying theorem for context-free languages. This has a rather simple proof, which is omitted here in the interests of space. The implication (i) ⇒ (iii) may be proved using the pumping lemma for context-free languages; it also follows from a closely related result proved by Ito and Katsura [8].

Theorem 1. *Let* $L = \{\, w\#w \mid w \in L_0 \,\}$. *The following are equivalent:*

[3] See [3] for the definition of EDT0L.

(i) *L is a context-free language.*
(ii) *L is a linear context-free language.*
(iii) *L_0 is a finite union of languages of the form rRs, where $r, s \in \Sigma^*$ and R is a regular subset of t^* for some $t \in \Sigma^+$.*

3 Combinatorics on Words

The statement of the double copying theorem for well-nested multiple context-free languages is similar to that for context-free languages, but we do not need to invoke the pumping lemma for well-nested MCFLs in order to prove it. Instead, we rely on some basic results in the combinatorics on words.

A string x is a *conjugate* of a string y if $x = uv$ and $y = vu$ for some u, v. Elements of u^* are called *powers* of u. A nonempty string is *primitive* if it is not a power of another string. For every nonempty string x, there is a unique primitive string u such that x is a power of u; this string u is called the *primitive root* of x. When two nonempty strings are conjugates, their primitive roots are also conjugates.

We use the following basic results from the combinatorics on words (see, e.g., [19]):

Lemma 2. *Let $x, y, z \in \Sigma^+$. Then $xy = yz$ if and only if there exist $u \in \Sigma^+$, $v \in \Sigma^*$, and an integer $k \geq 0$ such that $x = uv$, $z = vu$, and $y = (uv)^k u = u(vu)^k$.*

Lemma 3. *Let $x, y \in \Sigma^+$. The following are equivalent:*

(i) *$xy = yx$.*
(ii) *There exist $z \in \Sigma^+$ and $i, j \geq 1$ such that $x = z^i$ and $y = z^j$.*
(iii) *There exist $i, j \geq 1$ such that $x^i = y^j$.*

4 Multiple Context-Free Grammars

A *ranked alphabet* is a finite set $\Delta = \bigcup_{n \geq 0} \Delta^{(n)}$ such that $\Delta^{(i)} \cap \Delta^{(j)} = \varnothing$ if $i \neq j$. An element d of Δ has *rank* n if $d \in \Delta^{(n)}$. A *tree* over a ranked alphabet Δ is an expression of the form $(dT_1 \dots T_n)$, where $d \in \Delta^{(n)}$ and $T_1, \dots, T_n$ are trees over Δ; the parentheses are omitted when $n = 0$. In writing trees, we adopt the abbreviatory convention of dropping the outermost parentheses.

Let Δ be a ranked alphabet and Σ an unranked alphabet. Let $\mathcal{X}$ be a countably infinite set of *variables* ranging over Σ^*. We use boldface italic letters $\boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{z}_1$, etc., as variables in $\mathcal{X}$. A *rule* over Δ, Σ is an expression of the form

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

where $n \geq 0$, $A \in \Delta^{(q)}$, $B_i \in \Delta^{(q_i)}$, $\boldsymbol{x}_{i,j}$ are pairwise distinct variables, and α_i is a string over $\Sigma \cup \{\, \boldsymbol{x}_{i,j} \mid i \in [1, n], j \in [1, q_i] \,\}$ satisfying the following condition:

– for each i, j, the variable $\boldsymbol{x}_{i,j}$ occurs in $\alpha_1 \dots \alpha_q$ at most once.

Rules with $n = 0$ are called *terminating* and written without the :− symbol. When we deal with rules over Δ, Σ, we view elements of Δ as predicates, and call q the *arity* of A if $A \in \Delta^{(q)}$. Thus, rules are *definite clauses* (in the sense of logic programming) built from strings and predicates on strings.

A *multiple context-free grammar* (MCFG) is a quadruple $G = (N, \Sigma, P, S)$, where N is a ranked alphabet of *nonterminals*, Σ is an unranked alphabet of *terminals*, P is a finite set of rules over N, Σ, and $S \in N^{(1)}$. When $A \in N^{(q)}$ and $w_1, \dots, w_q \in \Sigma^*$, we write $\vdash_G A(w_1, \dots, w_q)$ to mean that $A(w_1, \dots, w_q)$ is derivable using the following inference schema:

$$\frac{\vdash_G B_1(w_{1,1}, \dots, w_{1,q_1}) \quad \dots \quad \vdash_G B_n(w_{n,1}, \dots, w_{n,q_n})}{\vdash_G A(\alpha_1, \dots, \alpha_q)\sigma}$$

where $A(\alpha_1, \dots, \alpha_q) :\!- B_1(\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,q_1}), \dots, B_n(\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,q_n})$ is in P and σ is the substitution mapping each $\boldsymbol{x}_{i,j}$ to $w_{i,j}$. The language of G is defined as $L(G) = \{\, w \in \Sigma^* \mid \vdash_G S(w) \,\}$.

In order to speak of derivation trees of derivable facts, we put the elements of P in one-to-one correspondence with the elements of a ranked alphabet Δ_P, so that a rule $\pi \in P$ with n occurrences of nonterminals on the right-hand side corresponds to a symbol in $\Delta_P^{(n)}$, which we confuse with π itself. In order to refer to contexts in which derivation trees appear, we augment Δ_P with a set $\mathbf{Y}$ of variables ($\mathbf{y}, \mathbf{z}$, etc.), whose rank is always 0. The following inference system associates derivation trees (trees over Δ_P) with derivable facts and derivation tree contexts (trees over $\Delta_P \cup \mathbf{Y}$) with facts derivable from some premises:

$$\frac{}{\mathbf{y} : A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \vdash_G \mathbf{y} : A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)}$$

$$\frac{\Gamma_1 \vdash_G T_1 : B_1(\beta_{1,1}, \dots, \beta_{1,q_1}) \quad \dots \quad \Gamma_n \vdash_G T_n : B_n(\beta_{n,1}, \dots, \beta_{n,q_n})}{\Gamma_1, \dots, \Gamma_n \vdash_G \pi T_1 \dots T_n : A(\alpha_1, \dots, \alpha_q)\sigma}$$

In the second schema, π is the rule

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

and σ is the substitution mapping each $\boldsymbol{x}_{i,j}$ to $\beta_{i,j}$; each Γ_i is a finite sequence of premises of the form $\mathbf{z} : C(\boldsymbol{y}_1, \dots, \boldsymbol{y}_p)$, and it is understood that Γ_i and Γ_j do not share any variables if $i \neq j$. It is clear that $\vdash_G A(w_1, \dots, w_q)$ if and only if $\vdash_G T : A(w_1, \dots, w_q)$ for some tree T over Δ_P. The set $\{\, T \mid \vdash_G T : S(w)$ for some $w \in \Sigma^* \,\}$ is a recognizable set of trees; as a consequence, the Parikh image of $L(G)$ is semilinear [21].

A nonterminal $A \in N^{(q)}$ is *useful* if $\vdash_G A(w_1, \dots, w_q)$ for some $w_1, \dots, w_q$ and $\mathbf{y} : A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \vdash_G T : S(\alpha)$ for some T and α; otherwise it is *useless*.

Example 4. Let G be the MCFG consisting of the following rules:

$$\pi_1 \colon S(\boldsymbol{x}_1 \boldsymbol{y}_1 \mathtt{b\#a} \boldsymbol{y}_2 \boldsymbol{x}_2) :\!- A(\boldsymbol{x}_1, \boldsymbol{x}_2), B(\boldsymbol{y}_1, \boldsymbol{y}_2).$$

$$\pi_2 \colon A(\mathtt{a}, \epsilon). \quad \pi_3 \colon A(\mathtt{ab}\boldsymbol{x}_1\mathtt{ba}, \mathtt{ab}\boldsymbol{x}_2\mathtt{ab}) :\!- A(\boldsymbol{x}_1, \boldsymbol{x}_2).$$

$$\pi_4 \colon B(\epsilon, \mathtt{b}). \quad \pi_5 \colon B(\mathtt{ba}\boldsymbol{y}_1\mathtt{ba}, \mathtt{ba}\boldsymbol{y}_2\mathtt{ab}) :\!- B(\boldsymbol{y}_1, \boldsymbol{y}_2).$$

For example,

$$\vdash_G \pi_1(\pi_3(\pi_3\pi_2))(\pi_5\pi_4) : S(\texttt{ababababababab\#ababababababab}),$$
$$\mathbf{y} : A(\boldsymbol{x}_1, \boldsymbol{x}_2) \vdash_G \pi_1(\pi_3\mathbf{y})(\pi_5\pi_4) : S(\texttt{ab}\boldsymbol{x}_1\texttt{bababab\#abababab}\boldsymbol{x}_2\texttt{ab}),$$

and we have $L(G) = \{\, (\texttt{ab})^n\texttt{\#}(\texttt{ab})^n \mid n \geq 1 \,\}$.

The *dimension* of an MCFG G is the maximal arity of nonterminals of G. The *branching factor* (or *rank*) of G is the maximal number of occurrences of nonterminals on the right-hand side of rules of G. We write m-MCFG(f) for the class of MCFGs whose dimension is at most m and whose branching factor is at most f. (Note that this notation is the opposite of the one used by Seki and Kato [17], but is more consistent with [18].) We write m-MCFG and MCFG for $\bigcup_f m$-MCFG(f) and $\bigcup_m \bigcup_f m$-MCFG(f), respectively. The corresponding classes of languages are denoted by m-MCFL(f), m-MCFL, etc.[4]

An MCFG rule

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

is *non-deleting* if each $\boldsymbol{x}_{i,j}$ occurs in $\alpha_1 \dots \alpha_q$; it is *non-permuting* if $j < k$ implies that the occurrence (if any) of $\boldsymbol{x}_{i,j}$ in $\alpha_1 \dots \alpha_q$ precedes the occurrence (if any) of $\boldsymbol{x}_{i,k}$ in $\alpha_1 \dots \alpha_q$. It is known that every $G \in m$-MCFG(f) has an equivalent $G' \in m$-MCFG(f) whose rules are all non-deleting and non-permuting. A non-deleting and non-permuting rule is *well-nested* if it moreover satisfies the following condition:

– if $i \neq i'$, $j < q_i$, and $j' < q_{i'}$, then $\alpha_1 \dots \alpha_q \notin (\Sigma \cup \mathcal{X})^* \boldsymbol{x}_{i,j} (\Sigma \cup \mathcal{X})^* \boldsymbol{x}_{i',j'} (\Sigma \cup \mathcal{X})^* \boldsymbol{x}_{i,j+1} (\Sigma \cup \mathcal{X})^* \boldsymbol{x}_{i',j'+1} (\Sigma \cup \mathcal{X})^*$.

In other words, if $\boldsymbol{x}_{i',j'}$ occurs between $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{i,j+1}$ in $\alpha_1 \dots \alpha_q$, then $\boldsymbol{x}_{i',1}, \dots, \boldsymbol{x}_{i',q_{i'}}$ must all occur between $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{i,j+1}$.

We attach the subscript "wn" to "MCFG" and "MCFL" to denote classes of well-nested MCFGs and corresponding classes of languages, as in m-MCFG$_{\mathrm{wn}}(f)$, m-MCFG$_{\mathrm{wn}}$, m-MCFL$_{\mathrm{wn}}(f)$, m-MCFL$_{\mathrm{wn}}$, etc. The grammar in Example 4 belongs to 2-MCFG$_{\mathrm{wn}}(2)$. Note that m-MCFL$(1) = m$-MCFL$_{\mathrm{wn}}(1)$.

Lemma 5. *For each $m \geq 1$, m-MCFL$_{wn}$ = m-MCFL$_{wn}(2)$.*

Proof (sketch). A well-nested rule

$$\pi = A(\alpha_1, \dots, \alpha_q) \mathrel{:-} B_1(\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,q_1}), \dots, B_n(\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,q_n})$$

with $n \geq 3$ can always be replaced by two rules whose right-hand side has at most $n-1$ nonterminals, as follows. The replacement introduces one new nonterminal C, whose arity does not exceed $\max\{q, q_1, \dots, q_n\}$. We assume without loss of generality that $1 \leq i < j \leq n$ implies that $\boldsymbol{x}_{i,1}$ occurs to the left of $\boldsymbol{x}_{j,1}$ in

[4] See [14] and [16] for relations among the classes m-MCFL(f) with different values of m and f.

$\alpha_1 \dots \alpha_q$. Since π is well-nested, there must be an $l \in [1,n]$ such that $\alpha_1 \dots \alpha_q \in (\Sigma \cup \mathcal{X})^* \boldsymbol{x}_{l,1} \Sigma^* \boldsymbol{x}_{l,2} \Sigma^* \dots \Sigma^* \boldsymbol{x}_{l,q_l} (\Sigma \cup \mathcal{X})^*$. Let i,j be such that $\alpha_i \in (\Sigma \cup \mathcal{X})^* \boldsymbol{x}_{l,1} (\Sigma \cup \mathcal{X})^*$ and $\alpha_j \in (\Sigma \cup \mathcal{X})^* \boldsymbol{x}_{l,q_l} (\Sigma \cup \mathcal{X})^*$.

Case 1. $i < j$. We can write $\alpha_i = \beta_1 \boldsymbol{x}_{l,1} \beta_2$ and $\alpha_j = \gamma_1 \boldsymbol{x}_{l,q_l} \gamma_2$. Let C be a new nonterminal of arity $q' = i + q - j + 1 \leq q$. We can replace π with the following two rules:

$$\begin{aligned}
&B(\boldsymbol{y}_1, \dots, \boldsymbol{y}_{i-1}, \boldsymbol{y}_i \boldsymbol{x}_{l,1} \beta_2, \alpha_{i+1}, \dots, \alpha_{j-1}, \gamma_1 \boldsymbol{x}_{l,q_l} \boldsymbol{y}_j, \boldsymbol{y}_{j+1}, \dots, \boldsymbol{y}_q) \text{ :- } \\
&\qquad C(\boldsymbol{y}_1, \dots, \boldsymbol{y}_i, \boldsymbol{y}_j, \dots, \boldsymbol{y}_q), B_l(\boldsymbol{x}_{l,1}, \dots, \boldsymbol{x}_{l,q_l}). \\
&C(\alpha_1, \dots, \alpha_{i-1}, \beta_1, \gamma_2, \alpha_{j+1}, \dots, \alpha_q) \text{ :- } \\
&\qquad B_1(\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,q_1}), \dots, B_{l-1}(\boldsymbol{x}_{l-1,1}, \dots, \boldsymbol{x}_{l-1,q_{l-1}}), \\
&\qquad B_{l+1}(\boldsymbol{x}_{l+1,1}, \dots, \boldsymbol{x}_{l+1,q_{l+1}}), \dots, B_n(\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,q_n}).
\end{aligned}$$

Case 2. $i = j$. We can write $\alpha_i = \beta_1 \boldsymbol{x}_{l,1} \beta_2 \boldsymbol{x}_{l,q_l} \beta_3$.

Case 2a. $\beta_1 \beta_3 \in \Sigma^*$. Let C be a new nonterminal of arity $q-1$. We can replace π with the following two rules:

$$\begin{aligned}
&A(\boldsymbol{y}_1, \dots, \boldsymbol{y}_{i-1}, \alpha_i, \boldsymbol{y}_{i+1}, \dots, \boldsymbol{y}_q) \text{ :- } \\
&\qquad C(\boldsymbol{y}_1, \dots, \boldsymbol{y}_{i-1}, \boldsymbol{y}_{i+1}, \dots, \boldsymbol{y}_q), B_l(\boldsymbol{x}_{l,1}, \dots, \boldsymbol{x}_{l,q_l}). \\
&C(\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_q) \text{ :- } \\
&\qquad B_1(\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,q_1}), \dots, B_{l-1}(\boldsymbol{x}_{l-1,1}, \dots, \boldsymbol{x}_{l-1,q_{l-1}}), \\
&\qquad B_{l+1}(\boldsymbol{x}_{l+1,1}, \dots, \boldsymbol{x}_{l+1,q_{l+1}}), \dots, B_n(\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,q_n}).
\end{aligned}$$

Case 2b. $\beta_1 = \gamma \boldsymbol{x}_{k,p} w$ with $w \in \Sigma^*$. Let C be a new nonterminal of arity q_k. We can replace π with the following two rules:

$$\begin{aligned}
&A(\alpha_1, \dots, \alpha_{i-1}, \gamma \boldsymbol{x}_{k,p} \beta_3, \alpha_{p+1}, \dots, \alpha_q) \text{ :- } \\
&\qquad B_1(\boldsymbol{x}_{1,1}, \dots, \boldsymbol{x}_{1,q_1}), \dots, B_{k-1}(\boldsymbol{x}_{k-1,1}, \dots, \boldsymbol{x}_{k-1,q_{k-1}}), C(\boldsymbol{x}_{k,1}, \dots, \boldsymbol{x}_{k,q_k}), \\
&\qquad B_{k+1}(\boldsymbol{x}_{k+1,1}, \dots, \boldsymbol{x}_{k+1,q_{k+1}}), \dots, B_{l-1}(\boldsymbol{x}_{l-1,1}, \dots, \boldsymbol{x}_{l-1,q_{l-1}}), \\
&\qquad B_{l+1}(\boldsymbol{x}_{l+1,1}, \dots, \boldsymbol{x}_{l+1,q_{l+1}}), \dots, B_n(\boldsymbol{x}_{n,1}, \dots, \boldsymbol{x}_{n,q_n}). \\
&C(\boldsymbol{x}_{k,1}, \dots, \boldsymbol{x}_{k,p-1}, \boldsymbol{x}_{k,p} w \boldsymbol{x}_{l,1} \beta_2 \boldsymbol{x}_{l,q_l}, \boldsymbol{x}_{k,p+1}, \dots, \boldsymbol{x}_{k,q_k}) \text{ :- } \\
&\qquad B_k(\boldsymbol{x}_{k,1}, \dots, \boldsymbol{x}_{k,q_k}), B_l(\boldsymbol{x}_{l,1}, \dots, \boldsymbol{x}_{q_l}).
\end{aligned}$$

Case 2c. $\beta_3 = w \boldsymbol{x}_{k,p} \gamma$ with $w \in \Sigma^*$. Similar to Case 2b. □

Seki and Kato [17] show that for all $m \geq 2$, $\mathrm{RESP}_m \in m\text{-MCFL}(2) - m\text{-MCFL}_{\mathrm{wn}}$, where RESP_m is defined by

$$\mathrm{RESP}_m = \{\, \mathtt{a}_1^i \mathtt{a}_2^i \mathtt{b}_1^j \mathtt{b}_2^j \dots \mathtt{a}_{2m-1}^i \mathtt{a}_{2m}^i \mathtt{b}_{2m-1}^j \mathtt{b}_{2m}^j \mid i,j \geq 0 \,\}.$$

It is easy to see that $\mathrm{RESP}_m \in 2m\text{-MCFL}(1) = 2m\text{-MCFL}_{\mathrm{wn}}(1)$.

5 The Double Copying Theorem for Well-Nested Multiple Context-Free Languages

The following theorem about possibly non-well-nested MCFGs is easy to prove. For part (ii), note that there is a rational transduction that maps L to L_0.[5]

Theorem 6. *Let* $L = \{\, w\#w \mid w \in L_0 \,\}$.

(i) *If* $L_0 \in m$-*MCFL*(f), *then* $L \in 2m$-*MCFL*(f).
(ii) *If* $L \in m$-*MCFL*(f), *then* $L_0 \in m$-*MCFL*(f).

A consequence of Theorem 6 is that the class of all MCFGs has an unlimited copying power in the sense that $L = \{\, w\#w \mid w \in L_0 \,\}$ is an MCFL whenever L_0 is. We will see that the copying power of well-nested MCFGs is much more restricted (Corollary 9).

The following lemma is used in the proof of our main theorem (Theorem 8). Its proof is straightforward and is omitted.

Lemma 7. *Let M be a semilinear subset of $\mathbb{N}^{2m}$ and $r_i, s_i, t_i, u_i, v_i \in \Sigma^*$ for $i \in [1, m]$. Then there are some $G = (N, \Sigma, P, S) \in m$-MCFG$(1)$ and nonterminal $A \in N^{(m)}$ such that*

$$\{\, (x_1, \ldots, x_m) \mid \vdash_G A(x_1, \ldots, x_m) \,\} = \\ \{\, (r_1 s_1^{n_1} t_1 u_1^{n_2} v_1, \ldots, r_m s_m^{n_{2m-1}} t_m u_m^{n_{2m}} v_m) \mid (n_1, \ldots, n_{2m}) \in M \,\}.$$

Theorem 8. *Let* $L = \{\, w\#w \mid w \in L_0 \,\}$. *The following are equivalent:*

(i) $L \in m$-*MCFL*$_{wn}$.
(ii) $L \in m$-*MCFL*(1).

Proof. The implication from (ii) to (i) immediately follows from m-MCFL$(1) = m$-MCFL$_{\mathrm{wn}}(1)$. To show that (i) implies (ii), suppose that $L = L(G)$ for some $G = (N, \Sigma \cup \{\#\}, P, S) \in m$-MCFG$_{\mathrm{wn}}(2)$. If L is finite, L clearly belongs to 1-MCFL(1), so we assume that L is infinite. Without loss of generality, we may suppose that G has no useless nonterminal and satisfies the following property:

– For each nonterminal $A \in N^{(q)}$, the set $\{\, (x_1, \ldots, x_q) \mid \vdash_G A(x_1, \ldots, x_q) \,\}$ is infinite.

To show that L belongs to m-MCFL(1), we prove that for each binary rule

$$\pi = A(\alpha_1, \ldots, \alpha_q) \mathrel{:-} B(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k), C(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_l)$$

in P, there are $G_\pi = (N_\pi, \Sigma \cup \{\#\}, P_\pi, S_\pi) \in m$-MCFG(1) and a nonterminal $A_\pi \in N_\pi^{(q)}$ such that

$$\{\, (x_1, \ldots, x_q) \mid \vdash_G \pi T_1 T_2 : A(x_1, \ldots, x_q) \text{ for some derivation trees } T_1, T_2 \,\} \\ = \{\, (x_1, \ldots, x_q) \mid \vdash_{G_\pi} A_\pi(x_1, \ldots, x_q) \,\}. \quad (1)$$

This is a consequence of the following claim. We assume without loss of generality that $\boldsymbol{y}_1$ occurs to the left of $\boldsymbol{z}_1$ in $(\alpha_1, \ldots, \alpha_q)$.

[5] See the discussion following the proof of Theorem 8 for a possible strengthening of part (ii) of Theorem 6.

Claim. There exist $t \in \Sigma^+$ and $r, s \in \Sigma^*$ such that if

$$\vdash_G \pi T_1 T_2 : A(x_1, \dots, x_q)$$

for some T_1, T_2, then $x_1, \dots, x_q$ are non-overlapping substrings of $rt^i s\#rt^i s$ for some $i \geq 0$.

Proof. We write $\Sigma_\#$ for $\Sigma \cup \{\#\}$. Let $U[\mathbf{x}]$ be a (smallest, for concreteness) derivation tree context such that for some $\gamma \in \Sigma_\#^* \boldsymbol{x}_1 \Sigma_\#^* \dots \Sigma_\#^* \boldsymbol{x}_q \Sigma_\#^*$,

$$\mathbf{x} : A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \vdash_G U[\mathbf{x}] : S(\gamma).$$

We write $\gamma[\vec{\beta}]$ for $\gamma[\boldsymbol{x}_1 := \beta_1, \dots, \boldsymbol{x}_q := \beta_q]$. Our goal is to find $t \in \Sigma^+$ and $r, s \in \Sigma^*$ such that

$$\vdash_G \pi T_1 T_2 : A(x_1, \dots, x_q) \text{ implies } \gamma[\vec{x}] = rt^i s\#rt^i s \text{ for some } i \geq 0. \tag{2}$$

We have

$$\mathbf{y} : B(\boldsymbol{y}_1, \dots, \boldsymbol{y}_k), \mathbf{z} : C(\boldsymbol{z}_1, \dots, \boldsymbol{z}_l) \vdash_G U[\pi \mathbf{y}\mathbf{z}] : S(\gamma[\vec{\alpha}]).$$

Let us write $\gamma[\vec{\alpha}][\vec{y}, \vec{z}]$ for the result of substituting $y_1, \dots, y_k, z_1, \dots, z_l$ for $\boldsymbol{y}_1, \dots, \boldsymbol{y}_k, \boldsymbol{z}_1, \dots, \boldsymbol{z}_l$ in $\gamma[\vec{\alpha}]$. Since π is well-nested, either

$$\gamma[\vec{\alpha}] \in \Sigma_\#^* \boldsymbol{y}_1 \Sigma_\#^* \dots \Sigma_\#^* \boldsymbol{y}_k \Sigma_\#^* \boldsymbol{z}_1 \Sigma_\#^* \dots \Sigma_\#^* \boldsymbol{z}_l \Sigma_\#^*$$

or else

$$\gamma[\vec{\alpha}] \in \Sigma_\#^* \boldsymbol{y}_1 \Sigma_\#^* \dots \Sigma_\#^* \boldsymbol{y}_h \Sigma_\#^* \boldsymbol{z}_1 \Sigma_\#^* \dots \Sigma_\#^* \boldsymbol{z}_l \Sigma_\#^* \boldsymbol{y}_{h+1} \Sigma_\#^* \dots \Sigma_\#^* \boldsymbol{y}_k \Sigma_\#^*$$

for some $h \in [1, k-1]$. Since $\gamma[\vec{\alpha}][\vec{y}, \vec{z}] \in L$ for all $y_1, \dots, y_k, z_1, \dots, z_l$ such that $\vdash_G B(y_1, \dots, y_k)$ and $\vdash_G C(z_1, \dots, z_l)$, and $y_1, \dots, y_k$ and $z_1, \dots, z_l$ can vary independently, it is easy to see that the former possibility is ruled out; thus we must have

$$\gamma[\vec{\alpha}] = \delta_1 \delta_2 \delta_3$$

where

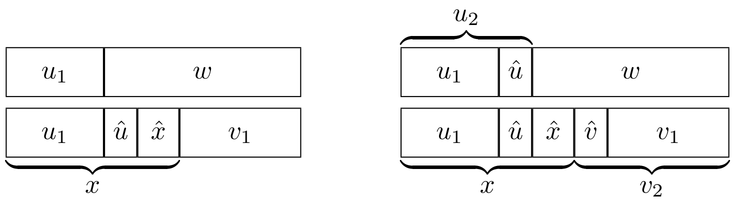
$$\delta_1 \in \Sigma^* \boldsymbol{y}_1 \Sigma^* \dots \Sigma^* \boldsymbol{y}_h \Sigma^*, \quad \delta_2 \in \boldsymbol{z}_1 \Sigma_\#^* \dots \Sigma_\#^* \boldsymbol{z}_l, \quad \delta_3 \in \Sigma^* \boldsymbol{y}_{h+1} \Sigma^* \dots \Sigma^* \boldsymbol{y}_k \Sigma^*.$$

Let

$$L_B = \{\, \delta_1 \# \delta_3[\vec{y}] \mid \vdash_G B(y_1, \dots, y_k) \,\} \quad \text{and} \quad L_C = \{\, \delta_2[\vec{z}] \mid \vdash_G C(z_1, \dots, z_l) \,\}.$$

Note that both L_B and L_C are infinite subsets of $\Sigma^* \# \Sigma^*$, and for every $u\#v \in L_B$ and $w\#x \in L_C$, the string $uw\#xv$ is an element of L. Let $u\#v, u'\#v' \in L_B$ with $|u| \leq |u'|$. By taking $w\#x \in L_C$ with $|w| \geq |v'|$ (or equivalently, $|x| \geq |u'|$), we see that u must be a prefix of u', since both are prefixes of x. We also see that $|u'| - |u| = |v'| - |v|$. By the same token, v must be a suffix of v'.

Let $u_1\#v_1$ and $u_2\#v_2$ be the two shortest strings in L_B. Then $u_2 = u_1\hat{u}$ and $v_2 = \hat{v}v_1$ for some $\hat{u}, \hat{v} \in \Sigma^+$ such that $|\hat{u}| = |\hat{v}|$. Let $w\#x \in L_C$, and suppose $|x| > |u_2|$.



From $u_1 w = x v_1$ and $u_2 w = x v_2$, we see that there is an $\hat{x} \in \Sigma^+$ such that

$$x = u_2\hat{x}, \quad w = \hat{x}v_2, \quad \text{and} \quad \hat{u}\hat{x} = \hat{x}\hat{v}.$$

By Lemma 2, there are $\hat{u}_1 \in \Sigma^+, \hat{u}_2 \in \Sigma^*$ such that

$$\hat{u} = \hat{u}_1\hat{u}_2, \quad \hat{v} = \hat{u}_2\hat{u}_1, \quad \text{and} \quad \hat{x} = \hat{u}^k\hat{u}_1 = \hat{u}_1(\hat{u}_2\hat{u}_1)^k \quad \text{for some } k \geq 0.$$

Now let t be the primitive root of $\hat{u}$. There are some $i_1, i_2 \geq 0$ and t_1, t_2 such that $t_1 \neq \epsilon$ and

$$t = t_1 t_2, \quad \hat{u}_1 = t^{i_1} t_1, \quad \hat{u}_2 = t_2 t^{i_2}.$$

Then

$$\hat{u}\hat{x} = \hat{x}\hat{v} = \hat{u}^{k+1}\hat{u}_1 \in t^* t_1.$$

It follows that for all $w\#x \in L_C$ such that $|x| > |u_2|$,

$$w \in t^* t_1 v_1, \tag{3}$$
$$x \in u_1 t^* t_1. \tag{4}$$

Now let $u\#v$ be an arbitrary element of L_B. Take $w\#x \in L_C$ such that $|x| > |u_2|$ and $|w| \geq |t| + |v|$. Since $uw = xv$, there is an x' such that $|x'| \geq |t|$ and

$$w = x'v, \tag{5}$$
$$x = ux'. \tag{6}$$

Since $|v| \geq |v_1|$ and $|x'| \geq |t|$, (3) and (5) implies

$$x' = t_1(t_2 t_1)^j t_3$$

for some $j \geq 0$ and some prefix t_3 of $t_2 t_1$ such that $t_3 \neq t_2 t_1$. Let t_4 be such that $t_3 t_4 = t_2 t_1$. Since (4) and (6) imply that x' ends in $t_2 t_1$, we see

$$t_4 t_3 = t_3 t_4.$$

Since $t_3 t_4 = t_2 t_1$ is a conjugate of t and hence is primitive, Lemma 3 implies that $t_3 = \epsilon$. Hence $x' \in t^* t_1$. By (4) and (6), we see

$$u \in u_1 t^*. \tag{7}$$

By a reasoning symmetric to that leading up to (7), we can infer that there exist some primitive non-empty string $\tilde{t}$ and some string w_1 such that for all $w\texttt{\#}x \in L_C$,

$$w \in \tilde{t}^* w_1. \tag{8}$$

By taking sufficiently long w, (3) and (8) together imply

$$t^{|\tilde{t}|} = \tilde{t}^{|t|}.$$

Since t and $\tilde{t}$ are both primitive, Lemma 3 implies $t = \tilde{t}$. Thus, for all $w\texttt{\#}x \in L_C$,

$$w \in t^* w_1. \tag{9}$$

From (7) and (9), we obtain

$$uw \in u_1 t^* w_1$$

for all $u\texttt{\#}v \in L_B$ and all $w\texttt{\#}x \in L_C$. Now (2) follows with $r = u_1$ and $s = w_1$. $\square$

We continue with the proof of Theorem 8. Let $c = \max\{|r|, |s|, |t|\}$. By the above claim, one of the following two cases must obtain.

Case 1. Every $(x_1, \dots, x_q)$ such that $\vdash_G \pi T_1 T_2 : A(x_1, \dots, x_q)$ for some T_1, T_2 is of the form

$$(r_1 t^{n_1} s_1, \dots, r_q t^{n_q} s_q).$$

for some $r_1, \dots, r_q, s_1, \dots, s_q \in \Sigma^{\leq c}$.

Case 2. Every $(x_1, \dots, x_q)$ such that $\vdash_G \pi T_1 T_2 : A(x_1, \dots, x_q)$ for some T_1, T_2 is of the form

$$\begin{aligned}(r_1 t^{n_1} s_1, \dots, r_{j-1} t^{n_{j-1}} s_{j-1}, r_j t^{n_j} s_j \texttt{\#} r_{j+1} t^{n_{j+1}} s_{j+1},\\ r_{j+2} t^{n_{j+2}} s_{j+2}, \dots, r_{q+1} t^{n_{q+1}} s_{q+1})\end{aligned}$$

for some $r_1, \dots, r_{q+1}, s_1, \dots, s_{q+1} \in \Sigma^{\leq c}$.

In Case 1, for any fixed $r_1, \dots, r_q, s_1, \dots, s_q$, the set

$$\{ (n_1, \dots, n_q) \mid \vdash_G \pi T_1 T_2 : A(r_1 t^{n_1} s_1, \dots, r_q t^{n_q} s_q) \text{ for some } T_1, T_2 \}$$

is semilinear. To see this, it suffices to note that $L_\pi = \{ x_1 \texttt{\$} \dots \texttt{\$} x_q \mid \vdash_G \pi T_1 T_2 : A(x_1, \dots, x_q) \text{ for some } T_1, T_2 \}$ is an m-MCFL and there is a rational transduction that relates $r_1 t^{n_1} s_1 \texttt{\$} \dots \texttt{\$} r_q t^{n_q} s_q$ to $\mathtt{a}_1^{n_1} \dots \mathtt{a}_q^{n_q}$. Thus, by Lemma 7, there are a $G_\pi = (N_\pi, \Sigma \cup \{\texttt{\#}\}, P_\pi, S_\pi) \in q\text{-MCFG}(1)$ and a nonterminal $A_\pi \in N_\pi^{(q)}$ such that (1) holds.

In Case 2, we can derive the same conclusion in a similar way.

Let P_2 be the set of all binary rules of G. We can now form a $G' = (N', \Sigma \cup \{\texttt{\#}\}, P', S) \in m\text{-MCFG}(1)$ generating L by setting

$$N' = N \cup \bigcup_{\pi \in P_2} N_\pi,$$

$$\begin{aligned}P' = (P - P_2) \cup \bigcup_{\pi \in P_2} P_\pi \cup \{ A(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \text{ :- } A_\pi(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \mid \pi \in P_2 \\ \text{and } A \in N^{(q)} \text{ is the head nonterminal of } \pi \}.\end{aligned}$$

This completes the proof of Theorem 8. $\square$

It would be desirable to have a precise characterization of the class of languages L_0 for which $L = \{ w\#w \mid w \in L_0 \}$ belongs to m-MCFL$_{\mathrm{wn}}$, as in the double copying theorem for context-free languages (Theorem 1). In a previous version of the paper, we hastily stated that $L \in m$-MCFL(f) implies $L_0 \in \lceil m/2 \rceil$-MCFL$(f)$ (compare part (ii) of Theorem 6), which would give us such a characterization for even m. While this still seems to us to be a reasonable conjecture, we currently see no easy way to prove it.

Since it is easy to see[6]

$$m\text{-MCFL}(1) = \text{EDT0L}_{\text{FIN}(m)},$$

Theorems 6 and 8 give

Corollary 9. *Let $L = \{ w\#w \mid w \in L_0 \}$. The following are equivalent:*

(i) $L \in \mathit{MCFL}_{wn}$.
(ii) $L \in \mathit{EDT0L}_{FIN}$.
(iii) $L_0 \in \mathit{EDT0L}_{FIN}$.

Since CFL $-$ EDT0L $\neq \varnothing$ and $\{ w\#w \mid w \in L_0 \} \in$ 2-MCFL for all $L_0 \in$ CFL $-$ EDT0L, Corollary 9 implies

Corollary 10. $2\text{-}\mathit{MCFL} - \mathit{MCFL}_{wn} \neq \varnothing$.

6 Conclusion

We have shown that imposing the well-nestedness constraint on the rules of multiple context-free grammars causes severe loss of the copying power of the formalism. The restriction on the languages L_0 that can be copied is similar to the restriction in Engelfriet and Skyum's [4] triple copying theorem for OI. It is worth noting that the crucial claim in the proof of Theorem 8 does not depend on the non-duplicating nature of the MCFG rules, and one can indeed prove that an analogous claim also holds of OI. This leads us to conjecture that a double copying theorem holds of OI with the same restriction on L_0 as in Engelfriet and Skyum's triple copying theorem (namely, membership in EDT0L).[7] We hope to resolve this open question in future work.

References

1. Arnold, A., Dauchet, M.: Un théorem de duplication pour les forêts algébriques. Journal of Computer and System Science 13, 223–244 (1976)
2. Ehrenfeucht, A., Rozenberg, G.: On some context-free languages that are not deterministic ET0L languages. R.A.I.R.O. Informatique théorique/Theoretical Computer Science 11, 273–291 (1977)

[6] See [3] for the definition of EDT0L$_{\text{FIN}(m)}$ and EDT0L$_{\text{FIN}}$.
[7] Arnold and Dauchet [1] prove a copying theorem for *OI context-free tree languages*, which is an exact tree counterpart to this conjecture.

3. Engelfriet, J., Rozenberg, G., Slutzki, G.: Tree transducers, L systems, and two-way machines. Journal of Computer and System Sciences 20, 150–202 (1980)
4. Engelfriet, J., Skyum, S.: Copying theorems. Information Processing Letters 4, 157–161 (1976)
5. Fisher, M.J.: Grammars with Macro-Like Productions. Ph.D. thesis, Harvard University (1968)
6. Hayashi, T.: On derivation trees of indexed gramamrs —an extension of the uvwxy-theorem—. Publications of the Research Institute for Mathematical Sciences 9, 61–92 (1973)
7. Hotz, G., Pitsch, G.: On parsing coupled-context-free languages. Thoretical Computer Science 161, 205–253 (1996)
8. Ito, M., Katsura, M.: Context-free languages consisting of non-primitive words. International Journal of Computer Mathematics 40, 157–167 (1991)
9. Joshi, A.K.: Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In: Dowty, D.R., Karttunen, L., Zwicky, A.M. (eds.) Natural Language Parsing: Psychological, Computational and Theoretical Perspectives, pp. 206–250. Cambridge University Press, Cambridge (1985)
10. Kanazawa, M.: The convergence of well-nested mildly context-sensitive grammar formalisms (July 2009), an invited talk given at the 14th Conference on Formal Grammar, Bordeaux, France. Slides available at `http://research.nii.ac.jp/~kanazawa/`.
11. Kanazawa, M.: The pumping lemma for well-nested multiple context-free languages. In: Diekert, V., Nowotka, D. (eds.) Developments in Language Theory: 13th International Conference, DLT 2009. pp. 312–325. Springer, Berlin (2009)
12. Kuhlmann, M.: Dependency Structures and Lexicalized Grammars. Ph.D. thesis, Saarland University (2007)
13. Michaelis, J.: An additional observation on strict derivational minimalism. In: Rogers, J. (ed.) Proceedings of FG-MoL 2005: The 10th conference on Formal Grammar and the 9th Meeting on Mathematics of Language. pp. 101–111. CSLI Publications, Stanford, CA (2009)
14. Rambow, O., Satta, G.: Independent parallelism in finite copying parallel rewriting systems. Theoretical Computer Science 223, 87–120 (1999)
15. Rozoy, B.: The Dyck language ${D'_1}^*$ is not generated by any matrix grammar of finite index. Information and Computation 74, 64–89 (1987)
16. Satta, G.: Trading independent for synchronized parallelism in finite copying parallel rewriting systems. Journal of Computer and System Sciences 56, 27–45 (1998)
17. Seki, H., Kato, Y.: On the generative power of multiple context-free grammars and macro grammars. IEICE Transactions on Information and Systems E91–D, 209–221 (2008)
18. Seki, H., Matsumura, T., Fujii, M., Kasami, T.: On multiple context-free grammars. Theoretical Computer Science 88, 191–229 (1991)
19. Shallit, J.: A Second Course in Formal Langauges and Automata Theory. Cambridge University Press, Cambridge (2009)
20. Staudacher, P.: New frontiers beyond context-freeness: DI-grammars and DI-automata. In: 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL '93). pp. 358–367 (1993)
21. Vijay-Shanker, K., Weir, D.J., Joshi, A.K.: Characterizing structural descriptions produced by various grammatical formalisms. In: 25th Annual Meeting of the Association for Computational Linguistics. pp. 104–111 (1987)