Proceedings of the Society for Computation in Linguistics

Volume 2 Article 14

2019

Learnability and Overgeneration in Computational Syntax

Yiding Hao
Yale University, yiding.hao@yale.edu

Learnability and Overgeneration in Computational Syntax

Yiding Hao
Department of Linguistics
Department of Computer Science
Yale University
yiding.hao@yale.edu

Abstract

This paper addresses the hypothesis that unnatural patterns generated by grammar formalisms can be eliminated on the grounds that they are unlearnable. I consider three examples of formal languages thought to represent dependencies unattested in natural language syntax, and show that all three can be learned by grammar induction algorithms following the Distributional Learning paradigm of Clark and Eyraud (2007). While learnable language classes are restrictive by necessity (Gold, 1967), these facts suggest that learnability alone may be insufficient for addressing concerns of overgeneration in syntax.

1 Introduction

A longstanding debate in linguistics concerns the division of labor in language acquisition between innate universal assumptions about natural language and the learner's ability to recognize patterns in data. The *rationalist* position, famously championed by the Principles and Parameters framework, follows the Poverty of the Stimulus argument (POS, Chomsky, 1965, 1968, 1971, 1980) in assuming a rich Universal Grammar (UG) that allows individual languages to vary along a narrow range of dimensions. On the other hand, the Distributional Learning paradigm of grammatical inference (Clark and Eyraud, 2007) has shown that it is possible to create *empiricist* representations of grammar optimized for extracting generalizations from data with theoretical guarantees of convergence.

Recent advances in mathematical linguistics have suggested that the rationalist–empiricist debate may be of interest to the program of formally characterizing the typology of syntax. Shieber's (1985) argument that Swiss German is not context-free shows that substantial expressive power is needed in order to adequately describe syntactic phenomena. At the same time, the class of context-free languages and its extensions include pathological dependencies unattested in natural language. Kobele (2011), for instance, shows that the context-free Merge operation allows Minimalist Grammars (MGs) to define languages that require every syntactically well-formed sentence to have at least one semantic type conflict. In light of this overgeneration problem, learnability has been proposed as a possible way to refine existing language classes so as to better align with empirical facts. Under such an approach, UG specifies the formalism in which grammars are represented, while language acquisition is modelled by a grammar induction algorithm that correctly converges on a subset of the possible grammars. Since no strict superclass of the finite languages admits a general learning procedure (Gold, 1967), there necessarily exist languages that are permitted by UG but that cannot be learned by the language acquisition algorithm.

This paper takes some preliminary steps toward evaluating the potential of learnability to produce restricted language classes that exclude unnatural patterns. Recent work in Distributional Learning has produced a hierarchy of context-free and multiple context-free language classes defined by learning algorithms. I examine three examples of unnatural patterns—structure-independent constraints on sentence length, free word order with unbounded crossing dependencies, and unlimited copying of deep context-free structure—and show that these patterns appear in small classes of the learnable hierarchy. This suggests that current approaches to grammar induction for syntax may fail to yield learnability-based accounts for the absence of these patterns in syntactic typology.

After basic definitions and notation are presented in Section 2, Section 3 introduces the learnable language classes considered in this paper.

Proceedings of the Society for Computation in Linguistics (SCiL) 2019, pages 124-134.
New York City, New York, January 3-6, 2019

The three unnatural patterns, drawn from Graf's (2013) discussion of overgeneration in MGs, are defined in Section 4. There, it will be shown that the three patterns exist within the language classes from Section 3. Section 5 concludes with a discussion of these facts and their relationship with the rationalist–empiricist debate.

2 Preliminaries

As usual, $\mathbb{N}$ denotes the set of nonnegative integers, and for any set A, $\mathcal{P}(A)$ denotes the power set of A. Unless otherwise specified, the letter Σ denotes a finite alphabet. The length of a string x is denoted by $|x|$, and ε denotes the empty string. For each $a \in \Sigma$, $|x|_a$ denotes the number of occurrences of a in x. Alphabet symbols are identified with strings of length 1. For strings a and b, ab denotes the concatenation of a and b. As usual, this notation is extended elementwise to sets of strings. For $k \in \mathbb{N}$, α^k denotes α concatenated with itself k-many times; $\alpha^{\leq k}$ denotes $\bigcup_{i=0}^{k} \alpha^i$; α^* denotes $\bigcup_{i=0}^{\infty} \alpha^i$; and α^+ denotes $\alpha^* \backslash \{\varepsilon\}$. This notation does not apply to $(\Sigma^*)^k$, which denotes the cartesian product $\prod_{i=1}^{k} \Sigma^*$. The *length of a tuple* $\mathbf{x} = \langle x_1, x_2, \ldots, x_k \rangle$ is defined as $|\mathbf{x}| := \sum_{i=1}^{k} |x_i|$. For $a \in \Sigma$, $|\mathbf{x}|_a$ denotes $|x_1 x_2 \ldots x_k|_a$.

For $k \in \mathbb{N}$, a *k-context over* Σ is a $(k+1)$-tuple of strings $\langle c_0, c_1, \ldots, c_k \rangle \in (\Sigma^*)^k$, denoted $c_0 \square c_1 \square \ldots \square c_k$. For sets $L_1, L_2, \ldots, L_k \subseteq \Sigma^*$, $L_0 \square L_1 \square \ldots \square L_k$ denotes the cartesian product $\prod_{i=0}^{k} L_i$. The *wrapping operation* $\odot$ between k-contexts and k-tuples of strings is defined by

$$
\begin{aligned}
&c_0 \square c_1 \square \ldots \square c_k \odot \langle x_1, x_2, \ldots, x_k \rangle \\
:= {} & c_0 x_1 c_1 x_2 c_2 \ldots x_k c_k
\end{aligned}
$$

and extended elementwise to sets of contexts and sets of strings. For $\alpha \in (\Sigma^*)^k \cup \mathcal{P}\left((\Sigma^*)^k\right)$ and $L \subseteq \Sigma^*$, the *contexts of α with respect to L* are defined to be the set

$$
\alpha^{|L\rangle} := \left\{ \mathbf{c} \in (\Sigma^*)^{k+1} \middle| \mathbf{c} \odot \alpha \subseteq L \right\},
$$

with the "$\subseteq$" above replaced by "$\in$" when $\alpha \in (\Sigma^*)^k$. For $\gamma \in (\Sigma^*)^{k+1} \cup \mathcal{P}\left((\Sigma^*)^{k+1}\right)$, we define the set

$$
\gamma^{\langle L|} := \left\{ \mathbf{x} \in (\Sigma^*)^k \middle| \gamma \odot \mathbf{x} \subseteq L \right\},
$$

with the "$\subseteq$" above replaced by "$\in$" when $\gamma \in (\Sigma^*)^{k+1}$. When the identity of the language L is clear from context, we may denote $\alpha^{|L\rangle}$ by $\alpha^{\triangleright}$ and $\gamma^{\langle L|}$ by $\gamma^{\triangleleft}$.

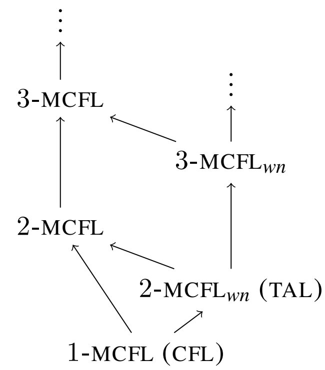

Figure 1: The MCFL hierarchy.

2.1 Multiple Context-Free Grammars

This paper considers *multiple context-free grammars* (MCFGs, Seki et al., 1991), a mildly context-sensitive (MCS) formalism equivalent to MGs (Harkema, 2001; Michaelis, 2001). MCFGs are a generalization of context-free grammars (CFGs) in which nonterminals derive tuples of strings. Whereas CFG rules concatenate strings derived from nonterminals on their right-hand sides, MCFGs interleave nonterminal-derived tuples. Let us consider an example to illustrate how MCFGs generate strings.

Example 1. The following four rules define an MCFG generating the copy language $L = \{ww \mid w \in \{a, b\}^*\}$. The start symbol is S.

$$
\begin{aligned}
S(xy) &\leftarrow T(x, y) && \text{(2a)} \\
T(ax, ay) &\leftarrow T(x, y) && \text{(2b)} \\
T(bx, by) &\leftarrow T(x, y) && \text{(2c)} \\
T(\varepsilon, \varepsilon) &\leftarrow && \text{(2d)}
\end{aligned}
$$

A rule of the form

$$
A(\mathbf{y}) \leftarrow B_1(\mathbf{x}_1) B_2(\mathbf{x}_2) \ldots B_n(\mathbf{x}_n)
$$

is interpreted as an axiom stating that if each nonterminal B_i on the right-hand side generates the tuple $\mathbf{x}_i$, then the nonterminal A on the left-hand side generates the tuple $\mathbf{y}$. In rule (2d), the right-hand side is empty; this means that we assume T to generate the tuple $\langle \varepsilon, \varepsilon \rangle$.

The string $abab \in L$ is derived as follows. By rule (2d), T generates $\langle \varepsilon, \varepsilon \rangle$. By (2c), T generates $\langle b, b \rangle$. By (2b), T generates $\langle ab, ab \rangle$. By (2a), the start symbol S generates $abab$.

An MCFG rule may be thought of as a function that describes how tuples generated by nonterminals on the right-hand side may be combined with one another. These functions must satisfy a condition known as *linearity*, which asserts that MCFG rules cannot copy their inputs.[1]

Definition 3. Fix $k \in \mathbb{N}$. Consider a function $f : \prod_{i=1}^{k} (\Sigma^*)^{d_i} \to (\Sigma^*)^{d_0}$. For each i, write $\mathbf{x}_i = \langle x_{i,1}, x_{i,2}, \ldots, x_{i,d_i} \rangle$. We say that f is a *linear function* if it is of the form

$$f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k) = \langle \alpha_1, \alpha_2, \ldots, \alpha_{d_0} \rangle,$$

where the concatenated string $\alpha = \alpha_1 \alpha_2 \ldots \alpha_{d_0}$ satisfies the following criteria:

- α contains exactly one occurrence of each variable $x_{i,j}$; and
- for each i and j, $x_{i,j}$ occurs to the left of $x_{i,j+1}$ in α.[2]

Furthermore, we say that f is *well-nested* if there are no indices i, j, i', j', and j'', with $i \neq i'$ and $j' \neq j''$, such that the variable $x_{i',j'}$ occurs between $x_{i,j}$ and $x_{i,j+1}$ in α, but $x_{i',j''}$ does not.

Definition 4. A *multiple context-free grammar* (MCFG) is an ordered quadruple $G = \langle N, \Sigma, R, I \rangle$, where

- N is a finite set of *nonterminals*;
- Σ is a finite set of *terminals*;
- $I \subseteq N$ is the set of *start symbols*; and
- letting $\mathcal{F}$ be the set of all linear functions, $R \subseteq N \times \mathcal{F} \times N^*$ is a finite set of *rules*.

We always assume that N and Σ are disjoint. Each nonterminal $A \in N$ is associated with a number $\dim(A)$ known as its *dimension*. All start symbols must have dimension 1. We denote each rule $r = \langle A, f, B_1 B_2 \ldots B_k \rangle$ by

$$A(\mathbf{y}) \leftarrow B_1(\mathbf{x}_1) B_2(\mathbf{x}_2) \ldots B_k(\mathbf{x}_k),$$

where each $\mathbf{x}_i$ is a $\dim(B_i)$-tuple of variables and $\mathbf{y} = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$. We say that r is *well-nested* if f is well-nested.

[1]While individual MCFG rules cannot copy, Example 1 shows that MCFGs can perform copying by combining several different rules.

[2]Technically, the definition of linear functions only requires that α contain at least one occurrence of each $x_{i,j}$. Seki et al. (1991) and Kracht (2003) show that the other assumptions can be made without changing the generative capacity of MCFGs.

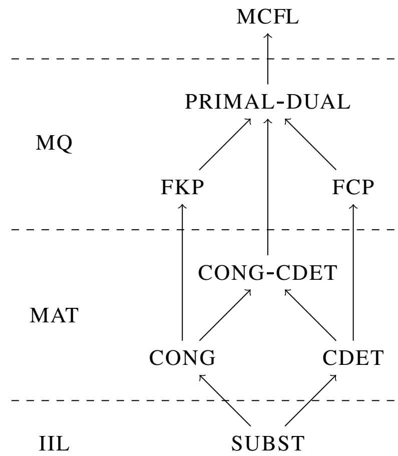

Figure 2: The hierarchy of Distributionally Learnable MCFLs (see Clark and Yoshinaka 2016).

We say that G is a k-*multiple context free grammar* (k-MCFG) if every nonterminal has dimension at most k. We say that G is *well-nested* if every rule in R is well-nested. For each nonterminal A, we define $\mathcal{L}(G, A) \subseteq (\Sigma^*)^{\dim(A)}$ as follows. For each rule $A(\mathbf{y}) \leftarrow B_1(\mathbf{x}_1) B_2(\mathbf{x}_2) \ldots B_k(\mathbf{x}_k)$ with $\mathbf{y} = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$, if $\mathbf{z}_i \in \mathcal{L}(G, B_i)$ for each i, then $f(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) \in \mathcal{L}(G, A)$. Identifying 1-tuples with strings, the *language generated by* G is the language $\mathcal{L}(G) := \bigcup_{S \in I} \mathcal{L}(G, S)$. We say that a language $L \subseteq \Sigma^*$ is a k-*multiple context-free language* (k-MCFL) if it is generated by a k-MCFG. We say that L is *well-nested* if G is well-nested.

MCFLs naturally subsume other common language classes: the class of context-free languages (CFLs) is the same as the class of 1-MCFLs, while the class of tree-adjoining languages (TALs) is the same as the class of well-nested 2-MCFLs (Kanazawa, 2009b). Seki et al. (1991) and Rambow and Satta (1999) prove a separation result showing that k-MCFLs are strictly contained within the class of $(k + 1)$-MCFLs. This MCFL hierarchy, along with its well-nested counterpart (Kanazawa, 2009a), is shown in Figure 1.

3 Learnable Classes of Languages

The theory of learnability considered here is based on the *Identification in the Limit* (IIL) model of Gold (1967). Under this paradigm, the learner receives an infinite data stream containing all possible strings drawn from a target language L, arranged in an unspecified order. After observing each string, the learner must guess a grammar for L. We consider a class of languages $\mathcal{C} \subseteq \mathcal{P}(\Sigma^*)$ to be learnable if there is a learner whose guesses converge to a correct grammar for L for any target $L \in \mathcal{C}$ and any presentation of the strings of L.

IIL-learning of formal languages was pioneered by Angluin (1982), who gave an algorithm that learns the class of regular languages satisfying a criterion known as *reversibility*. While the full class of regular languages is not learnable under the IIL paradigm, Angluin (1987) showed that learner can learn all regular languages if it is equipped with a *minimally adequate teacher* (MAT): a black-box oracle that answers certain questions about the target language. These algorithms were extended to CFLs by Clark and Eyraud (2007) and Clark (2010), respectively, and to MCFLs by Yoshinaka (2011a) and Yoshinaka and Clark (2012), respectively. Since the MAT-learning algorithm fails to learn the full classes of CFLs and MCFLs, further results by Yoshinaka (2011b), Yoshinaka (2012), and Clark and Yoshinaka (2012) expand the MAT-learnable classes via algorithms using *membership queries* (MQs). These learnable language classes are visually summarized in Figure 2.

The remainder of this section formally defines several classes of MCFLs. Subsection 3.1 defines the IIL-learnable *substitutable* MCFLs ("SUBST" in Figure 2), the MCFL analogue of Angluin's reversible regular languages. Subsection 3.2 defines the MAT-learnable *congruential* MCFLs ("CONG" in Figure 2). Subsection 3.3 defines two MQ-learnable classes of MCFLs: those generated by MCFGs with the *finite kernel property* ("FKP" in Figure 2) and those generated by MCFGs with the *finite context property* ("FCP" in Figure 2).

3.1 Substitutable Languages

The IIL algorithms of Clark and Eyraud (2007) and Yoshinaka (2011a) rely upon the strong assumption that whenever two k-tuples $\mathbf{x}$ and $\mathbf{y}$ appear in the same context—i.e., $\mathbf{x}^{\triangleright} \cap \mathbf{y}^{\triangleright} \neq \varnothing$—they must be generated by the same k-dimensional nonterminal. This property is known as *substitutability*.

Definition 5. For $k \in \mathbb{N}$, a language $L \subseteq \Sigma^*$ is *k-substitutable* if for all k-tuples $\mathbf{x}, \mathbf{y} \in (\Sigma^*)^k$, either $\mathbf{x}^{|L\rangle} = \mathbf{y}^{|L\rangle}$ or $\mathbf{x}^{|L\rangle} \cap \mathbf{y}^{|L\rangle} = \varnothing$.

For each k, a language L induces an equivalence relation $\equiv_L^k$ on $(\Sigma^*)^k$ in which $\mathbf{x} \equiv_L^k \mathbf{y}$ if and only if $\mathbf{x}^{\triangleright} = \mathbf{y}^{\triangleright}$. The grammar G constructed by the algorithm identifies each nonterminal A with an equivalence class $[a]$ of $\equiv_L^{\dim(A)}$, so that $a \in \mathcal{L}(G, A) \subseteq [a]$. It turns out that linear functions map tuples of subsets of equivalence classes to subsets of equivalence classes, allowing nonterminals to combine with one another via MCFG rules.

When L is a k-substitutable k-MCFL, the learner determines whether or not $\mathbf{x}$ and $\mathbf{y}$ are equivalent based on whether or not a common context in $\mathbf{x}^{\triangleright} \cap \mathbf{y}^{\triangleright}$ has been observed so far, with the understanding that such a context must appear in the data eventually. At each time step, the learner finds all equivalence classes seen so far, and constructs all rules such that a representative from the equivalence class on the left-hand side has been observed in the data. The learner converges when the data contain enough equivalence classes to construct a correct grammar for L.

3.2 Congruential Languages

In Clark (2010) and Yoshinaka and Clark (2012), the learner again identifies each nonterminal A with an equivalence class of $\equiv_L^{\dim(A)}$ and seeks to find all equivalence classes needed to construct the grammar. Without the assumption of substitutability, the learner relies on the minimally adequate teacher to determine which tuples are equivalent. To do this, the learner asks the teacher two types of questions: *membership queries* (Is $x \in L$?) and *equivalence queries* (What is an example of a string in $L \backslash \mathcal{L}(G)$?). At each time step, an equivalence query is used to identify an equivalence class not covered by an existing nonterminal, and membership queries are used to construct all possible rules involving the new nonterminal. Note that training data are not needed, since the learner asks the teacher for data through equivalence queries. MCFGs constructed through this procedure are known as *congruential* MCFGs.

Definition 6. A k-MCFG G is *congruential* if for every nonterminal A, $\mathcal{L}(G, A)$ is completely contained within an equivalence class of $\equiv_{\mathcal{L}(G)}^{\dim(A)}$. A k-MCFL is *congruential* if it is generated by a congruential k-MCFG.

3.3 The FKP and the FCP

The learning algorithms for the substitutable and congruential CFLs and MCFLs attempt to find equivalence classes of $\equiv_L^k$. Yoshinaka (2011b) and

Clark and Yoshinaka (2012) generalize beyond this approach by dropping the requirement that nonterminals correspond to equivalence classes. Instead, each nonterminal A is identified with a finite set of strings or contexts with the same distribution as A.

Definition 7. Fix $k \in \mathbb{N}$. An MCFG G has the *k-finite kernel property* (k-FKP) if for every nonterminal A of G, there exists $K_A \subseteq (\Sigma^*)^{\dim A}$ such that $|K_A| \leq k$ and $\mathcal{L}(G, A)^{\rhd} = K_A^{\rhd}$. G has the *k-finite context property* (k-FCP) if for every nonterminal A, there exists $C_A \subseteq (\Sigma^*)^{\dim(A)+1}$ such that $|C_A| \leq k$ and $\mathcal{L}(G, A)^{\rhd\lhd} = C_A^{\lhd}$.[3]

At each time step, the learner constructs nonterminals by considering all possible size-k sets of tuples or contexts. The learner constructs rules $r = A(\mathbf{y}) \leftarrow B_1(\mathbf{x}_1)B_2(\mathbf{x}_2)\ldots B_n(\mathbf{x}_n)$ by determining whether or not the right-hand side and the left-hand side have the same distribution. This is done heuristically by asking membership queries for strings of the form $\mathbf{c} \odot f(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$, where f is the linear function associated with r. When L is assumed to have the k-FKP, $\mathbf{c}$ is drawn from the contexts in $K_A^{\rhd}$ observed so far, and for each i, $\mathbf{b}_i \in K_{B_i}$. When L is assumed to have the k-FCP, $\mathbf{c} \in C_A$, while each $\mathbf{b}_i$ is drawn from the substrings in $C_{B_i}^{\lhd}$ observed so far.

4 Overgeneration

The formal problem of overgeneration has a long history in linguistics, starting from Peters and Ritchie's (1973) proof that Transformational Grammars can generate all recursively enumerable languages. Recent interest in the overgeneration problem arises from the model-theoretic approach pioneered by Rogers (1994, 1998) for the formalization of syntactic theories. Relying upon the equivalence of monadic second-order (MSO) logic over trees with tree automata (Thatcher and Wright, 1968), Rogers constructs a context-free implementation of Rizzi's (1990) constraint-based Relativized Minimality theory by stating the constraints in MSO logic. Morawietz (2008) extends the model-theoretic approach to MCS formalisms by combining MSO constraints on derivation trees with MSO-definable transductions from derivation trees to derived structures. Considering the case of MGs, Kobele (2011) and Graf (2011) show that MG derivation trees are closed under intersection with arbitrary MSO constraints, and Graf (2012) shows that complex movement operations can be added to MGs by enhancing the transduction from derivation trees to derived trees. These closure properties allow Graf (2013) to develop a full MG-based treatment of Minimalist syntax within Morawietz's framework, showing that Minimalist constraints on derivation trees and derived trees, as well as transderivational constraints, can be implemented using Merge by carefully choosing categories and selection features for lexical items.

These findings have highlighted the relevance of overgeneration to mainstream Minimalist syntax. As discussed in Graf (2017), existing constraints on movement can be circumvented by using Merge to create non-local dependencies. On the other hand, any pathological pattern is a logically possible MG as long as it is MSO-definable. Illustrating the latter point, Graf (2013, pp. 117–118) identifies three kinds of unnatural dependencies that can be generated by MGs:

(8) a. patterns sensitive to "non-linguistic" information, such as the number of words in a sentence;
   b. languages with completely free word order, subject to no restrictions; and
   c. creating arbitrarily many copies of context-free structure.

In Subsections 4.1 and 4.2, I show that the patterns described in (8a) and (8b), respectively, can be represented as substitutable MCFLs. In Subsection 4.3, we will see that the arbitrary copying of (8c) is not congruential, but it can be generated by an MCFG with both the 2-FKP and the 1-FCP.

4.1 Structure-Independent Patterns

One of the earliest motivations for the rationalist position is the observation that syntactic dependencies are universally sensitive to constituency structure. Chomsky (1968) argues:

> ... grammatical transformations are invariably *structure-dependent* in the sense that they apply to a string of words by virtue of the organization of these words into phrases. It is easy to imagine *structure-independent* operations that apply to a string of elements quite independently of its ab-

[3] The definition presented here is for the *weak* versions of the FKP and the FCP. Other versions of these properties are discussed in Kanazawa and Yoshinaka (2017).

> stract structure as a system of phrases. ... Yet no human language contains structure-independent operations .... The language-learner knows that [such operations] need not be considered as tentative hypotheses.

Examples of "structure-independent" patterns typically consist of operations whose targets are based on arithmetic criteria. One such example appears in the passage quoted above, where Chomsky imagines an auxiliary-fronting operation targeting the first auxiliary in a sentence. This produces the ungrammatical question (9b) from the corresponding declarative (9a).

(9) a. The subjects who will act as controls will be paid.

b. * Will the subjects who t act as controls will be paid?

c. Will the subjects who will act as controls t be paid?

Along these lines, Graf imagines a requirement that the length of a sentence or phrase be a multiple of some fixed number n.

Definition 10. For each $n > 0$, let us define

$$\mathrm{MOD}_n := \{x \mid |x| \equiv 0 \mod n\}.$$

Despite its structure-independence, MOD_n is easily shown to be substitutable.

Proposition 11. MOD_n *is* k*-substitutable for every* $k \in \mathbb{N}$ *and* $n > 0$*.*

Proof. For any k-tuple $\mathbf{x}$ and k-context $\mathbf{c}$, $\mathbf{c} \odot \mathbf{x} \in \mathrm{MOD}_n$ if and only if $|\mathbf{c}| + |\mathbf{x}|$ is a multiple of n. Therefore,

$$\mathbf{x}^{\triangleright} = \{\mathbf{c} \mid |\mathbf{c}| \equiv -|\mathbf{x}| \mod n\}.$$

It is clear that for any $\mathbf{x}, \mathbf{y} \in (\Sigma^*)^k$, $\mathbf{x}^{\triangleright}$ and $\mathbf{y}^{\triangleright}$ are either equal or disjoint, so MOD_n is k-substitutable for every k. □

4.2 Free Word Order

Following Shieber's (1985) argument that Swiss German is not context-free, Joshi (1985) proposed the MCS languages as a characterization of the possible natural languages. This class is defined by grammar formalisms that admit a polynomial-time parsing algorithm, exhibit constant growth, and express limited cross-serial dependencies. The notion of "limited cross-serial dependencies" was left vague, but Joshi et al. (1990, 1991) provide some elaboration:

> [MCS grammars] capture only certain kinds of dependencies, e.g., nested dependencies and certain limited kinds of crossing dependencies (e.g., in the subordinate clause constructions in Dutch or some variations of them, but perhaps not in the so-called MIX (or Bach) language ... [).]

The language MIX mentioned above is the focus of this subsection.

Definition 12. The language MIX is defined as

$$\mathrm{MIX} := \{x \in \{a, b, c\}^* \mid |x|_a = |x|_b = |x|_c\}.$$

According to Joshi (1985), MIX "represents the extreme case of the degree of free word order permitted in a language," and is therefore "linguistically not relevant." The fact that MIX is a 2-MCFL but not a TAL (Salvati, 2011, 2015; Kanazawa and Salvati, 2012) has been used to argue that the TALs are a more suitable formalization of the MCS languages than the MCFLs. This kind of reasoning may be seen as a rationalist position asserting that MIX is not a possible natural language because UG requires natural languages to be TALs. An empiricist account for the absence of MIX-like natural languages might claim that the learners fail to converge on MIX even though MIX is allowed by UG. However, it turns out that MIX is substitutable, so such an account would not be supported by Distributional Learning as a model of language acquisition.

Proposition 13 (Clark and Yoshinaka, 2016)**.** MIX *is* k*-substitutable for every* $k \in \mathbb{N}$*.*

Proof. Suppose $\mathbf{c} \odot \mathbf{x}, \mathbf{c} \odot \mathbf{y}, \mathbf{d} \odot \mathbf{x} \in \mathrm{MIX}$. We want to show that $\mathbf{d} \odot \mathbf{y} \in \mathrm{MIX}$. To that end, observe that for any context γ and symbol $i \in \{a, b, c\}$,

$$|\gamma \odot \mathbf{y}|_i = |\gamma \odot \mathbf{x}|_i - |\mathbf{x}|_i + |\mathbf{y}|_i. \tag{14}$$

Taking $\gamma = \mathbf{c}$, we obtain

$$-|\mathbf{x}|_i + |\mathbf{y}|_i = |\mathbf{c} \odot \mathbf{y}|_i - |\mathbf{c} \odot \mathbf{x}|_i.$$

Next, we take $\gamma = \mathbf{d}$ and substitute the above into (14), giving us

$$|\mathbf{d} \odot \mathbf{y}|_i = |\mathbf{d} \odot \mathbf{x}|_i + |\mathbf{c} \odot \mathbf{y}|_i - |\mathbf{c} \odot \mathbf{x}|_i.$$

Since the terms on the right-hand side have the same value for all i, so must the left-hand side. This means that $\mathbf{d} \odot \mathbf{y} \in \mathrm{MIX}$, as desired. □

4.3 Copying

The third unnatural dependency that Graf considers is represented by the *double-copying language* $\mathrm{COPY}_3(D_1)$.

Definition 15. For $L \subseteq \Sigma^*$ and $n \in \mathbb{N}$, define

$$\mathrm{COPY}_n(L) := \{(x\#)^n \mid x \in L\},$$

where $\# \notin \Sigma$.[4]

Definition 16. The language D_1 is the language generated by the following 1-MCFG.

$$\begin{aligned} S(xy) &\leftarrow S(x)S(y) \\ S(\,[\,x\,]\,) &\leftarrow S(x) \\ S(\varepsilon) &\leftarrow \end{aligned}$$

Copy languages have two interpretations in mathematical linguistics. On the one hand, $\mathrm{COPY}_2(\{a, b\}^*)$ represents the cross-serial dependencies found in Swiss German, since it is a homomorphic image of the embedded-clause verb–argument sequences that Shieber shows is not context-free. On the other hand, the idea of copying structure often appears explicitly in syntactic analyses. Kobele (2006), for instance, argues that Yoruba has a relative clause construction that involves copying VPs. Apart from overtly attested instances of copying, Merchant (1999, 2001) develops a theory of sluicing in which CPs are copied and their TP complements are deleted. It turns out that Distributional Learning can distinguish between these two interpretations of copying.

Proposition 17. *$\mathrm{COPY}_n(L)$ is a congruential n-MCFL if and only if L is regular.*[5]

If we consider non-regular Ls to represent copying of structure, then only the former interpretation is captured by the congruential n-MCFLs.

Lemma 18. *Let $\mathbf{x} = \langle x_1, x_2, \ldots, x_m\rangle$, with $m \leq n$. If $|\mathbf{x}|_\# < n$, then either $|x_i|_\# \geq 2$ for some i, $\mathbf{x}^{|\,\mathrm{COPY}_n(L)\rangle} = \varnothing$, or $\mathbf{x}$ belongs to a finite equivalence class of $\equiv^m_{\mathrm{COPY}_n(L)}$.*

Proof. Suppose $\mathbf{x}^\triangleright \neq \varnothing$, $|x_i|_\# \leq 1$ for all i, and $|\mathbf{x}|_\# \leq n-2$. Then, $\mathbf{x}$ has a context $\mathbf{c} = c_0\Box c_1\Box \ldots \Box c_n$ such that $|c_i|_\# \geq 2$ for some i. Writing $c_i = l\#w\#r$, observe that every $\mathbf{y} \in \mathbf{c}^\triangleleft$ satisfies $\mathbf{c} \odot \mathbf{y} = (w\#)^n$. There are only finitely many such $\mathbf{y}$s, so only finitely many strings may share the context $\mathbf{c}$ with $\mathbf{x}$.

Next, suppose that $\mathbf{x}^\triangleright \neq \varnothing$, $|x_i|_\# \leq 1$ for all i, and $|\mathbf{x}|_\# = n-1$. Then, we have $|x_p|_\# = 0$ for some p, and for $i \neq p$ we can write $x_i = y_i\#z_i$ with $|y_i|_\# = |z_i|_\# = 0$. Let y be the longest y_i and z be the longest z_i, and let $y_{m+1} = z_0 := \varepsilon$. Observe that $\mathbf{c} \in x^\triangleright$ if and only if $\mathbf{c}\odot\mathbf{x} = (w\#)^n$ for some $w \in z\Sigma^*y \cap z_{p-1}\Sigma^*x_p\Sigma^*y_{p+1}$. Thus, if $\mathbf{x'}^\triangleright = \mathbf{x}^\triangleright$, then $\mathbf{x}' = \langle x'_1, x'_2, \ldots, x'_m\rangle$, where $x'_q = x_p$ for some q, $x'_i = y'_i\#z'_i$ for $i \neq q$, y is the longest y'_i, z is the longest z'_i, $y'_{q+1} = y_{q+1}$, and $z'_{q-1} = z_{p-1}$. There are only finitely many such $\mathbf{x}'$, so the lemma follows. $\Box$

Proof of Proposition 17. First, suppose L is regular. Let M be the minimal right-to-left deterministic finite-state automaton recognizing L. We can construct an MCFG G for $\mathrm{COPY}_n(L)$ as follows. The nonterminals of G are the states of M. If A is the start state of M, then G has the rule $A(\#, \#, \ldots, \#) \leftarrow$. If M transitions from state B to state A after reading a, then G has a rule $A(ax_1, ax_2, \ldots, ax_n) \leftarrow B(x_1, x_2, \ldots, x_n)$. Finally, for each accept state S of M, G has a start symbol I_S and a rule $I_S(x_1x_2\ldots x_n) \leftarrow S(x_1, x_2, \ldots, x_n)$. It is clear that for each $x \in \mathcal{L}(G, I_S)$, $x^\triangleright = \{\Box\}$. Observe that for each nonterminal A of G, $\mathcal{L}(G, A)$ is the set of strings $(w\#)^n$ such that M is in state A after reading w. Thus, elements of $\mathcal{L}(G, A)$ are of the form $(w\#)^n$. For each such $(w\#)^n$, we have

$$((w\#)^n)^\triangleright = \{c\Box c\Box \ldots \Box c \mid cw \in L\}.$$

Since M is minimal, if $(u\#)^n, (v\#)^n \in \mathcal{L}(G, A)$, then by the Myhill–Nerode Theorem we must have

$$\{c \mid cu \in L\} = \{c \mid cv \in L\},$$

thus

$$\begin{aligned} ((u\#)^n)^\triangleright &= \{c\Box c\Box \ldots \Box c \mid cu \in L\} \\ &= \{c\Box c\Box \ldots \Box c \mid cv \in L\} \\ &= ((v\#)^n)^\triangleright. \end{aligned}$$

This means that G is congruential.

Now, suppose $\mathrm{COPY}_n(L)$ is generated by a congruential n-MCFG G. By Lemma 18, without loss of generality each copy of $w \in L$ in $(w\#)^n \in \mathrm{COPY}_n(L)$ is generated exclusively using rules of the form

$$A(l_1x_1r_1, l_2x_2r_2, \ldots, l_nx_nr_n)$$

[4] The language considered in Graf (2013) does not have a final $\#$. However, adding the final $\#$ allows $\mathrm{COPY}_n(L)$ to be congruential when L is regular (Proposition 17).

[5] However, it is easy to show that $\mathrm{COPY}_n(\{a^ib^i \mid i \geq 0\})$ is a congruential $(n+1)$-MCFL.

$\leftarrow B(x_1, x_2, \ldots, x_n)$,

where $l_i, r_i \in \Sigma^*$ for each i and $\mathcal{L}(G, B) \subseteq (\Sigma^*\#\Sigma^*)^n$. Since x_1 must already contain a $\#$, such rules can only append a constant string to the left of the first copy of w, so the set of possible ws must be regular. $\square$

Graf argues that $\mathrm{COPY}_3(D_1)$ is unnatural because "embeddings of unbounded depth are copied and fully realized in three distinct positions in the utterance." According to Proposition 17, the property of unbounded depth disqualifies $\mathrm{COPY}_3(D_1)$ from congruentiality, but the existence of more than two copies does not, as long as $\mathrm{COPY}_3(D_1)$ is generated by a 3-MCFG. $\mathrm{COPY}_3(D_1)$ is still learnable, however, because it belongs to the class of 3-MCFLs defined by the FKP and the FCP.

Proposition 19. $\mathrm{COPY}_3(D_1)$ *is generated by a* 3-*MCFG* G *with the* 2-*FKP and the* 1-*FCP.*

Proof. G is defined as follows.

$$\begin{aligned} S(x\#y\#z\#) &\leftarrow T(x, y, z) \\ T(x_1x_2, y_1y_2, z_1z_2) &\leftarrow T(x_1, y_1, z_1) \\ & T(x_2, y_2, z_2) \\ T(\texttt{[}x\texttt{]}, \texttt{[}y\texttt{]}, \texttt{[}z\texttt{]}) &\leftarrow T(x, y, z) \\ T(\varepsilon, \varepsilon, \varepsilon) &\leftarrow \end{aligned}$$

We have

$$\begin{aligned} \mathcal{L}(G, S)^{\triangleright} &= \mathrm{COPY}_3(D_1)^{\triangleright} = \{\square\} = \{\#\#\#\}^{\triangleright} \\ \mathcal{L}(G, T)^{\triangleright} &= \{\langle w, w, w\rangle \mid w \in D_1\}^{\triangleright} \\ &= \left\{ l\square r\#l\square r\#l\square r\# \,\middle|\, l\square r \in D_1^{\langle D_1 |} \right\} \\ &= \{\langle \texttt{[[]]}, \texttt{[[]]}, \texttt{[[]]}\rangle, \\ &\quad \langle \texttt{[][]}, \texttt{[][]}, \texttt{[][]}\rangle\}^{\triangleright}, \end{aligned}$$

so G has the 2-FKP.[6] We also have

$$\begin{aligned} \mathcal{L}(G, S)^{\triangleright\triangleleft} &= \{\square\}^{\triangleleft} \\ \mathcal{L}(G, T)^{\triangleright\triangleleft} &= \{\square\#\square\#\square\#\}^{\triangleleft}, \end{aligned}$$

so G has the 1-FCP. $\square$

5 Conclusion

We have seen that MOD_n and MIX are k-substitutable for every k, while $\mathrm{COPY}_3(D_1)$ is generated by a 3-MCFG with the 2-FKP and the 1-FCP. If the Distributional Learning hierarchy of Figure 2 is taken to be a measure of complexity, then we may conclude from these facts that MOD_n and MIX are very simple from the perspective of learnability. Proposition 17 gives us the interesting result that in general, the complexity of $\mathrm{COPY}_n(L)$ with respect to learnability is related to the language-theoretic complexity of L, so that $\mathrm{COPY}_3(D_1)$ is slightly more complex than the congruential languages. Since natural language grammars are not congruential,[7] any class in the learnable hierarchy that might plausibly include natural language syntax would also likely include MOD_n, MIX, and $\mathrm{COPY}_3(D_1)$.

The discussions in Subsections 4.2 and 4.3 should be contrasted with language-theoretic analyses of MIX and $\mathrm{COPY}_3(D_1)$, respectively. As mentioned previously, MIX falls outside the TALs, which is identical to the well-nested 2-MCFLs. Similarly, Kanazawa and Salvati (2010) show that $\mathrm{COPY}_3(D_1)$ is not a well-nested n-MCFL for any n. The criterion of well-nestedness, then, provides an elegant rationalist explanation for the absence of MIX- or $\mathrm{COPY}_3(D_1)$-like natural languages. Such a criterion could also be justified on functionalist grounds, since well-nested MCFGs admit a more efficient parsing algorithm than MCFGs in general (Gómez-Rodríguez et al., 2010). While the well-nestedness requirement does not eliminate MOD_n, an anonymous reviewer observes that intersecting the language of a CFG with MOD_n increases the size of that CFG by a factor of n^2. Thus, when n is large, languages with MOD_n-like dependencies may be eliminated by functionalist considerations regarding grammar size.

The case of MOD_n shows that the regular languages capture many patterns that do not resemble natural language dependencies. Although the inclusion of the regular languages in natural language has traditionally been taken for granted,[8] recent work on the *subregular hierarchy* has shown that markedness constraints in phonotactics and morphotactics typically belong to restricted, IIL-learnable subclasses of the regular languages (Heinz et al., 2011; Aksënova et al.,

[6] G does not have the 1-FKP because for any $\langle x, x, x\rangle \in \mathcal{L}(G, T)^{\triangleright\triangleleft}$, $x\square\#\square x\#x\square\# \in \langle x, x, x\rangle^{\triangleright} \backslash \mathcal{L}(G, T)^{\triangleright}$.

[7] For example, congruentiality would preclude the possibility of a single word such as *effect* or *affect* having the distribution of both a noun and a verb if nonterminals are identified with syntactic categories.

[8] Additionally, Joshi et al. (1990, 1991) mention strict inclusion of the CFLs as a fourth defining property of the MCS languages.

2016) that formalize notions of locality. It may be the case that a refinement of the regular languages is needed for syntax as well. While the study of the equivalence relation $\equiv_L^k$ may be seen as an algebraic treatment of the notion of structure (Clark, 2015), the learnability of MOD_n may reveal a point of divergence between the algebraic approach and the intuitive notion of syntactic constituencies.

In conclusion, this paper has shown that the restricted language classes in the Distributional Learning hierarchy are rich enough to raise the very questions of overgeneration that they were hypothesized to solve. While Distributional Learning does not provide a learnability-based account for the typological absence of patterns modelled by MOD_n, MIX, or $\mathrm{COPY}_3(D_1)$, all three patterns can plausibly be eliminated on rationalist or functionalist grounds. These findings suggest that learnability may play a smaller role in determining natural language typology than once expected.

References

Alëna Aksënova, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as Tier-Based Strictly Local Dependencies. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 121–130, Berlin, Germany. Association for Computational Linguistics.

Dana Angluin. 1982. Inference of Reversible Languages. *Journal of the Association for Computing Machinery*, 29(3):741–765.

Dana Angluin. 1987. Learning Regular Sets from Queries and Counterexamples. *Information and Computation*, 75(2):87–106.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*, 1 edition. MIT Press, Cambridge, MA.

Noam Chomsky. 1968. *Language and Mind*, 1 edition. Harcourt, Brace & World, New York, NY.

Noam Chomsky. 1971. *Problems of Knowledge and Freedom: The Russell Lectures*, 1 edition. Pantheon Books, New York, NY.

Noam Chomsky. 1980. On Cognitive Structures and their Development: A Reply to Piaget. In Piatelli M. Palmerini, editor, *Language and Learning: The Debate Between Jean Piaget and Noam Chomsky*, pages 35–54. Routledge & Kegan Paul, London, United Kingdom.

Alexander Clark. 2010. Distributional Learning of Some Context-Free Languages with a Minimally Adequate Teacher. In *Grammatical Inference: Theoretical Results and Applications*, pages 24–37, Valencia, Spain. Springer Berlin Heidelberg.

Alexander Clark. 2015. The syntactic concept lattice: Another algebraic theory of the context-free languages? *Journal of Logic and Computation*, 25(5):1203–1229.

Alexander Clark and Rémi Eyraud. 2007. Polynomial Identification in the Limit of Substitutable Context-free Languages. *Journal of Machine Learning Research*, 8(Aug):1725–1745.

Alexander Clark and Ryo Yoshinaka. 2012. Beyond Semilinearity: Distributional Learning of Parallel Multiple Context-free Grammars. In *Proceedings of the Eleventh International Conference on Grammatical Inference, PMLR 21:1-3, 2012*, volume 21 of *Proceedings of Machine Learning Research*, pages 84–96, College Park, MD. PMLR.

Alexander Clark and Ryo Yoshinaka. 2016. Distributional Learning of Context-Free and Multiple Context-Free Grammars. In Jeffrey Heinz and José M. Sempere, editors, *Topics in Grammatical Inference*, pages 143–172. Springer Berlin Heidelberg, Berlin, Germany.

E Mark Gold. 1967. Language identification in the limit. *Information and Control*, 10(5):447–474.

Carlos Gómez-Rodríguez, Marco Kuhlmann, and Giorgio Satta. 2010. Efficient Parsing of Well-Nested Linear Context-Free Rewriting Systems. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 276–284, Los Angeles, CA. Association for Computational Linguistics.

Thomas Graf. 2011. Closure Properties of Minimalist Derivation Tree Languages. In *Logical Aspects of Computational Linguistics: 6th International Conference, LACL 2011, Montpellier, France, June 29 – July 1, 2011. Proceedings*, pages 96–111, Berlin, Germany. Springer Berlin Heidelberg.

Thomas Graf. 2012. Movement-Generalized Minimalist Grammars. In *7th International Conference, LACL 2012, Nantes, France, July 2-4, 2012. Proceedings*, volume 7351 of *Lecture Notes in Computer Science*, pages 58–73, Berlin, Germany. Springer Berlin Heidelberg.

Thomas Graf. 2013. *Local and Transderivational Constraints in Syntax and Semantics*. PhD Dissertation, University of California, Los Angeles, Los Angeles, CA.

Thomas Graf. 2017. A computational guide to the dichotomy of features and constraints. *Glossa: a journal of general linguistics*, 2(1):18.1–36.

Henk Harkema. 2001. A Characterization of Minimalist Languages. In *Logical Aspects of Computational Linguistics: 4th International Conference, LACL 2001 Le Croisic, France, June 27–29, 2001 Proceedings*, pages 193–211, Berlin, Germany. Springer Berlin Heidelberg.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based Strictly Local Constraints for Phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 58–64, Portland, OR. Association for Computational Linguistics.

Aravind K. Joshi. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In Arnold M. Zwicky, David R. Dowty, and Lauri Karttunen, editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, Studies in Natural Language Processing, pages 206–250. Cambridge University Press, Cambridge, United Kingdom.

Aravind K. Joshi, K. Vijay Shanker, and David Weir. 1990. The Convergence of Mildly Context-Sensitive Grammar Formalisms. Technical Report MS-CIS-90-01, University of Pennsylvania Department of Computer and Information Science, Philadelphia, PA.

Aravind K. Joshi, K. Vijay Shanker, and David Weir. 1991. The Convergence of Mildly Context-Sensitive Grammar Formalisms. In Stuart M. Shieber, Peter Sells, and Thomas Wasow, editors, *Foundational Issues in Natural Language Processing*, System Development Foundation Benchmark Series, pages 31–81. MIT Press, Cambridge, MA.

Makoto Kanazawa. 2009a. The Convergence of Well-Nested Mildly Context-Sensitive Grammar Formalisms.

Makoto Kanazawa. 2009b. The Pumping Lemma for Well-Nested Multiple Context-Free Languages. In *Developments in Language Theory: 13th International Conference, DLT 2009, Stuttgart, Germany, June 30-July 3, 2009. Proceedings*, volume 5583 of *Lecture Notes in Computer Science*, pages 312–325, Berlin, Germany. Springer Berlin Heidelberg.

Makoto Kanazawa and Sylvain Salvati. 2010. The Copying Power of Well-Nested Multiple Context-Free Grammars. In *Language and Automata Theory and Applications: 4th International Conference, LATA 2010, Trier, Germany, May 24-28, 2010. Proceedings*, volume 6031 of *Lecture Notes in Computer Science*, pages 344–355, Berlin, Germany. Springer Berlin Heidelberg.

Makoto Kanazawa and Sylvain Salvati. 2012. MIX Is Not a Tree-Adjoining Language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 666–674, Jeju, South Korea. Association for Computational Linguistics.

Makoto Kanazawa and Ryo Yoshinaka. 2017. The Strong, Weak, and Very Weak Finite Context and Kernel Properties. In *Language and Automata Theory and Applications: 11th International Conference, LATA 2017, Umeå, Sweden, March 6-9, 2017, Proceedings*, pages 77–88, Cham, Switzerland. Springer International Publishing.

Gregory M. Kobele. 2006. *Generating Copies: An Investigation into Structural Identity in Language and Grammar*. PhD Dissertation, University of California, Los Angeles, Los Angeles, CA.

Gregory M. Kobele. 2011. Minimalist Tree Languages Are Closed Under Intersection with Recognizable Tree Languages. In *Logical Aspects of Computational Linguistics: 6th International Conference, LACL 2011, Montpellier, France, June 29 – July 1, 2011. Proceedings*, pages 129–144, Berlin, Germany. Springer Berlin Heidelberg.

Marcus Kracht. 2003. *The Mathematics of Language*. Number 63 in Studies in Generative Grammar. De Gruyter Mouton, Berlin, Germany.

Jason Merchant. 1999. *The Syntax of Silence: Sluicing, Islands, and Identity in Ellipsis*. PhD Dissertation, University of California, Santa Cruz, Santa Cruz, CA.

Jason Merchant. 2001. *The Syntax of Silence: Sluicing, Islands, and the Theory of Ellipsis*, 1 edition. Oxford Studies in Theoretical Linguistics. Oxford University Press, Oxford, United Kingdom.

Jens Michaelis. 2001. Derivational Minimalism Is Mildly Context–Sensitive. In *Logical Aspects of Computational Linguistics: Third International Conference, LACL'98 Grenoble, France, December 14–16, 1998 Selected Papers*, pages 179–198, Berlin, Germany. Springer Berlin Heidelberg.

Frank Morawietz. 2008. *Two-Step Approaches to Natural Language Formalism*, 1 edition. Number 64 in Studies in Generative Grammar. De Gruyter Mouton, Berlin, Germany.

P. Stanley Peters and R. W. Ritchie. 1973. On the generative power of transformational grammars. *Information Sciences*, 6(Supplement C):49–83.

Owen Rambow and Giorgio Satta. 1999. Independent parallelism in finite copying parallel rewriting systems. *Theoretical Computer Science*, 223(1):87–120.

Luigi Rizzi. 1990. *Relativized Minimality*. Number 16 in Linguistic Inquiry Monographs. MIT Press, Cambridge, MA.

James Rogers. 1994. *Studies in the Logic of Trees with Applications to Grammar Formalisms*. PhD Dissertation, University of Delaware, Newark, DE.

James Rogers. 1998. *A Descriptive Approach to Language-Theoretic Complexity*. Studies in Logic, Language, and Information. CSLI Publications, Stanford, CA.

Sylvain Salvati. 2011. MIX is a 2-MCFL and the word problem in $\mathbb{Z}^2$ is solved by a third-order collapsible pushdown automaton.

Sylvain Salvati. 2015. MIX is a 2-MCFL and the word problem in Z2 is captured by the IO and the OI hierarchies. *Journal of Computer and System Sciences*, 81(7):1252–1277.

Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. 1991. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229.

Stuart M. Shieber. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–343.

J. W. Thatcher and J. B. Wright. 1968. Generalized Finite Automata Theory with an Application to a Decision Problem of Second-Order Logic. *Mathematical systems theory*, 2(1):57–81.

Ryo Yoshinaka. 2011a. Efficient learning of multiple context-free languages with multidimensional substitutability from positive data. *Theoretical Computer Science*, 412(19):1821–1831.

Ryo Yoshinaka. 2011b. Towards Dual Approaches for Learning Context-Free Grammars Based on Syntactic Concept Lattices. In *Developments in Language Theory: 15th International Conference, DLT 2011, Milan, Italy, July 19-22, 2011. Proceedings*, pages 429–440, Berlin, Germany. Springer Berlin Heidelberg.

Ryo Yoshinaka. 2012. Integration of the Dual Approaches in the Distributional Learning of Context-Free Grammars. In *6th International Conference, LATA 2012, A Coruña, Spain, March 5-9, 2012. Proceedings*, volume 7183 of *Lecture Notes in Computer Science*, pages 538–550, Berlin, Germany. Springer Berlin Heidelberg.

Ryo Yoshinaka and Alexander Clark. 2012. Polynomial Time Learning of Some Multiple Context-Free Languages with a Minimally Adequate Teacher. In *Formal Grammar: 15th and 16th International Conferences, FG 2010, Copenhagen, Denmark, August 2010, FG 2011, Ljubljana, Slovenia, August 2011, Revised Selected Papers*, Lecture Notes in Computer Science, pages 192–207, Berlin, Germany. Springer Berlin Heidelberg.