Proceedings of Machine Learning Research 153:18–31, 2021 Proceedings of the 15th ICGI

A Hierarchy of Context-Free Languages Learnable from Positive Data and Membership Queries

Makoto Kanazawa KANAZAWA@HOSEI.AC.JP
Hosei University

Ryo Yoshinaka RYOSHINAKA@TOHOKU.AC.JP
Tohoku University

Editors: Jane Chandlee, Rémi Eyraud, Jeffrey Heinz, Adam Jardine, and Menno van Zaanen

Abstract

We consider a generalization of the "dual" approach to distributional learning of context-free grammars, where each nonterminal A is associated with a string set X_A characterized by a finite set C of contexts. Rather than letting X_A be the set of all strings accepted by all contexts in C as in previous works, we allow more flexible uses of the contexts in C, using some of them positively (contexts that accept the strings in X_A) and others negatively (contexts that do not accept any strings in X_A). The resulting more general algorithm works in essentially the same way as before, but on a larger class of context-free languages.
Keywords: distributional learning; membership queries; context-free grammars

1. Introduction

In the "dual" approach to distributional learning of context-free grammars (Clark and Yoshinaka, 2016), the learner uses finite sets of contexts (i.e., pairs of strings) as nonterminals of the hypothesized grammar. A hypothesized production

$$C_0 \to w_0\, C_1\, w_1\, \dots\, C_n\, w_n\,,$$

where each C_i is a finite set of contexts qua nonterminal and each w_i is a terminal string, is deemed by the learner to be compatible with available evidence about the target language if

$$C_0^{\triangleleft} \supseteq w_0\,(E \cap C_1^{\triangleleft})\,w_1\,\dots\,(E \cap C_n^{\triangleleft})\,w_n\,, \tag{1}$$

where E is the set of substrings contained in the positive data given to the learner so far. The operation $(\cdot)^{\triangleleft}$ takes a set C of contexts to the set $C^{\triangleleft} = \{\, x \mid \text{for all } (u,v) \in C,\ uxv \in L \,\}$ consisting of all the strings that are accepted by all contexts in C in the target language L. If $C = \{(u_1, v_1), \dots, (u_k, v_k)\}$, then

$$x \in C^{\triangleleft} \iff u_1xv_1 \in L \wedge \cdots \wedge u_kxv_k \in L\,, \tag{2}$$

so that whether a string x belongs to $C^{\triangleleft}$ is determined by k queries to the membership oracle. With a fixed bound on the cardinality of the C_i and the number n of right-hand side nonterminals, as well as an appropriate restriction on the elements of the C_i and the sequence of terminal strings $(w_0, w_1, \dots, w_n)$ in the right-hand side of productions, a polynomial number of queries suffice to test all hypothesized productions as to their compatibility, in the sense of (1).

From the computational efficiency perspective, however, there is no reason to want to restrict the use of finite context sets to conjunctions of "membership atoms" $uxv \in L$, as in (2). One could partition a finite context set representing a nonterminal into two sets $C = \{(u_1, v_1), \dots, (u_k, v_k)\}$ and $D = \{(y_1, z_1), \dots, (y_l, z_l)\}$, and use the contexts in C positively and those in D negatively:

$$x \in C^{\triangleleft} \cap D^{\overline{\triangleleft}} \Longleftrightarrow u_1 x v_1 \in L \wedge \cdots \wedge u_k x v_k \in L \wedge y_1 x z_1 \notin L \wedge \cdots \wedge y_l x z_l \notin L. \qquad (3)$$

The operation $(\cdot)^{\overline{\triangleleft}}$ takes a context set D to the set $D^{\overline{\triangleleft}} = \{\, x \mid \text{for all } (y,z) \in D,\ yxz \notin L \,\}$. With each nonterminal represented by such a pair of finite context sets, the production

$$(C_0, D_0) \to w_0\, (C_1, D_1)\, w_1\, \dots\, (C_n, D_n)\, w_n$$

would be compatible with available evidence just in case

$$C_0^{\triangleleft} \cap D_0^{\overline{\triangleleft}} \supseteq w_0\, (E \cap C_1^{\triangleleft} \cap D_1^{\overline{\triangleleft}})\, w_1\, \dots\, (E \cap C_n^{\triangleleft} \cap D_n^{\overline{\triangleleft}})\, w_n\,.$$

In fact, one could go a step further and allow all Boolean combinations of membership atoms $uxv \in L$, not just conjunctions of positive and negative membership literals as in (3).

In this paper, we show that these generalizations of the existing distributional learning algorithm indeed meet the same criteria of efficient learning from positive data and membership queries as before, while significantly enlarging the class of context-free languages that can be learned. For the case of allowing conjunctions of positive and negative membership literals, the bound k on the number of positive literals and the bound l on the number of negative literals give rise to a two-dimensional hierarchy of context-free languages, where an increment of either k or l cannot be matched by any amount of increase in the other parameter.

Although still further uses of membership queries are conceivable, we hope that this paper leads to a better understanding of the limits of efficient learning algorithms for context-free languages utilizing positive data and membership queries.

2. Preliminaries

We allow a context-free grammar to have multiple initial nonterminals. Thus, a CFG is a 4-tuple $G = (N, \Sigma, P, I)$, where N is the set of nonterminals, Σ is the terminal alphabet, P is the set of productions, and I is the set of initial nonterminals. A sequence $(X_A)_{A \in N}$ of subsets of Σ^* indexed by nonterminals is a *pre-fixed point* of G if for each production $B_0 \to w_0\, B_1\, w_1\, \dots\, B_n\, w_n$ of G, we have $X_{B_0} \supseteq w_0\, X_{B_1}\, w_1\, \dots\, X_{B_n}\, w_n$. If $(X_A)_{A \in N}$ is the least pre-fixed point (under componentwise inclusion) of G (which must exist), then we write $L(G, A)$ for X_A. The language of G is then $L(G) = \bigcup_{A \in I} L(G, A)$. A pre-fixed point $(X_A)_{A \in N}$ of G is *sound* if $\bigcup_{A \in I} X_A \subseteq L(G)$ (or, equivalently, if $\bigcup_{A \in I} X_A = L(G)$).

We take for granted the standard notions of derivation and derivation tree of a CFG G. If A is a nonterminal of G, then the *context set* of A is $C(G, A) = \{\, (u, v) \in \Sigma^* \times \Sigma^* \mid S \Rightarrow_G^* u\, A\, v \text{ for some } S \in I \,\}$. It is easy to see that if $(X_A)_{A \in N}$ is a sound pre-fixed point (SPP) of G, then $L(G) \supseteq \bigcup_{(u,v) \in C(G,A)} u\, X_A\, v$ for every $A \in N$. A nonterminal A is *unreachable* if $C(G, A) = \varnothing$ and *unproductive* if $L(G, A) = \varnothing$. A nonterminal is *useless* if it is either unproductive or unreachable.

A *pump* in a derivation tree τ of G is a part of τ that corresponds to a derivation of the form $A \Rightarrow_G^+ uAv$, where $A \in N$ and $uv \in \Sigma^+$. The *yield* of such a pump is the pair (u, v). The *pumping number* for G is the least natural number p such that every derivation tree τ for a string x with $|x| \geq p$ contains a pump with yield (u, v) such that $|u| + |v| \leq p$.

Let $L \subseteq \Sigma^*$. We write $\overline{L}$ for $\Sigma^* - L$. For $C \subseteq \Sigma^* \times \Sigma^*$, define

$$C^{\langle L |} = \{\, x \in \Sigma^* \mid \text{for all } (u, v) \in C,\ uxv \in L \,\}$$

For $X \subseteq \Sigma^*$, define

$$X^{| L \rangle} = \{\, (u, v) \in \Sigma^* \times \Sigma^* \mid u\, X\, v \subseteq L \,\}.$$

When L is understood from context, we write $C^{\triangleleft}$ and $X^{\triangleright}$ for $C^{\langle L|}$ and $X^{|L\rangle}$, and $C^{\overline{\triangleleft}}$ and $X^{\overline{\triangleright}}$ for $C^{\langle \overline{L}|}$ and $X^{|\overline{L}\rangle}$. The map $(\cdot)^{\triangleright\triangleleft} \colon \mathscr{P}(\Sigma^*) \to \mathscr{P}(\Sigma^*)$ is a closure operator in the sense that (i) $X \subseteq X^{\triangleright\triangleleft}$; (ii) $X \subseteq Y$ implies $X^{\triangleright\triangleleft} \subseteq Y^{\triangleright\triangleleft}$; and (iii) $X^{\triangleright\triangleleft\triangleright\triangleleft} = X^{\triangleright\triangleleft}$. A set $X \subseteq \Sigma^*$ is *closed* when $X^{\triangleright\triangleleft} = X$, or equivalently, when $X = C^{\triangleleft}$ for some $C \subseteq \Sigma^* \times \Sigma^*$.

For $k \geq 1$, $l \geq 0$, and $m \geq 1$, define[1]

$$\begin{aligned} \mathrm{FC}_L(k, l) &= \{\, C^{\triangleleft} \cap D^{\overline{\triangleleft}} \mid C, D \subseteq \Sigma^* \times \Sigma^*, 1 \leq |C| \leq k, 0 \leq |D| \leq l \,\}, \\ \mathrm{FC}_L(k, l, m) &= \{\, X_1 \cup \dots \cup X_m \mid X_1, \dots, X_m \in \mathrm{FC}_L(k, l) \,\}. \end{aligned}$$

If membership of x in X is determined by a fixed Boolean combination φ of a fixed finite set of queries of the form "$uxv \in L$?", then X is in $\mathrm{FC}_L(k, l, m)$ for some k, l, m. A bound on k, l, m is obtained by converting φ into disjunctive normal form.

A context-free language L belongs to $\mathbf{FCP}_r(k, l)$ (resp. $\mathbf{FCP}_r(k, l, m)$) iff there is a context-free grammar $G = (N, \Sigma, P, I)$ for L such that each production in P has at most r nonterminals on its right-hand side and G has an SPP $(X_A)_{A \in N}$ satisfying $X_A \in \mathrm{FC}_L(k, l)$ (resp. $X_A \in \mathrm{FC}_L(k, l, m)$) for all $A \in N$. We write $\mathbf{FCP}(k, l)$ and $\mathbf{FCP}(k, l, m)$ for $\bigcup_{r \geq 0} \mathbf{FCP}_r(k, l)$ and $\bigcup_{r \geq 0} \mathbf{FCP}_r(k, l, m)$, respectively.[2]

3. Learnability of $\mathbf{FCP}_r(k, l, m)$

We give an algorithm for learning context-free languages in $\mathbf{FCP}_r(k, l, m)$ in the limit from positive data and membership queries.

For $K \subseteq \Sigma^*$, let

$$\begin{aligned} \mathrm{Sub}(K) &= \{\, w \in \Sigma^* \mid uwv \in K \text{ for some } u, v \,\}, \\ \mathrm{Sub}^n(K) &= \{\, (w_1, \dots, w_n) \in (\Sigma^*)^n \mid u_0 w_1 u_1 \dots w_n u_n \in K \text{ for some } u_0, u_1, \dots, u_n \,\}, \\ \mathrm{Sub}^{\leq r}(K) &= \bigcup_{n=1}^{r} \mathrm{Sub}^n(K), \\ \mathrm{Con}(K) &= \{\, (u, v) \in \Sigma^* \times \Sigma^* \mid uwv \in K \text{ for some } w \,\}. \end{aligned}$$

We first observe a simple fact.

1. A set of the form $D^{\overline{\triangleleft}}$ contains all strings that do not occur as substrings of any elements of L and so cannot be a component of an SPP of a grammar G of L unless all strings occur as substrings of elements of L or G has an unreachable nonterminal. This is why we do not consider $\mathrm{FC}_L(0, l)$ or $\mathrm{FC}_L(0, l, m)$.
2. In the terminology of Kanazawa and Yoshinaka (2017), $L \in \mathbf{FCP}(k, 0)$ iff L has a CFG with the very weak k-FCP. We do not consider the "weak" and "strong" versions of $\mathbf{FCP}(k, l)$ and $\mathbf{FCP}(k, l, m)$ in this paper.

Proposition 1 *If $X \in \mathrm{FC}_L(k,l)$ and $X \neq \varnothing$, then $X = C^{\langle L|} \cap D^{\langle \overline{L}|}$ for some $C, D \subseteq \mathrm{Con}(L)$.*

Proof It is easy to see that if $C \not\subseteq \mathrm{Con}(L)$, then $C^{\langle L|} = \varnothing$, and for all $D \subseteq \Sigma^* \times \Sigma^*$, $D^{\langle \overline{L}|} = (D \cap \mathrm{Con}(L))^{\langle \overline{L}|}$. ■

A learner for $\mathbf{FCP}_r(k,l,m)$ is listed in Algorithm 1. A *positive presentation* of L_* is an infinite sequence of strings $t_1, t_2, \ldots$ enumerating exactly the elements of L_*. If $B = \{(C_1, D_1), \ldots, (C_m, D_m)\}$, where $C_j, D_j \subseteq \mathrm{Con}(L_*)$ for $j = 1, \ldots, m$, define

$$[\![B]\!]^{L_*} = \left(C_1^{\langle L_*|} \cap D_1^{\langle \overline{L_*}|}\right) \cup \cdots \cup \left(C_m^{\langle L_*|} \cap D_m^{\langle \overline{L_*}|}\right).$$

Whether a string x belongs to $[\![B]\!]^{L_*}$ can be decided by at most $m(k+l)$ queries to the membership oracle for L_*. A production $B_0 \to w_0\, B_1\, w_1 \,\ldots\, B_n\, w_n$ is *valid* if

$$[\![B_0]\!]^{L_*} \supseteq w_0\, [\![B_1]\!]^{L_*}\, w_1 \,\ldots\, [\![B_n]\!]^{L_*}\, w_n\,,$$

and *valid on E* if

$$[\![B_0]\!]^{L_*} \supseteq w_0\, (E \cap [\![B_1]\!]^{L_*})\, w_1 \,\ldots\, (E \cap [\![B_n]\!]^{L_*})\, w_n\,.$$

Note that a production is valid if and only if it is valid on $\mathrm{Sub}(L_*)$.

Algorithm 1: Learner for $\mathbf{FCP}_r(k,l,m)$.

Parameters: Positive integers r, k, m; a natural number l;
Data: A positive presentation $t_1, t_2, \ldots$ of $L_* \subseteq \Sigma^*$; membership oracle for L_*;
Result: A sequence of grammars $G_1, G_2, \ldots$;

$T_0 := \varnothing$; $E_0 := \varnothing$; $J_0 := \varnothing$; $H_0 := \varnothing$; $G_0 := (\varnothing, \Sigma, \varnothing, \varnothing)$;
for $i = 1, 2, \ldots$ **do**
  $T_i := T_{i-1} \cup \{t_i\}$; $E_i := \mathrm{Sub}(T_i)$;
  if $T_i \not\subseteq L(G_{i-1})$ **then**
    $J_i := \mathrm{Con}(T_i)$; $H_i := \mathrm{Sub}^{\leq r+1}(T_i)$;
  else
    $J_i := J_{i-1}$; $H_i := H_{i-1}$;
  end
  output $G_i := (N_i, \Sigma, P_i, I_i)$ where

$$N_i := \{\, \{(C_1, D_1), \ldots, (C_m, D_m)\} \mid C_j, D_j \subseteq J_i, 1 \leq |C_j| \leq k, 0 \leq |D_j| \leq l \,\},$$
$$P_i := \{\, B_0 \to w_0\, B_1\, w_1 \,\ldots\, B_n\, w_n \mid (w_0, w_1, \ldots, w_n) \in H_i,$$
$$B_0, B_1, \ldots, B_n \in N_i, B_0 \to w_0\, B_1\, w_1 \,\ldots\, B_n\, w_n \text{ is valid on } E_i \,\},$$
$$I_i := \{\, B \in N_i \mid E_i \cap [\![B]\!]^{L_*} \subseteq L_* \,\};$$

end

Proposition 2 *At each stage i, the number of queries to the membership oracle made by Algorithm 1 is bounded by a polynomial in the total lengths of the strings in T_i.*

Proof Let $n = \sum_{j=1}^{i} |t_j|$ (the sum of the lengths of the strings in T_i, accounting for repetitions). Then

$$
\begin{aligned}
|E_i| &\leq \binom{n+1}{2} + 1 = \frac{(n+1)n+2}{2}, \\
|J_i| &\leq \binom{n+1}{2} + 2(n+1) = \frac{(n+1)(n+4)}{2}, \\
|H_i| &\leq \sum_{j=1}^{r+1} \binom{n+2j}{2j} \leq (r+1)\binom{n+2r+2}{2r+2} \leq \frac{(r+1)(n+2r+2)^{2r+2}}{(2r+2)!}, \\
|N_i| &\leq \frac{1}{m!}\left(\left(\sum_{j=1}^{k}\binom{|J_i|}{j}\right)\left(\sum_{j=0}^{l}\binom{|J_i|}{j}\right)\right)^m \leq \frac{1}{m!}\left(|J_i|^k\,(|J_i|^l+1)\right)^m \leq \frac{|J_i|^{(k+l+1)m}}{m!}\,.
\end{aligned}
$$

The number of productions that are considered for inclusion in P_i is $|H_i|\,|N_i|^{r+1}$, and to test each of them for validity on E_i requires at most $|E_i|^r\,(m(k+l))^{r+1}$ queries to the membership oracle, so the construction of the set P_i requires at most $|H_i|\,|N_i|^{r+1}|E_i|^r\,(m(k+l))^{r+1}$ queries. Finally, to test each $B \in N_i$ for inclusion in I_i requires at most $|E_i|\,(m(k+l)+1)$ queries to the membership oracle, so to determine the set I_i requires at most $|N_i|\,|E_i|\,(m(k+l)+1)$ queries. All these numbers are polynomial in n. ■

The proof of correctness of Algorithm 1 is virtually identical to the corresponding proofs in Clark et al. (2016) and Kanazawa and Yoshinaka (2017).

Theorem 3 *If $L_* \subseteq \Sigma^*$ is in $\mathbf{FCP}_r(k,l,m)$, the output of Algorithm 1 converges to a grammar $G = (N, \Sigma, P, I)$ for L_*. Moreover, $([\![B]\!]^{L_*})_{B \in N}$ is an SPP of G consisting entirely of sets in $\mathrm{FC}_{L_*}(k,l,m)$.*

Proof Since L_* is in $\mathbf{FCP}_r(k,l,m)$, Proposition 1 implies that there is a CFG $G_* = (N_*, \Sigma, P_*, I_*)$ for L_* with the following properties:

- there are at most r nonterminals on the right-hand side of every production in P_*,
- G_* has an SPP $(X_A)_{A \in N_*}$ consisting of sets of the form
$$\left(C_1^{\langle L_* |} \cap D_1^{\langle \overline{L_*} |}\right) \cup \cdots \cup \left(C_m^{\langle L_* |} \cap D_m^{\langle \overline{L_*} |}\right), \tag{4}$$
  where for $j = 1, \ldots, m$, $C_j, D_j \subseteq \mathrm{Con}(L_*)$, $1 \leq |C_j| \leq k$, and $0 \leq |D_j| \leq l$.

Let J be the union of all the sets C_j, D_j that appear in the description (4) of the components of the SPP $(X_A)_{A \in N_*}$ for G_*. Since $t_1, t_2, \ldots$ enumerates L_*, there exists an i such that $J \subseteq \mathrm{Con}(T_i)$.

Case 1. $T_l \subseteq L(G_{l-1})$ for all $l \geq i$. In this case, $L_* \subseteq L(G_l)$ for all $l \geq i-1$. Also, for all $l \geq i$, we have $J_l = J_{i-1}$, $H_l = H_{i-1}$, $N_l = N_{i-1}$, and since $E_l \supseteq E_{l-1}$,

$$P_l \subseteq P_{l-1}\,, \quad I_l \subseteq I_{l-1}\,.$$

Since P_{i-1} and I_{i-1} are finite, P_l and I_l, and hence G_l, will eventually stabilize. When that happens, all productions in P_l will be valid on $\bigcup_l E_l = \mathrm{Sub}(L_*)$, and all nonterminals

$B \in I_l$ will satisfy $\bigcup_l E_l \cap [\![B]\!]^{L_*} = \mathrm{Sub}(L_*) \cap [\![B]\!]^{L_*} = [\![B]\!]^{L_*} \subseteq L_* \subseteq L(G_l)$. It follows that $([\![B]\!]^{L_*})_{B \in N_l}$ is an SPP of G_l. Since $(L(G_l, B))_{B \in N_l}$ is the least pre-fixed point of G_l, we also have $L(G_l) = \bigcup_{B \in I_l} L(G_l, B) \subseteq \bigcup_{B \in I_l} [\![B]\!]^{L_*} \subseteq L_*$. So $L(G_l) = L_*$.

Case 2. $T_l \not\subseteq L(G_{l-1})$ for some $l \geq i$. Then $J_l = \mathrm{Con}(T_l) \supseteq J$. For each $A \in N_*$, N_l contains a nonterminal $\hat{A} = \{(C_1, D_1), \ldots, (C_m, D_m)\}$ corresponding to the description of the form (4) of X_A, which is to say $[\![\hat{A}]\!]^{L_*} = X_A$. The fact that $([\![\hat{A}]\!]^{L_*})_{A \in N_*}$ is an SPP of G_* implies the following:

- for each production $A_0 \to w_0\, A_1\, w_1\, \ldots\, A_n\, w_n$ in P_*, the corresponding production

$$\hat{A}_0 \to w_0\, \hat{A}_1\, w_1\, \ldots\, \hat{A}_n\, w_n$$

  is valid and hence is in P_l;

- for each $A \in I_*$, the corresponding nonterminal $\hat{A}$ satisfies $[\![\hat{A}]\!]^{L_*} \subseteq L_*$ and hence is in I_l.

It follows that G_l contains a "homomorphic image" of G_*, which implies $L_* \subseteq L(G_l)$. It is easy to see that this will continue to be the case at all stages $j \geq l$. The same reasoning as in Case 1 shows that G_j will eventually stabilize to a correct grammar for L_* and $([\![B]\!]^{L_*})_{B \in N_j}$ will be an SPP of G_j. ■

4. A Language Outside of the Hierarchy

We show that there is a context-free language that Algorithm 1 does not learn for any choice of r, k, l, m.

For $x \in \{a, b\}^*$, let $\delta(x) = |x|_a - |x|_b$, where $|x|_c$ denotes the number of occurrences of c in x. For $X \subseteq \{a, b\}^*$, we let $\delta(X) = \{\, \delta(x) \mid x \in X \,\}$.

Proposition 4 *The language* $\overline{O_2} = \{\, x \in \{a, b\}^* \mid \delta(x) \neq 0 \,\}$ *does not belong to* $\mathbf{FCP}(k, l, m)$ *for any* k, l, m.

Proof Let $G = (N, \{a, b\}, P, I)$ be a CFG for $\overline{O_2}$. Applying the pumping lemma to a sufficiently long string in $\overline{O_2}$ of the form a^p, we obtain

$$S \Rightarrow_G^* a^{h_1} A a^{h_2},$$
$$A \Rightarrow_G^+ a^{i_1} A a^{i_2},$$
$$A \Rightarrow_G^+ a^j$$

such that $S \in I$, $A \in N$, and $i_1 + i_2 > 0$. Let $(X_B)_{B \in N}$ be any SPP of G. Then we must have

$$a^{h_1 + n i_1} X_A a^{n i_2 + h_2} \subseteq \overline{O_2} \qquad \text{for all } n \geq 0, \tag{5}$$
$$a^{n i_1 + j + n i_2} \in X_A \qquad \text{for all } n \geq 0. \tag{6}$$

The property (5) implies that $\delta(X_A)$ is co-infinite and the property (6) implies that $\delta(X_A)$ is infinite. Note that for every $(u, v) \in \{a, b\}^* \times \{a, b\}^*$, we have $\delta(\{(u, v)\}^{\langle \overline{O_2} |}) = \mathbb{Z} -$

$\{-\delta(uv)\}$, which is a co-finite set. Since any set $X \in \mathrm{FC}_{\overline{O_2}}(k,l,m)$ is a Boolean combination of sets of the form $\{(u,v)\}^{\langle \overline{O_2}|}$ and it is easy to see that δ commutes with Boolean operations on $\mathrm{FC}_{\overline{O_2}}(k,l,m)$, it follows that $\delta(X)$ is either finite or co-finite. So X_A cannot belong to $\mathrm{FC}_{\overline{O_2}}(k,l,m)$ for any k,l,m. ■

5. Hierarchy Theorems

Theorem 5 *For each $l \geq 1$,*

$$\mathbf{FCP}_1(1,l) - \bigcup_{k \geq 1} \mathbf{FCP}(k, l-1) \neq \varnothing.$$

Proof Consider the context-free grammar $G_l = (\{S, S_0, S_1, \ldots, S_l\}, \{a, b, \#, d\}, P_l, \{S\})$, with the following productions:

$$\begin{aligned}
S &\to S_0 \\
S &\to S_i \mid S_i\#d^i \qquad (1 \leq i \leq l) \\
S_0 &\to ab \mid ab^2 \mid aS_0b \mid aS_0b^2 \\
S_i &\to ab^{2i+1} \mid ab^{2i+2} \mid aS_ib^{2i+1} \mid aS_ib^{2i+2} \quad (1 \leq i \leq l)
\end{aligned}$$

Let $L_l = L(G_l)$. Writing A for $L(G_l, A)$, we have

$$\begin{aligned}
S &= S_0 \cup S_1 \cup S_1\#d \cup \cdots \cup S_l \cup S_l\#d^l = L_l = \{(\varepsilon,\varepsilon)\}^{\triangleleft}, \\
S_0 &= \{\, a^{j_1}b^{j_2} \mid j_1 \leq j_2 \leq 2j_1 \,\} = \{(\varepsilon,\varepsilon)\}^{\triangleleft} \cap \overline{\{(\varepsilon,\#d), \ldots, (\varepsilon,\#d^l)\}^{\triangleleft}}, \\
S_i &= \{\, a^{j_1}b^{j_2} \mid (2i+1)j_1 \leq j_2 \leq (2i+2)j_1 \,\} = \{(\varepsilon,\#d^i)\}^{\triangleleft}.
\end{aligned}$$

This shows that $L_l \in \mathbf{FCP}_1(1,l)$.

To prove that $L_l \notin \mathbf{FCP}(k,l-1)$ for any k, let $G = (N, \{a,b,\#,d\}, P, I)$ be a context-free grammar for L_l. Let $p-1$ be the pumping number for G.

Claim 1 If $p \leq q \leq r \leq 2q$, then any pump in a derivation tree for $a^q b^r$ has a yield of the form (a^{j_1}, b^{j_2}), where $1 \leq j_1 \leq j_2 \leq 2j_1$.

Let (v,w) be the yield of a pump in a derivation tree for $a^q b^r$. It is clear that (v,w) must be of the form (a^{j_1}, b^{j_2}), where $0 < j_1 + j_2$. We have

$$a^{q+(n-1)j_1} b^{r+(n-1)j_2} \in L_l \qquad \text{for every } n \geq 0. \tag{7}$$

Since (7) implies $q + (n-1)j_1 \leq r + (n-1)j_2 \leq (2l+2)(q + (n-1)j_1)$ for all n, it is easy to see that $1 \leq j_1 \leq j_2$. To see $j_2 \leq 2j_1$, suppose by way of contradiction $2j_1 < j_2$. Letting $n = 0$ in (7), we have $a^{q-j_1}b^{r-j_2} \in L_l$, which implies $q - j_1 \leq r - j_2$, and hence $j_2 - j_1 \leq r - q \leq 2q - q = q$. Since $2j_1 < j_2$, there is a natural number m such that

$$2(q + (n-1)j_1) < r + (n-1)j_2 \qquad \text{for all } n \geq m.$$

So let m be the least such number. Clearly, $m \geq 2$. We have

$$\begin{aligned} r + (m-1)j_2 &= r + (m-2)j_2 + j_2 \\ &\leq 2(q + (m-2)j_1) + j_2 \qquad \text{(by the minimality of } m\text{)} \\ &= 2(q + (m-1)j_1) + j_2 - 2j_1 \\ &= 2(q + (m-1)j_1) + (j_2 - j_1) - j_1 \\ &\leq 2(q + (m-1)j_1) + q - j_1 \\ &< 2(q + (m-1)j_1) + q + (m-1)j_1 \\ &= 3(q + (m-1)j_1), \end{aligned}$$

so we have

$$2(q + (m-1)j_1) < r + (m-1)j_2 < 3(q + (m-1)j_1),$$

which contradicts (7). This proves Claim 1.

Claim 2

(i) For all q, any derivation tree for $a^{p+q}b^{2p+q}$ contains a pump with yield (a^{j_1}, b^{j_2}) such that $1 \leq j_1 < j_2 \leq 2j_1$.

(ii) For all q, any derivation tree for $a^{p+q}b^{p+2q}$ contains a pump with yield (a^{j_1}, b^{j_2}) such that $1 \leq j_1 \leq j_2 < 2j_1$.

Both parts are proved by induction on q.

(i). Any derivation tree τ for $a^{p+q}b^{2p+q}$ contains a pump with yield (v, w) such that $|v| + |w| < p$. By Claim 1, $(v, w) = (a^{j_1}, b^{j_2})$ for some j_1, j_2 such that $1 \leq j_1 \leq j_2 \leq 2j_1$. Note that $2j_1 \leq j_1 + j_2 < p$. If $j_1 < j_2$, we are done. If $j_1 = j_2$, then consider the result of deleting the pump from τ, which is a derivation tree τ_1 for $a^{p+q-j_1}b^{2p+q-j_1}$. If $q < j_1$, then

$$2(p + q - j_1) < 2p + q - j_1 < 2p + q - j_1 + p - 2j_1 \leq 3(p + q - j_1),$$

but this contradicts $a^{p+q-j_1}b^{2p+q-j_1} \in L_l$. If $j_1 \leq q$, then $q - j_1 \geq 0$ and we can apply the induction hypothesis to τ_1. We thus get a pump with yield (a^{k_1}, b^{k_2}) in τ_1 such that $1 \leq k_1 < k_2 \leq 2k_1$. In τ, the part that corresponds to the pump with yield (a^{k_1}, b^{k_2}) in τ_1 may or may not overlap with the pump with yield (a^{j_1}, b^{j_1}). If they overlap, the former must wholly contain the latter and is a pump with yield $(a^{j_1+k_1}, b^{j_1+k_2})$, which has the required property. If they do not overlap, the part of τ that corresponds to the pump with yield (a^{k_1}, b^{k_2}) in τ_1 is a pump with yield (a^{k_1}, b^{k_2}), again satisfying the required property.

The proof of (ii) is similar.

Claim 3 The grammar G has an initial nonterminal S and a nonterminal A such that

$$S \Rightarrow_G^* a^{i_1+nl_1}Ab^{nl_2+i_2} \qquad \text{for all } n, \tag{8}$$
$$A \Rightarrow_G^* a^{nm_1+i_3}b^{i_4+nm_2} \qquad \text{for all } n, \tag{9}$$

where

$$1 \leq l_1 \leq l_2 \leq 2l_1,$$
$$1 \leq m_1 < m_2 < 2m_1.$$

We apply Claim 2 to a derivation tree τ for the string

$$a^{2p}b^{3p}$$

and obtain a pump with yield (a^{j_1}, b^{j_2}) such that $1 \leq j_1 < j_2 \leq 2j_1$ and a pump with yield (a^{k_1}, b^{k_2}) such that $1 \leq k_1 \leq k_2 < 2k_1$. If the two pumps are the same, then $1 \leq j_1 < j_2 < 2j_2$, and we are done by letting $l_1 = m_1 = j_1$ and $l_2 = m_2 = j_2$ and A be the nonterminal at the root the pump. Otherwise, the two pumps may or may not overlap. If they do not, let (a^{l_1}, b^{l_2}) be the yield of the upper pump and A be the nonterminal at its root. We then have (8) for some i_1, i_2. By repeating both pumps n times, we get (9) for some i_2, i_4 with $m_1 = j_1 + k_1$ and $m_2 = j_2 + k_2$. If the two pumps overlap, then one wholly contains the other. Letting (a^{l_1}, b^{l_2}) be the yield of the outer pump, we get (8) for some i_1, i_2. Repeating the inner pump, we get a pump with yield $(a^{j_1+k_1}, b^{j_2+k_2})$ in a derivation tree for $a^{2p+j_1}b^{3p+j_2}$ or $a^{2p+k_1}b^{3p+k_2}$, depending on which pump is the inner one. Letting $m_1 = j_1 + k_1$ and $m_2 = j_2 + k_2$, we get (9). This proves Claim 3.

Now suppose that $(X_B)_{B \in N}$ is an SPP of G. Let S, A, l_1, l_2, m_1, m_2, etc., be as in Claim 3. By (8) and (9), we have

$$a^{i_1+nl_1} X_A b^{nl_2+i_2} \subseteq L_l \qquad \text{for all } n, \tag{10}$$

$$a^{nm_1+i_3} b^{i_4+nm_2} \in X_A \qquad \text{for all } n. \tag{11}$$

Assume that C_A and D_A are finite subsets of $\{a, b, \#, d\}^* \times \{a, b, \#, d\}^*$ such that $X_A = C_A^{\triangleleft} \cap D_A^{\overline{\triangleleft}}$. Our goal is to prove $|D_A| \geq l$.

Claim 4 $C_A \subseteq \{a\}^* \times \{b\}^*$ and $D_A \cap (\{a\}^* \times \{b\}^*) = \varnothing$.

Let

$$n = \max\{i_3, i_4\} + \max \bigcup_{(v,w) \in C_A \cup D_A} \{|v|, |w|\}.$$

Let $j_1, j_2 \leq \max \bigcup_{(v,w) \in C_A \cup D_A} \{|v|, |w|\}$. Since $m_1 < m_2 < 2m_1$,

$$\begin{aligned} j_1 + nm_1 + i_3 &\leq nm_1 + n \\ &= n(m_1 + 1) \\ &\leq nm_2 \\ &\leq i_4 + nm_2 + j_2 \\ &\leq nm_2 + n \\ &= n(m_2 + 1) \\ &\leq 2nm_1 \\ &\leq 2(j_1 + nm_1 + i_3). \end{aligned}$$

This means that

$$a^{j_1} a^{nm_1+i_3} b^{i_4+nm_2} b^{j_2} \in L_l,$$

$$a^{j_1} a^{nm_1+i_3} b^{i_4+nm_2} b^{j_2} \# d^i \notin L_l \qquad \text{for any } i.$$

It follows from (11) that $(a^{j_1}, b^{j_2} \# d^i) \notin C_A$ and $(a^{j_1}, b^{j_2}) \notin D_A$. This establishes Claim 4.

Claim 5 For each $i \in \{1, \ldots, l\}$, D_A contains a pair of the form $(a^{j_1}, b^{j_2}\#d^i)$.

Let

$$j = \max\left(\{i_1 + l_1\} \cup \bigcup_{(v,w)\in C_A \cup D_A} \{|v|, |w|\}\right).$$

Fix $i \in \{1, \ldots, l\}$, and consider the string

$$u_i = a^{(4i+3)j} b^{(2i+2)(4i+2)j},$$

which is in L_l, since $(2i+1)(4i+3)j < (2i+2)(4i+2)j < (2i+2)(4i+3)j$. If $j_1 \le j$ and $j_2 \le j$, we have

$$(2i+1)(j_1 + (4i+3)j) \le (2i+2)(4i+2)j + j_2 \le (2i+2)(j_1 + (4i+3)j), \tag{12}$$

since

$$\begin{aligned}
(2i+1)(j_1 + (4i+3)j) &\le (2i+1)(j + (4i+3)j) \\
&= (2i+1)((4i+4)j) \\
&= (2i+2)(4i+2)j \\
&\le (2i+2)(4i+2)j + j_2 \\
&\le (2i+2)(4i+2)j + j \\
&< (2i+2)(4i+3)j \\
&\le (2i+2)(j_1 + (4i+3)j).
\end{aligned}$$

So $a^{j_1} u_i b^{j_2} \in L_l$. This means that $u_i \in C_A^{\triangleleft}$. Since $l_2 \le 2l_1$, there is an m such that

$$(2i+2)(4i+2)j + nl_2 + i_2 < 3(i_1 + nl_1 + (4i+3)j)$$

for all $n \ge m$. So let m be the least such number. Then $m \ge 2$, since

$$\begin{aligned}
3(i_1 + l_1 + (4i+3)j) &\le 3(j + (4i+3)j) \\
&= 6(2i+2)j \\
&\le (2i+2)(4i+2)j + l_2 + i_2.
\end{aligned}$$

We have

$$\begin{aligned}
2(i_1 + ml_1 + (4i+3)j) &= 2(i_1 + (m-1)l_1 + (4i+3)j) + 2l_1 \\
&= 3(i_1 + (m-1)l_1 + (4i+3)j) - (i_1 + (m-1)l_1 + (4i+3)j) + 2l_1 \\
&\le (2i+2)(4i+2)j + (m-1)l_2 + i_2 \\
&\qquad - (i_1 + (m-1)l_1 + (4i+3)j) + 2l_1 \\
&= (2i+2)(4i+2)j + ml_2 + i_2 \\
&\qquad - (i_1 + (m-2)l_1 + (4i+3)j) + l_1 - l_2 \\
&< (2i+2)(4i+2)j + ml_2 + i_2,
\end{aligned}$$

since $m \geq 2$, $l_1 \leq l_2$, and $j \geq l_1 \geq 1$. So

$$2(i_1 + ml_1 + (4i+3)j) < (2i+2)(4i+2)j + ml_2 + i_2 < 3(i_1 + ml_1 + (4i+3)j),$$

which implies

$$a^{i_1+ml_1} u_i b^{ml_2+i_2} = a^{i_1+ml_1} a^{(4i+3)j} b^{(2i+2)(4i+2)j} b^{ml_2+i_2} \notin L_l.$$

By (10), it follows that $u_i \notin X_A$. Since we have already shown that $u_i \in C_A^{\triangleleft}$, it must be that $u_i \notin D_A^{\overline{\triangleleft}}$. So there is a pair $(v, w) \in D_A$ such that $vu_iw \in L_l$. Since $D_A \cap (\{a\}^* \times \{b\}^*) = \varnothing$, it is clear that (v, w) must be of the form $(a^{j_1}, b^{j_2}\#d^h)$. But (12) means that we must have $h = i$. This proves Claim 5.

Having established Claim 5, we can immediately conclude $|D_A| \geq l$. ■

Theorem 6 *For each* $k \geq 2$,

$$\mathbf{FCP}_1(k, 1) - \bigcup_{l \geq 0} \mathbf{FCP}(k-1, l) \neq \varnothing.$$

Proof Consider the context-free grammar $G_k = (\{S, S_0, S_1, \ldots, S_k\}, \{a, b, \#, d\}, P_k, \{S\})$, with the following productions:

$$\begin{aligned}
S &\to S_0\#d \mid \cdots \mid S_0\#d^k \\
S &\to S_i\#d \mid \cdots \mid S_i\#d^{i-1} \mid S_i\#d^{i+1} \mid \cdots \mid S_i\#d^k \qquad (1 \leq i \leq k) \\
S_0 &\to ab \mid abb \mid aS_0b \mid aS_0bb \\
S_i &\to ab^{2i+1} \mid ab^{2i+2} \mid aS_ib^{2i+1} \mid aS_ib^{2i+2} \qquad (1 \leq i \leq k)
\end{aligned}$$

Let $L_k = L(G_k)$. We have

$$\begin{aligned}
L(G_k, S) &= \{(\varepsilon, \varepsilon)\}^{\triangleleft}, \\
L(G_k, S_0) &= \{(\varepsilon, \#d), \ldots, (\varepsilon, \#d^k)\}^{\triangleleft}, \\
L(G_k, S_i) &= \{(\varepsilon, \#d), \ldots, (\varepsilon, \#d^{i-1}), (\varepsilon, \#d^{i+1}), \ldots, (\varepsilon, \#d^k)\}^{\triangleleft} \cap \{(\varepsilon, \#d^i)\}^{\overline{\triangleleft}}.
\end{aligned}$$

This shows that $L_k \in \mathbf{FCP}_1(k, 1)$. A proof analogous to that of Theorem 5 shows that $L_k \notin \mathbf{FCP}(k-1, l)$ for any l. ■

We also have

Theorem 7 *For each* $k \geq 5$,

$$\mathbf{FCP}(k, 0) - \bigcup_{l \geq 0} \mathbf{FCP}(k-1, l) \neq \varnothing. \tag{13}$$

Proof We can show that the context-free language

$$L_k = \{\, a^{n_0}ba^{n_1}b \ldots ba^{n_k}ba^{n_{k+1}} \mid \text{there are } i, j \text{ such that } n_i = n_j \text{ and } 1 \leq i < j \leq k \,\}.$$

belongs to $\mathbf{FCP}(k, 0) - \bigcup_{l \geq 0} \mathbf{FCP}(k-1, l)$. We omit the details. ■

We leave open the question of whether (13) holds for $2 \leq k \leq 4$.

6. Conclusion

In previous works on distributional learning, nonterminals of CFGs were associated with *closed* sets of strings, i.e., sets of the form $C^{\triangleleft}$ for some $C \subseteq \Sigma^* \times \Sigma^*$. The present work shows that the closedness of the associated sets in this sense was not essential to the success of the "dual" learning algorithms. Closed sets, and more generally sets of the form $C_1^{\triangleleft} \cup \cdots \cup C_m^{\triangleleft}$, are *upper sets* with respect to the *syntactic quasiorder* $\preccurlyeq_L$ on Σ^* defined by $x \preccurlyeq_L y \iff \{ (u,v) \in \Sigma^* \times \Sigma^* \mid uxv \in L \} \subseteq \{ (u,v) \in \Sigma^* \times \Sigma^* \mid uyv \in L \}$.[3] Sets of the form $C^{\triangleleft} \cap D^{\overline{\triangleleft}}$ satisfy the weaker property that $x \preccurlyeq_L y \preccurlyeq_L z$ and $\{x, z\} \subseteq C^{\triangleleft} \cap D^{\overline{\triangleleft}}$ together imply $y \in C^{\triangleleft} \cap D^{\overline{\triangleleft}}$. Sets of the form $(C_1^{\triangleleft} \cap D_1^{\overline{\triangleleft}}) \cup \cdots \cup (C_m^{\triangleleft} \cap D_m^{\overline{\triangleleft}})$ do not even satisfy this weaker property. None of this matters to Algorithm 1.[4]

Our first hierarchy theorem (Theorem 5) shows that the generalization from using sets of the form $C^{\triangleleft}$ to using sets of the form $C^{\triangleleft} \cap D^{\overline{\triangleleft}}$ significantly enlarges the class of context-free languages that can be learned. We do not know whether the further extension to sets of the form $(C_1^{\triangleleft} \cap D_1^{\overline{\triangleleft}}) \cup \cdots \cup (C_m^{\triangleleft} \cap D_m^{\overline{\triangleleft}})$ similarly leads to a wider class of context-free languages. We leave this question for further work.

As regards sets of the form $C^{\triangleleft} \cap D^{\overline{\triangleleft}}$, it is perhaps worth mentioning that while they are not closed in the sense that they are fixed points of the operator $(\cdot)^{\triangleright\triangleleft}$, they are closed with respect to another closure operator. For $X \subseteq \Sigma^*$, let

$$X^{\blacktriangleright} = (X^{\triangleright}, X^{\overline{\triangleright}}).$$

For $(C, D) \in \mathscr{P}(\Sigma^* \times \Sigma^*) \times \mathscr{P}(\Sigma^* \times \Sigma^*)$, let

$$(C, D)^{\blacktriangleleft} = C^{\triangleleft} \cap D^{\overline{\triangleleft}}.$$

Then we have

$$X^{\blacktriangleright\blacktriangleleft} = X^{\triangleright\triangleleft} \cap X^{\overline{\triangleright\triangleleft}},$$

and we can see that $(\cdot)^{\blacktriangleright\blacktriangleleft}$ is a closure operator. The two operators $\blacktriangleright$ and $\blacktriangleleft$ give rise to an "(antitone) Galois connection" between the partially ordered sets $(\mathscr{P}(\Sigma^*), \subseteq)$ and $(\mathscr{P}(\Sigma^* \times \Sigma^*) \times \mathscr{P}(\Sigma^* \times \Sigma^*), \subseteq)$ (the latter ordered by componentwise inclusion), with all the properties familiar from Clark's (2015) *syntactic concept lattice*. In particular, sets of

3. The syntactic quasiorder was introduced by Schützenberger (1956) and Pin (1995). See also Almeida et al. (2015); Almeida and Klíma (2019).

4. When $(X_A)_{A \in N}$ is an SPP of $G = (N, \Sigma, P, I)$, so is $(X_A^{\triangleright\triangleleft})_{A \in N}$, so one might be tempted to view the nonterminals

$$B = ((C_1, D_1), \dots, (C_m, D_m))$$

used by Algorithm 1 as representing the closed sets $(\llbracket B \rrbracket^{L_*})^{|L_*\rangle\langle L_*|}$, rather than $\llbracket B \rrbracket^{L_*}$. However, this view does not jibe with the behavior of the algorithm. When a hypothesized production

$$B_0 \to w_0\, B_1\, w_1\, \dots\, B_n\, w_n$$

is discarded by the algorithm because it is not valid on E_i, it might still be the case that

$$(\llbracket B_0 \rrbracket^{L_*})^{|L_*\rangle\langle L_*|} \supseteq w_0\, (\llbracket B_1 \rrbracket^{L_*})^{|L_*\rangle\langle L_*|}\, w_1\, \dots\, (\llbracket B_n \rrbracket^{L_*})^{|L_*\rangle\langle L_*|}\, w_n.$$

(To see this, take $n = 0$ and any $w_0 \in (\llbracket B_0 \rrbracket^{L_*})^{|L_*\rangle\langle L_*|} - \llbracket B_0 \rrbracket^{L_*}$.) In other words, when L_* has a grammar $G_* = (N_*, \Sigma, P_*, I_*)$ with an SPP each of whose components is of the form $(\llbracket B \rrbracket^{L_*})^{|L_*\rangle\langle L_*|}$, there is no guarantee that Algorithm 1 learns it.

the form $(C, D)^{\blacktriangleleft}$ are closed sets with respect to $(\cdot)^{\blacktriangleright\blacktriangleleft}$, and for any $X, Y \in \mathscr{P}(\Sigma^*)$, we have $(XY)^{\blacktriangleright\blacktriangleleft} = (X^{\blacktriangleright\blacktriangleleft} Y^{\blacktriangleright\blacktriangleleft})^{\blacktriangleright\blacktriangleleft}$. While these properties of the sets $C^{\triangleleft} \cap D^{\overline{\triangleleft}}$ may be interesting in their own right, we stress that they play no role in our learnability result (Theorem 3).

Although Algorithm 1 does not rely on the properties of closed sets, it is still a "distributional" learning algorithm in that it classifies strings according to their distributions (i.e., the contexts in which they occur) in the target language.

Finally, one might wonder how the idea of using contexts both positively and negatively might be adapted to the "primal" approach to distributional learning, where finite sets of strings are used as nonterminals of the hypothesized grammar.[5] For the primal learner, a hypothesized production

$$K_0 \to w_0 \, K_1 \, w_1 \, \dots \, K_n \, w_n \, ,$$

where each K_i is a finite subset of Σ^*, is valid if

$$K_0^{\triangleright\triangleleft} \supseteq w_0 \, K_1^{\triangleright\triangleleft} \, w_1 \, \dots \, K_n^{\triangleright\triangleleft} \, w_n \, .$$

Crucially, this is equivalent to

$$K_0^{\triangleright} \subseteq (w_0 \, K_1 \, w_1 \, \dots \, K_n \, w_n)^{\triangleright} \, ,$$

which can then be approximated by

$$J \cap K_0^{\triangleright} \subseteq (w_0 \, K_1 \, w_1 \, \dots \, K_n \, w_n)^{\triangleright} \, ,$$

where J is the set of contexts contained in the positive data given to the learner so far. The last inclusion can be determined by a finite number of membership queries. One could replace $\triangleleft$ and $\triangleright$ in this approach by $\blacktriangleleft$ and $\blacktriangleright$, and everything would work the same way as before. It is easy to see that $(K_A^{\blacktriangleright\blacktriangleleft})_{A \in N}$ is an SPP of $G = (N, \Sigma, P, I)$ if and only if $(K_A^{\triangleright\triangleleft})_{A \in N}$ is an SPP of G and $(K_A^{\overline{\triangleright\triangleleft}})_{A \in N}$ is a pre-fixed point of G, so this new primal learner would be less general than the original one. It may still be interesting to investigate it, since it would produce a more compact context-free grammar, with fewer productions.

Acknowledgments

We thank one of the reviewers for raising some of the points that are addressed in the concluding section. This work was supported by JSPS KAKENHI Grant Numbers 17K00026 and 18K11150.

References

J. Almeida, A. Cano, O. Klíma, and J.-É. Pin. On fixed points of the lower set operator. *International Journal of Algebra and Computation*, 25(01n02):259–292, 2015. doi: 10.1142/S021819671540010X. URL https://doi.org/10.1142/S021819671540010X.

5. One could also consider using two finite sets of strings, one positively and the other negatively, to define a set of contexts, completely symmetrically to how we used two sets of contexts to define a set of strings. The set of contexts obtained this way can then be used to define a set of strings, and we may study CFGs that have SPPs whose components are obtained this way from two finite sets of strings. While this seems to lead to another hierarchy of context-free languages, it is not clear how to design a learning algorithm for such CFGs.

Jorge Almeida and Ondřej Klíma. On the insertion of n-powers. *Discrete Mathematics and Theoretical Computer Science*, 21(3), 2019. doi: 10.23638/DMTCS-21-3-5. URL https://dmtcs.episciences.org/5155.

Alexander Clark. The syntactic concept lattice: Another algebraic theory of the context-free languages? *Journal of Logic and Computation*, 25(5):1203–1229, 2015. doi: 10.1093/logcom/ext037. First published online: July 30, 2013.

Alexander Clark and Ryo Yoshinaka. 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, 2016.

Alexander Clark, Makoto Kanazawa, Gregory M. Kobele, and Ryo Yoshinaka. Distributional learning of some nonlinear tree grammars. *Fundamenta Informaticae*, 146(4): 339–377, 2016.

Makoto Kanazawa and Ryo Yoshinaka. The strong, weak, and very weak finite context and kernel properties. In Frank Drewes, Carlos Martín-Vide, and Bianca Truthe, editors, *Language and Automata Theory and Applications*, pages 77–88, Cham, 2017. Springer International Publishing. ISBN 978-3-319-53733-7. doi: 10.1007/978-3-319-53733-7_5.

Jean-Eric Pin. A variety theorem without complementation. *Russian Mathematics*, 39: 80–90, 1995.

M. P. Schützenberger. Une théorie algébraique du codage. *Séminaire P. Dubreil et C. Pisot. Algèbre et théorie des nombres, Année 1955/1956*, 1956.