Identification in the Limit of k, l-Substitutable Context-Free Languages

Ryo Yoshinaka

Graduate School of Information Science and Technology, Hokkaido University,
North-14 West-9, Sapporo, Japan
ry@ist.hokudai.ac.jp

Abstract. Recently Clark and Eyraud (2005, 2007) have shown that substitutable context-free languages are polynomial-time identifiable in the limit from positive data. Substitutability in context-free languages can be thought of as the analogue of reversibility in regular languages. While reversible languages admit a hierarchy, namely k-reversible regular languages for each nonnegative integer k, Clark and Eyraud targeted the subclass of context-free languages that corresponds to zero-reversible regular languages only. Following Clark and Eyraud's proposal, this paper introduces a hierarchy of substitutable context-free languages as the analogue of that of k-reversible regular languages and shows that each class in the hierarchy is also polynomial-time identifiable in the limit from positive data.

1 Introduction

Efficient learning of context-free languages is a topical issue on grammatical inference (see e.g. de la Higuera [12], Lee [18]), but not many techniques are known to be applicable to identification in the limit from positive data of non-regular subclasses of context-free languages, in comparison with subclasses of regular languages (see e.g. Lange et al. [17]). Recently Clark and Eyraud [6, 7] have shown that *substitutable context-free languages* are polynomial-time identifiable in the limit from positive data. Their work is remarkable among other achievements on learning context-free languages in several regards. One is the efficiency of the learning algorithm. Their algorithm for substitutable context-free languages runs in time polynomial in the size of the given data and it admits a set of positive examples of polynomial cardinality in the description size of the target grammar on which the conjecture converges to the target language. The second virtue is that the notion of substitutability can explain an aspect of natural language phenomena, which meets the very first motivation of grammatical inference [9]. A language L is said to be substitutable if and only if

$$x_1y_1z_1, x_1y_2z_1, x_2y_1z_2 \in L \text{ implies } x_2y_2z_2 \in L$$

for any strings $x_1, y_1, z_1, x_2, y_2, z_2$. From the point of view of formal language theory, substitutability of context-free languages can be thought of as the exact analogue of *zero-reversibility* in regular languages. Angluin [1] introduced

A. Clark, F. Coste, and L. Miclet (Eds.): ICGI 2008, LNAI 5278, pp. 266–279, 2008.

the hierarchy of k-reversible languages for nonnegative integers k and showed polynomial-time learnability of k-reversible regular languages.[1] A language L is k-reversible if and only if

$$x_1vy_1, x_1vy_2, x_2vy_1 \in L \text{ implies } x_2vy_2 \in L$$

where the length of v is k. As the literature has paid much attention to reversible regular languages and their variants and obtained many fruitful results (e.g., [2, 14,16,15,13,19,21,23]), the close relation of substitutable context-free languages to reversible regular languages also seems an advantage of their study. In fact Clark and Eyraud [7] suggested that one may define for context-free languages the exact analogue of k-reversibility in regular languages and that such classes would be still polynomial-time identifiable in the limit from positive data. This paper answers to those expectations in the affirmative. We call a language L k, l-substitutable if and only if

$$x_1vy_1uz_1, x_1vy_2uz_1, x_2vy_1uz_2 \in L \text{ implies } x_2vy_2uz_2 \in L$$

where the length of v is k and that of u is l. This paper proves that k, l-substitutable context-free languages are identifiable in the limit from positive data by a polynomial-time algorithm that is a natural generalization of Clark and Eyraud's one.

2 Definitions

We start by some standard notation, most of which follows Clark and Eyraud [7]. Let Σ be a non-empty finite set. $|\Sigma|$ denotes its cardinality. If x is a finite sequence consisting of elements of Σ, it is called a *string (over Σ)* and $|x|$ denotes its length. λ is the *empty string*. Σ^* denotes the set of all strings over Σ, $\Sigma^+ = \Sigma^* - \{\lambda\}$, $\Sigma^k = \{ x \in \Sigma^* \mid |x| = k \}$, $\Sigma^{\leq k} = \{ x \in \Sigma^* \mid |x| \leq k \}$ and $\Sigma^{<k} = \Sigma^{\leq k} - \Sigma^k$. For $x \in \Sigma^*$ and $a \in \Sigma$, $|x|_a$ denotes the number of occurrences of a in x. Any subset of Σ^* is called a *language (over Σ)*. If L is a finite language over Σ, its size is defined as $\|L\| = \sum_{w \in L} |w|$. We shall assume an order $\prec$ or $\preceq$ on Σ which we shall extend to Σ^* in the canonical way by saying that $u \prec v$ if either $|u| < |v|$ or $|u| = |v|$ and u is lexicographically before v.

A *context-free grammar (*CFG*)* is denoted by a quadruple $G = \langle \Sigma, V, P, S \rangle$, where Σ is the finite set of *terminal symbols*, V, disjoint from Σ, is the finite set of *nonterminal symbols*, P is the finite set of *production rules* and $S \in N$ is the *start symbol*. A production rule in P has the form $A \to \beta$ for some $A \in V$ and $\beta \in (\Sigma \cup V)^+$. If $A \to \beta \in P$, we write $\alpha A \gamma \Rightarrow_G \alpha\beta\gamma$ for any $\alpha, \gamma \in (\Sigma \cup V)^*$. $\Rightarrow_G^+$ is the transitive closure of $\Rightarrow_G$ and $\Rightarrow_G^*$ is the reflexive and transitive closure of $\Rightarrow_G$. The subscript G of $\Rightarrow_G$ is omitted if it is understood from the context. The *context-free language* (CFL) $\mathcal{L}(G)$ generated by G is the set $\mathcal{L}(G, S)$, where $\mathcal{L}(G, \alpha) = \{ w \in \Sigma^* \mid \alpha \stackrel{*}{\Rightarrow} w \}$ for $\alpha \in (\Sigma \cup V)^*$. Two grammars G_1 and

[1] Angluin defined k-reversible languages as a subclass of regular languages, while this paper calls any language satisfying k-reversibility a k-reversible language.

G_2 are *equivalent* iff $\mathcal{L}(G_1) = \mathcal{L}(G_2)$. The description size of G is defined as $\|G\| = \sum_{A\to\beta\in P}(|A\beta|)$. A symbol $A \in \Sigma \cup V$ is *useless* in G if there are no $x, y, z \in \Sigma^*$ such that $S \overset{*}{\Rightarrow} xAz \overset{*}{\Rightarrow} xyz$. A CFG G is *reduced* iff every $A \in \Sigma \cup V$ is not useless. We assume all grammars to be reduced in this paper. Note that we do not allow empty right hand side to production rules, and thus any CFLs dealt with in this paper are λ-free.

In the following, terminal symbols will be indicated by $a, b, c, \ldots$, nonterminal symbols by A, B, strings over Σ by $u, v, \ldots, z$, and strings over $(\Sigma \cup V)^*$ by $\alpha, \beta, \gamma, \delta$.

We now define our learning criterion. This is *identification in the limit from text* (or equivalently *from positive data*) as defined by Gold [9]. Let $\mathbb{R}$ be any recursive set of finite descriptions, say CFGs, and $\mathcal{L}$ be a function from $\mathbb{R}$ to non-empty languages over Σ. A learning algorithm $\mathcal{A}$ on $\mathbb{R}$, is an algorithm that computes a function from finite sequences of strings $w_1, \ldots, w_n \in \Sigma^*$ to $\mathbb{R}$. We define a *presentation* of a language L to be an infinite sequence of elements (called *positive examples*) of L such that every element of L occurs at least once. Given a presentation, we can consider the sequence of hypotheses that the algorithm produces, writing $R_n = \mathcal{A}(w_1, \ldots, w_n)$ for the nth such hypothesis. The algorithm $\mathcal{A}$ is said to *identify the class* $\mathbb{L}$ *of languages in the limit from positive data* if for every $L \in \mathbb{L}$, for every presentation of L, there is an integer n_0 such that for all $n > n_0$, $R_n = R_{n_0}$ and $L = \mathcal{L}(R_{n_0})$. For $\mathbb{R}' \subseteq \mathbb{R}$ satisfying that $\mathbb{L} = \{\, \mathcal{L}(R) \mid R \in \mathbb{R}' \,\}$, one also says $\mathcal{A}$ *identifies* $\mathbb{R}'$ *in the limit from positive data*. For convenience, we often allow the learner to refer to the previous hypothesis R_n for computing R_{n+1} in addition to $w_1, \ldots, w_{n+1}$. Obviously this relaxation does not effect the learnability of language classes. Moreover, learning algorithms in this paper compute hypotheses from a set of positive examples by identifying a sequence with the set consisting of the elements of the sequence.

We further require that the algorithm needs only polynomially bounded amounts of data and computation. De la Higuera's proposal is to measure the efficiency by a set of examples on which the learner converges to a representation of the target language [11].

Definition 1 (de la Higuera [11]). A representation class $\mathbb{R}$ is *identifiable in the limit from positive data with polynomial time and data* if and only if there exist two polynomials p and q and an algorithm $\mathcal{A}$ such that

1. Given a set S of positive examples of size $\|S\| = m$, $\mathcal{A}$ returns a hypothesis in time $p(m)$,
2. For each representation $R \in \mathbb{R}$ of size n, there exists a *characteristic set* CS of size less than $q(n)$ such that if $CS \subseteq S$, $\mathcal{A}$ returns a representation R_0 such that $\mathcal{L}(R) = \mathcal{L}(R_0)$.

The first condition (*polynomial updating time*) is widely accepted as a necessary condition for efficient learning. On the other hand, the second condition is somehow unsuitable as a model for efficient learning of CFGs, as this definition was initially designed for learning of regular languages. Even in a very restricted

kind of CFGs[2], like very simple grammars [25], the length of a shortest string in the language cannot be bounded by any polynomial in the size of a grammar. At present there is no consensus on the most appropriate modification of this criterion for learning of CFGs. Several ideas have been formulated to tackle this problem. Carme et al. [4] count the cardinality $|CS|$ of a characteristic set instead of the size $\|CS\|$. Wakatsuki and Tomita [24] have proposed to measure the complexity of an algorithm dealing with CFGs by another parameter τ_G, called *thickness*, defined by

$$\tau_G = \max\{\, |\omega(A)| \mid A \in V \,\} \text{ where } \omega(A) = \min\{\, w \in \Sigma^* \mid A \stackrel{*}{\underset{G}{\Rightarrow}} w \,\}$$

where "min" is with respect to $\prec$. We will show that our learning algorithm for k, l-substitutable context-free languages admits a characteristic set whose cardinality is bounded by a polynomial in the size of the target grammar and whose size is bounded by a polynomial in the thickness and the size of the target grammar.

We would like to remark that the notion of characteristic sets by de la Higuera [11] differs from that of *characteristic samples* by Angluin [1]. Let K be a finite subset of a language L and $\mathbb{L}$ a class of languages. We say that K is a *characteristic sample* of L with respect to $\mathbb{L}$ if it holds that

$$K \subseteq L' \text{ iff } L \subseteq L'$$

for any $L' \in \mathbb{L}$. The definition of a characteristic sample does not depend on any specific learning algorithm.

3 k, l-Substitutable Languages

Definition 2 (k, l-substitutability). Let k and l be nonnegative integers. A language L is said to be *k, l-substitutable* if and only if for any $x_1, y_1, z_1, x_2, y_2, z_2 \in \Sigma^*$, $v \in \Sigma^k$, $u \in \Sigma^l$ such that $vy_1u, vy_2u \neq \lambda$,

$$x_1vy_1uz_1, x_1vy_2uz_1, x_2vy_1uz_2 \in L \text{ implies } x_2vy_2uz_2 \in L.$$

The notion of substitutability by Clark and Eyraud [7] is exactly $0, 0$-substitutability in this paper and the condition $vy_1u, vy_2u \neq \lambda$ is essential only when $k = l = 0$ (otherwise trivially $vy_1u, vy_2u \neq \lambda$ holds). k, l-substitutability says nothing about strings of length shorter than $k + l$. $L \cup \{w\}$ is k, l-substitutable if and only if $L - \{w\}$ is for any $L \subseteq \Sigma^*$, $w \in \Sigma^{<k+l}$ and integers k and l.

It is obvious that if a language is k, l-substitutable, then it is m, n-substitutable for any $m \geq k$ and $n \geq l$. It is easy to see that the hierarchy is strict. For each $k, l \in \mathbb{N}$, there is a k, l-substitutable regular language that is m, n-substitutable if and only if $m \geq k$ and $n \geq l$. If a language is $k, 0$-substitutable or $0, k$-substitutable, then it is k-reversible. Recall that a language L is k-reversible if and only if for any $x_1, y_1, x_2, y_2 \in \Sigma^*$ and $v \in \Sigma^k$, $x_1vy_1, x_1vy_2, x_2vy_1 \in L$ implies $x_2vy_2 \in L$ [1].

[2] One exception is subclasses of linear grammars.

Proposition 1. *k,l-substitutable languages are not closed under intersection with regular sets, union, concatenation, complement, Kleene closure ($+,*$), λ-free homomorphism, inverse homomorphism. k,l-substitutable languages are closed under reversal if and only if $k=l$. k,l-substitutable languages are closed under intersection and λ-free inverse homomorphism.*

Proof. Let us say that a pair of strings (called a *context*) $\langle x,z\rangle$ is *applicable* to y in L if and only if $xyz\in L$.

Intersection with regular sets: Let $L_0 = ae^*ce^*a\cup ae^*de^*a\cup be^*ce^*b$ and $L_1 = L_0\cup be^*de^*b$. L_0 is regular and L_1 is $0,0$-substitutable. Clearly $L_1\cap L_0 = L_0$ is not k,l-substitutable for any k,l. The context $\langle a,a\rangle$ is applicable to e^kce^l and e^kde^l in L_0, but $\langle b,b\rangle$ is applicable only to e^kce^l.

Union: Let $L_2 = ae^*ce^*a\cup ae^*de^*a$ and $L_3 = be^*ce^*b$. L_2 and L_3 are both $0,0$-substitutable, but the union $L_2\cup L_3 = L_0$ is not k,l-substitutable for any k,l.

Concatenation: Languages $L_4 = ae^*c\cup ae^*d\cup b$ and $L_5 = e^*a\cup e^*ce^*b$ are $0,0$-substitutable. The concatenation L_4L_5 is not k,l-substitutable for any k,l, because $\langle a,a\rangle$ is applicable to e^kce^l and e^kde^l, but $\langle b,b\rangle$ is applicable only to e^kce^l.

Complement: $L_6 = a^*b$ is $0,0$-substitutable, but the complement $\overline{L_6}$ is not k,l-substitutable for any k,l, because while $\langle b,\lambda\rangle$ is applicable to both a^kaa^l and a^kba^l in $\overline{L_6}$, $\langle\lambda,b\rangle$ is applicable only to a^kba^l.

Kleene closure: $L_7 = \{\, a^nba^n \mid n\geq 0\,\}$ is $0,0$-substitutable, but neither L_7^+ nor L_7^* is k,l-substitutable. Let $m=\max\{k,l\}$. The context $\langle a^m,a^{m+1}\rangle$ is applicable to $ba^{2m+1}b$ and $ba^mba^{m+1}b$ in L_7^+, but $\langle a^{m+1},a^m\rangle$ is applicable only to $ba^{2m+1}b$ in L_7^*.

λ-free homomorphism: $L_8 = ae^*ce^*a\cup be^*ce^*b\cup fe^*de^*f$ is $0,0$-substi-tutable. Let h be the homomorphism that is almost the identify but $h(f)=a$, i.e., $h(a)=a$, $h(b)=b$, $h(c)=c$, $h(d)=d$, $h(e)=e$, $h(f)=a$. $h(L_8)=L_0$ is not k,l-substitutable.

Inverse homomorphism: $L_6 = a^*b$ is $0,0$-substitutable. Let h be such that $h(a)=a$, $h(b)=b$, $h(e)=\lambda$. $h^{-1}(L_6)$ is not k,l-substitutable, because $\langle\lambda,b\rangle$ is applicable to both e^kee^l and e^kae^l in $h^{-1}(L_6)$, but $\langle b,\lambda\rangle$ is applicable only to e^kee^l.

Reversal: If L is k,l-substitutable, its reversal L^R is trivially l,k-substi-tutable. It is enough to show that k,l-substitutable languages are not closed under reversal for $k>l\geq 0$. $L_9 = e^{k-1}ce^*\cup e^{k-1}de^*\cup ae^{k-1}ce^*a$ is $k,0$-substitutable, but its reversal L_9^R is not l,m-substitutable for any $m<k$ and l. In L_9^R, $\langle\lambda,\lambda\rangle$ is applicable to both e^lce^{k-1} and e^lde^{k-1}, but $\langle a,a\rangle$ is applicable only to e^lce^{k-1}.

Intersection: Let L and L' be k,l-substitutable. If $x_1vy_1uz_1, x_1vy_2uz_1$, $x_2vy_1uz_2\in L\cap L'$ for some $v\in\Sigma^k$, $u\in\Sigma^l$ and $vy_1u, vy_2u\in\Sigma^+$, then those are in both L and L'. Since L and L' are k,l-substitutable, $x_2vy_2uz_2$ is in both L and L' and thus in $L\cap L'$.

λ-FREE INVERSE HOMOMORPHISM: Let L be a k, l-substitutable language and h a λ-free homomorphism. We denote $h(w)$ by $\overline{w}$ for readability. If $x_1vy_1uz_1$, $x_1vy_2uz_1, x_2vy_1uz_2 \in h^{-1}(L)$ for some $v \in \Sigma^k$, $u \in \Sigma^l$ and $vy_1u, vy_2u \in \Sigma^+$, then $\overline{x_1}\overline{v}\overline{y_1}\overline{u}\overline{z_1}, \overline{x_1}\overline{v}\overline{y_2}\overline{u}\overline{z_1}, \overline{x_2}\overline{v}\overline{y_1}\overline{u}\overline{z_2} \in L$. Since L is k, l-substitutable and $|\overline{v}| \geq |v| = k$, $|\overline{u}| \geq |u| = l$, $|\overline{v}\overline{y_1}\overline{u}|, |\overline{v}\overline{y_2}\overline{u}| \geq 1$, we have $\overline{x_2}\overline{v}\overline{y_2}\overline{u}\overline{z_2} \in L$. This entails that $x_2vy_2uz_2 \in h^{-1}(L)$. □

We are particularly concerned with *k, l-substitutable context-free languages* (k, l-SCFLs) in this paper. As Clark and Eyraud [7] conjecture that all $0, 0$-SCFLs are NTS languages (see [22, 3] for the definition and properties of NTS languages), we conjecture all k, l-SCFLs are NTS too. The simple NTS example $\{ a^n b^n \mid n \geq 1 \}$ presented by Clark and Eyraud as a non-$0, 0$-substitutable language is $1, 1$-substitutable. The class of very simple languages is also an important subclass of CFLs due to the efficient identifiability in the limit from positive data [25, 26]. Clark and Eyraud show that the class of very simple languages and that of $0, 0$-SCFLs are incomparable. It is also the case for k, l-SCFLs. The language generated by the very simple grammar G consisting of two rules $S \to aSS$ and $S \to b$ is not k, l-substitutable for any k, l.

We note that Proposition 1 holds of classes of k, l-SCFLs except that k, l-SCFLs are not closed under intersection.

4 Learning Algorithm for k, l-Substitutable Context-Free Languages

Let us arbitrarily fix nonnegative integers k and l. Our learning target is the class of all *k, l-substitutable context-free languages* (k, l-SCFLs). However we do not yet have any grammatical characterization of this class. For mathematical completeness, yet we have to define our learning target by saying that our target representations are CFGs generating k, l-substitutable languages, though this property is not decidable. We remark that the class $\{ L \mid L \text{ is a } k, l\text{-SCFL for some } k, l \in \mathbb{N} \}$ is not identifiable in the limit from positive data, because this class is superfinite modulo λ, that is, it contains at least one infinite language and all the finite languages that do not contain λ. Obviously the absence of λ does not effect Gold's theorem [9] that any superfinite class is not identifiable in the limit from positive data.

Our learning algorithm for k, l-SCFLs is a natural generalization of Clark and Eyraud's original algorithm for $0, 0$-SCFLs [7]. However we omit the procedure in the original algorithm that constructs "the substitution graph" where potential nonterminal symbols that generate the same languages are merged. Though the procedure is important for making the output grammar more compact, we present a simpler learning algorithm and a simpler proof for the learnability instead.

Algorithm 1 is our learning algorithm k, l-SGL (k, l-Substitutable Grammar Learner) for learning k, l-SCFLs. If the new positive example is generated by the previous hypothesis by k, l-SGL, it keeps the hypothesis. Otherwise, let K be

the set of positive examples given so far. k,l-SGL computes the following CFG $\hat{G} = \langle \Sigma, V_K, P_K, S\rangle$ defined by

$$\begin{aligned} V_K &= \{\, [y] \mid xyz \in K,\, y \neq \lambda \,\} \cup \{S\}, \\ P_K &= \{\, [vyu] \to [vy'u] \mid xvyuz, xvy'uz \in K,\, |v| = k,\, |u| = l,\, vyu, vy'u \neq \lambda \,\} \\ &\quad \cup \{\, S \to [w] \mid w \in K \,\} \\ &\quad \cup \{\, [xy] \to [x][y] \mid [xy], [x], [y] \in V_K \,\} \\ &\quad \cup \{\, [a] \to a \mid a \in \Sigma \,\}. \end{aligned}$$

We note that k,l-SGL is specific to fixed nonnegative integers k and l. In other words, k and l are known to k,l-SGL a priori.

Algorithm 1. k,l-SGL

Data: A sequence of strings $w_1, w_2, \ldots$
Result: A sequence of CFGs $G_1, G_2, \ldots$
let $\hat{G}$ = CFG generating the empty language;
for $n = 1, 2, \ldots$ **do**
  read the next string w_n;
  if $w_n \notin \mathcal{L}(G)$ **then**
    let $\hat{G} = \langle \Sigma, V_K, P_K, S\rangle$ where $K = \{w_1, \ldots, w_n\}$;
  end if
  output $\hat{G}$;
end for

This section will establish the following main theorem of this paper.

Theorem 1. *The learning algorithm k,l-SGL identifies k,l-SCFLs in the limit from positive data with polynomial updating time. k,l-SGL admits a characteristic set K_G of polynomial cardinality in $\|G\|$ and of polynomial size in $\|G\|_{\tau_G}$ for the target grammar G.*

4.1 Proof That Hypothesized Language Is Not Too Large

First of all we shall show that k,l-SGL never hypothesizes too large a language.

Lemma 1. *If K is a finite subset of a k,l-substitutable language L, then $\mathcal{L}(\hat{G}) \subseteq L$.*

Proof. Let $\overline{(\cdot)}$ be the homomorphism from $(\Sigma \cup V_K - \{S\})^*$ to Σ^* such that $\bar{a} = a$ for all $a \in \Sigma$ and $\overline{[w]} = w$ for all $[w] \in V_K - \{S\}$. We prove by induction on the length of derivation that $S \Rightarrow_{\hat{G}}^{+} \alpha \in (\Sigma \cup V_K - \{S\})^*$ implies $\bar{\alpha} \in L$. Suppose that the last rule used in the derivation is of the form $S \to [w]$. Then $\overline{[w]} = w \in K \subseteq L$ by definition. Suppose that $S \Rightarrow_{\hat{G}}^{+} \alpha B \gamma \Rightarrow \alpha\beta\gamma$ for some rule $B \to \beta$ with $B \neq S$. The only nontrivial case is when $B = [vyu]$ and $\beta = [vy'u]$ for some $v \in \Sigma^k$, $u \in \Sigma^l$ and $y, y' \in \Sigma^*$. In this case, there are $x, z \in \Sigma^*$ such that $xvyuz, xvy'uz \in K \subseteq L$ by the definition of $\hat{G}$. By induction hypothesis, we have $\overline{\alpha B \gamma} = \bar{\alpha} vyu \bar{\gamma} \in L$. Since L is k,l-substitutable, this entails that $\bar{\alpha} vy'u \bar{\gamma} = \overline{\alpha\beta\gamma} \in L$. □

For some finite language K, $\hat{G}$ does not define a k,l-substitutable language.

Example 1. Let $k = l = 0$ and $K = \{ a, ab, abbc \}$. Because a and ab occur in the same context $\langle \lambda, \lambda \rangle$, if L is a $0,0$-substitutable language including K, then L is closed under substituting a for ab and we have $abc, ac \in L$ by $abbc \in L$. On the other hand, the output grammar $\hat{G}$ by the algorithm for the input K is equivalent to the grammar G consisting of the following rules:

$$S \to A,\ A \to a \mid AB \mid ABc,\ B \to b \mid BBc.$$

We have $ac \in L - \mathcal{L}(G)$. That is, $\mathcal{L}(\hat{G})$ is not $0,0$-substitutable.

Actually the least $0,0$-substitutable language including K of the above example is $a\{b,c\}^*$, which is indeed context-free and thus in the target class of our algorithm $0,0$-SGL. This means that even if a characteristic sample (in Angluin's sense [1]) of the target $0,0$-SCFL is given, our and Clark and Eyraud's learning algorithms do not necessarily converge to the target language.

4.2 Proof That Hypothesized Language Is Large Enough

To prove that the hypothesized language is large enough, we first need to define a characteristic set, that is to say a subset of a target language L_* which will ensure that the algorithm k,l-SGL will output a grammar $\hat{G}$ such that $\mathcal{L}(\hat{G}) = L_*$. We define a characteristic set in terms of a CFG in the following normal form, while we do not yet have any grammatical characterization on CFGs generating k,l-substitutable languages. Because Clark and Eyraud have already given a characteristic set of $0,0$-SCFLs for their algorithm and it works for our algorithm $0,0$-SGL, which is essentially the same as theirs, hereafter (including the next subsection) we target k,l-SCFLs with $\langle k,l \rangle \neq \langle 0,0 \rangle$.

Definition 3. Let k and l be nonnegative integers such that at least one of them is not zero. We say that a CFG $G = \langle \Sigma, V, P, S \rangle$ is in k,l-GNF if every production has the form $A \to w$ for some $w \in \Sigma^{\leq k+l} - \{\lambda\}$ or $A \to x\alpha z$ for some $x \in \Sigma^k$, $z \in \Sigma^l$ and $\alpha \in V^+$.

The notion of k,l-GNF is a generalization of Greibach normal form [10] and double Greibach normal form [20, 8]. Standard Greibach normal form is $1,0$-GNF and double Greibach normal form is $1,1$-GNF.

Lemma 2. *Let k and l be nonnegative integers such that at least one of them is not zero. For any* CFG *G, there is an equivalent* CFG *$G' = \langle \Sigma, V', P', S' \rangle$ in k,l-GNF such that $P' \subseteq V' \times (\Sigma^{\leq k+l} \cup \Sigma^k V'^{\leq 7(k+l)} \Sigma^l)$, $\|G'\|$ is polynomial in $\|G\|$ and $\tau_{G'}$ is polynomial in $\|G\|\tau_G$.*

Proof. CASE 1. $k,l \neq 0$. We would like to refer the reader to Engelfriet's conversion to double Greibach normal form of CFGs [8]. Observing his proof, one can see that every CFG in Chomsky normal form can be converted into an equivalent CFG whose productions have one of the following forms:

$$A \to a \text{ or } A \to a\alpha b$$

for some $A \in V$, $a, b \in \Sigma$ and $\alpha \in V^{\leq 7}$. Moreover, the size of the obtained grammar by his conversion is bounded by a polynomial in the size of the original grammar. Together with the well-known fact that any CFG can be transformed into Chomsky normal form of polynomial size, we may assume that $G = \langle \Sigma, V, P, S\rangle$ satisfies $P \subseteq V \times (\Sigma \cup \Sigma V^{\leq 7}\Sigma)$ without loss of generality.

Here we introduce a subrelation $\Rrightarrow$ of $\Rightarrow_G$. We write $\alpha \Rrightarrow \beta$ if either

- $\alpha = xA\delta$ and $\beta = x\gamma\delta$ for some $x \in \Sigma^{<k}$, $A \to \gamma \in P$ and $\delta \in (\Sigma \cup V)^*$,
- $\alpha = \delta Ax$ and $\beta = \delta\gamma x$ for some $x \in \Sigma^{<l}$, $A \to \gamma \in P$ and $\delta \in (\Sigma \cup V)^*$.

Let us define a CFG $G' = \langle \Sigma, V, P', S\rangle$ with

$$P' = \{\, A \to \alpha \mid A \overset{+}{\Rrightarrow} \alpha \in \Sigma^+ \cup (\Sigma^k(\Sigma \cup V)^*\Sigma^l) \,\}$$

where $\overset{+}{\Rrightarrow}$ is the transitive closure of $\Rrightarrow$. Some productions in P' may violate the condition of k,l-GNF, as some terminal symbols occur in α in a rule of the form $A \to x\alpha z$ with $x \in \Sigma^k$ and $z \in \Sigma^l$. A solution is trivial. For each terminal symbol $a \in \Sigma$, let us introduce a new nonterminal symbol N_a and a new production $N_a \to a$. Then we replace violating occurrences of terminal symbols a in productions by N_a. It is easy to see that $\mathcal{L}(G') = \mathcal{L}(G)$.

We evaluate the size of G'. Because G is in 1, 1-GNF, if $\alpha \Rrightarrow \beta$ and $\alpha \in \Sigma^m(\Sigma \cup V)^*\Sigma^n$, then either $\beta \in \Sigma^{m+1}(\Sigma \cup V)^*\Sigma^n$ or $\beta \in \Sigma^m(\Sigma \cup V)^*\Sigma^{n+1}$. Therefore, when A has n derivation steps induced by $\Rrightarrow$, i.e., $A \Rrightarrow \alpha_1 \Rrightarrow \ldots \Rrightarrow \alpha_n$, we have $n < k + l$ (note $\alpha_1 \in \Sigma V^+\Sigma \cup \Sigma^+$). Because the maximum length of production rules in P is at most 9, $\alpha \Rrightarrow \beta$ implies $|\beta| \leq |\alpha| + 8$. Thus if $A \Rrightarrow^+ \alpha$, then $|\alpha| \leq 1 + 8(k + l - 1) = 8(k + l) - 7$. If $\alpha = v\alpha' u$ for some $v \in \Sigma^k$ and $u \in \Sigma^l$, then $|\alpha'| \leq 7(k + l - 1)$. Moreover we see that $|P'| \leq |P|^{k+l-1} + |\Sigma|$. We have $\|G'\| \leq (8(k + l) - 7)(|P|^{k+l-1} + |\Sigma|) \in O(|P|^{k+l-1})$.

Moreover, it is not hard to see that when Engelfriet's conversion is applied to a CFG in Chomsky normal form obtained from a general CFG G'' by a reasonable method, then the thickness τ_G of the resultant grammar G in 1, 1-GNF is bounded by a polynomial in $\|G''\|\tau_{G''}$. By the fact $\tau_{G'} = \tau_G$, we get the lemma.

CASE 2. $k > 0$ and $l = 0$. Apply the similar conversion to Case 1 to CFG G in Greibach normal form such that $P \subseteq V \times \Sigma V^{\leq 2}$.

CASE 3. $k = 0$ and $l > 0$. This case is just symmetric to Case 2. □

It is easy to get rid of useless nonterminals in G' obtained by the above method if any.

Now we define a characteristic set K_G of a k,l-SCFL in terms of a reduced CFG $G = \langle \Sigma, V, P, S\rangle$ in k,l-GNF generating it as follows, where "min" is with respect to $\prec$, which is extended from Σ^* to $\Sigma^* \times \Sigma^*$ in some reasonable way:

$$\omega(\alpha) = \min\{\, w \in \Sigma^* \mid \alpha \overset{*}{\underset{G}{\Rightarrow}} w \,\} \text{ for } \alpha \in (\Sigma \cup V)^*,$$

$$\chi(A) = \min\{\, \langle x, z\rangle \in \Sigma^* \times \Sigma^* \mid S \overset{*}{\underset{G}{\Rightarrow}} xAz \,\} \text{ for } A \in V,$$

$$K_A = \{\, vw_1 \dots w_n u \in \Sigma^* \mid A \to vB_1 \dots B_n u, B_i \to \beta_i \in P,\ w_i = \omega(\beta_i) \,\}$$
$$\cup \{\, y \in \Sigma^* \mid A \to y \in P \,\} \text{ for } A \in V,$$
$$K_G = \{\, xyz \in \Sigma^* \mid \chi(A) = \langle x, z \rangle,\ y \in K_A,\ A \in V \,\}.$$

The following trivial lemma is implicitly used in the proof of Lemma 4.

Lemma 3. *$K_A \subseteq \mathcal{L}(G, A)$ and $K_S \subseteq K_G \subseteq \mathcal{L}(G)$. K_G is finite.*

Let k, l-SGL compute $\hat{G} = \langle \Sigma, V_K, P_K, S \rangle$ from K such that $K_G \subseteq K \subseteq \mathcal{L}(G)$. Then for any $w \in K_A$, $[w] \in V_K$. If $[w_1 \dots w_m] \in V_K$ with $w_1, \dots, w_m \in \Sigma^+$, then $[w_1 \dots w_m] \Rightarrow_{\hat{G}}^* [w_1] \dots [w_m] \stackrel{*}{\Rightarrow} w_1 \dots w_m$.

Lemma 4. *Suppose that the algorithm outputs $\hat{G}$ for the input K including K_G. Then $\mathcal{L}(G) \subseteq \mathcal{L}(\hat{G})$.*

Proof. We first show that if $A \to vB_1 \dots B_n u \in P$ with $v \in \Sigma^k$, $u \in \Sigma^l$ and $w_i \in K_{B_i}$, then there is $w \in K_A$ such that $[w] \Rightarrow_{\hat{G}}^* v[w_1] \dots [w_n]u$. Let β_i be such that $B_i \Rightarrow_G \beta_i \stackrel{*}{\Rightarrow} w_i$ and

$$I = \{\, i \mid w_i \neq \omega(\beta_i),\ 1 \leq i \leq n \,\}.$$

For each $i \in I$, we have $\beta_i \in \Sigma^k V^+ \Sigma^l$. Thus there are $v_i \in \Sigma^k$, $u_i \in \Sigma^l$ and $y_i, y_i' \in \Sigma^*$ such that $w_i = v_i y_i u_i$ and $\omega(\beta_i) = v_i y_i' u_i$. The fact $\omega(\beta_i), w_i \in K_{B_i}$ entails that $x_i v_i y_i' u_i z_i, x_i v_i y_i u_i z_i \in K_G$ where $\langle x_i, z_i \rangle = \chi(B_i)$. By definition, $\hat{G}$ has rule $[\omega(\beta_i)] \to [w_i] \in P_K$. We have $v\omega(\beta_1) \dots \omega(\beta_n)u \in K_A$ and

$$[v\omega(\beta_1) \dots \omega(\beta_n)u] \stackrel{*}{\underset{\hat{G}}{\Rightarrow}} v[\omega(\beta_1)] \dots [\omega(\beta_n)]u \stackrel{*}{\Rightarrow} v[w_1] \dots [w_n]u.$$

By using this claim inductively, we see that for any $A \Rightarrow_G^* w \in \Sigma^*$, there is $w' \in K_A$ such that $[w'] \Rightarrow_{\hat{G}}^* w$. Since $\hat{G}$ has rule $S \to [w'] \in P_K$ for any $w' \in K_S$, we obtain the lemma. □

Clark and Eyraud [7] define a characteristic set of $0, 0$-SCFLs $\mathcal{L}(G)$ by

$$CS(G) = \{\, xyz \mid A \to \beta \in P,\ \langle x, z \rangle \in \chi(A),\ y = \omega(\beta) \,\},$$

where G is not assumed to be in any special form. This set $CS(G)$ is more compact than K_G. However, $CS(G)$ can be too small as a characteristic set of a k, l-SCFL in general. Let G be a CFG in $1, 0$-GNF consisting of production rules $S \to aSC$, $S \to b$ and $C \to c$. Then $\mathcal{L}(G) = \{\, a^n bc^n \mid n \geq 0 \,\}$ is $1, 0$-substitutable. On the other hand, $CS(G) = \{\, b, abc \,\}$ is also $1, 0$-substitutable, and thus $CS(G)$ cannot be a characteristic set of $\mathcal{L}(G)$ for any algorithm learning $1, 0$-SCFLs.

4.3 Polynomial Time and Data

Now we discuss the efficiency of our learning algorithm k, l-SGL. Though the class of k, l-SCFLs is not identifiable in the limit from positive data with polynomial time and data in de la Higuera's sense (Definition 1), k, l-SGL satisfies

de la Higuera's definition if we accept the thickness τ_G of the target grammar as a fundamental parameter (Lemma 7). Besides, k,l-SGL identifies k,l-SCFLs in the limit from positive data with polynomial time and data in Carme et al.'s sense [4], i.e., k,l-SGL admits a characteristic set of polynomial cardinality (Lemma 6). Although we have no grammatical characterization of k,l-SCFLs, Lemma 2 justifies evaluating the characteristic set K_G where G is in k,l-GNF such that $P \subseteq V \times (\Sigma^{\leq k+l} \cup \Sigma^k V^{\leq 7(k+l)} \Sigma^l)$.

Lemma 5. *Computation of $\hat{G}$ from a finite language K is done in polynomial time in the description size of K.*

Proof. Let $\ell_K = \max\{\, |w| \mid w \in K \,\}$. For fixed $w, w' \in K$, the cost for enumerating all pairs vyu and $vy'u$ such that $w = xvyuz$, $w' = xvy'uz$, $|v| = k$, $|u| = l$, $vyu, vy'u \neq \lambda$ for some $x, z \in \Sigma^*$ is bounded by $O(\ell_K^2)$. Thus computing all the rules of the form $[vyu] \to [vy'u]$ takes $O(|K|^2 \ell_K^2)$ time. Computing all the rules of the form $S \to [w]$ for $w \in K$ takes $O(\|K\|)$ time. For each $[w] \in V_K$, there are $(|w|-1)$ pairs $\langle x, y\rangle$ such that $w = xy$ and $x, y \neq \lambda$. Thus computing all the rules of the form $[xy] \to [x][y]$ takes $O(|V_K|\ell_K)$ time. Together with $\ell_K, |K| \leq \|K\|$ and $|V_K| \leq \|K\|^2$, totally the algorithm updates its hypothesis in $O(\|K\|^4)$ time. □

Therefore, k,l-SGL updates its hypothesis quickly even for large k and l. However, the amount of data for letting k,l-SGL converge increases depending on k and l. For instance, to learn the k,l-SCFL $\Sigma^{\leq k+l} - \{\lambda\}$, the learner requires all elements of $\Sigma^{\leq k+l} - \{\lambda\}$ to be given as positive examples, because any subset of $\Sigma^{\leq k+l} - \{\lambda\}$ is also a k,l-SCFL.

Lemma 6. *$|K_G|$ is bounded by a polynomial in $\|G\|$.*

Proof. Let $n = \max\{\, |\beta| \mid A \to x\beta z \in P \text{ with } |x| = k,\ |z| = l \,\}$. Then we have $|K_G| \leq |P|^{n+1}$. By Lemma 2, we have $n \leq 7(k+l)$ (constant). $|K_G|$ is bounded by a polynomial. □

Lemma 7. *The description size of $\|K_G\|$ is bounded by a polynomial in $\|G\|$ and τ_G.*

Proof. By Lemma 6, it is enough to prove that the length of each element in K_G is bounded by a polynomial in $\|G\|$ and τ_G. Suppose that $xyz \in K_G$ where $\chi(A) = \langle x, z\rangle$ and $y \in K_A$ for $A \in V$. We have a derivation

$$A_0 \underset{G}{\Rightarrow} \alpha_1 A_1 \gamma_1 \Rightarrow \cdots \Rightarrow \alpha_1 \dots \alpha_m A_m \gamma_m \dots \gamma_1 \overset{*}{\Rightarrow} x A_m z$$

where $A_0 = S$, $A_{i-1} \to \alpha_i A_i \gamma_i$ for $i = 1, \dots, m$, $A_m = A$, $x = \omega(\alpha_1 \dots \alpha_m)$ and $z = \omega(\gamma_m \dots \gamma_1)$. We see $A_i \neq A_j$ if $i \neq j$ by the definition of $\chi(A)$. Thus $|\alpha_1 \dots \alpha_m \gamma_m \dots \gamma_1| \leq \|G\|$ and $|xz| \leq \|G\|\tau_G$. If $y \in K_A$, then either $A \to y \in P$, or there are productions $A \to vB_1 \dots B_n u \in P$, $B_i \to \beta_i \in P$ for $i = 1, \dots, n$ and $y = v\omega(\beta_1 \dots \beta_n)u$. Let $p = k + l$ (constant). By Lemma 2, we have $n \leq 7p$ and $\beta_i \in \Sigma^{\leq p} \cup \Sigma^k V^{\leq 7p} \Sigma^l$. Therefore $|\omega(\beta_i)| \leq p + 7p\tau_G$ and $|y| \leq p + 7p(p + 7p\tau_G) \in O(\tau_G)$. All in all we have $|xyz| \in O(\|G\|\tau_G)$. □

5 Discussion

Following the proposal given by Clark and Eyraud [7], this paper gave a formal definition of a hierarchy of substitutable languages by generalizing the original notion of substitutability and showed that each class of context-free languages in the hierarchy is polynomial-time identifiable in the limit from positive data. While this generalization can be thought of as the exact analogue of k-reversibility introduced by Angluin [1], some properties that hold of k-reversible regular languages do not hold of k, l-SCFLs, or are not known to hold of k, l-SCFLs.

One is a grammatical characterization of k, l-SCFLs, as already pointed out by Clark and Eyraud. The original definition of k-reversible languages is given in terms of finite state automata and the syntactic characterization of them is a theorem [1].

Kobayashi and Yokomori [14] have shown that the least k-reversible language including a finite language is always regular. In fact Angluin's learning algorithm always hypothesizes the least k-reversible regular language including the given data. On the other hand, the least $0, 0$-substitutable language including $\{abc, acb, bac, bca, aabbcc\}$ is MIX $= \{ w \in \{a, b, c\}^+ \mid |w|_a = |w|_b = |w|_c \}$, which is known to be non-context-free.[3] Moreover, MIX does not have a least $0, 0$-SCFL including it. $L_1 = \{ w \in \{a, b, c\}^+ \mid |w|_a = |w|_b \}$ and $L_2 = \{ w \in \{a, b, c\}^+ \mid |w|_a = |w|_c \}$ are $0, 0$-SCFLs and MIX $= L_1 \cap L_2$. This shows that some set of positive examples does not admit a least consistent $0, 0$-SCFL.

The literature has established many results on reversible regular languages and their variants (e.g., [2, 14, 16, 15, 13, 19, 21, 23]). It would be interesting to investigate whether or not analogous results hold of k, l-SCFLs.

Clark and Eyraud's algorithm SGL for $0, 0$-SCFLs [7] bases Clark's PAC learning algorithm for unambiguous NTS languages [5]. Though some unambiguous NTS languages are not $0, 0$-substitutable, taking into account the difference of context distributions of substrings, he successes learning non-$0, 0$-SCFLs using SGL. Our learning algorithm k, l-SGL is more powerful than SGL for $k, l > 0$, but we still conjecture all k, l-SCFLs are NTS. It is doubtful whether an application of Clark's method to k, l-SGL could enable a PAC learning algorithm that is more efficient or more powerful.

Acknowledgement

The author is deeply grateful to Rémi Eyraud, Alexander Clark and Thomas Zeugmann for their valuable comments and advice. He also appreciates the anonymous reviewers for their helpful comments and suggestions.

This work was supported by Grant-in-Aid for Young Scientists (B-20700124) and a grant from the Global COE Program, "Center for Next-Generation Information Technology based on Knowledge Discovery and Knowledge Federation", from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

[3] It is Eyraud and Clark who gave the author a critical clue to find this example in personal communication.

References

1. Angluin, D.: Inference of reversible languages. Journal of the Association for Computing Machinery 29(3), 741–765 (1982)
2. Angluin, D.: Negative results for equivalence queries. Machine Learning 5, 121–150 (1990)
3. Boasson, L., Sénizergues, G.: NTS languages are deterministic and congruential. Journal of Computer and System Sciences 31(3), 332–342 (1985)
4. Carme, J., Gilleron, R., Lemay, A., Niehren, J.: Interactive learning of node selecting tree transducer. Machine Learning 66(1), 33–67 (2007)
5. Clark, A.: PAC-learning unambiguous NTS languages. In: Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T., Tomita, E. (eds.) ICGI 2006. LNCS (LNAI), vol. 4201, pp. 59–71. Springer, Heidelberg (2006)
6. Clark, A., Eyraud, R.: Identification in the limit of substitutable context-free languages. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 283–296. Springer, Heidelberg (2005)
7. Clark, A., Eyraud, R.: Polynomial identification in the limit of context-free substitutable languages. Journal of Machine Learning Research 8, 1725–1745 (2007)
8. Engelfriet, J.: An elementary proof of double Greibach normal form. Information Processing Letters 44(6), 291–293 (1992)
9. Gold, E.M.: Language identification in the limit. Information and Control 10(5), 447–474 (1967)
10. Greibach, S.A.: A new normal-form theorem for context-free phrase structure grammars. Journal of the Association for Computing Machinery 12(1), 42–52 (1965)
11. de la Higuera, C.: Characteristic sets for polynomial grammatical inference. Machine Learning 27, 125–138 (1997)
12. de la Higuera, C.: A bibliographical study of grammatical inference. Pattern Recognition 38(9), 332–1348 (2005)
13. Kobayashi, S.: Iterated transductions and efficient learning from positive data: A unifying view. In: Oliveira, A.L. (ed.) ICGI 2000. LNCS (LNAI), vol. 1891, pp. 157–170. Springer, Heidelberg (2000)
14. Kobayashi, S., Yokomori, T.: On approximately identifying concept classes in the limit. In: Zeugmann, T., Shinohara, T., Jantke, K.P. (eds.) ALT 1995. LNCS, vol. 997, pp. 298–312. Springer, Heidelberg (1995)
15. Kobayashi, S., Yokomori, T.: Identifiability of subspaces and homomorphic images of zero-reversible languages. In: Li, M., Maruoka, A. (eds.) ALT 1997. LNCS, vol. 1316, pp. 48–61. Springer, Heidelberg (1997)
16. Kobayashi, S., Yokomori, T.: Learning approximately regular languages with reversible languages. Theoretical Computer Science 174(1-2), 251–257 (1997)
17. Lange, S., Zeugmann, T., Zilles, S.: Learning indexed families of recursive languages from positive data: A survey. Theoretical Computer Science 397(1-3), 194–232 (2008)
18. Lee, L.: Learning of context-free languages: A survey of the literature. Technical Report TR-12-96, Harvard University (1996), `ftp://deas-ftp.harvard.edu/techreports/tr-12-96.ps.gz`
19. Mäkinen, E.: On inferring zero-reversible languages. Acta Cybernetica 14(3), 479–484 (2000)
20. Rosenkrantz, D.J.: Matrix equations and normal forms for context-free grammars. Journal of ACM 14(3), 501–507 (1967)

21. Sempere, J.M.: Learning reversible languages with terminal distinguishability. In: Sakakibara, Y., Kobayashi, S., Sato, K., Nishino, T., Tomita, E. (eds.) ICGI 2006. LNCS (LNAI), vol. 4201, pp. 354–355. Springer, Heidelberg (2006)
22. Sénizergues, G.: The equivalence and inclusion problems for NTS languages. Journal of Computer and System Sciences 31(3), 303–331 (1985)
23. Tîrnauca, C., Knuutila, T.: Polynomial time algorithms for learning k-reversible languages and pattern languages with correction queries. In: Hutter, M., Servedio, R.A., Takimoto, E. (eds.) ALT 2007. LNCS (LNAI), vol. 4754, pp. 272–284. Springer, Heidelberg (2007)
24. Wakatsuki, M., Tomita, E.: A fast algorithm for checking the inclusion for very simple deterministic pushdown automata. IEICE transactions on information and systems E76-D(10), 1224–1233 (1993)
25. Yokomori, T.: Polynomial-time identification of very simple grammars from positive data. Theoretical Computer Science 298, 179–206 (2003)
26. Yokomori, T.: Erratum to Polynomial-time identification of very simple grammars from positive data. Theoret. Comput. Sci. 298, 179–206 (2003); Theoretical Computer Science 377(1-3), 282–283 (2007)