Identification in the Limit with Polynomial Time and Data of (k,l)-Local-Context Substitutable Grammars

Arnaud Patey ARNAUD.PATEY@ENS-RENNES.FR
François Coste FRANCOIS.COSTE@INRIA.FR
Univ Rennes, CNRS, Inria, IRISA, F-35000 Rennes

Editor: Editor's name

Abstract

Clark and Eyraud (2007) introduced the class of substitutable languages and showed that context-free substitutable languages are identifiable in polynomial time in the limit from positive data. Yoshinaka (2008) generalized this result to (k, l)-substitutable context-free languages, and Coste et al. (2012) extended it to (k, l)-local substitutable context-free languages. However, these classes are only learnable under a polynomial thick-data relaxation of the Identifiable in the Limit with Polynomial Time and Data (IPTD) criterion.

In this paper, we introduce a class of grammars providing a canonical representation of (k, l)-local-context substitutable languages, which combine (k, l)-substitutability and (k, l)-local substitutability, together with a linear-time inference algorithm. This allows us to prove that this class is IPTD, yielding the first IPTD result for a class of substitutable languages without thickness assumptions.

Keywords: Learnability, substitutable languages, context-free grammar, canonical form

1. Introduction

Building on Gold's model for identification from data (Gold, 1978) and Pitt's polynomial-time constraints on inductive inference (Pitt, 1989), de la Higuera (1997) formalized the learnability criterion of Identifiable in the Limit from Polynomial Time and Data (IPTD), requiring polynomial bounds on both time and data with respect to representation size and having since become a standard criterion for identifiability in feasible grammatical inference from data. de la Higuera (1997) showed that deterministic finite automata are IPTD, whereas context-free grammars, linear grammars, simple deterministic grammars, and even nondeterministic finite automata are not. These results highlight both the difficulty of learning expressive language classes and the crucial role played by the choice of representation in this learning framework. In particular, they place the learnability frontier near context-free grammars, motivating subsequent work on identifying grammar classes that retain some of the expressive power of context-free grammars while remaining efficiently learnable. A prominent line of research in this direction is due to Clark and Eyraud (2007), who introduced the class of substitutable languages as a simple formalism for studying learnability under the principle of substitutability and the distributional hypothesis from linguistics.

Substitutable languages are defined by the property that if two strings occur in the same context, they are substitutable in all their other contexts. Clark and Eyraud (2007) showed that the class of context-free substitutable languages is identifiable in polynomial time in the limit from positive data, opening the door to similar results for more expressive formalisms (for instance: Clark and Yoshinaka, 2016; Numaya et al., 2023; Kanazawa and

© 2026 A. Patey & F. Coste.

Yoshinaka, 2023) and to alternative definitions of substitutability for context-free inference (Yoshinaka, 2008; Coste et al., 2012). Yoshinaka (2008) introduced (k, l)-substitutability, hereafter referred to as (k, l)-context substitutability, as a generalisation of k-reversibility from regular languages to the context-free setting, restricting the contexts in which strings are substitutable. Motivated by the inference of context-free grammars for modelling biological sequences, Coste et al. (2012) showed that the criterion for determining whether two strings are substitutable should be relaxed from global to local contexts when considering such sequences. This led them to define the class of (k, l)-local substitutable languages and, in combination with the definition of Yoshinaka (2008), the class of (k, l)-local-context substitutable languages.

The result of Clark and Eyraud (2007) addresses polynomial-time learnability but not the polynomial data requirement of IPTD. Clark and Eyraud (2007) show that the size of the required training set cannot be bounded polynomially in the size of a context-free grammar representing substitutable languages, by considering languages consisting of the single string a^{2^n}, represented by the grammar $G_n = \langle a, \{N_i \colon 1 \leq i \leq n\}, \{N_i \to N_{i-1}N_{i-1} \colon 1 < i \leq n\} \cup \{N_1 \to aa\}, \{N_n\}\rangle$. For these languages, any training set has size exponential in the size of the grammar. Since these languages are also (k, l)-context substitutable for all $k, l \geq 0$, this limitation also applies to these classes. To obtain a polynomial data bound, a weaker criterion had to be used: the Identification in the Limit from Polynomial Time and Thick Data (IPTtD) (Yoshinaka, 2009; Eyraud et al., 2016), which incorporates the thickness of the grammar into the bound on the size of the required training set. Yoshinaka (2008) showed that (k, l)-context substitutable languages are IPTtD, and the same property was later obtained for (k, l)-local substitutable languages in reduced normal form by Coste and Nicolas (2018).

In this paper, we study the learnability of (k, l)-local-context substitutable languages, the more expressive class introduced by Coste et al. (2012), whose learnability has not yet been characterized. This enables us to provide the first IPTD result without thickness for a class of substitutable languages. Our contributions are threefold: (i) the introduction of (k, l)-step local-context grammars, (ii) the proof that they form a canonical representation of (k, l)-local-context substitutable languages, and (iii) a linear-time learning algorithm from finite data. As a consequence, we show that the (k, l)-step local-context grammar representation class of (k, l)-local-context substitutable languages, and thus (k, l)-local substitutable languages, is IPTD.

2. Definitions and notations

We remind first the definition of IPTD by de la Higuera (1997):

Definition 1 (IPTD) *A representation class $\mathcal{R}$ is identifiable in the limit with polynomial time and data (IPTD) iff there exists two polynomials p, q and an algorithm $\mathcal{A}$ such that:*

- *Given a training sample S of size $||S||$, $\mathcal{A}$ returns a representation R in $\mathcal{R}$ consistent with S in $O(p(||S||))$.*
- *For each representation R in $\mathcal{R}$ of size $||R||$, there exists a set CS (named characteristic set) of size at most $O(q(||R||))$, such that $\mathcal{A}$ returns a representation R' equivalent with R for any set S containing CS.*

To study the learnability of substitutable languages represented by context-free grammars from positive examples, we introduce in this section the related standard definitions and notations used throughout the paper.

Formal Languages

Let Σ be a non-empty finite set of symbols, called an alphabet. A string over Σ is a finite sequence of symbols from Σ. The length of a string w is denoted by $|w|$, and the empty string, i.e., the unique string of length 0, is denoted by λ. The set of all strings over Σ is denoted by Σ^*. We write $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$. For $k \in \mathbb{N}$, we define $\Sigma^k = \{w \in \Sigma^* \mid |w| = k\}$, $\Sigma^{\leq k} = \{w \in \Sigma^* \mid |w| \leq k\}$, and $\Sigma^{<k} = \Sigma^{\leq k} \setminus \Sigma^k$. We choose an arbitrary order $\prec$ over Σ that we extend canonically to Σ^* by defining that, for $u, v \in \Sigma^*$, $u \prec v$ if $|u| < |v|$, or $|u| = |v|$ and u is lexicographically smaller than v.

A subset of Σ^* is called a language over Σ. For a finite language $L \subseteq \Sigma^*$, $|L|$ denotes its cardinality, and its size is defined as $||L|| = \sum_{w \in L} |w|$. For a string $w \in \Sigma^+$, we define $Sub(w) = \{w' \in \Sigma^+ \colon \exists u, v \in \Sigma^*,\ uw'v = w\}$, and for a language $L \subseteq \Sigma^+$, $Sub(L) = \bigcup_{w \in L} Sub(w)$, the set of all factors of strings in L. For a string $y \in \Sigma^+$, we define its distribution $D_L(y)$ as the set of contexts in which it occurs in L. Formally, $D_L(y) = \{(u, v) \in \Sigma^* \times \Sigma^* \colon uyv \in L\}$. We define the (k,l)-distribution of $y \in \Sigma^+$ for L as $D_L^{k,l}(y) = \{(u, v) \in \Sigma^k \times \Sigma^l \colon uyv \in Sub(L)\}$. For a set of strings $X \subseteq \Sigma^+$, we define $D_L^{k,l}(X) = \bigcup_{y \in X} D_L^{k,l}(y)$. For a language $L \subseteq \Sigma^+$, we define the set of (k,l)-contexts in L by $C^{k,l}(L) = D_L^{k,l}(Sub(L))$. Finally, for $(u, v) \in C^{k,l}(L)$, we define $I_L(u, v) = \{y \in \Sigma^+ \colon uyv \in Sub(L)\}$, the set of strings occurring in context (u, v) in L.

Substitutable Languages

L is a substitutable language (Clark and Eyraud, 2007) if for any $y_1, y_2 \in L$,

$$D_L(y_1) \cap D_L(y_2) \neq \emptyset \implies D_L(y_1) = D_L(y_2).$$

For non negative integers k and l, it is a (k,l)-substitutable language (Yoshinaka, 2008) if for any $y_1, y_2 \in L$, for any $(u, v) \in \Sigma^k \times \Sigma^l$,

$$D_L(uy_1v) \cap D_L(uy_2v) \neq \emptyset \implies D_L(uy_1v) = D_L(uy_2v).$$

Since this substitutability is restricted to (u, v) contexts, we refer to this notion as (k,l)-context substitutability rather than (k,l)-substitutability. This helps to contrast with the definition of L being a (k,l)-local substitutable language (Coste et al., 2012) if for any $x, y \in \Sigma^+$,

$$D_L^{k,l}(x) \cap D_L^{k,l}(y) \neq \emptyset \implies D_L(x) = D_L(y)$$

in which local contexts of size (k,l) rather than global contexts are used to identify substitutable strings.

Combining these two notions balances broader, and sometimes excessive, generalization enabled by locality with the specificity provided by contextual constraints. This leads us to the definition of the class studied in this paper.

Definition 2 ((k,l)-LCS) *Let $k, l \in \mathbb{N}$. A language L is (k,l)-local-context substitutable ((k,l)-LCS) if for any $x, y \in \Sigma^+$ and any $(u,v) \in \Sigma^k \times \Sigma^l$,*

$$(u,v) \in D_L^{k,l}(x) \cap D_L^{k,l}(y) \implies D_L(uxv) = D_L(uyv).$$

Since the set of (k,l)-contexts is finite, there are only finitely many congruence classes for a (k,l)-local or (k,l)-local-context-substitutable language. Hence, these languages are regular.

If a language is (k,l)-LCS, then it is also (m,n)-LCS for any $m \geq k$ and $n \geq l$. The (k,l)-LCS classes form a strictly increasing hierarchy as k and l increase. To see that the hierarchy is strict, consider the language L defined by the regular expression $a^{k+1}b^+a^{l+1}$. This language is $(k+1, l+1)$-LCS since each string x with $D_L^{k+1,l+1}(x) \neq \emptyset$ belongs to b^+ and we have for all $x, y \in b^+$, $D_L(x) = D_L(y)$ implying for all $(u,v) \in D_L^{k,l}(x) \cap D_L^{k,l}(y)$, $D_L(uxv) = D_L(uyv)$. But L is not (k,l)-LCS since $(a^k, b^l) \in D_L^{k,l}(a) \cap D_L^{k,l}(b)$, while $(\lambda, a^{l+1}) \in D_L(a^kab^l)$ and $(\lambda, a^{l+1}) \notin D_L(a^kbb^l)$.

Context free grammars

A context-free grammar (CFG) is a quadruple $G = \langle \Sigma, V, P, S \rangle$, where Σ and V are disjoint sets of terminal and non-terminal symbols, respectively, P is a set of production rules, and $S \in V$ is the start symbol. We restrict ourselves to λ-free grammars to simplify the presentation: a production rule in P has the form $A \to \beta$, where $A \in V$ and $\beta \in (\Sigma \cup V)^+$. The size of G is defined as $||G|| = \sum_{A \to \beta \in P}(|A| + |\beta|) = \sum_{A \to \beta \in P}(1 + |\beta|)$.

For a grammar $G = (\Sigma, V, P, S)$, if $A \to \beta \in P$, we write $\alpha A \gamma \Rightarrow_G \alpha\beta\gamma$ for any $\alpha, \gamma \in (\Sigma \cup V)^+$. We denote by $\Rightarrow_G^+$ the transitive closure of $\Rightarrow_G$, and by $\Rightarrow_G^*$ its reflexive and transitive closure. For $n \in \mathbb{N}^*$, we write $\alpha_0 \Rightarrow_G^n \alpha_n$ if there exist $\alpha_0, \ldots, \alpha_n \in (\Sigma \cup V)^+$ such that $\alpha_i \Rightarrow_G \alpha_{i+1}$ for all $i \in \{0, \ldots, n-1\}$. For $\alpha \in (\Sigma \cup V)^+$, we define $L(G, \alpha) = \{w \in \Sigma^+ \colon \alpha \Rightarrow_G^* w\}$. The language generated by G is $L(G) = L(G, S)$. Two grammars G_1 and G_2 are equivalent if $L(G_1) = L(G_2)$. A language is context-free if it is generated by a context-free grammar.

A symbol $A \in \Sigma \cup V$ is useless in G if there are no $u, v, w \in \Sigma^*$ such that $S \Rightarrow_G^* uAw \Rightarrow_G^* uvw$ (Hopcroft et al., 2003). A CFG G is trim if $\Sigma \cup V$ contains no useless symbols. A trim grammar is entirely determined by its set of production rules and its start symbol. We assume that target grammars are trim. In the following, terminal symbols are denoted by $a, b, c, \ldots$, non-terminal symbols by $A, B, \ldots$, strings over Σ by $u, v, \ldots, z$, and strings over $\Sigma \cup V$ by α, β, γ.

3. (k,l)-step local-context grammar

3.1. Definition

To define the subclass of context-free grammars representing (k,l)-LCS languages, we focus on (k,l)-contexts and introduce a hierarchical decomposition of a string into nested (k,l)-contexts obtained by recursively removing its prefix of length k and suffix of length l.

Definition 3 ((k,l)-context step decomposition) *Let $k, l \geq 0$ with $k + l > 0$. The (k,l)-context step decomposition of a string $w \in \Sigma^+$, denoted by $\mathcal{D}^{k,l}(w)$, is defined recur-*

sively as follows:

$$\mathcal{D}^{k,l}(w) = \begin{cases} (w), & \text{if } |w| \leq k+l, \\ (u, \mathcal{D}^{k,l}(x), v), & \text{if } |w| > k+l, \text{ where } w = uxv,\ |u| = k,\ |v| = l. \end{cases}$$

For example, for $k = 1$ and $l = 2$, $\mathcal{D}^{1,2}(\texttt{aabacbaca}) = (\texttt{a}, (\texttt{a}, (\texttt{bac}), \texttt{ba}), \texttt{ca})$.

We represent the (k,l)-context-step decomposition of w as a recursively nested tuple $\mathcal{D}^{k,l}(w) = (u_1, (\ldots (u_n, (x), v_n) \ldots), v_1)$. We call x the *core*, denoted $\mathcal{I}^{k,l}(w)$; each (u_i, v_i) the *i-th context*, denoted $\mathcal{D}^{k,l}_i(w)$; and n the *depth*, denoted $d^{k,l}(w)$, given by $d^{k,l}(w) = \left\lfloor \frac{|w|-1}{k+l} \right\rfloor$. We define the set of all context-step pairs of w: $\mathcal{D}^{k,l}_*(w) = \{(u_i, v_i) \mid 1 \leq i \leq d^{k,l}(w)\}$. We extend these notions to a language $L \subseteq \Sigma^+$ by $\mathcal{I}^{k,l}(L) = \bigcup_{w \in X} \mathcal{I}^{k,l}(w)$, $\mathcal{D}^{k,l}_i(L) = \bigcup_{w \in X} \mathcal{D}^{k,l}_i(w)$ and $\mathcal{D}^{k,l}_*(L) = \bigcup_{w \in X} \mathcal{D}^{k,l}_*(w)$, where we define $\mathcal{D}^{k,l}_i(w) = \emptyset$ if i exceeds the depth of $\mathcal{D}^{k,l}(w)$.

We can now define the (k,l)-step local-context grammar for a language L as follows.

Definition 4 ((k,l)-step local-context grammar) *Let $k, l \geq 0$ with $k + l > 0$. The (k,l)-step local-context grammar for a language $L \subseteq \Sigma^+$ is $\mathcal{G}^{k,l}(L) = (\Sigma, V, P, N_S)$, where:*

$$\begin{aligned} V &= \{N_{u,v} \colon (u,v) \in \mathcal{D}^{k,l}_*(L)\} \cup \{N_S\}, \\ P &= \{N_S \to y \colon y \in L \cap \mathcal{I}^{k,l}(L)\} \\ &\quad \cup \{N_S \to u N_{u,v} v \colon (u,v) \in \mathcal{D}^{k,l}_1(L)\} \\ &\quad \cup \{N_{u,v} \to y \colon y \in \mathcal{I}^{k,l}(L), (u,v) \in \mathcal{D}^{k,l}_*(L), uyv \in Sub(L)\} \\ &\quad \cup \{N_{u,v} \to u' N_{u',v'} v' \colon (u,v), (u',v') \in \mathcal{D}^{k,l}_*(L), \exists y \in \Sigma^+, uu'yv'v \in Sub(L)\}. \end{aligned}$$

This grammar is well defined since the set of strings $\mathcal{I}^{k,l}(L)$ of length at most $k+l$, as well as the set of possible contexts (u,v) with $|u| = k$ and $|v| = l$, are finite, and therefore V and P are finite.

This grammar is in (k,l)-Greibach normal form ((k,l)-GNF) (Yoshinaka, 2008), which generalizes Greibach normal form (Greibach, 1965) and double Greibach normal form (Rosenkrantz, 1967; Engelfriet, 1992). It uses only (k,l)-contexts from $\mathcal{D}^{k,l}_*(L)$, i.e., those required by the (k,l)-context-step decompositions of strings in L, to define non-terminals generating each the language inside their k,l-context, thereby inducing a closure under local substitutability on these contexts.

3.2. Canonicity

Let k, l be non-negative integers with $k + l > 0$, let L be a (k,l)-LCS language, and let $\mathcal{G}^{k,l}(L) = (\Sigma, V, P, N_S)$ denote its associated (k,l)-step local-context grammar. To show that the (k,l)-step local-context grammar is a canonical form for (k,l)-LCS languages, we begin by establishing some properties of the production rules of $\mathcal{G}^{k,l}(L)$ before characterizing the languages generated by the non-terminals.

Proposition 5 *For $(u,v) \in \mathcal{D}^{k,l}_*(L), y \in \Sigma^{\leq k+l}$, $N_{u,v} \to y \in P$ if and only if there exists $w \in L$ such that $y = \mathcal{I}^{k,l}(w)$ and $(u,v) = \mathcal{D}^{k,l}_{d^{k,l}(w)}(w)$.*

Proof Let $(u,v) \in \mathcal{D}_*^{k,l}(L)$ and $y \in \Sigma^{\leq k+l}$.

$\implies$: Suppose that $N_{u,v} \to y \in P$. Then by definition of P, $uyv \in Sub(L)$. This means that there exists $w_1 \in L$ such that $w_1 = x_1uyvz_1$ for some $x_1, z_1 \in \Sigma^*$. Moreover, $(u,v) \in \mathcal{D}_*^{k,l}(L)$ so by definition, there exists $w_2 \in L$ and $i \in \{1, ..., d^{k,l}(w_2)\}$ such that $(u,v) = \mathcal{D}_i^{k,l}(w_2)$. Let $x_2, z_2 \in \Sigma^*$ and $y_2 \in \Sigma^+$ such that $w_2 = x_2uy_2vz_2$. We have $(u,v) \in D_L^{k,l}(y) \cap D_L^{k,l}(y_2)$, so as $(x_2, z_2) \in D_L(uy_2v) = D_L(uyv)$ (because L is (k,l)-LCS), $w = x_2uyvz_2 \in L$, and as $|y| \leq k+l$ and $(u,v) = \mathcal{D}_i^{k,l}(w_2)$, then $d^{k,l}(w) = i, (u,v) = \mathcal{D}_{d^{k,l}(w)}^{k,l}(w)$ and $y = \mathcal{I}^{k,l}(w)$.

$\impliedby$: Suppose that there exists $w \in L$ such that $y = \mathcal{I}^{k,l}(w)$ and $(u,v) = \mathcal{D}_{d^{k,l}(w)}^{k,l}(w)$. By definition, $y \in \mathcal{I}^{k,l}(L), (u,v) \in \mathcal{D}_*^{k,l}(L)$ and there exists $x, z \in \Sigma^*$ such that $w = xuyvz$ so $uyv \in Sub(L)$ so by definition of P, $N_{u,v} \to y \in P$. ■

Proposition 6 *For $(u,v) \in \mathcal{D}_*^{k,l}(L), (u',v') \in \mathcal{D}_*^{k,l}(L)$, $N_{u,v} \to u'N_{u',v'}v' \in P$ if and only if there exists $w \in L$ and $i \in \{1, ..., d^{k,l}(w) - 1\}$ such that $(u,v) = \mathcal{D}_i^{k,l}(w)$ and $(u',v') = \mathcal{D}_{i+1}^{k,l}(w)$.*

Proof $\implies$: Suppose that $N_{u,v} \to u'N_{u',v'}v' \in P$. By definition of P, there exists $y \in \Sigma^+$ such that $uu'yv'v \in Sub(L)$. Moreover, $(u,v) \in \mathcal{D}_*^{k,l}(L)$ so there exist $w_1 \in L$ and $i_1 \in \{1, ..., d^{k,l}(w_1)\}$ such that $(u,v) \in \mathcal{D}_{i_1}^{k,l}(w_1)$. We then have $w_1 = x_1uy_1vz_1$ for some $x_1, z_1 \in \Sigma^*$ and $y_1 \in \Sigma^+$. Hence $uy_1v \in Sub(L)$, $uu'yv'v \in Sub(L)$, which implies that $(u,v) \in D_L^{k,l}(y_1) \cap D_L^{k,l}(u'yv')$. And as L is (k,l)-LCS, then by definition, as $(x_1, z_1) \in D_L(uy_1v) = D_L(uu'yv'v)$, we have $x_1uu'yv'vz_1 \in L$. Let $w = x_1uu'yv'vz_1$. Then $w \in L$, $(u,v) = \mathcal{D}_{i_1}^{k,l}(w)$ because $(u,v) = \mathcal{D}_{i_1}^{k,l}(w_1)$, and as $(u',v') \in \mathcal{D}_*^{k,l}(L)$, then $(u',v') \in \Sigma^k \times \Sigma^l$ so $(u',v') = \mathcal{D}_{i_1+1}^{k,l}(w)$.

$\impliedby$: Suppose that there exists $w \in L$ and $i \in \{1, ..., d^{k,l}(w) - 1\}$ such that $(u,v) = \mathcal{D}_i^{k,l}(w)$ and $(u',v') = \mathcal{D}_{i+1}^{k,l}(w)$. Then by definition, $(u,v) \in \mathcal{D}_*^{k,l}(L), (u',v') \in \mathcal{D}_*^{k,l}(L)$ and there exists $x, z \in \Sigma^*$ and $y \in \Sigma^+$ such that $w = xuu'yv'vz$, so $uu'yv'v \in Sub(L)$ so by definition of P, $N_{u,v} \to u'N_{u',v'}v' \in P$. ■

Lemma 7 *For all $(u,v) \in \mathcal{D}_*^{k,l}(L)$, $L(N_{u,v}) \subseteq \{x \in Sub(L) \colon uxv \in Sub(L)\}$.*

Proof We show by induction that for all $p \geq 1$, H_p: "$\forall (u,v) \in \mathcal{D}_*^{k,l}(L), \forall y \in L(N_{u,v}), (N_{u,v} \Rightarrow_G^p y) \implies uyv \in Sub(L)$".

$p = 1$: Let $(u,v) \in \mathcal{D}_*^{k,l}(L)$, and $y \in L(N_{u,v})$ such that $N_{u,v} \Rightarrow_G^1 y$. Then y was produced by the rule $N_{u,v} \to y$ in $\mathcal{G}^{k,l}(L)$. As this rule is in P, by definition, $uyv \in Sub(L)$.

$p - 1 \to p$: Let $p \geq 2$ such that H_{p-1} holds. Let $(u,v) \in \mathcal{D}_*^{k,l}(L)$ and $y \in L(N_{u,v})$ such that $N_{u,v} \Rightarrow_G^p y$. As $p \geq 2 > 1$, y could not be produced by the rule $N_{u,v} \to y$, so the first production rule that produced it is one of the form $N_{u,v} \to u'N_{u',v'}v'$ for some $(u',v') \in \mathcal{D}_*^{k,l}(L)$. So we have $y = u'y'v'$ for some $y' \in L(N_{u',v'})$, and we have $N_{u',v'} \Rightarrow_G^{p-1} y'$. By the induction hypothesis H_{p-1}, $y = u'y'v' \in Sub(L)$. Moreover, as

the rule $N_{u,v} \to u'N_{u',v'}v'$ is in P, this means by Proposition 6 that there exists a string $w \in L$ and some $j \in \mathbb{N}$ such that $(u,v) = \mathcal{D}_j^{k,l}(w)$ and $(u',v') = \mathcal{D}_{j+1}^{k,l}(w)$. So we can write $w = xuu'y''v'vz$ for some $x,z \in \Sigma^*, y'' \in \Sigma^+$.
As $u'y'v' \in Sub(L)$, $(u',v') \in D_L^{k,l}(y'') \cap D_L^{k,l}(y')$. Since L is assumed to be k,l-local context substitutable, by definition, $D_L(u'y'v') = D_L(u'y''v')$. As $(xu,vz) \in D_L(u'y''v')$ (because $w \in L$), $(xu,vz) \in D_L(u'y'v')$ so $xuu'y'v'uz \in L$. By this fact, $uyv = uu'y'v'v \in Sub(L)$.
As this works for every $(u,v) \in \mathcal{D}_*^{k,l}(L)$, we get H_p.

By the induction principle, H_p holds for all $p \geq 1$. Moreover, by definition, for every $(u,v) \in \mathcal{D}_*^{k,l}(L)$, for every $y \in L(N_{u,v})$, there exists $p \in \mathbb{N}$ such that $N_{u,v} \Rightarrow_G^p y$ so the lemma is proved. ■

This implies that if L is (k,l)-LCS, the language recognized by $L(\mathcal{G}^{k,l}(L))$ is at most L.

Corollary 8 $L(\mathcal{G}^{k,l}(L)) \subseteq L$

Proof Let $w \in L(\mathcal{G}^{k,l}(L))$. If the first production rule used to derive w from N_S is $N_S \to y$ for some $y \in L \cap \mathcal{I}^{k,l}(L)$, then $w = y \in L \cap \mathcal{I}^{k,l}(L) \subseteq L$ so $w \in L$.

Otherwise, the first derivation is $N_S \to uN_{u,v}v$ where $(u,v) = \mathcal{D}_1^{k,l}(w')$ for some $w' \in L$ (by definition of P). So there exists $y \in L(N_{u,v})$ such that $w = uyv$. By Lemma 7, $uyv \in Sub(L)$. Moreover, we have $w' = uy'v \in L$ for some $y' \in \Sigma^+$. As $(u,v) \in D_L^{k,l}(y) \cap D_L^{k,l}(y')$ and as L is (k,l)-LCS, $(\lambda,\lambda) \in D_L(uy'v) = D_L(uyv)$, so $w = uyv = \lambda uyv\lambda \in L$. So $L(\mathcal{G}^{k,l}(L)) \subseteq L$. ■

Let us prove now that the generated language is large enough.

Lemma 9 $L \subseteq L(\mathcal{G}^{k,l}(L))$.

Proof Let $w \in L$.

If $|w| \leq k+l$, the rule $N_S \to w$ is in P by definition, so $w \in L(\mathcal{G}^{k,l}(L))$.

Otherwise, let $(u_1,(...,(u_n,(x),v_n),...),v_1)$ be the (k,l)-context-step decomposition of w. We show by induction that for all $i \in \{1,...,n\}$, H_i : "$N_S \Rightarrow_G^i u_1...u_iN_{u_i,v_i}v_i...v_1$".

$\boldsymbol{i=1}$: $w \in L$ and $(u_1,v_1) = \mathcal{D}_1^{k,l}(w)$, so $N_S \to u_1N_{u_1,v_1}v_1$ in P by definition, and thus H_1 holds.

$\boldsymbol{i \to i+1}$: Let $i < n$ such that H_i holds. We have $w \in L$, $(u_i,v_i) = \mathcal{D}_i^{k,l}(w)$ and $(u_{i+1},v_{i+1}) = \mathcal{D}_{i+1}^{k,l}(w)$, so by Proposition 6, P contains the rule $N_{u_i,v_i} \to u_{i+1}N_{u_{i+1},v_{i+1}}v_{i+1}$ and by the induction hypothesis H_i, $N_S \Rightarrow_G^i u_1...u_iN_{u_i,v_i}v_i...v_1$. So by definition, $N_S \Rightarrow_G^{i+1} u_1...u_{i+1}N_{u_{i+1},v_{i+1}}v_{i+1}...v_1$. So H_{i+1} holds.

By the induction principle, we can thus derive $u_1...u_nN_{u_n,v_n}v_n...v_1$ from N_S in P.

Then, $w \in L$, $(u_n,v_n) = \mathcal{D}_{d^{k,l}(w)}^{k,l}(w)$ and $x = \mathcal{I}^{k,l}(w)$, so $N_{u_n,v_n} \to x \in P$, so we can derive $u_1...u_nxv_n...v_1 = w$.

Finally, $w \in L(\mathcal{G}^{k,l}(L))$. This works for every $w \in L$, so we have $L \subseteq L(\mathcal{G}^{k,l}(L))$. ■

In the proof of this second lemma, we did not use the fact that L is (k,l)-LCS, so the result applies to any language. For a language L, $L(\mathcal{G}^{k,l}(L))$ generalizes L, but it does not guarantee that it is (k,l)-LCS when L is not itself (k,l)-LCS. For example, take $k=l=1$ and $L = \{aba, aaaba\}$, then $L(\mathcal{G}^{1,1}(L)) = \{aba, aaa, aaba, aaaba, aaaaba\}$. Here

b and ab share the same $(1,1)$-context (a,a), but substituting b by ab in $aaaaba$ yields $aaaaaba \notin L(\mathcal{G}^{1,1}(L))$, so $L(\mathcal{G}^{1,1}(L))$ is not $(1,1)$-LCS.

If L is (k,l)-LCS, we get by Corollary 8 and Lemma 9 that $L(\mathcal{G}^{k,l}(L)) = L$, so the class of grammars of the form $\mathcal{G}^{k,l}(L)$ is a normal representation for the class of (k,l)-LCS languages. We now show that it is a canonical form.

Theorem 10 *The (k,l)-step local-context grammars form a canonical representation for the class of (k,l)-LCS languages.*

Proof We already proved that it is a normal representation, so to prove the theorem, it suffices to show that there is a unique (k,l)-step local-context grammar that recognizes a fixed (k,l)-LCS language.

Let L be a (k,l)-LCS language, and let $\mathcal{G}^{k,l}(L') = \langle \Sigma, V', P', N'_S \rangle$ be a (k,l)-step local-context grammar such that $L(\mathcal{G}^{k,l}(L')) = L$. Let also $\langle \Sigma, V, P, N_S \rangle = \mathcal{G}^{k,l}(L)$. By Lemma 9, $L' \subseteq L(\mathcal{G}^{k,l}(L')) = L$ so by definition, $P' \subseteq P$. We show that $P \subseteq P'$

Let $y \in \Sigma^*$ such that $N_S \to y \in P$. Then $y \in \Sigma^{\leq k+l}$, $y \in L = L(\mathcal{G}^{k,l}(L'))$ and $\mathcal{G}^{k,l}(L')$ is a (k,l)-step local-context grammar, so $N_S \to y \in P'$.

Let $(u,v) \in \mathcal{D}_1^{k,l}(L)$ such that $N_S \to uN_{u,v}v \in P$. Then there is a word $w = uyv \in L = L(\mathcal{G}^{k,l}(L'))$ for some $y \in \Sigma^+$, and as $|w| > k+l$ and $\mathcal{G}^{k,l}(L')$ being a (k,l)-step local-context grammar, $N_S \to uN_{u,v}v \in P'$.

Let $(u,v) \in \mathcal{D}_*^{k,l}(L), y \in \mathcal{I}^{k,l}(L)$ be such that $N_{u,v} \to y \in P$. By Proposition 5, there exists a word $w \in L$ such that $y = \mathcal{I}^{k,l}(w)$ and $(u,v) = \mathcal{D}^{k,l}_{d^{k,l}(w)}$. $w \in L = L(\mathcal{G}^{k,l}(L'))$, and $\mathcal{G}^{k,l}(L')$ is a (k,l)-step local-context grammar, so $N_{u,v} \to y \in P'$.

Let $(u,v),(u',v') \in \mathcal{D}_*^{k,l}(L)$ be such that $N_{u,v} \to u'N_{u',v'}v' \in P$. By Proposition 6, there exists a word $w \in L$ and an integer $i \in \{0,...,d^{k,l}(w)-1\}$ such that $(u,v) = \mathcal{D}_i^{k,l}(w)$ and $(u',v') = \mathcal{D}_{i+1}^{k,l}(w)$. We have $w \in L = L(\mathcal{G}^{k,l}(L'))$, and $\mathcal{G}^{k,l}(L')$ is a (k,l)-step local-context grammar, so $N_{u,v} \to u'N_{u',v'}v' \in P'$.

So $P = P'$, and the unique (k,l)-step local-context grammar that recognizes L is $\mathcal{G}^{k,l}(L)$. ■

4. Identification in the limit with polynomial time and data

4.1. Algorithm

Let $S = \{s_1, \ldots, s_n\}$ be a finite training sample consisting of strings from a language L over an alphabet Σ. Algorithm 1 presents the pseudocode of the (k,l)-step learner algorithm, which constructs a grammar $G^{k,l}(S)$ directly from the (k,l)-context-step decompositions of the strings in S. We refer to $G^{k,l}(S)$ as the *(k,l)-step grammar* of S.

Lemma 11 *Algorithm 1 terminates.*

Proof There is only one while loop in the algorithm, and $|y|$ is strictly decreasing in every iteration, so the algorithm terminates. ■

Lemma 12 *Algorithm 1 computes $G^{k,l}(S)$ in linear time $O(\|S\|)$.*

Algorithm 1: (k,l)-step learner

Input: Set of sequence S on Σ; $k, l \in \mathbb{N}$ such that $k + l > 0$
Output: Grammar $G^{k,l}(S) = \langle \Sigma, V_S, P_S, N_{\lambda,\lambda} \rangle$
$V_S \leftarrow \{N_{\lambda,\lambda}\}$
$P_S \leftarrow \emptyset$
for $y \in S$ **do**
  $u, v \leftarrow \lambda, \lambda$
  while $\exists (u', y', v') \in \Sigma^k \times \Sigma^+ \times \Sigma^l, y = u'y'v'$ **do**
    $V_S \leftarrow V_S \cup \{N_{u',v'}\}$
    $P_S \leftarrow P_S \cup \{N_{u,v} \to u' N_{u',v'} v'\}$
    $u, y, v \leftarrow u', y', v'$
  end
  $P_S \leftarrow P_S \cup \{N_{u,v} \to y\}$
end
return $\langle \Sigma, V_S, P_S, N_{\lambda,\lambda} \rangle$

Proof The algorithm processes each string $s \in S$ exactly once. For each s, either $|s| \leq k+l$, in which case a single production is added in $O(|s|)$ time, or it computes the (k,l)-context-step decomposition of s, which takes $O(|s|)$ time. In the latter case, one production of size $O(k+l)$ is created per step of the decomposition. The number of steps is at most $\lfloor |s|/(k+l) \rfloor \leq |s|$, hence the total cost per string remains $O(|s|)$. Summing over all $s \in S$ yields $O\big(\sum_{s \in S} |s|\big) = O(\|S\|)$, so the algorithm runs in linear time. ■

4.2. Characteristic set

For a target (k,l)-step local-context grammar G_* recognizing a (k,l)-LCS language, we define, as in Clark and Eyraud (2007), its characteristic set by

$$CS(G_*) = \{lwr \colon (N \to \alpha) \in P, (l,r) = c(N), w = w(\alpha)\},$$

where for a nonterminal $N \in V$, $c(N) = \min\{(l', r') \colon S \Rightarrow_G^* l'Nr'\}$ and for a word $\alpha \in (\Sigma \cup N)^*$, $w(\alpha) = \min\{w' \colon \alpha \Rightarrow_G^* w'\}$.

Definition 13 *Let k, l be nonnegative integers such that $k + l > 0$. We say that a CFG $G = \langle \Sigma, V, P, S \rangle$ is in (k,l)-RGNF (Restrictive Greibach Normal Form) if every production is of the form $A \to y$ with $y \in \Sigma^{\leq k+l} \setminus \{\lambda\}$ or $A \to uNv$ with $u \in \Sigma^k, v \in \Sigma^l$ and $N \in V$.*

(k,l)-RGNF grammars form the subclass of (k,l)-GNF grammars (Yoshinaka, 2008) that are linear, i.e., whose right-hand sides contain at most one nonterminal. Since (k,l)-step local-context grammars form a subclass of (k,l)-RGNF grammars, the following lemmas will allow us to derive polynomial bounds on the size of $CS(G_*)$.

Lemma 14 *Let $G = \langle \Sigma, V, P, S \rangle$ be a (k,l)-RGNF grammar and let $w \in L(G)$. The only possible derivations of w from S have the form $S \Rightarrow_G u_1 N_1 v_1 \Rightarrow_G u_1 u_2 N_2 v_2 v_1 \Rightarrow_G \ldots \Rightarrow_G u_1 u_2 \ldots u_n N_n v_n \ldots v_2 v_1 \Rightarrow_G w = u_1 u_2 \ldots u_n x v_n \ldots v_2 v_1$ for some $N_i \in V$, $i \in \{1, 2, \ldots, n\}$, where $(u_1, (u_2, (\ldots, (u_n, (x), v_n), \ldots), v_2), v_1)$ is the (k,l)-context-step decomposition of w.*

Proof Let $G = \langle \Sigma, V, P, S\rangle$ be a (k,l)-RGNF grammar, let $w \in L(G)$ and let the (k,l)-context-step decomposition of w be $(u_1, (u_2, (\dots, (u_n, (x), v_n), \dots), v_2), v_1)$. If $n = 0$, then $|w| \leq k + l$ so the only possible derivation of w from S is $S \Rightarrow_G w$, by definition of (k,l)-RGNF. Otherwise, we proceed by induction on $i \in \{1, \dots, n\}$ and prove H_i: "the first i derivation steps have the form $S \Rightarrow_G u_1 N_1 v_1 \Rightarrow_G u_1 u_2 N_2 v_2 v_1 \Rightarrow_G \dots \Rightarrow_G u_1 u_2 \dots u_i N_i v_i \dots v_2 v_1$ where $N_j \in V$ for $j \in \{1, \dots, i\}$".

$i = 1$: Since $|w| > k + l$, by definition of (k,l)-RGNF, the first derivation step is of the form $S \to u N_1 v$ with $|u| = k, |v| = l$, so H_1 holds with $u_1 = u$ and $v_1 = v$.

$i \to i + 1$: Let $i \in \{1, \dots, n-1\}$ such that H_i holds. By the induction hypothesis, the first i derivation steps can only be of the form $S \Rightarrow_G u_1 N_1 v_1 \Rightarrow_G u_1 u_2 N_2 v_2 v_1 \Rightarrow_G \dots \Rightarrow_G u_1 u_2 \dots u_i N_i v_i \dots v_2 v_1$ where $N_j \in V$ for $j \in \{1, \dots, i\}$. Moreover, since $i < n$ and using the fact that $|w| = n \cdot (k + l) + |c|$ with $|c| < k + l$, the $(i+1)$-th derivation can't be of the form $N_i \to y$, otherwise the derived word would have length $i \cdot (k + l) + |y| < (i+1) \cdot (k+l) \leq n \cdot (k+l) < |w|$. So the only possible derivation is of the form $N_i \to u N_{i+1} v$ with $|u| = k, |v| = l$ and it follows that H_{i+1} is verified with $u_{i+1} = u, v_{i+1} = v$. Finally, the last derivation can only be of the form $N_n \to c$, otherwise the derived word would be longer than $|w|$, thus $x = c$ because it is the last derivation and the final word is w. ■

Lemma 15 *Let G_* be a target (k,l)-step local-context grammar, and let $L = L(G_*)$. For every $S \subseteq L$, if $CS(G_*) \subseteq S$, then $\mathcal{G}^{k,l}(L) = \mathcal{G}'^{k,l}(S) = \mathcal{G}^{k,l}(S)$, where $\mathcal{G}'^{k,l}(S) = \langle \Sigma, V', P', N_S\rangle$ with $V' = V$ and P' defined as follows:*

$$\begin{aligned} P' = &\{N_S \to y \colon y \in S \cap \mathcal{I}^{k,l}(S)\} \\ &\cup \{N_S \to u N_{u,v} v \colon (u,v) \in \mathcal{D}_1^{k,l}(S)\} \\ &\cup \{N_{u,v} \to y \colon \exists w \in S,\ y = \mathcal{I}^{k,l}(w),\ (u,v) = \mathcal{D}^{k,l}_{d^{k,l}(w)}(w)\} \\ &\cup \{N_{u,v} \to u' N_{u',v'} v' \colon \exists w \in S,\ 1 \leq i < d^{k,l}(w),\ (u,v) = \mathcal{D}_i^{k,l}(w),\ (u',v') = \mathcal{D}_{i+1}^{k,l}(w)\}. \end{aligned}$$

Proof Let $S \subseteq L$ such that $CS(G_*) \subseteq S$. As $S \subseteq L$ and as the grammar $\mathcal{G}^{k,l}$ is defined in terms of existence of words in the language, we have by construction $\mathcal{G}'^{k,l}(S) \subseteq \mathcal{G}^{k,l}(S) \subseteq \mathcal{G}^{k,l}(L)$. We now show that $\mathcal{G}^{k,l}(L) = (\Sigma, V, P, N_S) \subseteq \mathcal{G}'^{k,l}(S)$.

- $\forall y \in L \cap \Sigma^{\leq k+l}$, $N_S \to y \in P$. Moreover, $c(N_S) = (\lambda, \lambda)$ and $w(y) = y$ so $y \in S$ and $|y| \leq k + l$ so $N_S \to y \in P'$ by definition.
- $\forall (u,v) \in \mathcal{D}_1^{k,l}(L)$, $N_S \to u N_{u,v} v \in P$. Moreover, $c(N_S) = (\lambda, \lambda)$ so $w(u N_{u,v} v) = u w(N_{u,v}) v \in S$ so by definition, $(u,v) \in \mathcal{D}_1^{k,l}(S)$ and finally $N_S \to u N_{u,v} v \in P'$.
- Let $N_{u,v} \to y \in P$. By Proposition 5, there exists $w \in L$ such that $y = \mathcal{I}^{k,l}(w)$ and $(u,v) = \mathcal{D}^{k,l}_{d^{k,l}(w)}(w)$. We have $w(y) = y$ so by writing $(u',v') = c(N_{u,v})$, $u'yv' \in S$. Whenever $N_{u,v}$ appears in the right side of a production rule, the right side is $u N_{u,v} v$, so by Lemma 14, $(u,v) = \mathcal{D}^{k,l}_{d^{k,l}(u'yv')}(u'yv')$ and $y = \mathcal{I}^{k,l}(u'yv')$ so by definition, $N_{u,v} \to y \in P'$.

- Let $N_{u,v} \to u'N_{u',v'}v' \in P$. By Proposition 6, there exists $w \in L, i \in \{1,\dots,d^{k,l}(w)-1\}$ such that $(u,v) = \mathcal{D}_i^{k,l}(w)$ and $(u',v') = \mathcal{D}_{i+1}^{k,l}(w)$. By writing $(u_0,v_0) = c(N_{u,v})$, by definition, $\exists j \in \mathbb{N}, N_S \Rightarrow^j u_0N_{u,v}v_0 \Rightarrow^* u_0w(N_{u,v})v_0$ and by Lemma 14, $(u,v) = \mathcal{D}_j^{k,l}(u_0w(N_{u,v})v_0)$. This implies that $(u,v) = \mathcal{D}_j^{k,l}(u_0u'w(N_{u',v'})v'v_0)$ and $(u',v') = \mathcal{D}_{j+1}^{k,l}(u_0u'w(N_{u',v'})v'v_0)$. As $w(u'N_{u',v'}v') = u'w(N_{u',v'})v'$, $u_0u'w(N_{u',v'})v'v_0 \in S$ so by definition and the facts above, $N_{u,v} \to u'N_{u',v'}v' \in P'$.

We just proved that every production in $\mathcal{G}^{k,l}(L)$ is also in $\mathcal{G}'^{k,l}(S)$ so $\mathcal{G}^{k,l}(L) \subseteq \mathcal{G}'^{k,l}(S)$. Finally, we have $\mathcal{G}^{k,l}(L) \subseteq \mathcal{G}'^{k,l}(S) \subseteq \mathcal{G}^{k,l}(S) \subseteq \mathcal{G}^{k,l}(L)$, so $\mathcal{G}^{k,l}(L) = \mathcal{G}'^{k,l}(S) = \mathcal{G}^{k,l}(S)$. ■

Lemma 16 *For every $S \subseteq L$, let $G^{k,l}(S)$ be the output of the (k,l)-step learner. Then $G^{k,l}(S) = \mathcal{G}'^{k,l}(S)$.*

Proof To prove this, since we iterate over every word in S, it is sufficient to show that, writing u_i, v_i for the u', v' chosen during the i-th iteration of the while loop, y' for the final value of y and y_0 for the initial value of y, $(u_1,(\dots,(y')\dots),v_1)$ is the (k,l)-context-step decomposition of y_0. If this is the case, for every word $w \in S$, $G^{k,l}(S)$ will contain only the derivation $N_{\lambda,\lambda} \to w$ if $|y| \leq k+l$. Otherwise it will contain the derivations $N_{\lambda,\lambda} \to u_1N_{u_1,v_1}v_1$, $N_{u_i,v_i} \to u_{i+1}N_{u_{i+1},v_{i+1}}v_{i+1}$ for $1 \leq i < d^{k,l}(w)$ and $N_{u_{d^{k,l}(w)},v_{d^{k,l}(w)}} \to y'$, which are exactly the derivations in $\mathcal{G}'^{k,l}(S)$ after replacing $N_{\lambda,\lambda}$ with N_S.

We do this by induction on the length of y at the beginning of the while loop, with the fact that at some point we will have $|y| \leq k+l$. If $|y| \leq k+l$ at the beginning of the while loop, then the loop is not executed, $y' = y$ and (y) is the (k,l)-context-step decomposition of y so the claim holds. If $|y| > k+l$, then $y = u'y''v'$ with $(u',v') \in \Sigma^k \times \Sigma^l$, and we set $y_1 = y$. At the end of the while loop, we have $y = y''$ with $|y''| < |y_1|$ so the induction hypothesis applies, $(u_2,(\dots(y')\dots),v_2)$ is the (k,l)-context-step decomposition of y'' so by definition of the (k,l)-context-step decomposition, the hypothesis is verified.

Thus by induction, the lemma holds. ■

Corollary 17 *For every $S \subseteq L$, if $CS(G_*) \subseteq S$, then the (k,l)-step learner algorithm outputs $\mathcal{G}^{k,l}(L)$ on input S.*

Proof By Lemma 16, the algorithm outputs $\mathcal{G}'^{k,l}(S)$. By Lemma 15, we have $\mathcal{G}^{k,l}(L) = \mathcal{G}'^{k,l}(S)$, which concludes the proof. ■

By Corollary 17, we proved that if the characteristic set is in the input set, then the algorithm outputs the target grammar $\mathcal{G}^{k,l}(L)$. To prove IPTD, we need now to bound polynomially the size of the characteristic set.

Lemma 18 *For a grammar $G = \langle \Sigma, V, P, S\rangle$ in (k,l)-RGNF and a production rule $N \to \alpha \in P$, the smallest word (with the lexicographic order) of $L(G)$ for which $N \to \alpha$ can appear in one of its derivation in G is of size at most $2 \cdot (k+l) \cdot (||G|| + 1)$. i.e, for every $N \to \alpha \in P$, if $(l,r) = c(N)$ and $w = w(\alpha)$, then $|lwr| \leq 2 \cdot (k+l) \cdot (||G|| + 1)$.*

Proof Let w be the smallest such word and let $(u_1, (u_2, (\dots, (u_n, (c), v_n), \dots), v_2), v_1)$ be its (k,l)-context-step decomposition. Let $S \Rightarrow_G u_1N_1v_1 \Rightarrow_G u_1u_2N_2v_2v_1 \Rightarrow_G \dots \Rightarrow_G u_1u_2\dots u_nN_nv_n\dots v_2v_1 \Rightarrow_G w$ be a derivation from S of w (by Lemma 14) such that $N \to \alpha$ appears in it. Suppose by contradiction that $|w| > 2 \cdot (k+l) \cdot (||G|| + 1)$. As every derivation adds at most $k+l$ terminal letters to the final word, this means that there are at least $2 \cdot (||G|| + 1)$ derivations, so $n > 2 \cdot (||G|| + 1) - 1 = 2 \cdot ||G|| + 1$. As $|V| \leq ||G||$, and as for every $i \in \{1, \dots, n\}, N_i \in V$, by the pigeonhole principle, there exists $i_1 < i_2 < i_3 \in \{1, \dots, n\}$ such that $N_{i_1} = N_{i_2} = N_{i_3}$. Now we distinguish two cases:
1) $N \to \alpha = S \to u_1N_1v_1$ or $N \to \alpha = N_n \to c$: since $i_1 \geq 1$ and $i_3 \leq n$, the derivation $S \Rightarrow_G u_1N_1v_1 \Rightarrow_G \dots \Rightarrow_G u_1\dots u_{i_1}N_{i_1}v_{i_1}\dots v_1 \Rightarrow_G u_1\dots u_{i_1}u_{i_3+1}N_{i_3+1}v_{i_3+1}v_{i_1}\dots v_1 \Rightarrow_G \dots \Rightarrow_G u_1\dots u_{i_1}u_{i_3+1}\dots u_ncv_n\dots v_{i_3+1}v_{i_1}\dots v_1$ is a derivation of a word smaller than w from S in G, so it is in $L(G)$ and its derivation contains $N \to \alpha$, so this contradicts the fact that w is the smallest such word.
2) $N \to \alpha = N_i \to u_{i+1}N_{i+1}v_{i+1}$ for some $i \in \{1, \dots, n-1\}$: In this case, we distinguish between two subcases:

a) $i < i_2$: In this subcase, as $N_{i_2} = N_{i_3}$, the derivation $S \Rightarrow_G u_1N_1v_1 \Rightarrow_G \dots \Rightarrow_G u_1\dots u_{i_2}N_{i_2}v_{i_2}\dots v_1 \Rightarrow_G u_1\dots u_{i_2}u_{i_3+1}N_{i_3+1}v_{i_2}\dots v_1 \Rightarrow_G \dots \Rightarrow_G u_1\dots u_{i_2}u_{i_3+1}\dots u_nwv_n\dots v_{i_3+1}v_{i_2}\dots v_1$ is a derivation of a word smaller than w from S in G, so it is in $L(G)$ and it contains $N \to \alpha$, so this contradicts the fact that w is the smallest such word.

b) $i \geq i_2$: In this subcase, as $N_{i_1} = N_{i_2}$, the derivation $S \Rightarrow_G u_1N_1v_1 \Rightarrow_G \dots \Rightarrow_G u_1\dots u_{i_1}N_{i_1}v_{i_1}\dots v_1 \Rightarrow_G u_1\dots u_{i_1}u_{i_2+1}N_{i_2+1}v_{i_2+1}v_{i_1}\dots v_1 \Rightarrow_G \dots \Rightarrow_G u_1\dots u_{i_1}u_{i_2+1}\dots u_ncv_n\dots v_{i_2+1}v_{i_1}\dots v_1$ is a derivation of a word smaller than w from S in G, so it is in $L(G)$ and it contains $N \to \alpha$, so this contradicts the fact that w is the smallest such word.

In every cases, we get a contradiction. This concludes the proof. ■

Lemma 19 *$||CS(G_*)|| \leq p(||G_*||)$ for some p polynomial.*

Proof By definition of $CS(G_*)$, $|CS(G_*)| \leq |P| \leq ||G_*||$. Moreover, by Lemma 18, every $w \in CS(G_*)$ satisfies $|w| \leq 2(k+l)(||G_*|| + 1)$. Hence, the encoding size of $CS(G_*)$ is bounded by the product of its cardinality and maximal word length, yielding $||CS(G_*)|| \leq 2(k+l)(||G_*|| + 1) \cdot ||G_*||$, which is a polynomial in $||G_*||$. ■

4.3. IPTD result

By Propositions 5 and 6 and Lemmas 16, 15 and 19, $CS(G)$ is a characteristic set for G whose size is polynomial in $||G||$. Together with Lemma 12, which establishes the linear-time complexity of the learning algorithm, this yields that the class of (k,l)-step local-context grammars generating (k,l)-LCS languages is IPTD.

Note that every (k,l)-LCS language is also an (m,n)-LCS language for all $m \leq k$ and $n \leq l$, with $m + n > 0$, so overestimating (k,l) preserves IPTD identifiability at the cost of a larger training sample.

5. Conclusion and perspectives

We introduced (k,l)-step local-context grammars as a representation class for (k,l)-local-context substitutable languages, and thus for (k,l)-local substitutable languages, together with a linear-time inference algorithm, and showed that this class is IPTD. Since these language classes are regular, owing to their finite number of congruence classes, the significance of this result lies in the choice and expressiveness of the representation rather than in the languages themselves. (k,l)-step local-context grammars provide a compact yet canonical context-free representation capturing nested dependencies and contextualized syntactic or semantic categories while enabling feasible grammar inference in polynomial time and data.

This work opens several perspectives. First, (k,l)-step local-context grammars form a subclass of context-free linear grammars. This class is not IPTD for any alphabet Σ of size at least 2 (de la Higuera, 1997) but some of its subclasses are IPTD, building on the seminal results on even linear grammars (Takada, 1988; Sempere and García, 1994). Understanding the relationship between these results, and in particular how the locality criterion contributes to IPTD learnability and whether it could yield similar results for other classes of linear grammars, warrants further investigation. From an applied grammatical inference perspective, (k,l)-local-context substitutable languages can be seen as an extension of locally testable regular languages, which are efficiently learnable and have been applied to pattern recognition and biological sequence modeling (Garcia and Vidal, 1990; Yokomori and Kobayashi, 1998; Peris et al., 2006). In this lineage, local strings are lifted to local contexts, yielding a context-free generalization of local testability. Investigating the connections between these formalisms, and more generally between local grammatical representations, is a promising direction for advancing expressive and feasible grammatical inference.

References

Alexander Clark and Rémi Eyraud. Polynomial identification in the limit of substitutable context-free languages. *Journal of Machine Learning Research*, 8:1725–1745, 2007.

Alexander Clark and Ryo Yoshinaka. Distributional learning of context-free and multiple context-free grammars. In *Topics in Grammatical Inference*. Springer Berlin Heidelberg, 2016.

François Coste and Jacques Nicolas. Learning local substitutable context-free languages from positive examples in polynomial time and data by reduction. In *ICGI*, volume 93 of *PMLR*, pages 155–168, 2018.

François Coste, Gaëlle Garet, and Jacques Nicolas. Local substitutability for sequence generalization. In *ICGI*, volume 21 of *JMLR Workshop and Conference Proceedings*, pages 97–111, 2012.

Colin de la Higuera. Characteristic sets for polynomial grammatical inference. *Machine Learning*, 27(2):125–138, 1997.

Joost Engelfriet. An elementary proof of double greibach normal form. *Inf. Process. Lett.*, 44:291–293, 1992.

Rémi Eyraud, Jeffrey Heinz, and Ryo Yoshinaka. Efficiency in the identification in the limit learning paradigm. In *Topics in Grammatical Inference*. Springer-Verlag Berlin Heidelberg, 2016.

P. Garcia and E. Vidal. Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):920–925, 1990.

E Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

Sheila A. Greibach. A new normal-form theorem for context-free phrase structure grammars. *J. ACM*, 12(1):42–52, 1965.

John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation - international edition, 2nd Edition*. Addison-Wesley, 2003.

Makoto Kanazawa and Ryo Yoshinaka. Extending distributional learning from positive data and membership queries. In *ICGI*, volume 217 of *PMLR*, pages 8–22, 2023.

Yutaro Numaya, Diptarama Hendrian, Ryo Yoshinaka, and Ayumi Shinohara. Identification of substitutable context-free languages over infinite alphabets from positive data. In *ICGI*, volume 217 of *PMLR*, pages 23–34, 2023.

Piedachu Peris, Damián López, Marcelino Campos, and José M. Sempere. Protein motif prediction by grammatical inference. In *ICGI*, volume 4201 of *LNCS*, pages 175–187, 2006.

Leonard Pitt. Inductive inference, dfas, and computational complexity. In *Analogical and Inductive Inference*, pages 18–44, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

Daniel J. Rosenkrantz. Matrix equations and normal forms for context-free grammars. *J. ACM*, 14(3):501–507, 1967.

José M. Sempere and Pedro García. A characterization of even linear languages and its application to the learning problem. In *ICGI*, volume 862 of *LNCS*, pages 38–44. Springer, 1994.

Yuji Takada. Grammatical interface for even linear languages based on control sets. *Inf. Process. Lett.*, 28(4):193–199, 1988.

Takashi Yokomori and Satoshi Kobayashi. Learning local languages and their application to dna sequence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1067–1079, 1998.

Ryo Yoshinaka. Identification in the limit of k, l-substitutable context-free languages. In *ICGI*, volume 5278 of *LNCS*, pages 266–279. Springer, 2008.

Ryo Yoshinaka. Learning mildly context-sensitive languages with multidimensional substitutability from positive data. In *ALT*, volume 5809 of *LNCS*, pages 278–292. Springer, 2009.