J Log Lang Inf (2015) 24:1–26
DOI 10.1007/s10849-014-9211-2

On the Effect of the IO-Substitution on the Parikh Image of Semilinear Full AFLs

Pierre Bourreau

Published online: 27 November 2014

Abstract Back in the 1980's, the class of mildly context-sensitive formalisms was introduced so as to capture the syntax of natural languages. While the languages generated by such formalisms are constrained by the constant-growth property, the most well-known and used ones—like tree-adjoining grammars or multiple context-free grammars—generate languages which verify the stronger property of being semilinear. In (Bourreau et al. 2012), the operation of IO-substitution was created so as to exhibit mildly-context sensitive classes of languages which are not semilinear. In the present article, we extend the notion of semilinearity, and characterize the Parikh image of the languages in **IO(L)**, the closure of a class **L** of semilinear languages under IO-substitution, as universally-semilinear. Based on this result and on the work of Fischer on macro-grammars, we then show that **IO(L)** is not closed under inverse homomorphism when **L** is closed under inverse homomorphism, and encompasses the class of regular languages. This result proves that **IO(MCFL)** is not a full AFL, where **MCFL** denotes the class of multiple context-free languages, closing an open question in Bourreau et al. (2012). More importantly, our proof gives an insight into the relation between the non-closure under inverse homomorphism of **IO**(**MCFL**) and how IO-substitution breaks semilinearity.

Keywords Formal languages · Mildly context-sensitive formalisms · Semilinearity · Constant-growth · IO macro-grammars · Multiple context-free grammars · Abstract family of languages

P. Bourreau (✉)
Institut für Sprache und Information, Heine-Heinrich Universität Düsseldorf,
Universitätstr. 1, 40225 Düsseldorf, Germany
e-mail: pierre.bourreau@gmail.com

1 Introduction

The mathematical description of natural language syntax is a problem which has captured the attention of scientists for many years. The initial work of Chomsky (1956) on formal languages led to the first approximation of natural languages as context-free languages. Nowadays, it is commonly accepted that the class of context-free languages is too weak to entirely capture the structure of syntax. This was first proved in Huybregts (1984) and Shieber (1985) through examples in Swiss-German, and later on confirmed in Michaelis and Kracht (1997) and discussed in Kobele (2006). In order to define a new family of formal languages that would approximate natural languages, Joshi (1985) defined a class of formalisms which he called *mildly context-sensitive*, in an attempt to answer the question: "How much context-sensitivity is needed to provide reasonable structural descriptions?"; mildly context-sensitive formalisms are defined through the following conditions:

1. the class of languages generated by such a formalism must encompass the class of context-free languages;
2. they must take into account some limited cross-serial dependencies;
3. they must be recognizable in polynomial-time;
4. and, the generated languages must verify the constant-growth property;

While this definition and the answer it gives to the initial question are under debate, we focus on the fourth point of the definition and the notion of constant-growth property. Indeed, many mildly context-sensitive formalisms are known to verify the stronger property of generating semilinear languages. That is for instance the case of tree-adjoining grammars, multiple context-free grammars (Seki et al. 1991) (or alternatively linear context-free rewriting systems Vijay-Shanker et al. 1987), or minimalist grammars (Stabler 1996; Michaelis 1998). In the following work, we investigate the gap between semilinear languages and languages which verify the constant-growth property.

In (Bourreau et al. 2012), an operation on languages called IO-substitution was defined. This operation allows one to enrich a class of languages with a limited copying mechanism. IO-substitution can indeed be seen as a bounded copying operation on strings. In this preliminary work, Bourreau et al. (2012) proved three main properties. First, given a full abstract family of semilinear languages $\mathbf{L}$, its closure under IO-substitution $\mathbf{IO}(\mathbf{L})$ forms a family of languages which is closed under union, concatenation, homomorphism and intersection with regular sets; an open question is therefore to prove whether $\mathbf{IO}(\mathbf{L})$ is a full abstract family of languages. Moreover, it was proved that if the languages in $\mathbf{L}$ verify the constant-growth property, so do the languages in $\mathbf{IO}(\mathbf{L})$. Finally, in the special case where $\mathbf{L} = \mathbf{MCFL}$, the class of multiple context-free languages, the authors showed that any language in $\mathbf{IO}(\mathbf{MCFL})$ can be recognized in polynomial-time; these first results lead to considering the formalisms whose generated languages fall within $\mathbf{IO}(\mathbf{MCFL})$ as candidates for being mildly context-sensitive.

In the present article, we investigate a precise characterization of the Parikh image of languages in $\mathbf{IO}(\mathbf{L})$, where $\mathbf{L}$ is a family of semilinear languages (i.e. as a particular case, the results we obtain apply when $\mathbf{L}$ is the family of regular, context-free, or multiple context-free languages of strings). In order to do so, we extend the notion of semilinearity in a natural way, by defining two new characterizations for sets of vectors: *existentially-semilinear sets* and *universally-semilinear sets*, and show that the Parikh image of $\mathbf{IO}(\mathbf{L})$ falls within the second one, leading, as a corollary, to an alternative proof that such languages verify the constant-growth property. In the second part of the article, we give a proof of the non-closure of $\mathbf{IO}(\mathbf{L})$ under inverse homomorphism, where $\mathbf{L}$ is a full abstract family of semilinear languages which contains **REG**, the set of regular languages. This result, which is obtained thanks to the previous characterization of the Parikh image for the considered languages and by generalizing the main ideas of Fischer's proof of the non-closure of IO-macro grammars under inverse homomorphism, shows that $\mathbf{IO}(\mathbf{MCFL})$ is not a full abstract family of languages.

The outline of this document is the following: Sect. 2 defines the fundamental notions needed from formal language theory: the Parikh image, semilinearity, the constant-growth property, and the IO-substitution. In Sect. 3, we introduce universal-semilinearity and existential-semilinearity as natural extensions of semilinearity, we compare these notions with each other and with the constant-growth property, and we show that the languages in $\mathbf{IO}(\mathbf{L})$ verify the constant-growth property. Finally, Sect. 4 is dedicated to proving the non-closure of $\mathbf{IO}(\mathbf{L})$ under inverse homomorphism, for $\mathbf{L}$ a semilinear full AFL such that $\mathbf{REG} \subseteq \mathbf{L}$; this proof will also bring the opportunity to study new structural properties of $\mathbf{IO}(\mathbf{L})$.

2 Semilinearity, Constant-Growth and IO-Substitution

2.1 Formal Languages, Constant-Growth and Semilinearity

We first introduce the notations for various usual notions related to formal languages. Given a set Σ (called an alphabet), we write Σ^* for the set of words built on Σ, and ϵ for the empty word. Given w in Σ^*, we write $|w|$ for its length, and $|w|_a$ for the number of occurrences of a letter a of Σ in w. A language on Σ is a subset of Σ^*. Given a language L, we will speak of the alphabet Σ of L to designate any set such that $L \subseteq \Sigma^*$. Given two languages $L_1, L_2 \subseteq \Sigma^*$, $L_1 \cdot L_2$, the concatenation of L_1 and L_2, is the language $\{w_1 w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$; the union of L_1 and L_2 is written $L_1 + L_2$.

We write $\mathbb{N}$ for the set of natural numbers. For a finite alphabet Σ, $\mathbb{N}^\Sigma$ is the set of vectors whose coordinates are indexed by the letters of Σ. Vectors will be denoted by $\overrightarrow{v}$, and an n-dimensional vector ($n \in \mathbb{N}$) will be written $\langle c_1, \ldots, c_n \rangle$ (where $c_1, \ldots, c_n \in \mathbb{N}$) when we wish to exhibit the values of the vector on each of its dimension. Given $a \in \Sigma$ and $\overrightarrow{v} \in \mathbb{N}^\Sigma$, $\overrightarrow{v}[a]$ will denote the value of $\overrightarrow{v}$ on the dimension a.

In (Joshi 1985), the constant-growth property was introduced as a condition languages generated by mildly context-sensitive formalisms must verify. This condition expresses some constraints on the distribution of the length of the words in a language:

Definition 1 (*Constant-growth*) A language $L \subseteq \Sigma^*$ is said to be *constant-growth* if there exist $k, c \in \mathbb{N}$ such that, for every $w \in L$, if $|w| > k$, then there is $w' \in L$ for which $|w| < |w'| \leq |w| + c$.

As mentioned in the introduction, most of the mildly context-sensitive formalisms commonly used in modeling natural language syntax generate languages which verify the stronger property of semilinearity.

Definition 2 (*Parikh image*) Let us consider a word w in a language $L \subseteq \Sigma^*$. The *Parikh image of* w, written $\overrightarrow{p}(w)$ is the vector of $\mathbb{N}^\Sigma$ such that, for every $a \in \Sigma$, $\overrightarrow{p}(w)[a] = |w|_a$. The Parikh image of L is defined as $\overrightarrow{p}(L) = \{\overrightarrow{p}(w) \mid w \in L\}$.

Definition 3 (*Semilinearity*) A set V of vectors of $\mathbb{N}^\Sigma$ is said to be *linear* when there are vectors $\overrightarrow{v_0}, \ldots, \overrightarrow{v_n}$ in $\mathbb{N}^\Sigma$ such that $V = \{\overrightarrow{v_0} + k_1\overrightarrow{v_1} + \ldots + k_n\overrightarrow{v_n} \mid k_1, \ldots, k_n \in \mathbb{N}\}$.

A set of vectors is said to be *semilinear* when it is a (possibly empty) finite union of linear sets.

Given two sets of vectors V_1 and V_2 of $\mathbb{N}^k$, for $k \in \mathbb{N}$, we will write $V_1 + V_2$ for the set $\{\overrightarrow{v_1} + \overrightarrow{v_2} \mid \overrightarrow{v_1} \in V_1, \overrightarrow{v_2} \in V_2\}$. Similarly, given $c \in \mathbb{N}$ and a set of vectors V of $\mathbb{N}^k$, we will write $cV = \{c\overrightarrow{v} \mid \overrightarrow{v} \in V\}$.

Definition 4 A language L is said *semilinear* when $\overrightarrow{p}(L)$ is a semilinear set.

Well-known classes of semilinear languages are the class **REG** of regular languages, the class **CFL** of context-free languages, the class **yTAL** of yields of tree-adjoining languages or the class **MCFL** of multiple context-free languages.

Definition 5 Given a class of languages **L** and a class of sets of vectors $\mathcal{V}$, we say that ***L*** yields $\mathcal{V}$ if the two following conditions are verified:

- for every language $L \in \mathbf{L}$, $\overrightarrow{p}(L) \in \mathcal{V}$;
- for every $V \in \mathcal{V}$, there exists $L \in \mathbf{L}$ such that $\overrightarrow{p}(L) = V$.

It is known that **REG** yields the class of semilinear sets. Consequently, **CFL**, **yTAL** and **MCFL** also verify this property as they include all languages in **REG**, and every language in these classes is a semilinear language.

Given two alphabets Σ_1 and Σ_2, a string homomorphism h from Σ_1^* to Σ_2^* is a function such that $h(\epsilon) = \epsilon$ and $h(w_1w_2) = h(w_1)h(w_2)$, where $w_1, w_2 \in \Sigma_1^*$. Given $L \subseteq \Sigma_1^*$, we write $h(L)$ for the language $\{h(w) \in \Sigma_2^* \mid w \in L\}$.

Let us consider a class **L** of languages. Given an n-ary operation $\mathtt{op} : (\Sigma^*)^n \to \Sigma^*$ on strings (where $n \in \mathbb{N}$), we say that **L** is closed under $\mathtt{op}$ if for every $L_1, \ldots, L_n \in \mathbf{L}$, $\mathtt{op}(L_1, \ldots, L_n) \in \mathbf{L}$.

Definition 6 (*Full AFLs*) A class of languages **L** is called a *full abstract family of languages* (written full AFL for concision) if it is closed under union, concatenation, Kleene star, homomorphism, inverse homomorphism and intersection with regular sets.

The previously defined classes **REG**, **CFL**, **yTAL**, and **MCFL** are known to be full AFLs. Moreover, because the languages in these classes are semilinear, we say that they are semilinear full AFLs.

2.2 IO-Substitution: Going Beyond Semilinearity

In (Bourreau et al. 2012), the operation of IO-substitution was defined so as to enrich languages with a limited copying operation.

Definition 7 (*IO-substitution*) Let us consider the alphabets Σ_1 and Σ_2, and two languages $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$. Given a word $w \in L_2$, and a symbol $a \in \Sigma_1$, we define the homomorphism $io_{a,w}$ based on the function

$$\begin{aligned} io_{a,w} : \Sigma_1 &\to \Sigma_2^* \\ c &\mapsto \begin{cases} w & \text{if } c = a \\ c & \text{otherwise} \end{cases} \end{aligned}$$

We define the language resulting from the IO-substitution of the symbol a by the language L_2 in the language L_1 as

$$L_1[a := L_2]_{IO} = \bigcup_{w \in L_2} io_{a,w}(L_1)$$

and we call $L_1[a := L_2]_{IO}$ the *IO-substitution of* a *by* L_2 *in* L_1.

Note that, if for every word $w \in L_1$, a has no occurrence in w (i.e. $|w|_a = 0$) then $L_1[a := L_2]_{IO} = L_1$ is verified. We adopt a convention of left-associativity for the IO-substitution operation: therefore $L_1[x_1 := L_2]_{IO}[x_2 := L_3]_{IO}$ will denote the language $(L_1[x_1 := L_2]_{IO})[x_2 := L_3]_{IO}$.

Example 1 Let us consider the languages $L_1 = a^*$ and $L_2 = ab + c$; the language $L = L_1[a := L_2]_{IO}$ is then defined as $(ab)^* + c^*$. In this particular case, L is a regular language, just like L_1 and L_2.

A more interesting example is $L_{nprime} = \{a^p \mid p \text{ is not a prime number}\}$, which verifies $L_{nprime} = xx^*x[x := aa^*a]_{IO}$. Such a language is not semilinear since its Parikh image is equal to $\{nm\langle 1\rangle \mid n, m > 1\}$. Therefore L_{nprime} does not belong to **REG**, while xx^*x and aa^*a do.

Definition 8 (*IO(L)*) Given a class of languages $\mathbf{L}$, we define the class $IO_n(\mathbf{L})$ by induction on $n \in \mathbb{N}$ as

1. $IO_0(\mathbf{L}) = \mathbf{L}$
2. for $n \geq 0$,

$$IO_{n+1}(\mathbf{L}) = IO_n(\mathbf{L}) \cup \bigcup_{L_1, L_2 \in IO_n(\mathbf{L})} \bigcup_{x \in \Sigma_1} L_1[x := L_2]_{IO}$$

where Σ_1 is the alphabet of L_1

The smallest class of languages containing $\mathbf{L}$ and closed under IO-substitution is defined by $\mathbf{IO}(\mathbf{L}) = \{L \in IO_n(\mathbf{L}) \mid n \in \mathbb{N}\}$

We introduce a notion of derivation tree associated to a language in **IO(L)**.

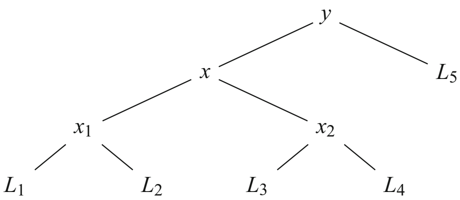


Fig. 1 Example of a derivation tree associated to a language in **IO(L)**

Definition 9 (*Derivation tree*) Given a language L in $\mathbf{IO}(\mathbf{L})$, we define the set of *derivation trees* $\mathcal{T}_L$ associated to L as the smallest set such that:

- if $L \in \mathbf{L}$, $t = n \in \mathcal{T}_L$, where n is a node labelled with L;
- if $L_1[x := L_2]_{IO} = L$ for $L_1, L_2 \in \mathbf{IO}(\mathbf{L})$, then, given n a node labelled with x, $\{n(t_1, t_2) \mid t_1 \in \mathcal{T}_{L_1}, t_2 \in \mathcal{T}_{L_2}\} \subseteq \mathcal{T}_L$.

Example 2 Let us consider some languages $L_i \in \mathbf{L}$ for $1 \leq i \leq 5$ and the language

$$((L_1[x_1 := L_2]_{IO})[x := (L_3[x_2 := L_4]_{IO})]_{IO})[y := L_5]_{IO}$$

The corresponding derivation tree is represented by the binary tree in Figure 1.

There is an obvious correspondence between the notation $x(t_1, t_2)$ of a derivation tree of a language L (where $t_1 \in \mathcal{T}_{L_1}$ and $t_2 \in \mathcal{T}_{L_2}$), and the notation $L_1[x := L_2]_{IO}$ of the language L itself; in the rest of the article, whenever we have an equality $L_1[x := L_2]_{IO} = L$, we will say that $L_1[x := L_2]_{IO}$ is a *representation* of L. More formally, given $L \in \mathbf{IO}(\mathbf{L}))$, L is a representation of L; and if $L = L_1[x := L_2]_{IO}$, where $L_1, L_2 \in \mathbf{IO}(\mathbf{L})$ for every representation r_1 of L_1, and every representation r_2 of L_2, $r_1[x := r_2]_{IO}$ is a representation of L.

As pointed out in Bourreau et al. (2012), the IO-substitution operation can be seen as a restriction of the copying power of IO-macro grammars in Fischer (1968a) and Fischer (1968b). Indeed, the authors gave a grammatical formalism in terms of abstract categorial grammars (de Groote 2001; Muskens 2001) which generates languages in **IO(MCFL)**, and the construction exhibits the use of copies in a non-recursive way, i.e. the use of a bounded number of copies; this restriction leads, for instance, to exclude languages like $L_{sq} = \{a^{n^2} \mid n \in \mathbb{N}\}$ from **IO(MCFL)**, while such a language is known to be generated by IO-macro grammars (and also by parallel MCFGs (Seki et al. 1991), another formalism that enriches MCFGs with deletion and copying operations).

One will note that L_{sq} is not a constant-growth language. In fact, one can show that the IO-substitution preserves the constant-growth property of languages under the constraints given in the following theorem:

Theorem 1 (Bourreau et al. 2012) *Given* $\mathbf{L}$ *a full abstract family of semilinear languages:*

- ***IO(L)*** *is a family of constant-growth languages*
- ***IO(L)*** *is closed under homomorphism, intersection with regular sets, finite union and concatenation.*

We now investigate a precise characterization of the Parikh image of languages in **IO(L)**, and give an alternative proof of the constant-growth property for the languages in this class. In order to do so, we will provide and discuss some natural extensions of semilinear sets.

3 IO-MCFLs have Factorized Parikh Images

3.1 Constant-Growth, ∃-Semilinear and ∀-Semilinear Sets

As mentioned in the previous section, the Parikh image of **IO(L)** goes beyond semilinear sets, while being captured by the notion of constant-growth. We present a generalization of the notion of semilinear sets with respect to the following remark: one can see a linear set $V = \{\overrightarrow{v_0} + x_1\overrightarrow{v_1} + \cdots + x_n\overrightarrow{v_n} \mid x_1, \ldots, x_n \in \mathbb{N}\}$, where $\overrightarrow{v_0}, \ldots, \overrightarrow{v_n} \in \mathbb{N}^p$, as the image of the function $f : \mathbb{N}^n \to \mathbb{N}^p$ such that $f(x_1, \ldots, x_n) = \overrightarrow{v_0} + x_1\overrightarrow{v_1} + \cdots + x_n\overrightarrow{v_n}$, hence parameterized by the variables $x_1, x_2, \ldots, x_n$. In this context, we will establish relaxed conditions of linearity on the function f such that the sets of vectors built on these new functions approximate the Parikh image of constant-growth languages, or those of languages in **IO(L)**, where **L** yields semilinear sets.

Definition 10 (*Functional Decomposition*) Given a vector set $E \subseteq \mathbb{N}^p$, (where $p \geq 1$), and a finite set of vector functions $F = \{F_1, \ldots, F_m\}$, $m \geq 0$ such that, for every $1 \leq k \leq m$, the codomain of F_k is $\mathbb{N}^p$, we call F *a functional decomposition of* E if $E = \bigcup_{1\leq k\leq m} \mathrm{Im}(F_k)$.

Following this definition, a semilinear set can be defined as a vector set for which there exists a functional decomposition made of linear functions. Remark that, in the decomposition F of a vector set, we will assume that the functions in F share the same codomain.

Next we define specific functions with the aim of approximating the ideas behind the constant-growth property.

Definition 11 (***i****-linear function*) Given a vector function $F : \mathbb{N}^n \to \mathbb{N}^m$ and $1 \leq i \leq n$, F is said to be *i-linear*, if there exists $(f_j, \overrightarrow{v_j})_{1\leq j\leq m}$, where for all $1 \leq j \leq m$, $f_j : \mathbb{N}^n \to \mathbb{N}$ and $\overrightarrow{v_j} \in \mathbb{N}^m$, such that $F(x_1, \ldots, x_n) = \sum_{1\leq j\leq m} f_j(x_1, \ldots, x_n)\overrightarrow{v_j}$ and for every $1 \leq j \leq m$, f_j is affine in x_i:

$$f_j(x_1, \ldots, x_n) = A_j(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)x_i + B_j(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

where $A_j, B_j : \mathbb{N}^{n-1} \to \mathbb{N}$, and A_j is not constantly 0.

We say that F is:

- *existentially-linear* (written ∃-linear) if there exists $1 \leq i \leq n$, for which F is i-linear.
- *universally-linear* (written ∀-linear) if for every $1 \leq i \leq n$, F is i-linear.

An important remark is that the notion of ∀-linear functions is a natural extension of that of linear functions; indeed, a linear function $f(x_1, \ldots, x_n)$ has an affine form on each of its parameters: $f(x_1, \ldots, x_n) = x_i \overrightarrow{v_i} + (v_o + \Sigma_{1 \leq k \leq i-1} x_k \overrightarrow{v_k} + \Sigma_{i+1 \leq k \leq n} x_k \overrightarrow{v_k})$; with respect to the notion of multilinear functions (a well-established notion in the domain of multilinear algebra), such a function could be called multi-affine. The notion of ∀-linear functions generalizes this property and seems to capture all multi-affine functions.

These definitions are then naturally extended to sets of vectors:

Definition 12 (*Existentially- and Universally-semilinear sets*) A vector set E is said to be:

- existentially-linear (written ∃-linear) if there exists a functional decomposition of E in which at least one function is ∃-linear.
- universally-linear (written ∀-linear) if there exists a functional decomposition of E in which all functions are ∀-linear.

Finally, a ∃-semilinear (*resp.* ∀-semilinear) set is a finite union of ∃-linear (*resp.* ∀-linear) sets.

It is trivial that a ∀-linear vector function is ∃-linear. Similarly, a ∀-semilinear set is ∃-semilinear. Following the terminology for semilinearity, we will speak of a ∃-semilinear (*resp.* ∀-semilinear) language when its Parikh image is ∃-semilinear (*resp.* ∀-semilinear).

Lemma 1 *Given a language L, if L is ∃-semilinear then L is constant-growth.*

Proof Let us consider such a language L; there exists a functional decomposition of $\overrightarrow{p}(L)$ such that $\bigcup_{1 \leq i \leq k} \mathrm{Im}(F_i) = \overrightarrow{p}(L)$ and there exists $1 \leq i \leq k$ such that F_i is ∃-linear, i.e., for $\mathrm{Dom}(F_i) = \mathbb{N}^n$, there exists a finite family of functions $(f_{i,j})_{1 \leq j \leq m_i}$ such that $F_i(x_1, \ldots, x_n) = \sum_{1 \leq j \leq m_i} f_{i,j}(x_1, \ldots, x_n) \overrightarrow{v_j}$, and there exists $1 \leq l \leq n$ for which, for every $1 \leq j \leq m_i$:

$$f_{i,j}(x_1, \ldots, x_n) = A_j(x_1, \ldots, x_{l-1}, x_{l+1}, \ldots, x_n) x_l + B_j(x_1, \ldots, x_{l-1}, x_{l+1}, \ldots, x_n)$$

Let us consider $c_1, \ldots, c_n \in \mathbb{N}$. We then write

$$\begin{aligned} A &= \sum_{1 \leq j \leq m_i} A_j(c_1, \ldots, c_{l-1}, c_{l+1}, \ldots, c_n) \\ B &= \sum_{1 \leq j \leq m_i} B_j(c_1, \ldots, c_{l-1}, c_{l+1}, \ldots, c_n) \\ K &= A c_l + B \end{aligned}$$

Then, because A cannot be constantly 0, we can build an (increasing) sequence of words $(w_i)_{i \in \mathbb{N}}$, such that $|w_i| = K + iA$, for every $i \in \mathbb{N}$. Therefore, given a word $w \in L$ such that $|w| > K$, we can find $i \in \mathbb{N}$ such that $|w_i| < |w| \leq |w_{i+1}|$, i.e. $|w_i| < |w| \leq |w_i| + A$. □

The family of ∃-semilinear sets seems to be the biggest family of constant-growth sets of vectors definable from the definition of functional decomposition of vector-sets. Moreover, ∃-semilinear sets gives an insight into the gap between constant-growth and semilinear languages generated by mildly context sensitive formalisms. This is illustrated, for instance, with the language $L = \{a^{n^2} b^m c^{nm} \mid n, m \in \mathbb{N}\}$, whose Parikh image is given by $\mathrm{Im}(F)$, where $F(x_1, x_2) = x_1^2\langle 1, 0, 0\rangle + x_2\langle 0, 1, 0\rangle + x_1 x_2\langle 0, 0, 1\rangle$. Then F is $\exists$-linear (for x_2) and therefore, L is constant-growth.

Languages whose Parikh images are $\exists$-semilinear vector sets can be seen as languages which have a "linear sub-basis". Indeed, the definition of a $\exists$-semilinear set of vectors states that an infinite subset of it verifies a linear growth. As a particular case, and if we only consider the formal definition of mildly context-sensitivity, formalisms which allow copy mechanisms should be considered as candidates for mildly-context sensitive formalisms as soon as they ensure such a linear sub-basis in the languages generated. Such a property might be interesting in the description of natural language syntax, in case one wants to describe ellipsis through copying operations (Sarkar and Joshi 1996; Kobele 2007; Bourreau 2013), or to integrate copying phenomena appearing in Yes-No questions in Mandarin (Radzinski 1990), or in relatives in Bambara (Culy 1987), for instance.

Finally, we can remark that $\forall$-semilinear languages seem to be closer to the ideas expressed in the revisited constant-growth property of Kallmeyer (2010), where a language is constant-growth if there exists a constant $c \in \mathbb{N}$, such that for every word $w \in L$ verifying $|w| > c$, there are vectors $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ for which $\overrightarrow{p}(w) = \overrightarrow{v_1} + \overrightarrow{v_2}$ and for every $k \geq 1$, $\overrightarrow{v_1} + k\overrightarrow{v_2} \in \overrightarrow{p}(L)$. Indeed, any word w in a $\forall$-semilinear language L belongs to a sublanguage of L which verifies the revised constant-growth property, and this sublanguage is given by the vector function associated to w.

3.2 Factored Parikh Image

We now give a precise characterization of the Parikh image of languages in **IO(L)**. We will prove that such an image is a particular case of $\forall$-semilinear sets. This result leads to a proof of the constant-growth property of these languages, which differs from the one given in Bourreau et al. (2012). Moreover, we show that **IO**(**L**) does not yield $\forall$-semilinear sets, and instead, we exhibit a less natural family of vector sets yielded by **IO**(**L**).

In what follows, we denote by $\mathcal{F}(\mathbb{N}^n, \mathbb{N}^m)$ the set of vector functions whose domain is $\mathbb{N}^n$ and whose codomain is $\mathbb{N}^m$. Moreover, given a vector $\overrightarrow{v} = \langle v_1, \dots, v_n\rangle$ on $\mathbb{N}^n$ and an integer $1 \leq k \leq n$, we use $\mathrm{On}_k(\overrightarrow{v})$ to denote the value v_k on the k^{th} projection of $\overrightarrow{v}$, and $\mathrm{Wtt}_k(\overrightarrow{v})$ to denote the vector $\langle v_1, \dots, v_{k-1}, 0, v_{k+1}, \dots, v_n\rangle$. These notations are extended in the following definition:

Definition 13 Let us consider a function $F : \mathbb{N}^n \to \mathbb{N}^m$, and $1 \leq k \leq m$. We define the functions Wtt_k (without the k^{th} projection) and On_k (value on the k^{th} projection) as:

- $\mathrm{Wtt}_k : \mathcal{F}(\mathbb{N}^n, \mathbb{N}^m) \to \mathcal{F}(\mathbb{N}^n, \mathbb{N}^m)$ such that $\mathrm{Wtt}_k(F)(x_1, \dots, x_n) = \mathrm{Wtt}_k(F(x_1, \dots, x_n))$
- $\mathrm{On}_k : \mathcal{F}(\mathbb{N}^n, \mathbb{N}^m) \to \mathcal{F}(\mathbb{N}^n, \mathbb{N})$ such that $\mathrm{On}_k(F)(x_1, \dots, x_n) = \mathrm{On}_k(F(x_1, \dots, x_n))$

When a vector will be associated to the Parikh image of a language, we will allow ourselves to index these two functions by the letter corresponding to the dimension, and use the notations Wtt_x and On_x.

From the functions Wtt_k and On_k, we define the following notion of a factored-semilinear Parikh image.

Definition 14 (*Factored function*) A function $F : \mathbb{N}^n \to \mathbb{N}^m$ is said to be *factored* if the following induction is verified:

1. F is a linear function, or
2. there exist $1 \leq k \leq m$, $F_1 : \mathbb{N}^{n_1} \to \mathbb{N}^m$ and $F_2 : \mathbb{N}^{n_2} \to \mathbb{N}^m$, factored vector functions such that $n = n_1 + n_2$ and

$$F(x_1, \dots, x_n) = \mathrm{Wtt}_k(F_1)(x_1, \dots, x_{n_1}) + \mathrm{On}_k(F_1)(x_1, \dots, x_{n_1})F_2(x_{n_1+1}, \dots x_n)$$

In the rest of the document, we allow ourselves to write $\mathrm{Wtt}_k(F_1) + \mathrm{On}_k(F_1)F_2$ for a function as in 2. in the definition above.

Definition 15 (*Synchronized set of factored functions*) A finite set of factored function $E = \{F_1, \dots, F_n\}$ is called a *synchronized set of factored functions* if the functions in E share the same codomain $\mathbb{N}^m$ and the following induction is verified:

1. for every $1 \leq i \leq n$, F_i is a linear function; or
2. there exist $1 \leq k \leq m$, and synchronized sets of factored vector functions $E_1 = \{F_{1,1}, \dots, F_{1,n_1}\}$ and $E_2 = \{F_{2,1}, \dots, F_{2,n_2}\}$ such that,

$$E = \bigcup_{1 \leq i_1 \leq n_1} \bigcup_{1 \leq i_2 \leq n_2} \mathrm{Wtt}_k(F_{1,i_1}) + \mathrm{On}_k(F_{1,i_1})F_{2,i_2}$$

A set S for which there exists a functional decomposition $S = \bigcup_{1 \leq i \leq n} \mathrm{Im}(F_i)$ such that $\{F_1, \dots, F_n\}$ forms a synchronized set of factored functions is called a *factored-semilinear set*.

We now prove that **IO**(**L**) yields factored-semilinear sets, if **L** yields linear sets.

Proposition 1 *If every language $L \in \mathbf{L}$ has a semilinear Parikh image, then every language in* **IO**(**L**) *has a factored-semilinear Parikh image.*

Proof By definition, there exists $n \in \mathbb{N}$ such that $L \in IO_n(\mathbf{L})$. We proceed by induction on n:

- if $n = 0$, then L belongs to **L**. By definition $\overrightarrow{p}(L)$ is a semilinear set, therefore a factored-semilinear vector set.
- otherwise, there exist $L_1, L_2 \in IO_{n-1}(\mathbf{L})$ such that $L_1[x := L_2]_{IO} = L$. By induction hypothesis, there exist $\{F_{1,1}, \dots, F_{1,n_1}\}$ and $\{F_{2,1}, \dots, F_{2,n_2}\}$, synchronized sets of factored vector functions such that $\overrightarrow{p}(L_1) = \bigcup_{1 \leq i_1 \leq n_1} \mathrm{Im}(F_{1,i_1})$ and $\overrightarrow{p}(L_2) = \bigcup_{1 \leq i_2 \leq n_2} \mathrm{Im}(F_{2,i_2})$. Let us show that

$$\overrightarrow{p}(L) = \bigcup_{1 \leq i_1 \leq n_1} \bigcup_{1 \leq i_2 \leq n_2} \mathrm{Im}(\mathrm{Wtt}_x(F_{1,i_1}) + \mathrm{On}_x(F_{1,i_1})F_{2,i_2})$$

First, if we consider a word $w \in L$, there exists $w_1 \in L_1$ and $w_2 \in L_2$ such that $w = io_{x,w_2}(w_1)$. Then, there exist $1 \leq k_1 \leq n_1$ and $1 \leq k_2 \leq n_2$ such that $\overrightarrow{p}(w_1) = F_{1,k_1}(c_{1,1}, \ldots, c_{1,m_1})$ and $\overrightarrow{p}(w_2) = F_{2,k_2}(c_{2,1}, \ldots, c_{2,m_2})$, where c_{i,j_i} belongs to $\mathbb{N}$, for $i \in \{1, 2\}$ and $1 \leq j_i \leq m_i$. Moreover, we observe that

$$\begin{aligned}\overrightarrow{p}(w) &= \mathrm{Wtt}_x(F_{1,k_1}(c_{1,1}, \ldots, c_{1,m_1})) \\ &\quad + \mathrm{On}_x(F_{1,k_1}(c_{1,1}, \ldots, c_{1,m_1}))F_{2,k_2}(c_{2,1}, \ldots, c_{2,m_2})\end{aligned}$$

We conclude that $\overrightarrow{p}(w)$ is in $\bigcup_{1 \leq i_1 \leq n_1} \bigcup_{1 \leq i_2 \leq n_2} \mathrm{Im}(\mathrm{Wtt}_x(F_{1,i_1}) + \mathrm{On}_x(F_{1,i_1})F_{2,i_2})$. Now consider a vector

$$\overrightarrow{v} = (\mathrm{Wtt}_x(F_{1,k_1}) + \mathrm{On}_x(F_{1,k_1})F_{2,k_2})(c_{1,1}, \ldots, c_{1,m_1}, c_{2,1}, \ldots, c_{2,m_2})$$

for some $1 \leq k_1 \leq n_1$, $1 \leq k_2 \leq n_2$. The element $w \in L$ such that $\overrightarrow{p}(w) = \overrightarrow{v}$ is trivially obtained by considering words $w_1 \in L_1$ and $w_2 \in L_2$ such that $\overrightarrow{p}(w_1) = F_{1,k_1}(c_{1,1}, \ldots, c_{1,n_1})$ and $\overrightarrow{p}(w_2) = F_{2,k_2}(c_{2,1}, \ldots, c_{2,n_2})$. □

Proposition 2 *If for every semilinear set E', there exists a language $L' \in \mathbf{L}$ such that $\overrightarrow{p}(L') = E'$, then for every factored-semilinear set E, there is a language L in $\mathbf{IO}(\mathbf{L})$ such that $\overrightarrow{p}(L) = E$.*

Proof Let us consider a synchronized set $F = \{F_1, \ldots, F_n\}$ of factored functions such that $E = \bigcup_{1 \leq i \leq n} \mathrm{Im}(F_i)$ is a factored-semilinear set. We proceed by induction on the construction of F:

- if for every $1 \leq i \leq n$, F_i is a linear function, then E is a semilinear set; hence, by hypothesis, there is a language $L \in \mathbf{L} = IO_0(\mathbf{L})$ such that $\overrightarrow{p}(L) = E$.
- otherwise, there exist $1 \leq k \leq p$ (where $\mathbb{N}^p$ is the codomain of every function in F), and two synchronized sets of factored vector functions $\{F_{1,1}, \ldots, F_{1,n_1}\}$ and $\{F_{2,1}, \ldots, F_{2,n_2}\}$ such that

$$E = \bigcup_{1 \leq i_1 \leq n_1} \bigcup_{1 \leq i_2 \leq n_2} \mathrm{Im}(\mathrm{Wtt}_k(F_{1,i_1}) + \mathrm{On}_k(F_{1,i_1})F_{2,i_2})$$

By induction hypothesis, there exist L_1 and L_2 in $\mathbf{IO}(\mathbf{L})$ such that

$$\overrightarrow{p}(L_1) = \bigcup_{1 \leq i_1 \leq n_1} \mathrm{Im}(F_{1,i_1}) \text{ and } \overrightarrow{p}(L_2) = \bigcup_{1 \leq i_2 \leq n_2} \mathrm{Im}(F_{2,i_2})$$

With the same reasoning as in the proof of the previous proposition, we see that, given $L = L_1[x := L_2]_{IO}$, we have $\overrightarrow{p}(L) = E$. □

From Propositions 1 and 2, we can deduce the following corollary, which establishes the strong relation between **IO(REG)** (or **IO(CFL)**, **IO(yTAL)** **IO(MCFL)**) and factored-semilinear languages.

Corollary 1 *Assume* **L** *is a family of languages which yields semilinear sets. Then,* **IO(L)** *yields factored-semilinear sets.*

This corollary leads to an alternative proof of the constant-growth property for languages in **IO**(**L**); it suffices to show that factored-semilinear sets are ∃-linear; we prove the stronger statement that these languages are ∀-linear.

Theorem 2 *Every factored-semilinear set is ∀-linear.*

Proof Let us consider an arbitrary factored-semilinear set E, and a synchronized set of factored vector functions $F = \{F_1, \dots, F_n\}$ such that $E = \bigcup_{1 \leq i \leq n} \mathrm{Im}(F_i)$. We proceed by induction on the construction of F:

- first, if every function in F is linear, then it is also ∀-linear. We conclude that E is then a ∀-semilinear set.
- otherwise, there exist $1 \leq k \leq p$, for $\mathbb{N}^p$ the codomain of the functions in F, and two synchronized sets of vector functions $\{F_{1,1}, \dots, F_{1,n_1}\}$ and $\{F_{2,1}, \dots, F_{2,n_2}\}$ such that

$$E = \bigcup_{1 \leq i_1 \leq n_1} \bigcup_{1 \leq i_2 \leq n_2} \mathrm{Im}(\mathrm{Wtt}_k(F_{1,i_1}) + \mathrm{On}_k(F_{1,i_1}) F_{2,i_2})$$

By induction hypothesis, every function in $\{F_{1,1}, \dots, F_{1,n_1}\}$ and $\{F_{2,1}, \dots, F_{2,n_2}\}$ is ∀-linear.

Let us consider a function $H_{k_1,k_2} = \mathrm{Wtt}_k(F_{1,k_1}) + \mathrm{On}_k(F_{1,k_1}) G_{2,k_2}$, for some $1 \leq k_1 \leq n_1$ and some $1 \leq k_2 \leq n_2$. We also suppose $F_{1,k_1} : \mathbb{N}^{p_1} \to \mathbb{N}^p$, $F_{2,k_2} : \mathbb{N}^{p_2} \to \mathbb{N}^p$ and $H_{k_1,k_2} : \mathbb{N}^{p_1+p_2} \to \mathbb{N}^p$. We show that H_{k_1,k_2} is q-linear, for every $1 \leq q \leq p_1 + p_2$. Let us write $H_{k_1,k_2,q}(x) = H_{k_1,k_2}(c_1, \dots, c_{q-1}, x, c_{q+1}, \dots, c_{p_1+p_2})$, where $c_r \in \mathbb{N}$ for every r such that $1 \leq r \leq q-1$ or $q+1 \leq r \leq p_1 + p_2$. The problem reduces to showing that $H_{k_1,k_2,q}$ is an affine function.

- if $1 \leq q \leq p_1$, then:

$$\begin{aligned} H_{k_1,k_2,q}(x) &= \mathrm{Wtt}_k(F_{1,k_1})(c_1, \dots, c_{q-1}, x, c_{q+1}, \dots, c_{p_1}) \\ &\quad + \mathrm{On}_k(F_{1,k_1})(c_1, \dots, c_{q-1}, x, c_{q+1}, \dots, c_{p_1}) \\ &\quad \times F_{2,k_2}(c_{p_1+1}, \dots, c_{p_1+p_2}) \\ &= (\overrightarrow{A} x + \overrightarrow{B}) + (A' x + B') \overrightarrow{C} \text{ as } F_{1,k_1} \text{ is ∀-linear} \\ &= (\overrightarrow{A} + A' \overrightarrow{C}) x + (\overrightarrow{B} + B' \overrightarrow{C}) \end{aligned}$$

  and $H_{k_1,k_2,q}$ is affine.
- if $p_1 + 1 \leq q \leq p_1 + p_2$, then:

$$\begin{aligned} H_{k_1,k_2,q}(x) &= \mathrm{Wtt}_k(F_{1,k_1})(c_1, \dots, c_{p_1}) \\ &\quad + \mathrm{On}_k(F_{1,k_1})(c_1, \dots, c_{p_1}) \\ &\quad \times F_{2,k_2}(c_{p_1+1}, \dots, c_{q-1}, x, c_{q+1}, \dots, c_{p_1+p_2}) \\ &= \overrightarrow{A} + A'(\overrightarrow{B} x + \overrightarrow{C}) \text{ as } F_{2,k_2} \text{ is ∀-linear} \\ &= A' \overrightarrow{B} x + (\overrightarrow{A} + A' \overrightarrow{C}) \end{aligned}$$

  and again $H_{k_1,k_2,q}$ is affine.

Therefore, $H_{k_1,k_2,q}$ is affine for every $1 \leq q \leq p_1 + p_2$, hence H_{k_1,k_2} is $\forall$-linear. We conclude that E is a $\forall$-linear set. □

We therefore proved that, given **L** a full abstract family of languages which yields semilinear sets, **IO(L)** yields factored-semilinear sets. It is then easy to see that **IO(L)** does not yield $\forall$-semilinear sets. For instance, consider the set $\text{Im}(F)$ where $F(x_1, x_2, x_3) = x_1x_2\langle 1, 0, 0\rangle + x_2x_3\langle 0, 1, 0\rangle + x_1x_3\langle 0, 0, 1\rangle$. According to the definition, F is $\forall$-linear but not factored-linear. It is therefore an open question to define a class of languages which yields $\forall$-linear sets, or $\exists$-linear sets. We hope these two newly introduced classes of sets can be relevant in the study of other classes of languages.

In the next section, we show that **IO(MCFL)** is not a full abstract family of languages, by proving it is not closed under inverse homomorphism. The proof is done similarly to the proof that IO macro-grammars are not closed under inverse homomorphism in Fischer (1968a), but differs from it in two related aspects; first IO-substitution is an operation on languages, while IO-macro languages are defined with respect to a grammatical formalism; the argument we need to exhibit is therefore related to the properties of the languages in **IO**(**L**). This leads to the second point as Fischer exhibits an IO-macro language L and a homomorphism h such that, if $h^{-1}(L)$ is an IO-macro language, then $L \in$ **CFL** which is impossible; we show the more general property that, for $L' \in IO(L)$, $h^{-1}(L')$ should verify some linearity constraints on its Parikh image. This argument reveals the relation between the way IO-copying breaks semilinearity, and the closure under inverse homomorphism for a class of languages built with IO-copying operations.

4 Non-closure of IO-MCFLs Under Inverse Homomorphism

In (Bourreau et al. 2012), the closure of **IO**(**L**), under homomorphism, concatenation, union and intersection with regular sets was proved for **L** a full abstract family of languages. We prove that the closure under inverse homomorphism is not satisfied, when **L** is a semilinear full AFL such that **REG** $\subseteq$ **L**, leading, as a corollary, to the proof that **IO**(**L**) is not a full abstract family of languages. In order to simplify the proof, we will first give some structural properties of **IO**(**L**).

4.1 Standard Representations for **IO(L)**

In this section, we introduce a specific representation of a language in **IO**(**L**), and prove that to every language in **IO**(**L**) we can associate such a representation.

We first introduce a convention on the naming of the symbols on which the IO-substitutions are performed. Given a language $L = L_1[x := L_2]_{IO}$ in **IO**(**L**), one can rename x into y if y has no occurrence in the words of L_1, i.e. for all words $w_1 \in L_1$ and $w_2 \in L_2$, $io_{x,w_2}(w_1) = io_{y,w_2}(io_{x,y}(w_2))$. In particular, assume that there exists a language $L_1[x := L_2]_{IO} \in$ **IO**(**L**), and a representation of L_2 that uses the languages $L'_1, \ldots, L'_n \in$ **L**; we can suppose x has no occurrence in any of the words of $\bigcup_{1\leq i\leq n} L'_i$.[1] In the rest of the article, we will assume that, without loss of generality, given a language $L \in \mathbf{IO}(\mathbf{L})$ and a representation of L, for each letter x on which an IO-substitution is performed, this IO-substitution is the only one on x for this representation of L, and x has no occurrence in L.[2] This will allow us, in particular, to designate an IO-substitution by the letter it is performed on without any ambiguity.

Definition 16 (*Right-representation*) Given a language $L \in \mathbf{IO}(\mathbf{L})$, a representation $L_0[x_1 := L_1]_{IO} \ldots [x_n := L_n]_{IO}$ of $L$is called a *right-representation* if for every $0 \leq i \leq n$, L_i belongs to $\mathbf{L}$. Moreover, we say that such a representation is of length n.

Lemma 2 *For every language $L \in \mathbf{IO}(\mathbf{L})$ there exists a right-representation.*

Moreover, if $\mathbf{L}$ is closed under homomorphism, for $L \in IO_n(\mathbf{L})$, there exists a right-representation of L of length n.

Proof Given a representation of L, we prove by induction on this representation that it can be transformed into a right-representation of L. First, if L belongs to $\mathbf{L} = IO_0(\mathbf{L})$, the result is trivial. Moreover, the representation of L is of length $n = 0$.

Otherwise, suppose $L = L_1[x := L_2]_{IO}$ and, by induction hypothesis, assume there exist a right-representation $L_{1,0}[x_{1,1} := L_{1,1}]_{IO} \ldots [x_{1,n} := L_{1,n_1}]_{IO}$ for L_1, and a right-representation $L_{2,0}[x_{2,1} := L_{2,1}]_{IO} \ldots [x_{2,m} := L_{2,n_2}]_{IO}$ for L_2, where $n_1, n_2 \in \mathbb{N}$.

For every $0 \leq p \leq n_2$, consider the language

$$L_p = L_1[x := L_{2,0}[x_{2,1} := L_{2,1}]_{IO} \ldots [x_{2,p} := L_{2,p}]_{IO}]_{IO}$$

We prove by induction on p that there exist symbols $y_1, \ldots, y_p$ such that

$$L_p = L_1[x := y_1]_{IO}[y_1 := L_{2,1}]_{IO} \ldots [x_p := y_p]_{IO}[y_p := L_p]_{IO}$$

For $p = 0$, the result is trivial. Consider a symbol y_p which has no occurrence in the words of $L_1 \cup \bigcup_{0\leq i\leq p-1} L_{2,i}$. Then $L_p = L_{p-1}[x_{2,p} := y_p]_{IO}[y_p := L_{2,p}]_{IO}$. Because of the induction hypothesis there exists a right-representation for L_{p-1}, hence a right-representation for L_p. Moreover, if $\mathbf{L}$ is closed under homomorphism, because $io_{x_{2,p},y_p}$ is a homomorphism, and because of the induction hypothesis, it is trivial to find a right-representation of L_p of length $n_1 + p$, and $L \in IO_{n_1+p}(\mathbf{L})$. □

Definition 17 (*Standard representation*) Given two languages $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$, we call the IO-substitution $L_1[x := L_2]_{IO}$:

- an *irrelevant IO-substitution* if for every word $w \in L_1$, $|w|_x = 0$.
- a *deleting IO-substitution* if $L_2 = \{\epsilon\}$.

[1] For readers familiar with the λ-calculus, the precise conditions under which a letter on which an IO-substitution is performed can be renamed are similar to the α-equivalence on λ-terms: variables can be renamed under the constraint that no other variable is "captured" by this renaming. We do not detail such constraints in the present work. The analogy with λ-calculus is made explicit in (Bourreau et al. 2012).

[2] Such a strict convention is to be compared with Barendregt's convention on variables in the λ-calculus.

A right-representation of a language L with no irrelevant or deleting IO-substitution is called a *standard representation* of L.

Lemma 3 *Let us consider a class* $\mathbf{L}$ *of languages, closed under homomorphism, and a language* $L \in \mathbf{IO}(\mathbf{L})$*. There exists a standard representation of* L*.*

Proof First, if $|w|_x = 0$ for every word $w \in L_1$, then $L_1[x := L_2]_{IO} = L_1$, and such a substitution can be trivially removed from any (right-)representation of L.

Now consider a language $L \in \mathbf{IO}(\mathbf{L})$ and $n \in \mathbb{N}$ such that $L \in IO_n(\mathbf{L}) \setminus IO_{n-1}(\mathbf{L})$ (if such a n does not exist, $L \in \mathbf{L}$ and we trivially have a right-representation of L with no deleting IO-substitution). Then there is a right-representation of the form $L_0[x_1 := L_1]_{IO} \dots [x_n := L_n]_{IO}$ for L according to Lemma 2. By induction hypothesis, $L_0[x_1 := L_1]_{IO} \dots [x_{n-1} := L_{n-1}]_{IO}$ has no deleting substitution; moreover $L_n \neq \{\epsilon\}$, because, according to Bourreau et al. (2012), for every $n \in \mathbb{N}$, $IO_n(\mathbf{L})$, is closed under homomorphism and $io_{x_n,\epsilon}$ being a homomorphism, we would have $L \in IO_{n-1}(\mathbf{L})$. □

4.2 Fully Effective Representations

In this section, we characterize a new kind of representation for languages in $\mathbf{IO}(\mathbf{L})$. This new form will allow us to exhibit only substitutions that are effective, i.e. $L_1[x := L_2]_{IO}$ such that $|w|_x > 0$ for every $w \in L_1$. In order to do so, we start by giving a fundamental lemma, which is a direct consequence of the Myhill-Nerode theorem:

Definition 18 Given an alphabet Σ, a right congruence $\cong$ on Σ^* is an equivalence relation such that for every $w_1, w_2, u \in \Sigma^*$, $w_1 \cong w_2$ implies $w_1 u \cong w_2 u$.

Such a congruence is said

- to be of finite index if $\Sigma/\cong$ is finite.
- to saturate a language $L \subseteq \Sigma^*$ if for every $w_1, w_2 \in \Sigma^*$, $w_1 \cong w_2$ implies $w_1 \in L$ iff $w_2 \in L$ (i.e. L is made of a union of congruence classes in $\Sigma^*/\cong$)

Theorem 3 (Myhill-Nerode) *A language* $L \subseteq \Sigma^*$ *is regular iff there exists a right congruence* $\cong$ *of finite index over* Σ^* *which saturates* L*.*

Corollary 2 *(***Separation Lemma**[3]*) Consider an alphabet* Σ*, a right congruence* $\cong$ *of finite index on* Σ^**, a class* $C \in \Sigma^*/\cong$ *and a language* $L \subseteq \Sigma^*$*, such that* L *belongs to a family* $\mathbf{L}$ *of languages closed under intersection with regular sets. Then* $L \cap C$ *belongs to* $\mathbf{L}$*; moreover,* $L = \bigcup_{C \in \Sigma^*/\cong} L \cap C$*.*

We are now in position of defining fully effective IO-substitutions, and fully effective representations of languages.

Definition 19 (*Full effectiveness*) An IO-substitution $L_1[x := L_2]_{IO}$ is said to be *fully effective* if for every word $w \in L_1$, we have $|w|_x > 0$.

[3] The corresponding version of this lemma is given as the factorization lemma in Fischer (1968a). We use a different name as we already used the terminology of factorization in Sect. 3.2.

Given a language $L \in \mathbf{IO}(\mathbf{L})$, a representation of L is in *fully effective form* when it is of the shape:

$$\bigcup_{i \in I} L_{i0}[x_{i_1} := L_{i1}]_{IO} \ldots [x_{i_{n_i}} := L_{in_i}]_{IO}$$

where

- I is a finite set,
- and for every $i \in I$, the representation $L_{i0}[x_{i_1} := L_{i1}]_{IO} \ldots [x_{i_n} := L_{in_i}]_{IO}$ is in standard form, and every IO-substitution in it is fully effective.

Given a language $L \in \mathbf{IO}(\mathbf{L})$, a fully effective representation of L is built by considering a decomposition $L = L_1 + \cdots + L_n$ of L, such that $L_1, \ldots, L_n \in \mathbf{IO}(\mathbf{L})$, and for every $1 \leq i \leq n$, there is a standard representation of L_i where each IO-substitution is fully effective.

As for standard representations, we show that every language $L \in \mathbf{IO}(\mathbf{L})$ has a fully effective representation. In order to do so, we first give the following property on IO-substitutions.

Lemma 4 *Given $L_{11}, L_{12}, L_{21}, L_{22} \in \mathbf{IO}(\mathbf{L})$:*

$$(L_{11} + L_{12})[x := L_{21} + L_{22}]_{IO} = \bigcup_{i,j \in \{1,2\}} L_{1i}[x := L_{2j}]_{IO}$$

Proof By definition,we have:

$$(L_{11} + L_{12})[x := L_{21} + L_{22}]_{IO} = \bigcup_{w \in L_{21}+L_{22}} io_{x,w}(L_{11} + L_{12})$$

Then, we easily establish:

$$\begin{aligned}\bigcup_{w \in L_{21}+L_{22}} io_{x,w}(L_{11} + L_{12}) &= \bigcup_{w \in L_{21}+L_{22}} io_{x,w}(L_{11}) \cup \bigcup_{w \in L_{21}+L_{22}} io_{x,w}(L_{12}) \\ &= \bigcup_{w \in L_{21}} io_{x,w}(L_{11}) \cup \bigcup_{w \in L_{22}} io_{x,w}(L_{11}) \\ &\quad \cup \bigcup_{w \in L_{21}} io_{x,w}(L_{12}) \cup \bigcup_{w \in L_{22}} io_{x,w}(L_{12}) \\ &= L_{11}[x := L_{21}]_{IO} + L_{11}[x := L_{22}]_{IO} \\ &\quad + L_{12}[x := L_{21}]_{IO} + L_{12}[x := L_{22}]_{IO}\end{aligned}$$

□

Lemma 5 *Given a family of languages* $\mathbf{L}$ *closed under intersection with regular sets, for every language L in **IO(L)**, there exists a representation in fully effective form.*

Proof Let us consider such a language $L \in \mathbf{IO}(\mathbf{L})$ and a representation in standard form for it: $L_0[x_1 := L_1]_{IO} \dots [x_n := L_n]_{IO}$. Given the alphabet Σ of L, we build the right congruence $\cong$ on $(\Sigma \cup \{x_1, \dots, x_n\})^*$ such that for every word $w_1, w_2 \in (\Sigma \cup \{x_1, \dots, x_n\})^*$

$$w_1 \cong w_2 \iff \text{for every } 1 \leq i \leq n, |w_1|_{x_i} = 0 \text{ iff } |w_2|_{x_i} = 0$$

This congruence has a finite index I of cardinality 2^n. For every $1 \leq i \leq n$, according to the separation lemma, we have $L_i = \bigcup_{k \in I} L_{ik}$, where $L_{ik} = L_i \cap C_k$, C_k being the k^{th} class in $(\Sigma \cup \{x_1, \dots, x_n\})^*/\cong$. Then we can write:

$$\begin{aligned} L &= \left(\bigcup_{k_0 \in I} L_{0k_0}\right) \left[x_1 := \bigcup_{k_1 \in I} L_{1k_1}\right]_{IO} \dots \left[x_n := \bigcup_{k_n \in I} L_{nk_n}\right]_{IO} && \text{(Corollary 2)} \\ &= \bigcup_{k_0, \dots, k_n \in I} L_{0k_0}[x_1 := L_{1k_1}]_{IO} \dots \left[x_n := L_{nk_n}\right]_{IO} && \text{(Lemma 4)} \end{aligned}$$

According to the separation lemma, $L_{ik_i} \in \mathbf{L}$, for every $1 \leq i \leq n$ and $k_i \in I$; therefore the language $L_{k_0 \dots k_n} = L_{0k_0}[x_1 := L_{1k_1}]_{IO} \dots [x_n := L_{nk_n}]_{IO}$, belongs to $\mathbf{IO(L)}$. Moreover, for every $0 \leq j \leq n$ and every $k_0, \dots, k_j \in I$, let us consider the language $L_{k_0 \dots k_j} = L_{0k_0}[x_1 := L_{1k_1}]_{IO} \dots [x_j := L_{jk_j}]_{IO}$; the construction ensures that for all the words w_1 and w_2 in $L_{k_0 \dots k_j}$, w_1 and w_2 are congruent (which is provable with a direct induction on j); therefore, the IO-substitution $L_{k_0 \dots k_j}[x_{j+1} := L_{(j+1)k_{j+1}}]_{IO}$ is fully effective iff there exists w in $L_{k_0 k_1 \dots k_j}$ such that $|w|_{x_{j+1}} > 0$. According to the construction of a standard representation of a language from a right representation of a language in Lemma 3, we can remove the irrelevant IO-substitutions, which are exactly the substitutions which are not fully effective. This leads to the existence of a representation of $L_{k_0 \dots k_n}$ in fully effective and standard form, for every $k_0, \dots, k_n \in I$; hence, the representation $\bigcup_{k_0, \dots, k_n \in I} L_{0k_0}[x_1 := L_{1k_1}]_{IO} \dots [x_n := L_{nk_n}]_{IO}$ of the language L is fully effective. □

4.3 a-Linearity

Finally, we study with more precision the copying effects of the IO-substitution. We already saw how this operation allows one to build non-semilinear languages which verify the constant-growth property. In this section, we study the effect of an IO-substitution on symbols.

We first define notions of linearity and universal-linearity with respect to the symbols of a given language. These definitions are natural extensions of the definitions given in Sect. 3.

Definition 20 (*a-linearity*) Let us consider a language $L \subseteq \Sigma^*$ and a letter $a \in \Sigma$. Assume that the Parikh image $\overrightarrow{p}(L)$ of L is of the form $\bigcup_{1 \leq i \leq k} \mathrm{Im}(F_i)$ for some vector functions $F_1 : \mathbb{N}^{n_1} \to \mathbb{N}^n, \dots, F_k : \mathbb{N}^{n_k} \to \mathbb{N}^n$, we say that L is:

- *a-constant* if for every $1 \leq i \leq k$,

$$\mathrm{On}_a(F_i)(x_1, \ldots, x_{n_i}) = c_i \in \mathbb{N}$$

- *a-linear* if for every $1 \leq i \leq k$,

$$\mathrm{On}_a(F_i)(x_1, \ldots, x_{n_i}) = c_i + \sum_{1 \leq j \leq n_i} d_j x_j$$

where $c_i \in \mathbb{N}$ and $d_j \in \mathbb{N}$ for every $1 \leq j \leq n_i$;

- *a-functional* if for every $1 \leq i \leq k$,

$$\mathrm{On}_a(F_i)(x_1, \ldots, x_{n_i}) = c_i + \sum_{1 \leq j \leq m_i} f_j(x_1, \ldots, x_{n_i})$$

where $f_j \in \mathbb{N}^{n_i} \rightarrow \mathbb{N}$ for every $1 \leq j \leq m_i$, and $c_i \in \mathbb{N}$.

It is immediate to see that a finite language $L \subseteq \Sigma^*$ is a-constant for every $a \in \Sigma$; similarly, L is a semilinear language if L is a-linear for every $a \in \Sigma$; and $\forall$-linear if a-functional for every $a \in \Sigma$, and if the functions f_j in the definition above are k-linear for every $1 \leq k \leq n_i$.

Based on the naming convention we adopted for the representations of languages in **IO(L)**, the following notion of introducers provides us a mean to precisely study the copying process which occurs along a sequence of IO-substitutions. Based on these definitions, we aim at studying conditions under which an IO-substitution leads to building a-constant, a-linear or a-functional languages.

Definition 21 (*a-introducers*) Let us consider $L \in \mathbf{IO(L)}$, the alphabet Σ of L, and a representation (denoted by r) $L_0[x_1 := L_1]_{IO} \ldots [x_n := L_n]_{IO}$ of L in standard form. We define the binary relation $In_r \subseteq (\Sigma \cup \{x_1, \ldots, x_n\}) \times (\Sigma \cup \{x_1, \ldots, x_n\})$ by $b\ In_r\ a$ iff

$$b = x_i \text{ for some } 1 \leq i \leq n \text{ and there exists } w \in L_i \text{ such that } |w|_a > 0.$$

We define In_r^* as the reflexive and transitive closure of In_r, and *the set of introducers of $a \in \Sigma \cup \{x_1, \ldots, x_n\}$ in r* will be written $In_r^a = \{b \in \Sigma \cup \{x_1, \ldots, x_n\} \mid b\ In_r^*\ a\}$.

The idea behind this definition is that a symbol b is an introducer of another symbol a in a representation r of some language, iff it is involved in creating occurrences of a at some IO-substitution of the representation: it is either the symbol a itself, or it introduces a symbol in $\{x_1, \ldots, x_n\}$ which will itself be substituted by a language that contains at least one word in which a has an occurrence. Note that the set In_r^a must be finite, as a representation of a language is finite.

Example 3 Let us consider the language represented by the following representation:

$$r : x_1 x_2 x_3 [x_1 := a]_{IO} [x_2 := b]_{IO} [x_3 := z^*]_{IO} [z := ab^* a]_{IO}$$

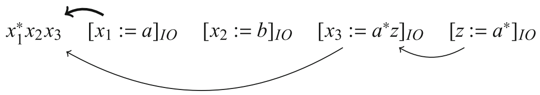
Then we have $x\ In_r\ a$ iff $x \in \{z, x_1\}$; moreover, $In_r^a = \{a, x_1, z, x_3\}$.

Definition 22 (*Chain of introducers*) Given a standard representation r of a language $L \in \mathbf{IO}(\mathbf{L})$, we inductively define a *chain of introducers of* a *in* r as a finite sequence C on In_r^a, such that

1. $C = (a)$, or
2. $C = (x_0, \ldots, x_n, x_{n+1})$, where $(x_0, \ldots, x_n)$ is a chain of introducers of a, and $x_{n+1}\ In_r\ x_n$

Moreover, for $r : L_0[x_1 := L_1]_{IO} \ldots [x_n := L_n]_{IO}$, a chain $C = (x_{i_1}, \ldots, x_{i_p})$ (where $0 \leq i_p < \ldots < i_1 \leq n$) is said *maximal* if there exists $w \in L_0$ such that $|w|_{x_{i_p}} > 0$.

Example 4 Let us consider the language associated to the representation in Example 3. The chains of a-introducers in it are (a), (a, z), (a, z, x_3); (a, x_3) and (a, x_1). The maximal chains of a-introducers are (a, z, x_3), (a, x_3) (represented by the arrows at the bottom in the following figure) and (a, x_1) (represented by the thick arrow at the top).



Intuitively, considering semilinear languages on which we perform some IO-substitutions, the only way, for the resulting language to be a-constant or a-semilinear for a symbol a, is related to the way languages which contain words with occurrences of a are copied along the IO-substitutions.

Lemma 6 *Consider a class of semilinear languages* $\mathbf{L}$*, languages* $L, L_1, L_2 \in \mathbf{IO}(\mathbf{L})$*, and a non-irrelevant IO-substitution* $L_1[x := L_2]_{IO}$ *such that* $L_1[x := L_2]_{IO} = L \subseteq \Sigma^*$*. Given a letter* $a \in \Sigma$*, suppose that* L_1 *and* L_2 *are* a*-linear, and that* L_1 *is* x*-linear:*

1. *if* L_1 *is* x*-constant or* L_2 *is* a*-constant, then* L *is* a*-linear.*
2. *if there exists* $w \in L_2$ *such that* $|w|_a > 0$*, then* L *is* a*-constant iff* L_1 *is* x*-constant and* a*-constant, and* L_2 *is* a*-constant.*

Proof We consider the Parikh images of L_1 and L_2 as, respectively $\bigcup_{1 \leq i \leq n_1} \mathrm{Im}(F_{1,i_1})$ and $\bigcup_{1 \leq j \leq n_2} \mathrm{Im}(F_{2,i_2})$, where F_{1,i_1} and F_{2,i_2} are vector functions, for $1 \leq i_1 \leq n_1$ and $1 \leq i_2 \leq n_2$. From the proof of Proposition 1, we know that

$$\overrightarrow{p}(L) = \bigcup_{1 \leq i_1 \leq n_1} \bigcup_{1 \leq i_2 \leq n_2} \mathrm{Im}(\mathrm{Wtt}_x(F_{1,i_1}) + \mathrm{On}_x(F_{1,i_1}) F_{2,i_2})$$

1. if L_1 is x-constant: $\mathrm{On}_a(\mathrm{Wtt}_x(F_{1,i_1}){+}\mathrm{On}_x(F_{1,i_1})F_{2,i_2})(x_1,\ldots,x_{m_{i_1}},y_1,\ldots,y_{m_{i_2}})$

$$= c_{1,i_1} + \sum_{1\leq l_1\leq m_{i_1}} d_{1,l_1}x_{l_1} + k_{1,i_1}(c_{2,i_2} + \sum_{1\leq l_2\leq m_{i_2}} d_{2,l_2}y_{l_2})$$

because L_1 and $L_2$are a-linear andL_1 is x-constant

$$= (c_{1,i_1} + k_{1,i_1}c_{2,i_2}) + \sum_{1\leq l_1\leq m_{i_1}} d_{1,l_1}x_{l_1} + \sum_{1\leq l_2\leq m_{i_2}} k_{1,i_1}d_{2,l_2}y_{l_2}$$

for every $1 \leq i_1 \leq n_1$ and every $1 \leq i_2 \leq n_2$; it then follows that L is a-linear. Similarly, if L_2 is a-constant, we obtain a similar equation, and the same conclusion.

2. L is a-constant iff $\mathrm{Im}(\mathrm{On}_a(\mathrm{Wtt}_x(F_{1,i_1}) + \mathrm{On}_x(F_{1,i_1})F_{2,i_2}) = \{c_{i_1,i_2}\}$, where $c_{i_1,i_2} \in \mathbb{N}$, for every $1 \leq i_1 \leq n_1$ and every $1 \leq i_2 \leq n_2$, which is verified iff the following equation is true, under the hypothesis that L_1 and L_2 are a-linear and L_1 is x-linear:

$$c_{i_1,i_2} = c_{1,i_1} + \sum_{1\leq l_1\leq m_i} d_{1,l_1}x_{l_1} + (c'_{1,i_1} + \sum_{1\leq l_1\leq m_i} d'_{1,l_1}x_{l_1})(c_{2,i_2} + \sum_{1\leq l_2\leq m_{i_2}} d_{2,l_2}y_{l_2}) \tag{1}$$

Under the assumptions that the substitution is not irrelevant, there exists $1 \leq i'_1 \leq n_1$ such that $\mathrm{On}_x(F_{i'_1}) \neq 0$; also, because there exists a word $w \in L_2$ s.t. $|w|_a > 0$, there exists $1 \leq i'_2 \leq n_2$ such that $\mathrm{On}_x(G_{i'_2}) \neq 0$. Therefore, for every $1 \leq i_1 \leq n_1$ and every $1 \leq i_2 \leq n_2$, $(c'_{1,i_1} + \sum_{1\leq l_1\leq m_{i_1}} d'_{1,l_1}x_{l_1})(c_{2,i_2} + \sum_{1\leq l_2\leq m_{i_2}} d_{2,l_2}y_{l_2})$ is constant iff $d_{1,l_1} = d'_{1,l_1} = d_{2,l_2} = 0$ for every $1 \leq l_1 \leq m_1$ and every $1 \leq l_2 \leq m_2$, so that Eq. (1) reduces to:

$$c_{i_1,i_2} = c_{1,i_1} + c'_{1,i_1}c_{2,i_2}$$

These conditions are equivalent to L_1 being both x-constant and a-constant, and L_2 being a-constant.

□

Example 5 Consider the languages

$$L_1 = \{b^n a^* x^n \mid n \in \mathbb{N}\}[x := ab^*]_{IO} = \{b^n a^*(ab^*)^n \mid n \in \mathbb{N}\}$$
$$L'_1 = b^* x a^*[x := a^*]_{IO} = b^* a^*$$

L_1 and L'_1 respect the conditions of Lemma 6.1, and are both a-linear. Remark that, if these conditions are not respected, it is possible to build a language which is not a-linear, just like $\{b^n x^n \mid n \in \mathbb{N}\}[x := a^*]_{IO} = \{b^n a^{nm} | n, m \in \mathbb{N}\}$, whose elements respect a dependence between the number of a's and the number of b's in them.

The second part of the Lemma can be illustrated with the language

$$x^4 b^* a[x := b^* a^3]_{IO} = \left\{b^{4n} a^{12} b^m a \mid n, m \in \mathbb{N}\right\}$$

Intuitively, because all elements which introduce an a in this substitution are constant in the two languages involved, the resulting language is a-constant.

One should remark that Lemma 6 cannot be reformulated into an equivalence: indeed, given two semilinear languages L_1 and L_2, Bourreau et al. (2012) gave some conditions under which a language L such that $L_1[x := L_2]_{IO} = L$ is itself semilinear.

The following lemma is obtained from Lemma 6 in the particular case of a standard representation for a language in **IO(L)**.

Lemma 7 *Consider a family* $\mathbf{L}$ *of semilinear languages, a language* $L \subseteq \Sigma^*$ *in* $\mathbf{IO}(\mathbf{L})$*, a standard representation* $r : L_0[x_1 := L_2]_{IO} \ldots [x_n := L_n]_{IO}$ *of* L *and a letter* $a \in \Sigma$*. Suppose that, for every chain* $C \in Ch_r^a$*, there exist at most one* $x_C \in C$ *and at most one* $0 \leq i_C \leq n$ *such that :*

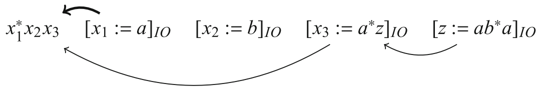
- L_{i_C} *is not* x_C*-constant, but* x*-constant for every* $x \in C - \{x_C\}$*, and*
- *for every* $x \in C$ *and every* $0 \leq j \leq n$ *such that* $j \neq i_C$*,* L_j *is* x*-constant*

Then L *is* a*-linear.*

Proof Let us denote by r the representation $L_0[x_1 := L_1]_{IO} \ldots [x_n := L_n]_{IO}$ of the language L. We inductively define $L'_i \in \mathbf{IO}(\mathbf{L})$, for every $1 \leq i \leq n$ as: $L'_0 = L_0$, and $L'_i = L'_{i-1}[x_i := L_i]_{IO}$. By induction on i, we show that L'_i is x-linear for every $x \in In_r^a$:

- if $i = 0$, then L'_i is a semilinear language by hypothesis, which implies that L'_i is in particular x-linear for every $x \in In_r^a$.
- suppose the result is true for every $0 \leq k \leq i$. We consider $L'_i[x_{i+1} := L_{i+1}]_{IO}$; by induction hypothesis, L'_i is x-linear for every $x \in In_r^a$, and by hypothesis, L_{i+1} is semilinear, hence b-linear for every $b \in \Sigma \cup \{x_1, \ldots, x_n\}$. Suppose first that $x_{i+1} \notin In_r^a$; then, for every b such that there exists $w \in L_{i+1}$ for which $|w|_b > 0$, we have $b \notin In_r^a$. Therefore, for every $x \in In_r^a$, L'_{i+1} is x-linear iff L'_i is x-linear, which is verified thanks to the induction hypothesis.
  Suppose now that $x_{i+1} \in In_r^a$. By hypothesis, L'_i is x_{i+1}-linear. If L'_i is x_{i+1}-constant, according to Lemma 6.1, L'_{i+1} is x-linear for every $x \in In_r^a$. Otherwise, L'_i is not x_{i+1}-constant; according to Lemma 6.2, there exist a chain C of x_{i+1}-introducers, $x_C \in In_r^{x_{i+1}}$ and $1 \leq i_C \leq i$ such that L_{i_C} is not x_C-constant. Because x_C is also an a-introducer, for every $x \in In_r^a$, L_{i+1} must be x-constant by hypothesis. Finally, by application of Lemma 6.1, we obtain that L'_{i+1} is x-linear for every $x \in In_r^a$. □

Example 6 Let us consider the language given in Example 3:



Every maximal chain of a-introducers verifies the conditions given in Lemma 7: for (a, x_1), only the language $x_1^* x_2 x_3$ is not constant on an a-introducer (i.e. x_1); similarly for (a, x_3, z) and (a, x_3), only the language a^*z is not constant on an a-introducer (i.e. a).

The conditions expressed in Lemma 7 are not complete as illustrated with the representation $a^*x[x := a^*]_{IO}$ of the language a^*. The maximal chains of a-introducers in it are $C' = (a)$ and $C = (a, x)$; and both a^*x and a^* are not a-constant.

4.4 **IO(L)** is not a Full AFL

Finally, based on the previous results, we prove that, given **L** a semilinear full AFL such that **REG** $\subseteq$ **L**, the family **IO**(**L**) is not closed under inverse homomorphism.

We first prove the following lemma, which states that, whenever a chain of IO-substitutions generates a language which is not a-linear (i.e. whenever such a chain can copy words/languages more than once), the derived words must verify a specific pattern.

Lemma 8 *Consider a language* $L \in \mathbf{IO}(\mathbf{L})$ *and:*

- *a representation* r *of* L*, of the form* $L_0[x_1 := L_1]_{IO} \dots [x_n := L_n]_{IO}$*, fully effective and in standard form;*
- *a symbol* $a \in \Sigma$*, a chain* $C \in Ch_r^a$*, distinct symbols* $y_1, y_2 \in C$ *and* $0 \leq i_1, i_2 \leq n$*,* $i_1 \neq i_2$ *such that for every* $u_1 \in L_{i_1}$ *and* $u_2 \in L_{i_2}$*,* $|u_1|_{y_1} > 1$ *and* $|u_2|_{y_2} > 1$

Then, for every $w \in L$*, there exists* $w', w_1, w_2, w_3 \in \Sigma^*$ *such that* $w = w_1aw'aw_2aw'aw_3$*.*

Proof First note that $y_1 \neq y_2$ and $y_1, y_2 \in C$ implies either $y_1 \in In_r^{y_2}$ or $y_2 \in In_r^{y_1}$, by definition of a chain of introducers; without loss of generality, let us assume that $y_1 \in In_r^{y_2}$. We consider a word u in the language represented by

$$L_0[x_1 := L_1]_{IO} \dots [x_{i_1-1} := L_{i_1-1}]_{IO}$$

and a word $u' \in L_{i_1}$. By assumption, u' is of the form $u'_1y_1u'_2y_1u'_3$; because the IO-substitution is not irrelevant, $io_{x_{i_1-1},u'}(u)$ is of the form $u_1y_1u_2y_1u_3$.

Because there is no deleting IO-substitution along the representation r, the words w in the language represented by $L_0[x_1 := L_1]_{IO} \dots [x_{i_2-1} := L_{i_2-1}]_{IO}$ must be of the form $w_1yw_2yw_3$, where $y_1In_r^*y$ and $yIn_r^*y_2$. Then, because every word in L_{i_2} must be of the general form $w' = w'_1y_2w'_2y_2w'_3$, the word $io_{y_2,w'}(w)$ is of the form $w''_1y_2w'_2y_2w''_2y_2w'_2y_2w''_3$.

Again, because the substitutions in r are not deleting, we can conclude that the words in L are of the form $u'_1au'au'_2au'au'_3$. □

We are now in position of proving our main result. Informally, a sketch of the proof is as follows: if we let $L_{anp,b} = \{w \in \{a, b\}^* \mid |w|_a = nm,\ \text{where } n, m > 1\}$ in **IO**(**L**), and $L_{diff} = \{b^{p_0}ab^{p_1}a \dots ab^{p_{nm}} \mid n, m > 1 \text{ and for every } 0 \leq i, j \leq nm, i \neq j \Rightarrow p_i \neq p_j\}$, then we can exhibit a language L such that $L_{diff} \subseteq L \subseteq L_{anp,b}$, by removing representations of the form given in Lemma 8 in a representation of $L_{anp,b}$. This means that no IO-substitution of a symbol x can be performed on words that contain multiple occurrences of that symbol; therefore, the language L must be a-linear, which is impossible.

Theorem 4 (Non-closure under inverse homomorphism) *Given a full abstract family of semilinear languages* $\mathbf{L}$ *such that* $\mathbf{REG} \subseteq \mathbf{L}$*, the family* $\mathbf{IO}(\mathbf{L})$ *is not closed under inverse homomorphism.*

Proof Let us consider the language consisting of all strings of a^* whose length is a non-prime number:

$$L_{nprime} = \left\{a^{nm} \mid n, m > 1\right\} = \left\{a^{4+2n+2m+nm} \mid n, m \in \mathbb{N}\right\}$$

This language is not semilinear since its Parikh image is equal to $\mathrm{Im}(F)$ where $F(x_1, x_2) = \langle 4\rangle + x_1\langle 2\rangle + x_2\langle 2\rangle + x_1x_2\langle 1\rangle$. Therefore L_{nprime} does not belong to $\mathbf{L}$, but belongs to $\mathbf{IO}(\mathbf{REG})$: indeed $a^2a^*[a := a^2a^*]_{IO} = L_{nprime}$ and a^2a^* is a regular language. Therefore, if $\mathbf{REG} \subseteq \mathbf{L}$ then L_{nprime} is in $\mathbf{IO}(\mathbf{L})$.

Now, consider the homomorphism $\phi : \{a, b\} \rightarrow a^*$ such that $\phi(a) = a$ and $\phi(b) = \epsilon$. Then we obtain:

$$\phi^{-1}(L_{nprime}) = L_{anp,b} = \{w \in \{a, b\}^* \mid |w|_a = nm, \text{ where } n, m > 1\}$$

Let us assume $L_{anp,b}$ belongs to $\mathbf{IO}(\mathbf{L})$. Then, according to Lemma 5, there exists a fully effective representation $r_{anp,b} : \bigcup_{i\in I} L_{i0}[x_{i_1} := L_{i1}]_{IO} \ldots [x_{i_{n_i}} := L_{in_i}]_{IO}$ for $L_{anp,b}$, where for every $0 \leq j \leq n_i$, L_{ij} belongs to $\mathbf{L}$. For every $i \in I$, let us denote by r_i the representation $L_{i0}[x_{i_1} := L_{i1}]_{IO} \ldots [x_{i_{n_i}} := L_{in_i}]_{IO}$ and by L_i the resulting language.

Let us consider the language $L_{diff} \subsetneq L_{anp,b}$ defined by:

$$L_{diff} = \{b^{p_0}ab^{p_1}a \ldots ab^{p_{nm}} \mid n, m > 1 \text{ and for every } 0 \leq i, j \leq nm, i \neq j \Rightarrow p_i \neq p_j\}$$

We aim at building a language L such that $L_{diff} \subseteq L \subseteq L_{anp,b}$. In order to do so, for every $i \in I$, let us consider the right congruence $\cong_i$ defined as:

$$w_1 \cong_i w_2 \text{ iff for every } y \in In^a_{r_i}, |w_1|_y > 1 \iff |w_2|_y > 1$$

Such a congruence is of finite index. According to the separation lemma and Lemma 4, we can write:

$$L_i = \bigcup_{C_0,\ldots,C_{n_i} \in \Sigma^*/\cong_i} (L_{i0} \cap C_0)[x_1 := (L_{i1} \cap C_1)]_{IO} \ldots [x_{n_i} := (L_{in_i} \cap C_{n_i})]_{IO}$$

Moreover, for every class of equivalence $C_0, \ldots, C_{n_i} \in \Sigma^*/\cong_i$, the representation $(L_{i0} \cap C_0)[x_1 := (L_{i1} \cap C_1)]_{IO} \ldots [x_{n_i} := (L_{in_i} \cap C_{n_i})]_{IO}$ is in standard form, and is fully effective, because $L_{ij} \cap C_j$ is a sublanguage of L_{ij}, for every $1 \leq j \leq n_i$, and because $r_{anp,b}$ is in fully effective form.

Now let us consider $R_i : (L_{i0} \cap C_0)[x_1 := (L_{i1} \cap C_1)]_{IO} \dots [x_{n_i} := (L_{in_i} \cap C_{n_i})]_{IO}$ where $C_0, \dots, C_{n_i} \in \Sigma^*/\cong_i$, such that there exist a chain $ch \in Ch^a_{R_i}$, symbols $y_1, y_2 \in ch$ with $y_1 \neq y_2$, and integers $0 \leq i_1, i_2 \leq n_i$, for which:

- for every word $w \in C_{i_1}$, $|w|_{y_1} > 1$;
- for every word $w \in C_{i_2}$, $|w|_{y_2} > 1$;

Then, according to Lemma 8, any word in L'_i, the language associated to the representation R_i, does not belong to L_{diff}. We can therefore build the language L such that the following representation R:

$$\bigcup_{i \in I'} \bigcup_{C_0 \in \Sigma^*/\cong_i} \dots \bigcup_{C_{n_i} \in \Sigma^*/\cong_i} (L_{i0} \cap C_0)[x_1 := (L_{i1} \cap C_1)]_{IO} \dots [x_{n_i} := (L_{in_i} \cap C_{n_i})]_{IO}$$

is a representation of L, and results from removing the representations of languages whose intersection with L_{diff} is empty.

But, for every $i \in I'$ and every class $C_0, \dots C_{n_i}$ of $\Sigma^*/\cong_i$, the representation

$$(L_{i0} \cap C_0)[x_1 := (L_{i1} \cap C_1)]_{IO} \dots [x_{n_i} := (L_{in_i} \cap C_{n_i})]_{IO}$$

must verify the assumptions of Lemma 7; therefore, the language associated to such a representation must be a-linear, and L is a finite union of a-linear languages, hence an a-linear language itself.

But, $\phi(L_{diff}) = L_{nprime} \subseteq \phi(L) \subseteq \phi(L_{anp,b}) = L_{nprime}$. Therefore, L_{nprime} should be a-linear, which is wrong, and we obtain a contradiction. □

Corollary 3 *If* **L** *is a semilinear full AFL such that* **REG** $\subseteq$ **L**, *then* **IO(L)** *is not a full AFL. In particular,* **IO(CFL)**, **IO(yTAL)** *and* **IO(MCFL)** *are not full AFLs.*

5 Conclusion

In the present paper, we proposed a study on the effect of IO-substitution on the Parikh image of languages in **L**, a full abstract family of semilinear languages. We first gave a full and complete characterization of these images in terms of factored semilinear Parikh images, and based on this result, we gave a new proof that languages in **IO(L)** verify the constant-growth property. This first step was also the opportunity to define universally- and existentially-semilinear Parikh images, and to prove that languages whose Parikh images belong to these classes also verify the constant-growth property. We gave some brief arguments in favour of the interest of the newly introduced classes of universally- and existentially-semilinear Parikh images in capturing natural language syntax, which would require further investigations. In the second part of the paper, we proved that **IO(L)** is not closed under inverse homomorphism, when **REG** $\subseteq$ **L** and **L** is a semilinear full AFL. The proof relies on the results obtained in the first section, and in particular in showing that the copying power brought by the IO-substitution operation forces the words to verify a certain pattern. As a consequence, we can conclude that **IO(MCFL)** is not a full abstract family of languages, which was an open question in Bourreau et al. (2012). Moreover, our result generalizes the one of Fischer on IO-macro languages, and, on a technical point of view, the argument of the proof reveals some connection between the way IO-copying breaks semilinearity and the non-closure under inverse homomorphism of a class of languages built with IO-copying operations.

This work gives space for further problems. First, the sketch of the proof of the non-closure property under inverse homomorphism can probably be reused to prove the same result on other formalisms in which copying material is allowed. In particular, we can conjecture that parallel multiple context-free languages are not closed under such an operation, which contradicts the conjecture in the seminal paper (Seki et al. 1991). The same question can be addressed on the language in the IO hierarchy (Damm 1982; Salvati and Kobele 2013), i.e. formalisms in which higher-order operations on strings can be copied. Some questions can also be addressed related to the first part of the present article. For instance, how can we create a class of languages which yields universally-semilinear sets? Addressing the same question on the existentially-semilinear sets seems less trivial as the functions used to build such sets are free but on one of their arguments.

Finally, some formal questions on the IO-substitution operation can be addressed. One of them is to characterize the languages obtained with infinite application of such an operation; in particular, IO-macro languages might be generated by recursive application of some IO-substitutions. For example, the language $\{a^{n^2} \mid n \in \mathbb{N}\}$ can be expressed as: $\epsilon + a([a := aa]_{IO})^* = \{\epsilon\} \cup \bigcup_{n\in\mathbb{N}} a\, \underbrace{[a := aa]_{IO}[a := aa]_{IO} \dots [a := aa]_{IO}}_{n}$. With such patterns, one might be able to express languages such as macro-languages, index languages or parallel multiple context-free languages. We will therefore investigate whether the IO-substitution can be used to revisit and classify classes of languages in which some copying mechanism is used.

Acknowledgments This work was funded by the Deutsche Forchungsgemeinschaft, under the project SFB 991 “Die Struktur von Repräsentationen in Sprache, Kognition und Wissenschaft”. I am thankful to Sylvain Salvati for the motivating discussions on this topic; to Laura Kallmeyer for her insights on the notions of universally-linear sets; and to Christian Wurm who helped me to improve the formal definitions with his feedbacks. The responsibility for any mistakes contained herein rests solely on me.

References

Bourreau, P. (2013). Traitement d’ellipses: Deux approches par les grammaires catégorielles abstraites. In *Actes de Traitement Automatique du Langage Naturel—TALN 2013*.

Bourreau, P., Kallmeyer, L., & Salvati, S. (2012). On IO-copying and mildly-context sensitive formalisms. In *Proceedings of Formal Grammar 2012*.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, *2*, 113–124.

Culy, C. (1987). The complexity of the vocabulary of bambara. In W. Savitch, E. Bach, W. Marsh, & G. Safran-Naveh (Eds.), *The formal complexity of natural language, Studies in linguistics and philosophy* (Vol. 33, pp. 349–357). Netherlands: Springer.

Damm, W. (1982). The IO- and OI-hierarchies. *Theoretical Computer Science*, *20*, 95–207.

de Groote, P. (2001). Towards abstract categorial grammars. In *Proceedings of the conference on Association for computational linguistics, 39th Annual meeting and 10th conference of the European chapter*, pp. 148–155.

Fischer, M. J. (1968a). *Grammars with macro-like productions*. Ph.D. thesis, Harvard University.

Fischer, M. J. (1968b). Grammars with macro-like productions. In *IEEE conference record of 9th annual symposium on switching and automata theory*, pp. 131–142.

Huybregts, R. (1984). The weak inadequacy of context-free phrase structure grammars. In *Van Preferie naar Kern*, pp. 81–90.

Joshi, A. K. (1985). Tree-adjoining grammars: How much context-sensitivity is required to provide reasonable strucutral descriptions? In *Natural language parsing: psychological, computational and theoretical perspectives*, pp. 206–250.

Kallmeyer, L. (2010). On mildly context-sensitive non-linear rewriting. *Research on Language and Computation*, *8*(2), 341–363.

Kobele, G. M. (2006). *Generating Copies: An investigation into structural identity in language and grammar*. Ph.D. thesis, UCLA.

Kobele, G. M. (2007). Parsing ellipsis. Unpublished Manuscript.

Michaelis, J. (1998). Derivational minimalism is mildly context-sensitive. In M. Moortgat (Ed.), *LACL, Lecture Notes in Computer Science* (Vol. 2014, pp. 179–198). Berlin: Springer.

Michaelis, J. & Kracht, M. (1997). Semilinearity as a syntactic invariant. In *Proceedings of logical aspects of computational linguistics*.

Muskens, R. (2001). Lambda Grammars and the syntax-semantics interface. In van Rooy, R. & Stokhof, M., (Eds.), *Proceedings of the thirteenth amsterdam colloquium*, (pp. 150–155). North-Holland: Amsterdam

Radzinski, D. (1990). Unbounded syntactic copying in mandarin chinese. *Linguistics and Philosophy*, *13*(1), 113–127.

Salvati, S. & Kobele, G. (2013). The IO and OI hierarchies revisited. In *Proceedings of the 40th international colloquium on automata, languages and programming* (to be published)

Sarkar, A. & Joshi, A. (1996). Coordination in tree adjoining grammars: Formalization and implementation. In *Proceedings of the 16th conference on Computational linguistics, COLING'96* (Vol. 2, pp. 610–615), Stroudsburg, PA: Association for Computational Linguistics.

Seki, H., Matsamura, T., Mamoru, F., & Kasami, T. (1991). On multiple context-free grammars. *Theoretical Computer Science*, *88*(2), 191–229.

Shieber, S. (1985). Evidence against the context-freeness of natural language. *Linguistic and Philosophy*, *8*, 333–343.

Stabler, E. P. (1996). Derivational minimalism. In Retoré, C., (Ed.), *LACL*, *Lecture Notes in Computer Science*, (Vol. 1328, pp. 68–95). Berlin: Springer.

Vijay-Shanker, K., Weir, D. J., & Joshi, A. K. (1987). Characterizing structural descriptions produced by various grammatical formalisms. In *Proceedings of the 25th annual meeting of the association for computational linguistics*, Stanford.