Theoretical Computer Science 5 (1977) 85–100

ITERATING ITERATED SUBSTITUTION

Joost ENGELFRIET

Department of Applied Mathematics, Twente University of Technology, P.O. Box 217, Enschede The Netherlands

Communicated by Arto Salomaa
Received 15 February 1977
Revised 30 April 1977

Abstract. By iterating iterated substitution not all regular languages can be copied. Hence the smallest full hyper (1)-AFL is properly contained in ETOL, the smallest full hyper-AFL. The number of iterations of iterated subsitution gives rise to a proper hierarchy. Consequently the smallest full hyper (1)-AFL is not a full principal AFL.

1. Introduction

The notion of iterated substitution has been the subject of many investigations in formal language theory. If L is a language over the alphabet V and f is a substitution over V, then we define $f^*(L) = \bigcup_{n=0}^{\infty} f^n(L)$. We call f^* an iterated substitution. If f is nested, i.e. $a \in f(a)$ for all symbols $a \in V$, then f^* is called a nested iterated substitution. If U is a finite set of substitutions over V, then we define $U^*(L) = \bigcup\{f_n \cdots f_2 f_1(L) \mid n \geq 0, f_i \in U\}$. We call U^* an iterated multiple substitution.

Nested iterated substitution was the first notion to be studied, in particular in connection with the context-free languages [14, 18, 21, 38] and the regular tree languages [32, 33]. It was shown that the context-free languages are the smallest full AFL closed under nested iterated substitution. This result can be "explained" by the Kleene Theorem for regular tree languages, where nested iterated substitution plays the role of the star operation (i.e. iterated concatenation) in the case of the regular languages. A general theory of AFLs closed under nested iterated substitution, called super-AFLs, was developed in [12, 13].

The investigation of (nonnested) iterated substitution started by the introduction of parallel rewriting systems, motivated by biological considerations, in [19, 27]. These so-called Lindenmayer systems consisted essentially of an iterated finite substitution applied to a singleton. Later the idea of a Lindenmayer system with "tables" was introduced [23, 24], in which each table is a finite substitution. Such systems consisted of an iterated multiple finite substitution applied to a singleton.

Many other variations have since been introduced (cf. [22, 25]) and the theory of Lindenmayer systems is now a well established part of formal language theory [16, 28, 20]. A general theory of AFLs closed under iterated subtitution, called hyper (1)-AFLs, and AFLs closed under iterated multiple substitution, called hyper-AFLs, was developed in [34, 29, 1]. As a formal device in these investigations the notion of K-iteration grammar (where K is a family of languages) was introduced [34, 29, 37, 26] consisting of an iterated (multiple) K-substitution applied to a singleton followed by intersection with Σ^* for some terminal alphabet Σ (a K-substitution satisfies $f(a) \in K$ for all symbols a). Nested iteration grammars were used in [35]. For a comparison of results on super- and hyper-AFLs see [2, 3, 4].

The properties of iterated substitution are rather poor in comparison to iterated multiple substitution and nested iterated substition. As an example, for a given family K closed under a few operations, it can be shown that the smallest full hyper-AFL containing K can already be obtained by applying the operation of iterated multiple substitution once to the elements of K (followed by intersection with Σ^*), i.e. by taking all languages generated by K-iteration grammars [29, 1]. In particular ETOL (where K is the family of finite languages) is the smallest full hyper-AFL [5, 6]. An analogous statement holds for the smallest super-AFL containing K [12]. The statement fails however in the case of iterated substitution. The family EOL (obtained by iteration grammars with one finite substitution) is not closed under iterated substitution and is not even an AFL, cf. [29]. (We shall show that even if K is a super-AFL, the statement need not be true for the smallest full hyper(1)-AFL containing K.) Thus, to obtain the smallest full hyper(1)-AFL, one has to iterate the process of applying an iterated substitution [4]. In this paper we investigate this iterated iterated substitution. We shall prove that not every full hyper(1)-AFL is a full hyper-AFL, in particular the smallest full hyper(1)-AFL is properly contained in ETOL, the smallest full hyper-AFL. Roughly speaking the idea involved is as follows. Let us say that a language L can be copied in a family K if $\{w\#w\#w \mid w \in L\}$ is in K. With the use of iterated multiple substitution (with finite substitutions) many languages can be copied in ETOL, in particular all regular languages (in fact precisely all EDTOL languages, as shown in [31]). However, if L has too much strings (of each length) then L cannot be copied by using iterated substitution iteratively. Thus $\{a, b\}^*$ cannot be copied into the smallest full hyper(1)-AFL. It was already shown in [31, Theorem 2(b)] that $\{a, b\}^*$ cannot be copied in EOL, i.e. by using one iterated finite substitution (in fact, that only HDOL languages can be copied in EOL). This paper is essentially a generalization of the proof of the latter result.

This paper is divided into 5 sections. Section 2 contains the necessary terminology and some useful facts. In Section 3 we prove a technical result needed in the next section. It shows that each K-iteration grammar with one substitution has an equivalent iteration grammar in which a "final" substitution is applied to the sentential forms rather than that they are intersected by some terminal Σ^*. This has the advantage that every derivation in the grammar yields a terminal word. The disadvantage is that the resulting grammar is in general only a K_∞-iteration grammar where K_∞ is the substitution closure of K. The proof of this result uses the technique of "slicing" (cf. [26]) and is a very weak generalization of [7], see also [36].

Section 4 contains our main general result concerning (iterated) iterated substitution. Let K be a family of languages. We show that languages with certain structural properties which are in the smallest full hyper(1)-AFL containing K, are in fact already in the smallest family that contains K and is closed under iterated λ-free homomorphism. Thus the problem of obtaining languages not in the former family is reduced to the problem of obtaining languages not in the latter, provided they have the mentioned properties (which are possessed by languages of the form $\{w\#w\#w \mid w \in L\}$ and similar ones). Thus the above result expresses that the nondeterminism of the iterated substitutions is of no help with regard to copying, so that they may be replaced by iterated homomorphisms. Such results have been proved in many other situations [31, 8, 9]. The proof consists of a generalization of the proof in [31] that in EOL only HDOL languages can be copied, together with the fact that taking the substitution closure K_∞ of a family K is of no help with respect to copying (under certain restrictions on K). The latter fact is needed to deal with the K_∞ languages that turn up in the previous section.

In Section 5 we apply the copying theorem of Section 4 to the case of the smallest full hyper(1)-AFL. A characterization is given of languages in the smallest family that contains the finite languages and is closed under iterated λ-free homomorphism. For such a language the number of words of length n is polynomial in n. Consequently languages with the above mentioned properties and such that the number of words is not of polynomial order, are not in the smallest full hyper(1)–AFL. An example of such a language is $\{w\#w\#w \mid w \in \{a, b\}^*\}$.

As mentioned before, the smallest family containing a given family K and closed under iterated substitution is obtained by iteratively applying the operation of iterated substitution. The same holds for iterated λ-free homomorphism. The result in Section 4 is proved in such a way that the number of iterations in this iterative process is preserved. In Section 5 we show (by the same argument concerning the number of strings) that this number of iterations gives rise to a proper hierarchy. From this it can be shown that the smallest full hyper(1)-AFL is not a full principal AFL.

2. Terminology and preliminaries

In this section we introduce the terminology needed in this paper. The reader is assumed to be familiar with the basic terminology and facts of formal language theory (cf. [17, 30]), in particular the theory of Lindenmayer systems [16]. We also state a number of useful facts taken from the literature.

For each word w, we identify $\{w\}$ and w. The length of w is denoted by $|w|$. The empty word is denoted by λ. A language is λ-free if it does not contain λ. A mapping f such that $f(x)$ is a language for every x in its domain, is said to be λ-free if all $f(x)$ are λ-free, and $\emptyset$-free if all $f(x)$ are nonempty. For two arbitrary mappings f and g their composition is denoted by fg. Thus $fg(x) = f(g(x))$ for all x. We denote $f \cdots f$(n times) by f^n, in particular the identity mapping by f^0. A family of languages is defined as usual, except that we shall always assume that it contains all singleton languages. The families of finite, regular and context-free languages are denoted by FIN, REG and CF respectively.

Let V be an alphabet. A *substitution* is (as usual) a mapping f from V into languages, extended to words over V by $f(\lambda) = \{\lambda\}$ and $f(a_1 \cdots a_n) = f(a_1) \cdots f(a_n)$, and extended to languages over V by $f(L) = \bigcup\{f(w) \mid w \in L\}$. It is said to be a substitution over V if $f(a)$ is a language over V for all a in V; it is said to be a K-*substitution* (for a family K of languages) if $f(a) \in K$ for each $a \in V$, and to be *nested* if $a \in f(a)$ for each $a \in V$.

Let f be a substitution over V. For a language L over V we define $f^*(L) = \bigcup_{n=0}^{\infty} f^n(L)$. The mapping f^* is called an *iterated substitution* [12]. If f is nested, then f^* is called a nested iterated substitution. Let U be a finite set of substitutions over V. For a language L over V we define $U^*(L) = \bigcup\{f_n \cdots f_2 f_1(L) \mid n \geq 0, f_i \in U\}$. We shall call the mapping U^* an *iterated multiple substitution*. It is called nested if all elements of U are nested. A family K is closed under substitution (iterated substitution, iterated multiple substititution) if $f(L)$ is in K whenever $L \in K$ and f is a K-substitution (iterated K-substitution, iterated multiple K-substitution respectively). We note that if K is closed under union, then K is closed under nested iterated multiple substitution iff it is closed under nested iterated substitution (given a nested U, define g such that $g(a) = \bigcup\{f(a) \mid f \in U\}$; then $g^* = U^*$), cf. [2].

Let K be a family of languages. A K-*iteration grammar* is a quadruple $G = (V, \Sigma, A, U)$, where V is an alphabet, Σ is a subset of V (the terminal alphabet), $A \in K$ is a language over V (the set of axioms) and U is a finite set of K-substitutions over V. The set of sentential forms generated by G is defined by $L_S(G) = U^*(A)$, and the language generated by G by $L(G) = U^*(A) \cap \Sigma^*$. If U has n elements then G will be called a K-(n) iteration grammar. G is said to be λ-free ($\emptyset$-free) if all elements of U are λ-free ($\emptyset$-free respectively). We note that our definition of K-iteration grammar differs from the usual one in [29] in that it has a whole set of axioms rather than just one. It is easy to see that if K contains all singleton languages (as is assumed throughout the paper), then the two definitions are equivalent (with preservation of the number of substitutitons). In the sequel we will mainly be interested in K-(1) iteration grammars G which will be denoted as (V, Σ, A, g) rather than $(V, \Sigma, A, \{g\})$. For such a grammar we shall also write

$w_1 \Rightarrow^k w_2$ if $w_2 \in g^k(w_1)$ and we shall talk about derivations in the usual way. Note that $L_S(G) = g^*(A)$ and $L(G) = g^*(A) \cap \Sigma^*$. The family of languages generated by K-iteration grammars will be denoted by $H(K)$. By $H_n(K)$ we denote the family of languages generated by K-(n) iteration grammars ($n \geq 1$). It can be shown that (under weak assumptions on K) $H_2(K) = H(K)$, see [1]. In this paper we deal with $H_1(K)$. We denote $\bigcup_{n=0}^{\infty} H_1^n(K)$ by $H_1^*(K)$.

The following terminology will be used concerning closure properties. Let K be a family of languages. K is a *pre-quasoid* [1] if it is closed under finite substitution and intersection with regular languages. K is a *quasoid* [34, 29] if it is a pre-quasoid containing all regular languages. We note that FIN is the only pre-quasoid which is not a quasoid. The next concept is only introduced for the purposes of this paper. K is an *SFL* (special family of languages) if it is a pre-quasoid closed under union and concatenation. Observe that FIN is an SFL. K is *substitution-closed* if it is closed under K-substitution. We denote by K_∞ the smallest substitution-closed family containing K. Note that $\text{FIN}_\infty = \text{FIN}$. K is a *super-AFL* if it is a full AFL closed under nested iterated substitution [12]. K is a *full hyper*(1)*-AFL* if it is a full AFL closed under iterated substitution (or, equivalently, a full AFL such that $H_1(K) = K$). Finally, K is a *full hyper-AFL* if it is a full AFL closed under iterated multiple substitution (or, equivalently, a full AFL such that $H(K) = K$; see [29], where the adjective full is not used).

Before continuing our terminology we state a number of facts from the literature. We note first that $H(\text{FIN}) = \text{ETOL}$, $H(\text{ONE}) = \text{EDTOL}$ (where ONE is the family of all singleton languages) and $H_1(\text{FIN}) = \text{EOL}$ [29]. It was shown in [5, 6] that ETOL is the smallest full hyper-AFL. This was generalized in [1]: $H(K)$ is a full hyper-AFL for every pre-quasoid K (and in fact the smallest one containing K). Thus, for a pre-quasoid K, $H(H(K)) = H(K)$, which means that iteration of H has no effect. EOL is an SFL, but not an AFL [15, 29]. $H_1(\text{REG})$ is a full AFL, not closed under (iterated) substitution [5], so that H_1 is not idempotent in general. In the following lemma we state closure properties of K_∞, $H_1(K)$ and $H_1^*(K)$ under suitable restrictions on K (together with a similar statement for the nested case).

Lemma 2.1. *Let K be a family of languages.*

(1) *If K is a quasoid, then* K_∞ *is the smallest substitution-closed full AFL containing K.*

(2) *If K is a full AFL, then* $\{f^*(L) \cap R \mid f^*$ *is a nested iterated substitution,* $L \in K$ *and R is a regular language*$\}$ *is the smallest super-AFL containing K.*

(3) *If K is a quasoid, then* $H_1(K)$ *is a full AFL.*

(4) *If K is a pre-quasoid, then* $H_1^*(K)$ *is the smallest full hyper*(1)*-AFL containing K.*

(5) $H_1^*(\mathit{FIN}) = H_1^*(\mathit{EOL})$ *is the smallest full hyper*(1)*-AFL.*

Proof. [11, 12, 29, 4, 4] respectively. We observe that (4) follows from (3) and (5) from (4). □

We now continue our terminology. An *NPDOL scheme* is a quadruple $S = (V, f, \Sigma, h)$, where f is a λ-free homomorphism over V and h is a λ-free homorphism from V^* into Σ^*. If L is a language over V, then we denote by $S(L)$ the language $h(f^*(L))$. Note that an NPDOL scheme is an NPDOL system [22] without axiom, thus, for $w \in V^+$, $S(w)$ is an NPDOL language. For a family K we denote by NPDOL(K) the family $\{S(L) \mid S$ is an NPDOL scheme and $L \in K\}$. We denote $\bigcup_{n=0}^{\infty} \mathrm{NPDOL}^n(K)$ by NPDOL*(K). Note that NPDOL*(K) is the smallest family containing K and closed under iterated λ-free homomorphism. It is left to the reader to show that NPDOL(K) $\subseteq H_1(K)$ and even NPDOL(K_∞) $\subseteq H_1(K)$. Consequently, for $n \geq 1$, $\mathrm{NPDOL}^n(K_\infty) \subseteq H_1^n(K)$. It should also be clear that NPDOL*(FIN) $\subseteq$ EDTOL.

We end this section by defining two properties, (F) and (S) of a language L over V.

(F) For all $u, u', x, x', v, v' \in V^*$, if uxv, $ux'v$, $u'xv'$ and $u'x'v'$ are in L, then $x = x'$ or both $u = u'$ and $v = v'$.

(S) For every integer t there exists an integer T such that for all $u, x, y, v \in V^*$, if $uxv \in L$, $|uxv| \geq T$, $|x| \leq t$ and $uyv \in L$, then $x = y$.

Property (F) was used by Fischer [10]; see also [9] where this property is discussed (as property (P1)). Property (S) was used implicitly by Skyum [31] to show that EOL languages having this property are in HDOL(= NPDOL(FIN), see [22] where NPDOL(FIN) is denoted by NPDFOL). Intuitively it says that one cannot change small subwords of a word in L without leaving L. It is easy to show that, for any language M, languages such as for instance $\{w\#w\#w \mid w \in M\}$, $\{w\#w^R\#w\#w^R\# \mid w \in M\}$ and $\{f(w)g(w)h(w) \mid w \in M\}$ have both properties (F) and (S) (where $\#$ is a new symbol, w^R is the reverse of w, f, g and h are length-preserving 1–1 mappings with disjoint target alphabets).

3. Change of filter

The language defined by a K-(1) iteration grammar $G = (V, \Sigma, A, g)$ is obtained by first generating the set of sentential forms of G and then putting this through the "filter" that only allows words over Σ. In this section we show that (apart from derivations of some bounded length this filter can be changed into one that applies an $\emptyset$-free substitution to the sentential forms. We have however, to pay the price of using K_∞- rather than K-substitutions. This result is expressed in the following theorem, in which we simultaneously show that λ-freeness can be obtained. The proof of the theorem is analogous to part of the proof in [7]. It uses the technique of slicing [26].

Theorem 3.1. *Let K be a pre-quasoid. Each language in $H_1(K)$ is the union of a language in K_∞ and a language of the form $f(L_S(G))$, where f is an $\emptyset$-free λ-free K_∞-substitution and G is an $\emptyset$-free λ-free K_∞-(1) iteration grammar.*

Proof. Since K_∞ is either FIN or a full AFL (cf. Lemma 2.1(1)), K_∞ is closed under union with $\{\lambda\}$. Therefore it suffices to prove the theorem for λ-free languages in $H_1(K)$. In [29] it is shown that, under the given conditions on K, each λ-free K-(1) iteration language can be generated by a λ-free K-(1) iteration grammar. Thus let $G_0 = (V, \Sigma, A, g)$ be a λ-free K-(1) iteration grammar. We shall show the theorem for $L(G_0)$. First we introduce several definitions taken from [7]. For $a \in V$ the spectrum of a, denoted by Spec(a), is defined as $\{n \geq 0 \mid g^n(a) \cap \Sigma^* \neq \emptyset\}$. Thus $n \in \text{Spec}(a)$ iff a generates a terminal word in n steps. A symbol a in V is said to be vital if Spec(a) is infinite. In [37] it is proved that, for each a in V, Spec(a) is an ultimately periodic set of integers. For an arbitrary ultimately periodic set I we denote by per(I) a period of I and by thres(I) a threshold of I, i.e. an integer such that $\{n \in I \mid n \geq \text{thres}(I)\}$ is periodic with period per(I). We now define the uniform period of G_0, denoted by m, to be an integer such that

(i) for all nonvital a in V, $\text{Spec}(a) \subseteq \{0, 1, \ldots, m-1\}$;

(ii) for all vital a in $V, m \geq \text{thres}(\text{Spec}(a))$ and per(Spec(a)) divides m.

We now construct the (1)iteration grammar $G = (\Delta, \Delta, B, h)$, where $\Delta = \{a \in V \mid m \in \text{Spec}(a)\}$, $B = \{w \in \Delta^* \mid v \Longrightarrow^k w$ for some $k < 2m$ and $v \in A\}$ and, for $a \in \Delta$, $h(a) = \{w \in \Delta^* \mid a \Longrightarrow^m w\}$. Since K_∞ is either FIN or a full AFL, G is a K_∞-(1) iteration grammar (note that $B = \bigcup\{g^k(A) \mid 0 \leq k \leq 2m\} \cap \Delta^*$ and $h(a) = g^m(a) \cap \Delta^*$). Since G_0 is λ-free, so is G. To see that G is $\emptyset$-free, consider $a \in \Delta$. Thus $m \in \text{Spec}(a)$ and therefore, by (ii), $2m \in \text{Spec}(a)$. Hence there exist $w \in V^*$ and $x \in \Sigma^*$ such that $w \in g^m(a)$ and $x \in g^m(w)$. Clearly $w \in \Delta^*$.

Next we define the K_∞-substitution f such that, for $a \in \Delta$, $f(a) = \{w \in \Sigma^* \mid a \Longrightarrow^m w$ in $G_0\}$. Obviously f is $\emptyset$-free and λ-free. Finally, let $M = \{w \in \Sigma^* \mid v \Longrightarrow^k w$ in G_0 for some $k < 2m$ and $v \in A\}$. Obviously $M \in K_\infty$. We now claim that $L(G_0) = M \cup f(L_S(G))$, which proves the theorem. Clearly $M \cup f(L_s(G))$ is included in $L(G_0)$. To show the converse, let $x \in g^p(V) \cap \Sigma^*$ for some $p \geq 2m$ and some $v \in A$. Let $p = qm + r$ for some q and r such that $q \geq 2$ and $0 \leq r < m$. Let

$$v \overset{m+r}{\Longrightarrow} w_1 \overset{m}{\Longrightarrow} w_2 \overset{m}{\Longrightarrow} w_3 \overset{m}{\Longrightarrow} \cdots \overset{m}{\Longrightarrow} w_{q-1} \overset{m}{\Longrightarrow} x$$

be a derivation in G_0 of x from v. Let the symbol a occur in w_i. It produces some terminal word in $(q-i)m \geq m$ steps. Hence a is vital by (i), and since $(q-i)m \in \text{Spec}(a)$, $m \in \text{Spec}(a)$ by (ii). Consequently all words w_i are in Δ^*. Hence $w_1 \Longrightarrow w_2 \Longrightarrow \cdots \Longrightarrow w_{q-1}$ is a derivation in G (note that $m + r < 2m$, so that $w_1 \in B$) and $x \in f(w_{q-1})$. Thus $x \in f(L_S(G))$. □

4. A copying theorem

Let K be an SFL and L a language with properties (F) and (S). In this section we want to show that if $L \in H_1^*(K)$ then $L \in \text{NPDOL}^*(K)$. More precisely, for every

$n \geq 0$, if $L \in H_1^n(K)$ then $L \in \text{NPDOL}^n(K)$ (Theorem 4.4). This theorem can easily be proved by induction from the result that, for every $n \geq 0$, if $L \in \text{NPDOL}^n(H_1(K))$ then $L \in \text{NPDOL}^n(\text{NPDOL}(K))$ (Lemma 4.3). The kernel of the proof of this result is an obvious generalization of the proof of the fact that if $L \in H_1(\text{FIN})$ then $L \in \text{NPDOL}(\text{FIN})$ (see [31]) and uses essentially property (S) of L. However, since in this proof we start by transforming the initial K-(1) iteration grammar according to Theorem 3.1, we need the following lemma to deal with the K_∞ languages turning up in that transformation (Lemma 4.2): for every $n \geq 0$, if $L \in \text{NPDOL}^n(K_\infty)$ then $L \in \text{NPDOL}^n(K)$. This is in fact a special case of Lemma 4.3 (recall that $\text{NPDOL}(K_\infty) \subseteq H_1(K)$). Its proof uses property (F) of L.

Before showing the above mentioned results we prove the following useful lemma, which roughly speaking provides us with a way of changing a substitution, involved in the generation of a language with property (F), into a homomorphism.

Lemma 4.1. *Let $S_1, \ldots, S_n$ ($n \geq 0$) be NPDOL schemes, f an $\emptyset$-free substitution and M a language. Let $L = S_n \cdots S_1(f(M))$. If L has property* (F), *then $L = S_n \cdots S_1(h(M) \cup \bigcup\{h(u_a)f(a)h(v_a) \mid a \in A\})$, where h is any homomorphism such that $h(a) \in f(a)$ for all a, A is the set of all symbols occurring in words of M, and u_a and v_a are any words such that $u_a a v_a \in M$.*

Proof. Denote $S_n \cdots S_1(h(M) \cup \bigcup\{h(u_a)f(a)h(v_a) \mid a \in A\})$ by N. It should be obvious that $N \subseteq L$. To show that $L \subseteq N$, let $z \in L$. Let, for $1 \leq j \leq n$, $S_j = (V_j, f_j, \Sigma_j, h_j)$. There exist words x, y and integers $k(1), \ldots, k(n)$ such that $x \in M$, $y \in f(x)$ and $z = g(y)$, where g denotes the homomorphism $h_n f_n^{k(n)} \cdots h_1 f_1^{k(1)}$. Let $x = a_1 \cdots a_m$ with $a_i \in A$ and $y = w_1 \cdots w_m$ with $w_i \in f(a_i)$.

We now consider three cases.

Case 1. $g(w_i) = g(h(a_i))$ for all i, $1 \leq i \leq m$. Then $g(h(x)) = g(h(a_1)) \cdots g(h(a_m)) = g(w_1) \cdots g(w_m) = z$. Hence $z \in S_n \cdots S_1(h(M))$ and so $z \in N$.

Case 2. There is exactly one i such that $g(w_i) \neq g(h(a_i))$. Let $u = a_1 \cdots a_{i-1}$ and $v = a_{i+1} \cdots a_m$. Denote a_i by a and w_i by w. Thus $x = uav$ and $z = g(h(u))g(w)g(h(v))$. It follows that $g(h(u))g(w)g(h(v))$ and $g(h(u))g(h(a))g(h(v))$ are in L, and (starting from the word $u_a a v_a$) $g(h(u_a))g(w)g(h(v_a))$ and $g(h(u_a))g(h(a))g(h(v_a))$ are in L. Consequently, since L has property (F) and $g(w) \neq g(h(a))$, we have that $g(h(u_a)) = g(h(u))$ and $g(h(v_a)) = g(h(v))$. Hence $z = g(h(u_a))g(w)g(h(v_a))$, and so $z \in S_n \cdots S_1(h(u_a)f(a)h(v_a))$, and $z \in N$.

Case 3. there are i and j such that $g(w_i) \neq g(h(a_i))$ and $g(w_j) \neq g(h(a_j))$. It is left to the reader to show that this case cannot occur due to property (F) of L (in fact, property (F) implies property (P2) of [9] which forbids L to have two different possibilities for two nonoverlapping subwords). $\square$

We now show that languages with property (F) which can be generated by a number of NPDOL schemes from a language in K_∞, can in fact be generated from a language in K.

Lemma 4.2. *Let K be an SFL, and let $n \geq 0$. If L has property (F) and $L \in \mathrm{NPDOL}^n(K_\infty)$, then $L \in \mathrm{NPDOL}^n(K)$.*

Proof. We first note that, by [11], $K_\infty = \bigcup_{m=1}^{\infty} K_m$, where $K_1 = K$ and $K_{m+1} = \{f(L) \mid L \in K_m$ and f is a K-substitution$\}$. We also note that $\mathrm{NPDOL}^n(K)$ is closed under union (If $L_1 = S_n \cdots S_1(M_1)$ and $L_2 = T_n \cdots T_1(M_2)$ then, after some necessary alphabetic changes, $L_1 \cup L_2 = R_n \cdots R_1(M_1 \cup M_2)$ where R_i is obtained by joining the alphabets and homomorphisms of S_i and T_i two by two). Thus it suffices to show that, for $m \geq 1$, if L has property (F) and $L \in \mathrm{NPDOL}^n(K_m)$, then L is a finite union of languages from $\mathrm{NPDOL}^n(K)$. We show this by induction on m.

For $m = 1$ the statement is trivial. Suppose it is true for m and let $L \in \mathrm{NPDOL}^n(K_{m+1})$ have property (F). Thus $L = S_n \cdots S_1(f(M))$, where $S_1, \ldots, S_n$ are NPDOL schemes, f is a K-substitution and $M \in K_m$. We may assume that f is $\emptyset$-free (otherwise we intersect M with Δ^*, where $\Delta = \{a \mid f(a) = \emptyset\}$; it is easy to see that K_m is closed under intersection with Δ^*). Thus Lemma 4.1 is applicable, so that $L = S_n \cdots S_1(h(M)) \cup S_n \cdots S_1(\bigcup\{h(u_a)f(a)h(v_a) \mid a \in A\})$. Obviously $\bigcup \{h(u_a)f(a)h(v_a) \mid a \subset A\}$ is in K and $h(M)$ is in K_m. Consequently L is the union of an $\mathrm{NPDOL}^n(K)$ language and an $\mathrm{NPDOL}^n(K_m)$ language. Since every subset of L also has property (F), it follows by induction that L is a finite union of languages from $\mathrm{NPDOL}^n(K)$. □

We now turn to the main stage in the proof of the copying theorem.

Lemma 4.3. *Let K be an SFL. Let L be a language with properties (F) and (S). For every $n \geq 0$, if $L \in \mathrm{NPDOL}^n(H_1(K))$ then $L \in \mathrm{NPDOL}^{n+1}(K)$.*

Proof. We observe that $\mathrm{NPDOL}^{n+1}(K)$ is closed under union and that $\mathrm{NPDOL}^n(K) \subseteq \mathrm{NPDOL}^{n+1}(K)$.

Let $L = S_n \cdots S_1(M_0)$, where $S_1, \ldots, S_n$ are NPDOL schemes and $M_0 \in H_1(K)$. By Theorem 3.1, $M_0 = M_1 \cup M_2$ with $M_1 \in K_\infty$ and $M_2 = f(L_S(G))$ for some K_∞-substitution f and K_∞-(1) iteration grammar G (both $\emptyset$-free and λ-free). Thus $L = S_n \cdots S_1(M_1) \cup S_n \cdots S_1(f(L_S(G)))$. Since every subset of L also has property (F), it follows from Lemma 4.2 that $S_n \cdots S_1(M_1) \in \mathrm{NPDOL}^n(K)$. By our observation above it now suffices to show that $L_1 = S_n \cdots S_1(f(L_S(G)))$ is in $\mathrm{NPDOL}^{n+1}(K)$. Note that, being a subset of L, L_1 also has properties (F) and (S). We now apply Lemma 4.1 to L_1 (with $M = L_S(G)$). Thus $L_1 = L_2 \cup L_3$ with $L_2 = S_n \cdots S_1(h(L_S(G)))$ and $L_3 = S_n \cdots S_1(\bigcup\{h(u_a)f(a)h(v_a) \mid a \in A\})$, where h is a λ-free homomorphism. Since f is a K_∞-substitution and K_∞ is an AFL (Lemma 2.1(i)), $L_3 \in \mathrm{NPDOL}^n(K_\infty)$. Hence, by Lemma 4.2, $L_3 \in \mathrm{NPDOL}^n(K)$.

It remains to show that $L_2 = S_n \cdots S_1(h(L_S(G)))$ is in $\mathrm{NPDOL}^{n+1}(K)$. Note that L_2 still has properties (F) and (S). Let $G = (V, V, B, g)$. We write $L(G)$ rather than $L_S(G)$. Let m be an integer such that if $x \in L(G)$ and $|x| \geq m$, then each symbol occurring in x occurs in infinitely many other words of $L(G)$. Define $D = \bigcup\{g^i(B) \mid 0 \leq i \leq p\}$, where p is chosen such that $\bigcup\{g^i(B) \mid 0 \leq i \leq p-1\}$ contains all $x \in L(G)$ with $|x| < m$. Clearly $D \in K_\infty$. Construct the NPDOL scheme $S_0 = (V, f_0, \Sigma_0, h_0)$, where f_0 is any (λ-free) homomorphism such that for all $a \in V$ $f_0(a) \in g(a)$, and $h_0 = h$ (Σ_0 being its target alphabet).

We will prove that $L_2 = S_n \cdots S_1 S_0(D)$. From that it follows that $L_2 \in \mathrm{NPDOL}^{n+1}(K_\infty)$ and so, by Lemma 4.2, $L_2 \in \mathrm{NPDOL}^{n+1}(K)$, which completes our proof. Obviously $S_n \cdots S_1 S_0(D) \subseteq L_2$. To show the converse, let $z \in L_2$. Let, for $1 \leq j \leq n$, $S_j = (V_j, f_j, \Sigma_j, h_j)$. There exist a word $y \in L(G)$ and integers $k(1), \ldots, k(n)$ such that $z = \psi(y)$, where ψ denotes the λ-free homomorphism $h_n f_n^{k(n)} \cdots h_1 f_1^{k(1)} h$. If $|y| < m$, then $y \in D$ and $h(y) \in S_0(D)$, so that $z \in S_n \cdots S_1 S_0(D)$. Now let $|y| \geq m$. By the definition of D and the λ-freeness of G there exists $x \in L(G)$ such that $x \in D$, $|x| \geq m$ and $x \Longrightarrow^i y$ for some $i \geq 0$. Let $x = a_1 \cdots a_r$, $y = w_1 \cdots w_r$ and $a_j \Longrightarrow^i w_j$ for $1 \leq j \leq r$. We now show that for each j, $1 \leq j \leq r$, $\psi(w_j) = \psi(f_0^i(a_j))$. Let $t = |\psi(f_0^i(a_j))|$, and let T correspond to t as in the statement of property (S) in Section 2. Since $|x| \geq m$, a_j occurs in infinitely many elements of $L(G)$. Consider a word $u = u_1 a_j u_2$ in $L(G)$ with $|u| \geq T$. Then both

$$\psi(f_0^i(u)) = \psi(f_0^i(u_1))\psi(f_0^i(a_j))\psi(f_0^i(u_2)) \quad \text{and} \quad \psi(f_0^i(u_1))\psi(w_j)\psi(f_0^i(u_2))$$

are in L_2. Moreover, since ψf_0^i is λ-free, $|\psi(f_0^i(u))| \geq |u| \geq T$. Hence, by property (S), $\psi(f_0^i(a_j)) = \psi(w_j)$, as we wanted to show. This implies that

$$z = \psi(y) = \psi(w_1 \cdots w_r) = \psi(f_0^i(a_1 \cdots a_r)) = \psi(f_0^i(x)),$$

and consequently $z \in S_n \cdots S_1 S_0(D)$, which proves the lemma. □

We finally state the copying theorem for $H_1^n(K)$.

Theorem 4.4. *Let K be an SFL. Let L be a language with properties (F) and (S). For every $n \geq 0$, if $L \in H_1^n(K)$, then $L \in \mathrm{NPDOL}^n(K)$.*

Proof. This follows easily from Lemma 4.3 by induction. To be able to apply this lemma we have to show that $H_1^m(K)$ is an SFL for every $m \geq 0$. It is obviously sufficient to show that if K_0 is an SFL, then so is $H_1(K_0)$. Now, if $K_0 = \mathrm{FIN}$, then $H_1(K_0) = \mathrm{EOL}$, which is an SFL. If $K_0 \neq \mathrm{FIN}$, then it is a quasoid and so $H_1(K_0)$ is a full AFL by Lemma 2.1(3). □

Another (weaker) way to express this theorem is to say that if K is an SFL and if L (with properties (F) and (S)) is in the smallest full hyper(1)-AFL containing K, then L is in the smallest family that contains K and is closed under iterated λ-free homomorphism (cf. Lemma 2.1(4)).

5. The smallest full hyper (1)-AFL

In this section we apply the copying theorem of Section 4 to the case that $K = \mathrm{FIN}$. Using a characterization of $\mathrm{NPDOL}^n(\mathrm{FIN})$, we then show that the families $H_1^n(\mathrm{FIN})$ form a proper hierarchy, properly contained in ETOL. Thus (cf. Lemma 2.1(4) and (5)) the smallest full hyper(1)-AFL $H_1^*(\mathrm{FIN})$ is properly contained in the smallest full hyper-AFL ETOL. It also follows that $H_1^*(\mathrm{FIN})$ is not a full principal AFL. At the end of the section we give an inclusion diagram of all families discussed.

Since FIN is an SFL, the next corollary follows directly from Theorem 4.4.

Corollary 5.1. *Let L be a language with properties (F) and (S). For every $n \geq 0$, if $L \in H_1^n(\mathrm{FIN})$ then $L \in \mathrm{NPDOL}^n(\mathrm{FIN})$.* □

To show that certain languages are not in $\mathrm{NPDOL}^n(\mathrm{FIN})$ we now give, for any L in $\mathrm{NPDOL}^n(\mathrm{FIN})$, an estimation of the number of words in L of a given length. Let, for any language L and integer k, $nw(L, k)$ denote the number of words in L of length k.

Theorem 5.2. *For every $n \geq 1$ and every language L, if $L \in \mathrm{NPDOL}^n(\mathrm{FIN})$ then $nw(L, k) = O(k^{n-1})$.*

Proof. We shall prove the statement by induction on n. For $n = 1$ we have to show that $nw(L, k) = O(1)$, i.e. that $nw(L, k)$ is bounded by a constant, for $L \in \mathrm{NPDOL}(\mathrm{FIN})$. This result is proved in [22, Lemma 5.10]. Now assume that the theorem holds for n and consider $L \in \mathrm{NPDOL}^{n+1}(\mathrm{FIN}))$. Thus $L = S(M)$, where S is an NPDOL scheme and $M \in \mathrm{NPDOL}^n(\mathrm{FIN})$. By induction, $nw(M, k) = O(k^{n-1})$. Since the homomorphisms of S are λ-free, $x \in L$ iff $x \in S(y)$ for some $y \in M$ with $|y| \leq |x|$. If we can show that there is a constant C such that for all words y and all k, $nw(S(y), k) \leq C$, then, for all k,

$$nw(L, k) \leq C \cdot \sum_{i=0}^{k} nw(M, i) = C \cdot \sum_{i=0}^{k} O(i^{n-1}) = O(k^n),$$

and the theorem is proved.

To prove this we note that it has been shown in the proofs of Lemma 3.1 and Theorem 4.12 of [22] that for each NPDOL scheme $S_1 = (V_1, f_1, \Sigma_1, h_1)$ there exist an NPDOL scheme $S_2 = (V_2, f_2, \Sigma_2, h_2)$ and an integer N such that (1) the homomorphism h_2 is length-preserving (i.e. a symbol to symbol coding), and (2) for each $w \in V_1^*$ there is a finite set $W \subseteq V_2^*$ of cardinality N such that $S_1(w) = S_2(W)$. It is straightforward to see that $nw(S_2(W), k) \leq N \cdot A$, where A is a constant depending on S_2 only (in fact, A is the maximum of the cardinality of V_2 and the product of all $\max\{nw(S_2(a), k) \mid k \leq 0\}$ for all $a \in V_2$ such that $S_2(a)$ is finite).

Hence $nw(S_1(w), k) \leq N \cdot A$, where $N \cdot A$ only depends on S_1 and not on w. This proves the statement and the theorem. □

Corollary 5.1 and Theorem 5.2 together lead to the next corollary.

Corollary 5.3. *For $n \geq 1$, if L has properties (F) and (S) and $nw(L, k)$ is not $O(k^{n-1})$, then $L \notin H_1^n(\mathrm{FIN})$.* □

It is now easy to find examples of languages not in any $H_1^n(\mathrm{FIN})$.

Theorem 5.4. *$H_1^*(\mathrm{FIN})$ is properly included in ETOL. In particular there exist EDTOL languages not in $H_1^*(\mathrm{FIN})$.*

Proof. Consider for instance $L = \{w\#w\#w \mid w \in \{a, b\}^*\}$. L has properties (F) and (S), and is in EDTOL. Obviously $nw(L, 3m+2) = 2^m$, and so $nw(L, k)$ is not $O(k^n)$ for any $n \geq 0$. Thus, by Corollary 5.3, L is not in $H_1^*(\mathrm{FIN})$. □

This theorem shows that the smallest full hyper(1)-AFL is properly included in the smallest full hyper-AFL. It expresses the fact that iterated iteration of one substitution is less powerful than a single iteration of a multiple substitution. The next result shows that the number of times the process of applying an iterated substitution is iterated gives rise to a proper hierarchy.

Theorem 5.5. *For $n \geq 0$, $H_1^n(\mathrm{FIN})$ is a proper subset of $H_1^{n+1}(\mathrm{FIN})$. In particular there exist $\mathrm{NPDOL}^{n+1}(\mathrm{FIN})$ languages not in $H_1^n(\mathrm{FIN})$.*

Proof. Let, for $n \geq 1$, $L_n = \{w\#w\#w\# \mid w \in a_1^* a_2^* \cdots a_n^*\}$, where $a_1, \ldots, a_n$, $\#$ are different symbols. We will show that $L_n \in \mathrm{NPDOL}^n(\mathrm{FIN}) - H_1^{n-1}(\mathrm{FIN})$. First we prove that L_n is in $\mathrm{NPDOL}^n(\mathrm{FIN})$. Let, for $1 \leq i \leq n$, S_i be the NPDOL scheme $(V_i, f_i, \Sigma_i, h_i)$, where $V_i = \Sigma_i = \{\#, a_1, \ldots, a_i\}$, h_i is the identity, $f_i(\#) = a_i\#$ and $f_i(a_j) = a_j$ for $1 \leq j \leq i$. Then clearly $L_n = S_n \cdots S_1(\#\#\#)$. Secondly we prove that $L_n \notin H_1^{n-1}(\mathrm{FIN})$. This is clear for $n = 1$. Now let $n \geq 2$. Obviously L_n has properties (F) and (S). Moreover it is easy to see that there is a positive constant C such that, for all sufficiently large m, $nw(L_n, 3m+3) \geq Cm^{n-1}$. Hence $nw(L_n, k)$ is not $O(k^{n-2})$. It now follows from Corollary 5.3 that $L_n \notin H_1^{n-1}(\mathrm{FIN})$. □

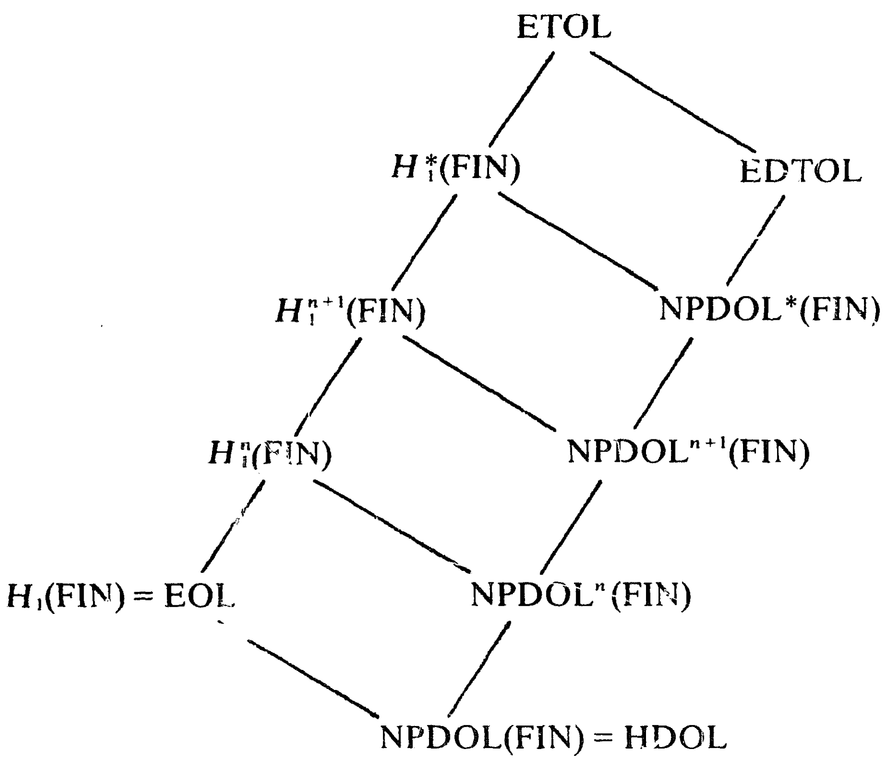
By this theorem, no family K containing FIN and contained in $H_1^n(\mathrm{FIN})$ for some n is closed under iterated λ-free homomorphism (otherwise $\mathrm{NPDOL}^{n+1}(\mathrm{FIN})$ would be included in $H_1^n(\mathrm{FIN})$). Consider for instance $H_1(\mathrm{CF})$. It is contained in $H_1^2(\mathrm{FIN})$ and contains FIN. Thus, $H_1(\mathrm{CF})$ is not closed under iterated λ-free homomorphism. This example shows that even if K is a super-AFL, $H_1(K)$ need not be a full hyper(1)-AFL (cf. Lemma 2.1).

Corollary 5.6. *There is a super-AFL K such that $H_1(K)$ is not closed under iterated λ-free homomorphism.* □

Since by Lemma 2.1(3) H_1^n(FIN) is a full AFL (for $n \geq 2$) and H_1^*(FIN) is their union, Theorem 5.5 proves that H_1^*(FIN) is not a full principal AFL.

Corollary 5.7. *H_1^*(FIN) is not a full principal AFL.*

Note that according to the results of [12], this also implies that there is no language L such that H_1^*(FIN) is the smallest super-AFL containing L. Thus H_1^*(FIN) is the union of a proper hierarchy of super-AFLs. In fact, using Lemma 2.1, it can easily be shown that the smallest super-AFLs containing H_1^n(FIN) form such a hierarchy. We finally put the language families discussed in this section in an inclusion diagram, the correctness of which follows from Theorem 5.4 and 5.5.



($n \geq 2$)

For readers of [31] we observe that this diagram can be inserted into the diagram of [31, Fig. 4]. To show the necessary incomparabilities to the other families in the latter diagram we note that the language $\{w\#w\#w \mid w \in \{a, b\}^*\}$ used in the proof of Theorem 5.4 is in IP and in ED, that $\{a, b\}^*$ is in REG — NPDOL*(FIN) by Theorem 5.2, and that the lamguage $\{a_1^k a_2^m \# b_1^k b_2^m \# c_1^k c_2^m \mid k, m \geq 0\}$ is in NPDOL^2(FIN) — ER (by a result concerning ER in [31]).

As far as H_1(REG) and H_1(CF) are concerned, they can easily be added to the above diagram, because EOL $\subsetneq H_1$(REG) $\subsetneq H_1$(CF) $\subsetneq H_1^2$(FIN). Let us prove this.

The inclusions are obvious from the fact that FIN $\subseteq$ REG $\subseteq$ CF $\subseteq$ EOL. Since H_1(REG) is an AFL whereas EOL is not, EOL $\subsetneq H_1$(REG). Proper inclusion of H_1(REG) in H_1(CF) can be shown in an analogous way as that used in [5] to show nonclosure of H_1(REG) under substitution. In fact let $L = \{w_1 \# w_2 \# \cdots \# w_k \mid k = 2^m$ for some $m \geq 0$, $w_i \in \{a^n b^n \mid n \geq 0\}\}$. Then $L \in H_1$(CF). By SFL operations one can obtain from L the language $\{w \in \{a, b\}^* \mid$ the number of b's is a power of 2$\}$ which is not in EOL [16]. Thus $L \notin H_1$(FIN). Suppose $L \in H_1$(REG) $- H_1$(FIN). By the pumping lemma for regular languages and the fact that REG-(1) iteration grammars can be made λ-free, there is a word in L with a nonempty subword u that can be iterated. Since the number of $\#$'s in words from L is exponential, the number of $\#$'s in U can only be 0. This implies that a subword of $a^n b^n$ can be pumped up, which is a contradiction. This proves that H_1(REG) is properly included in H_1(CF). To show that H_1(CF) is properly included in H_1(EOL) $= H_1^2$(FIN), consider the language L_2 used in the proof of Theorem 5.5. Thus $L_2 \in H_1^2$(FIN) $- H_1$(FIN). Assume that $L_2 \in H_1$(CF). Then, by Theorem 4.4, $L_2 \in$ NPDOL(CF). Using exactly the same technique as in the proof of Theorem 4.4 it can easily be shown that this implies that $L_2 \in$ NPDOL(FIN), which is a contradiction. (We note that in fact Lemma 4.2 can be generalized by replacing K_∞ in the statement of the lemma by the smallest super-AFL containing K). Hence H_1(CF) is properly included in H_1(EOL).

6. Conclusion

We have proved that the smallest full hyper(1)-AFL is properly contained in the smallest full hyper-AFL ETOL. Thus the operation of (iterated) iterated substitution is weaker than that of iterated multiple substitution. The smallest full hyper(1)-AFL is not a full principal AFL, i.e. no Chomsky–Schützenberger-like characterization holds for this family (unlike the smallest super-AFL and the smallest full hyper-AFL). It is open whether there exist other full hyper(1)-AFLs which are not full hyper-AFLs. Is there a whole hierarchy of full hyper(1)-AFLs included in ETOL? Is the smallest full hyper(1)-AFL containing EDTOL properly contained in ETOL?

Acknowledgement

I am grateful to Peter Asveld for many stimulating discussions. I wish to thank Sven Skyum for teaching me all about copying.

References

[1] P.R.J. Asveld, Controlled iteration grammars and full hyper-AFLs, Memorandum 114, Twente University of Technology, Holland (1976).

[2] P.R.J. Asveld, Rational, algebraic and hyper-algebraic extensions of families of languages, Memorandum 90, Twente University of Technology, Holland (1975).

[3] P.R.J. Asveld, Extensions of families of languages: Lattices of full X-AFLs, Memorandum 124, Twente University of Technology, Holland (1976).

[4] P.R.J. Asveld, Full Lindenmayer-AFLs and related language families; Memorandum 141, Twente University of Technology, Holland (1976).

[5] P.A. Christensen, Hyper-AFLs and ETOL-systems; DAIMI PB-35, University of Aarhus, Denmark (1974) (an abstract appeared in [28]).

[6] P.J. Downey, Formal languages and recursion schemes, Ph.D. Thesis, Harvard University (TR-16-74), Cambridge, MA (1974).

[7] A. Ehrenfeucht and G. Rozenberg, The equality of EOL and codings of OL languages; *Inter. J. Comput. Math.* **4** (1974) 95–104.

[8] A. Ehrenfeucht, G. Rozenberg and S. Skyum, A relationship between ETOL and EDTOL languages, *Theoretical Comput. Sci.* **1** (1976) 325–330.

[9] J. Engelfriet and S. Skyum, Copying theorems, *Inf. Proc. Lett.* **4** (1976) 157–161.

[10] M.J. Fischer, Grammars with macro-like productions, Ph.D. Thesis, Harvard University, Cambridge, MA (1968).

[11] S. Ginsburg and E.H. Spanier, Substitution in families of languages, *Inform. Sci.* **2** (1970) 83–110.

[12] S.A. Greibach, Full AFLs and nested iterated substitution, *Information and Control* **16** (1970) 7–35.

[13] S.A. Greibach, A generalization of Parikh's semilinear theorem, *Discrete Math.* **2** (1972) 347–355.

[14] J. Gruska, A characterization of context-free languages, *J. Comput. System Sci.* **5** (1971) 353–364.

[15] G.T. Herman, Closure properties of some families of languages associated with biological systems, *Information and Control* **24** (1974) 101–121.

[16] G.T. Herman and G. Rozenberg, *Developmental Systems and Languages* (North-Holland, Amsterdam, 1975).

[17] J.E. Hopcroft and J.D. Ullman, *Formal Languages and their Relation to Automata* (Addison-Wesley, Reading, MA, 1969).

[18] J. Král, A modification of a substitution theorem and some necessary and sufficient conditions for sets to be context-free, *Math. Syst. Theory* **4** (1970) 129–139.

[19] A. Lindenmayer, Developmental systems without cellular interactions, their languages and grammars, *J. Theoret. Biol.* **30** (1971) 455–484.

[20] A. Lindenmayer and G. Rozenberg, eds., *Automata, Languages and Development* (North-Holland, Amsterdam, 1976).

[21] I.P. McWirther, Substitution expressions, *J. Comput. System Sci.* **5** (1971) 629–637.

[22] M. Nielsen, G. Rozenberg, A. Salomaa and S. Skyum, Nonterminals, homomorphisms and codings in different variations of OL-systems, *Acta Informat.* **4** (1974) 87–106; **3** (1974) 357–364.

[23] G. Rozenberg, TOL systems and languages, *Information and Control* **23** (1973) 357–381.

[24] G. Rozenberg, Extension of tabled OL-systems and languages, *Inter. J. Comp. Inform. Sci.* **2** (1973) 311–336.

[25] G. Rozenberg, Theory of L systems: From the point of view of formal language theory, in: G. Rozenberg and A. Salomaa, eds., *L-Systems* (Springer-Verlag, Berlin, 1974) 1–23.

[26] G. Rozenberg, On slicing of K-iteration grammars, *Inf. Proc. Lett.* **4** (1976) 127–131.

[27] G. Rozenberg and P. Doucet, On OL-languages, *Information and Control* **19** (1971) 302–318.

[28] G. Rozenberg and A. Salomaa, eds., *L-systems*, LNCS 15 (Springer-Verlag, Berlin, 1974).

[29] A. Salomaa, Macros, iterated substitution and Lindenmayer-AFLs, DAIMI PB-18, University of Aarhus, Denmark (1973) (an abstract appeared in [28], 250–253).

[30] A. Salomaa, *Formal languages* (Academic Press, New York, 1973).

[31] S. Skyum, Decomposition theorems for various kinds of languages parallel in nature, *SIAM J. Comp.* **5** (1976) 284–296.

[32] J.W. Thatcher, Tree automata: an informal survey, in: A.V. Aho, ed., *Currents in the Theory of Computing*, (Prentice-Hall, Englewood Cliffs, NJ, 1973) 143–172.
[33] J.W. Thatcher and J.B. Wright, Generalized finite automata theory with an application to a decisionproblem of second-order logic, *Math. Syst. Theory* **2** (1968) 57–81.
[34] J. Van Leeuwen, *F*-iteration languages, Memorandum, University of California, Berkeley, CA (1973).
[35] J. Van Leeuwen, A generalization of Parikh's theorem in formal language theory, in: J. Loeckx, ed., *Automata, Languages and Programming, 2nd Colloquium*, LNCS 14 (Springer–Verlag, Berlin, 1974) 17–26.
[36] J. Van Leeuwen and D. Wood, A decomposition theorem for hyper-algebraic extensions of language families, *Theoretical Comp. Sci.* **1** (1976) 199–214.
[37] D. Wood, A note on Lindenmayer systems, Szilard languages, spectra, and equivalence, *Inter. J. Comp. Inf. Sci.* **4** (1975) 53–62.
[38] M.K. Yntema, Cap expressions for context-free languages, *Information and Control* **18** (1971) 311–318.