

S. A. GREIBACH

**The strong independence of substitution and
homomorphic replication**

RAIRO. Informatique théorique, tome 12, n° 3 (1978), p. 213-234

http://www.numdam.org/item?id=ITA_1978__12_3_213_0

© AFCET, 1978, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Informatique théorique » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

THE STRONG INDEPENDENCE OF SUBSTITUTION AND HOMOMORPHIC REPLICATION (*)

by S. A. GREIBACH ⁽¹⁾

Communicated by R. V. BOOK

Abstract. — *The operations of homomorphic replication and substitution are strongly independent for full semiAFLs in the sense that if \mathcal{L} is a full semiAFL closed under neither homomorphic replication nor substitution, then the closure of \mathcal{L} under homomorphic replication ($\hat{\mathcal{M}}_r(\mathcal{L})$) is incomparable with the closure of \mathcal{L} under substitution ($\hat{\mathcal{M}}_s(\mathcal{L})$). The least full AFL containing a full semiAFL \mathcal{L} and closed under homomorphic replication ($\hat{\mathcal{F}}_r(\mathcal{L})$) is closed under substitution if and only if either $\mathcal{L} \subseteq \hat{\mathcal{F}}_r(\text{REGL})$ (REGL is the family of regular languages) or $\mathcal{L} = \hat{\mathcal{M}}_r(\mathcal{L}) = \hat{\mathcal{F}}_r(\mathcal{L}) = \hat{\mathcal{M}}_s(\mathcal{L})$.*

1. The operation of homomorphic replication (first introduced in [13]) has been used in several recent papers, [3, 4, 5, 17, 19, 26, 27] to characterize a variety of classes of languages arising naturally in different situations—machines, grammars, string relations, complexity classes, etc. For example, the class of finite reversal checking automaton languages is the closure of the regular sets under homomorphic replication and also the closure of the regular sets under iteration of controls on linear context-free grammars [17, 19]. The class of recursively enumerable languages is the smallest class of languages containing the regular sets and closed under intersection and homomorphic replication [3].

The operation of substitution and some of its extensions and restrictions (such as nested iterated substitution [20], iterated substitution [2, 31], and insertion [21]) have also proved useful. For example, the class of derivation bounded languages is the substitution closure of the class of linear context-free languages [14].

Both substitution and homomorphic replication are syntactic operators in the sense that, if \mathcal{L} is a full semiAFL not closed under the operator, a proper hierarchy is obtained by iterating applications of the operator and if the closure

(*) Received 24 January 1978.

⁽¹⁾ Department of System Science, University of California, Los Angeles.

This work was supported in part by the National Science Foundation under Grants No. DCR74-15091, and MCS78-04725.

of a full semiAFL under the operator is not contained in some other full semiAFL, it cannot be contained in the corresponding AFL [21]. These are very useful properties in gaining strong results in a variety of situations without examining the details of particular machines or grammars, as shown in [15, 16, 17, 19, 20, 21 and 34]. For example, from the facts that the family of linear context-free languages is not closed under concatenation and that its substitution closure is the family of derivation-bounded (finite index) languages, one can conclude at once that the family of derivation-bounded languages is properly contained in the family of context-free languages [15]. Similarly, the fact that there are nonregular linear context-free languages allows one to conclude that iterating controls on linear context-free grammars, starting with the regular sets, produces a proper hierarchy [17, 24]. Perhaps the most impressive example of these techniques appears in Engelfriet's proof of the tree transducer hierarchy [34], which uses duplication (a special case of homomorphic replication) and other operators.

In this paper, we compare the closure of \mathcal{L} under homomorphic replication ($\hat{\mathcal{M}}_r(\mathcal{L})$) with its closure under substitution ($\hat{\mathcal{M}}_\sigma(\mathcal{L})$) for full semiAFLs \mathcal{L} . The two operations are strongly independent for full semiAFLs in the sense that, if \mathcal{L} is a full semiAFL closed under neither operation, $\hat{\mathcal{M}}_r(\mathcal{L})$ and $\hat{\mathcal{M}}_\sigma(\mathcal{L})$ are incomparable.

Turning to $\hat{\mathcal{F}}_r(\mathcal{L})$, the least full AFL containing \mathcal{L} and closed under homomorphic replication, we find a different situation. For the family REGL of regular languages, $\hat{\mathcal{F}}_r(\text{REGL})$ is closed under substitution. Further, for a full semiAFL \mathcal{L} , $\hat{\mathcal{F}}_r(\mathcal{L})$ is substitution closed if and only if either $\mathcal{L} \subseteq \hat{\mathcal{F}}_r(\text{REGL})$ [so $\hat{\mathcal{F}}_r(\mathcal{L}) = \hat{\mathcal{F}}_r(\text{REGL})$] or \mathcal{L} is closed under substitution and homomorphic replication. If \mathcal{L} is a full semiAFL closed under substitution but not homomorphic replication, $\hat{\mathcal{F}}_r(\mathcal{L})$ is properly contained in $\text{FINITE-VISIT}(\mathcal{L})$, the closure of \mathcal{L} under deterministic two-way finite state transductions.

The paper is organized as follows. In section 2, we establish notation and give formal definitions. Section 3 compares $\hat{\mathcal{M}}_r(\mathcal{L})$ and $\hat{\mathcal{M}}_\sigma(\mathcal{L})$. Section 4 discusses the closure of $\hat{\mathcal{F}}_r(\mathcal{L})$ under substitution. Section 5 compares $\hat{\mathcal{F}}_r(\text{REGL})$ and $\text{FINITE-VISIT}(\mathcal{L})$ by showing that a generalized Ogden's lemma holds for the former but not for the latter.

2. It is assumed that the reader is familiar with the basic concepts of automata and formal languages as found, for example, in [10] or [30]. Some of the concepts that are most important for this paper are reviewed here and notation is established.

NOTATION: For a string w , $|w|$ denotes the length of w . For a finite set S , $\#S$ denotes the number of members of S . The reversal w^R of a string w is the string

obtained by writing w in reverse order. Let $w^1 = w$, $w^{n+1} = ww^n$. For a language L , $L^R = \{w^R \mid w \text{ in } L\}$ and for a family of languages \mathcal{L} , $\mathcal{L}^R = \{L^R \mid L \text{ in } \mathcal{L}\}$.

Kleene+ is the operation which takes a language L into $L^+ = \{w_1 \dots w_n \mid n \geq 1, \text{ each } w_i \in L\}$. We use e for the empty string. Kleene* is the operation taking L into $L^* = L^+ \cup \{e\}$. Inverse homomorphism is the operation determined by a homomorphism h taking L into $h^{-1}(L) = \{w \mid h(w) \in L\}$. By homomorphism, we mean monoid homomorphism, i. e., a function $h: \Sigma^* \rightarrow \Delta^*$ such that for all $x, y \in \Sigma^*$, $h(xy) = h(x)h(y)$.

A homomorphism h is *nonerasing* if $h(w) \neq e$ for $w \neq e$. A homomorphism h is *linear erasing* on a language L if there is a $k > 0$ such that for all w in L with $|w| \geq k$, $|w| \leq k \mid h(w) \mid$. A class \mathcal{L} of languages is *closed under (nonerasing, linear erasing) homomorphism* if for every language L and any homomorphism h (that is nonerasing, linear erasing on L), $h(L) = \{h(w) \mid w \in L\}$ is in \mathcal{L} .

We shall reserve Σ with or without subscripts for finite alphabets.

Now we give the definitions and notation used for discussing semiAFLs.

DEFINITION: A *semiAFL* is a family of languages containing at least one nonempty set and closed under union, nonerasing homomorphism, inverse homomorphism, and intersection with regular sets. A *full semiAFL* is a semiAFL closed under arbitrary homomorphisms. An *AFL (full AFL)* is a semiAFL (full semiAFL) closed under concatenation and Kleene +.

For a family \mathcal{L} , we use the notation $\mathcal{M}(\mathcal{L})$ [respectively, $\hat{\mathcal{M}}(\mathcal{L})$, $\mathcal{F}(\mathcal{L})$, $\hat{\mathcal{F}}(\mathcal{L})$] for the least semiAFL (respectively, full semiAFL, AFL, full AFL) containing \mathcal{L} . If $\mathcal{L} = \{L\}$, we write $\mathcal{M}(L)$ [respectively, $\hat{\mathcal{M}}(L)$, $\mathcal{F}(L)$, $\hat{\mathcal{F}}(L)$] and call it a *principal semiAFL* (respectively, *full principal semiAFL*, *principal AFL*, *full principal AFL*). For families \mathcal{L}_1 and \mathcal{L}_2 , let

$$\mathcal{L}_1 \vee \mathcal{L}_2 = \{L_1 \cup L_2 \mid L_1 \in \mathcal{L}_1, L_2 \in \mathcal{L}_2\}.$$

DEFINITION: Let ρ be a function from $\{1, \dots, n\}$ into $\{1, R\}$ and for $1 \leq i \leq n$, let h_i be a homomorphism. The operation on languages defined by

$$\langle \rho, h_1, \dots, h_n \rangle(L) = \{(h_1(w))^{\rho(1)} \dots (h_n(w))^{\rho(n)} \mid w \in L\},$$

is a *homomorphic replication* of type ρ . It is *nonerasing* if each h_i is nonerasing. Let

$$\mathcal{L}_\rho = \{\langle \rho, h_1, \dots, h_n \rangle(L) \mid L \in \mathcal{L}, h_1, \dots, h_n \text{ homomorphisms}\}.$$

We shall add the subscript r to specify a family closed under homomorphic replication of the appropriate type. Thus, $\mathcal{M}_r(\mathcal{L})$ [respectively, $\hat{\mathcal{M}}_r(\mathcal{L})$, $\mathcal{F}_r(\mathcal{L})$, $\hat{\mathcal{F}}_r(\mathcal{L})$], is the least semiAFL (respectively, full semiAFL, AFL, full AFL)

containing \mathcal{L} and closed under nonerasing homomorphic replication. Clearly, a full semiAFL or AFL closed under nonerasing homomorphic replication is closed under homomorphic replication.

Two additional classes of operation we use are the a -transductions and the substitutions.

DEFINITION: An a -transducer is a tuple $M = (K, \Sigma, \Delta, H, q_0, F)$ where K is a finite set of states, $q_0 \in K$, $F \subseteq K$, Σ is a finite input alphabet, Δ is a finite output alphabet and H is a finite subset of $K \times \Sigma^* \times \Delta^* \times K$. An ID (Instantaneous Description) of M is any member of $K \times \Sigma^* \times \Delta^*$. If (q, uw, y) is an ID and $(q, u, v, q') \in H$, then we write $(q, uw, y) \vdash (q', w, yv)$. The relation \vdash^* among ID's is the transitive reflexive extension of \vdash .

For $w \in \Sigma^*$,

$$M(w) = \{ v \mid \exists q \in F, (q_0, w, e) \vdash^* (q, e, v) \}$$

and for a language L ,

$$M(L) = \{ v \mid \exists w \in L, v \in M(w) \}.$$

We call $M(L)$ an a -transducer mapping of L if.

$$H \subseteq K \times (\Sigma \cup \{e\}) \times (\Delta \cup \{e\}) \times K,$$

we call M 1-bounded.

Intuitively, an a -transducer is a nondeterministic one-way finite state transducer with accepting states; output is "legal" only when the machine is in an accepting state.

We shall use the fact that every full semiAFL is closed under a -transducer mapping and, more strongly, is characterized by union and a -transducer mapping [11, 12]. That is, for any family of languages \mathcal{L} , containing at least one nonempty language,

$$\hat{\mathcal{M}}(\mathcal{L}) = \{ M_1(L_1) \cup \dots \cup M_n(L_n) \mid L_1, \dots, L_n \in \mathcal{L}, M_1, \dots, M_n \text{ are } a\text{-transducers} \}$$

and for a language L ,

$$\mathcal{M}(L) = \{ M(L) \mid M \text{ is an } a\text{-transducer} \}.$$

Further, restriction to 1-bounded a -transducers causes no loss in power [12]. Properties of semiAFLs and AFLs can be found in [11, 12]; a -transducers are also described in [7].

DEFINITION: A substitution τ on a finite alphabet Σ takes each a in Σ into a language $\tau(a)$. We extend τ to words by $\tau(e) = \{e\}$ and $\tau(xy) = \tau(x)\tau(y)$ and to

languages L by $\tau(L) = \{u \mid \exists w \in L, u \in \tau(w)\}$. If $\tau(a)$ is in \mathcal{L} for each a in Σ , then τ is an \mathcal{L} -substitution. If ϵ is not in $\tau(a)$ for each a in Σ , then τ is nonerasing. For families of languages \mathcal{L}_1 and \mathcal{L}_2 , the families of languages obtained by substituting members of \mathcal{L}_2 into \mathcal{L}_1 are

$$\mathcal{L}_1 \hat{\sigma} \mathcal{L}_2 = \{ \tau(L) \mid L \in \mathcal{L}_1, \tau \text{ is an } \mathcal{L}_2\text{-substitution} \}$$

and

$$\mathcal{L}_1 \sigma \mathcal{L}_2 = \{ \tau(L) \mid L \in \mathcal{L}_1, \tau \text{ is a nonerasing } \mathcal{L}_2\text{-substitution} \}.$$

If $\mathcal{L} \hat{\sigma} \mathcal{L}_1 \subseteq \mathcal{L}$ ($\mathcal{L} \hat{\sigma} \mathcal{L}_1 \subseteq \mathcal{L}$), then \mathcal{L} is closed under substitution (nonerasing substitution) by \mathcal{L}_1 ; if $\mathcal{L}_1 \hat{\sigma} \mathcal{L} \subseteq \mathcal{L}$ ($\mathcal{L}_1 \sigma \mathcal{L} \subseteq \mathcal{L}$), then \mathcal{L} is closed under substitution (nonerasing substitution) into \mathcal{L}_1 . If $\mathcal{L} \hat{\sigma} \mathcal{L} \subseteq \mathcal{L}$ ($\mathcal{L} \sigma \mathcal{L} \subseteq \mathcal{L}$), then \mathcal{L} is closed under substitution (nonerasing substitution).

We add the subscript σ to indicate closure under nonerasing substitution. Thus $\mathcal{M}_\sigma(\mathcal{L})$ [respectively, $\hat{\mathcal{M}}(\mathcal{L})$, $\mathcal{F}_\sigma(\mathcal{L})$, $\hat{\mathcal{F}}_\sigma(\mathcal{L})$] is the least semiAFL (respectively, full semiAFL, AFL, full AFL) containing \mathcal{L} and closed under nonerasing substitution. A full semiAFL closed under nonerasing substitution is closed under substitution [22]. Full semiAFLs are closed under regular substitution (substitution by regular languages) while full AFLs are also closed under substitution into regular sets. A semiAFL closed under nonerasing substitution is an AFL, so $\mathcal{F}_\sigma(\mathcal{L}) = \mathcal{M}_\sigma(\mathcal{L})$ and we normally use just $\mathcal{M}_\sigma(\mathcal{L})$ or $\hat{\mathcal{M}}_\sigma(\mathcal{L})$.

We let REGL denote the family of regular languages and CF the family of context-free languages. Two useful facts about substitutions are that substitution (and nonerasing substitution) are associative on semiAFLs [e. g., $\mathcal{L}_1 \hat{\sigma} (\mathcal{L}_2 \hat{\sigma} \mathcal{L}_3) = (\mathcal{L}_1 \hat{\sigma} \mathcal{L}_2) \hat{\sigma} \mathcal{L}_3$] and that for any family of languages \mathcal{L} , $\hat{\mathcal{F}}(\mathcal{L}) = \text{REGL} \hat{\sigma} \hat{\mathcal{M}}(\mathcal{L})$ [11, 12, and 22].

3. In this section, we establish the strong independence of substitution and homomorphic replication using “syntactic lemmas” akin to those in references [15] through [21], which state that languages of certain forms can only be built up in certain ways.

The first syntactic lemma echoes example 3.1 of [21], the example on p. 27 of [20] and lemma 4.4 of [18]. It is similar to the “copying” theorems in [2, 8 and 9]; a close relative appears in [33].

LEMMA 3.1: *Let \mathcal{L} be a family of languages closed under concatenation by unit sets (i. e., sets of size 1), intersection with sets of the form Σ^* for Σ a finite vocabulary, and under union and (nonerasing) homomorphism. Let $L \subseteq \Sigma^*$ be a language with the following property:*

(★) *if xyz is in L then either $\#(L \cap \Sigma^* y \Sigma^*) \leq 1$ or else $\#(L \cap x \Sigma^* z) \leq 1$.*

If $L = \tau(L_1)$ for L_1 in \mathcal{L} and τ is an \mathcal{L} -substitution (a nonerasing \mathcal{L} -substitution), then L is in \mathcal{L} .

Proof: Let $L_1 \subseteq \Sigma^*$, let $\Sigma_1 = \{a \text{ in } \Sigma \mid \# \tau(a) = 1\}$ and let $\Sigma_2 = \Sigma - \Sigma_1$. The language $L_2 = \tau(L_1 \cap \Sigma_1^*)$ is in \mathcal{L} . Suppose uav is in L_1 with $\# \tau(a) \geq 2$ and $\tau(u) \neq \emptyset \neq \tau(v)$. Let x be in $\tau(u)$, y in $\tau(a)$ and z in $\tau(v)$. Then xyz is in L . If $\#(L \cap \Sigma^* y \Sigma^*) \geq 2$, then $\#(L \cap x \Sigma^* z) = 1$, so $\tau(a) = \{y\}$, a contradiction. Hence there are unique x_a and z_a such that $x_a y z_a$ is in L . Let

$$\Sigma_3 = \{a \text{ in } \Sigma_2 \mid (L_1 \cap \Sigma^* a \Sigma^*) \neq \emptyset\}$$

and let

$$L_3 = \bigcup_{a \text{ in } \Sigma_3} x_a \tau(a) y_a.$$

Then L_3 is in \mathcal{L} and so $L = L_2 \cup L_3$ is in \mathcal{L} . \square

Lemma 3.1 enables us to say that substitutions cannot "help" in building up certain types of homomorphic replications.

LEMMA 3.2: Let \mathcal{L} be a semiAFL. Let $A \subseteq \Sigma^*$, $A \in \mathcal{L}$, and let c be a symbol not in Σ . Then for any $k \geq 3$, t_1, \dots, t_k in $\{1, R\}$ the language $L = \{w^{t_1} c \dots c w^{t_k} \mid w \text{ in } A\}$ is in $\mathcal{M}_\sigma(\mathcal{L})$ if and only if it is in \mathcal{L} , and if \mathcal{L} is a full semiAFL, L is in $\hat{\mathcal{M}}_\sigma(\mathcal{L})$ if and only if it is in \mathcal{L} .

Proof: The language L has property (*) of lemma 3.1, since for any xyz in L either y contains at least two c 's and uniquely determines x and z or else x contains at least one c and uniquely determines yz or z contains at least one c and uniquely determines xy . For a semiAFL \mathcal{L} , $\mathcal{M}_\sigma(\mathcal{L})$ can be obtained from \mathcal{L} by repeated nonerasing \mathcal{L} -substitutions and $\hat{\mathcal{M}}_\sigma(\mathcal{L})$ by repeated \mathcal{L} -substitutions. But lemma 3.1 tells us that if L is not in \mathcal{L} , this will not suffice to obtain L in $\mathcal{M}_\sigma(\mathcal{L})$ or, if \mathcal{L} is already a full semiAFL, to obtain L in $\hat{\mathcal{M}}_\sigma(\mathcal{L})$. \square

THEOREM 3.3: Let \mathcal{L} and \mathcal{L}_1 be semiAFLs with \mathcal{L} closed under linear erasing homomorphism

$$(1) \quad \mathcal{M}_r(\mathcal{L}_1) \subseteq \mathcal{M}_\sigma(\mathcal{L}) \text{ if and only if } \mathcal{M}_r(\mathcal{L}_1) \subseteq \mathcal{L}$$

and

$$\hat{\mathcal{M}}_r(\mathcal{L}_1) \subseteq \hat{\mathcal{M}}_\sigma(\mathcal{L}) \text{ if and only if } \hat{\mathcal{M}}_r(\mathcal{L}_1) \subseteq \mathcal{L}.$$

(2) If \mathcal{L} is not closed under nonerasing homomorphic replication, then

$$\mathcal{M}_r(\mathcal{L}) - \mathcal{M}_\sigma(\mathcal{L}) \neq \emptyset$$

and if \mathcal{L} is also a full semiAFL,

$$\hat{\mathcal{M}}_r(\mathcal{L}) - \hat{\mathcal{M}}_\sigma(\mathcal{L}) \neq \emptyset.$$

(3) $\mathcal{M}_\sigma(\mathcal{L})$ is closed under nonerasing homomorphic replication if and only if $\mathcal{L} = \mathcal{M}_r(\mathcal{L}) = \mathcal{M}_\sigma(\mathcal{L})$ and if \mathcal{L} is a full semiAFL, $\hat{\mathcal{M}}_\sigma(\mathcal{L})$ is closed under homomorphic replication if and only if $\mathcal{L} = \hat{\mathcal{M}}_r(\mathcal{L}) = \hat{\mathcal{M}}_\sigma(\mathcal{L})$.

Proof: Suppose $\mathcal{L}_2 = \mathcal{M}_r(\mathcal{L}_1)$ is contained in $\mathcal{M}_\sigma(\mathcal{L})$. Let A be any member of \mathcal{L}_2 and c a new symbol. Since \mathcal{L}_2 is closed under nonerasing homomorphic replication, $\{wcw^Rcw \mid w \text{ in } A\}$ is in $\mathcal{M}_\sigma(\mathcal{L})$ and hence, by lemma 3.2, in \mathcal{L} . Since \mathcal{L} is closed under linear erasing homomorphism, A is in \mathcal{L} . A similar argument applies to $\hat{\mathcal{M}}_r(\mathcal{L}_1)$. This establishes (1). Statement (2) follows from (1), taking $\mathcal{L}_1 = \mathcal{L}$, and using the fact that if \mathcal{L} is a full semiAFL, $\mathcal{M}_\sigma(\mathcal{L}) = \hat{\mathcal{M}}_\sigma(\mathcal{L})$ [22].

From (2) we know that if $\mathcal{L} \neq \mathcal{M}_r(\mathcal{L})$, $\mathcal{M}_\sigma(\mathcal{L})$ is not closed under nonerasing homomorphic replication. A full semiAFL is closed under homomorphic replication if and only if it is closed under nonerasing homomorphic replication, so if \mathcal{L} is a full semiAFL with $\mathcal{L} \neq \hat{\mathcal{M}}_r(\mathcal{L})$, $\hat{\mathcal{M}}_\sigma(\mathcal{L})$ is not closed under homomorphic replication. If $\mathcal{L} \neq \mathcal{M}_\sigma(\mathcal{L})$, then $\mathcal{M}_r(\mathcal{M}_\sigma(\mathcal{L}))$ is not contained in \mathcal{L} [since $\mathcal{M}_\sigma(\mathcal{L}) \subseteq \mathcal{M}_r(\mathcal{M}_\sigma(\mathcal{L}))$], and hence not in $\mathcal{M}_\sigma(\mathcal{L})$ by (1). This establishes (3). \square

The analogous result for nonerasing substitution is established only for full semiAFLs, using the fact that if \mathcal{L} is a full semiAFL not closed under homomorphic replication, $\hat{\mathcal{M}}_r(\mathcal{L})$ is not an AFL.

THEOREM 3.4: *Let \mathcal{L} be a full semiAFL and let \mathcal{L}_1 be a semiAFL. Then $\hat{\mathcal{M}}_r(\mathcal{L})$ is closed under nonerasing substitution into \mathcal{L}_1 if and only if $\mathcal{L} = \hat{\mathcal{M}}_r(\mathcal{L}) = \hat{\mathcal{F}}_r(\mathcal{L})$ and \mathcal{L} is closed under nonerasing substitution into \mathcal{L}_1 .*

Proof: Now

$$\begin{aligned} \mathcal{L}_1 \sigma \hat{\mathcal{M}}_r(\mathcal{L}) &= (\mathcal{L}_1 \sigma \text{REGL}) \sigma \hat{\mathcal{M}}_r(\mathcal{L}) \\ &= \mathcal{L}_1 \sigma (\text{REGL} \sigma \hat{\mathcal{M}}_r(\mathcal{L})) = \mathcal{L}_1 \sigma \hat{\mathcal{F}}_r(\mathcal{M}_r(\mathcal{L})), \end{aligned}$$

so if $\hat{\mathcal{M}}_r(\mathcal{L})$ is closed under nonerasing substitution into \mathcal{L}_1 , then it is an AFL and equal to $\hat{\mathcal{F}}_r(\mathcal{L})$. But if $\mathcal{L} \neq \mathcal{M}_r(\mathcal{L})$, it is not an AFL [17, 19, 21]. \square

Putting together theorems 3.3 and 3.4, we obtain our strong independence theorem.

THEOREM 3.5: *Substitution and homomorphic replication are strongly independent operations on full semiAFLs in the sense that if \mathcal{L} is a full semiAFL not closed under substitution or homomorphic replication, then $\hat{\mathcal{M}}_\sigma(\mathcal{L})$ and $\hat{\mathcal{M}}_r(\mathcal{L})$ are incomparable.*

COROLLARY: *The family of derivation bounded languages is incomparable with the family of finite reversal checking automaton languages.*

If we try to extend theorem 3.5 to compare $\hat{\mathcal{F}}_r(\mathcal{L})$ and $\hat{\mathcal{F}}_\sigma(\mathcal{L})$, a different situation emerges. For any semiAFL \mathcal{L} , $\hat{\mathcal{M}}_\sigma(\mathcal{L})$ is an AFL while for a full

semiAFL \mathcal{L} , $\hat{\mathcal{M}}_r(\mathcal{L})$ cannot be an AFL unless $\mathcal{L} = \hat{\mathcal{M}}_r(\mathcal{L}) = \hat{\mathcal{F}}_r(\mathcal{L})$. If $\mathcal{L} \neq \hat{\mathcal{M}}_r(\mathcal{L})$, then not only is $\hat{\mathcal{M}}_r(\mathcal{L})$ not an AFL but $\hat{\mathcal{F}}_r(\mathcal{M}_r(\mathcal{L}))$ is not closed under homomorphic replication [17, 19, 21]. We can build $\hat{\mathcal{F}}_r(\mathcal{L})$ from \mathcal{L} by alternative applications of homomorphic replication and AFL closure as

$$\hat{\mathcal{F}}_r(\mathcal{L}) = \bigcup_{i \geq 0} \mathcal{L}_i \quad \text{for } \mathcal{L}_0 = \mathcal{L}$$

and

$$\mathcal{L}_{i+1} = \begin{cases} \hat{\mathcal{M}}_r(\mathcal{L}_i) & i \text{ odd,} \\ \mathcal{F}(\mathcal{L}_i) & i \text{ even} \end{cases}$$

and if $\mathcal{L}_0 \neq \mathcal{L}_1$, then $\mathcal{L}_i \neq \mathcal{L}_{i+1}$ for all i .

Examining the closure of $\hat{\mathcal{F}}_r(\mathcal{L})$ under substitution, a curious situation emerges. For $\hat{\mathcal{F}}_r(\text{REGL})$ is closed under substitution and this is basically the only situation in which substitution closure holds unless $\mathcal{L} = \hat{\mathcal{M}}_r(\mathcal{L}) = \hat{\mathcal{F}}_r(\mathcal{L})$, and \mathcal{L} is substitution closed, for example, if \mathcal{L} is the family of recursively enumerable languages which is closed under substitution and homomorphic replication [3]. We discuss this in the next section.

4. We first establish the closure of $\hat{\mathcal{F}}_r(\text{REGL})$ under substitution. The only construction needed already appears in [13], in the proof that \mathcal{L}_p is a full semiAFL whenever \mathcal{L} is a full semiAFL. We excerpt the basic idea as lemma 4.1.

LEMMA 4.1: *Let \mathcal{L}_1 and \mathcal{L}_2 be full semiAFLs and let $\rho: \{1, \dots, k\} \rightarrow \{1, R\}$. Then*

$$(\mathcal{L}_1)_\rho \hat{\sigma} \mathcal{L}_2 \subseteq (\mathcal{L}_1 \hat{\sigma} \mathcal{L}_2)_\rho.$$

Proof: Part (d) of the proof of theorem 3.1 of [13] shows how to express $\tau(\langle \rho, h_1, \dots, h_k \rangle(L))$ as $\langle \rho, g_1, \dots, g_k \rangle(\bar{\tau}(L))$ where τ and $\bar{\tau}$ are regular substitutions. The only property of REGL used is closure under inverse homomorphism, intersection with regular sets and concatenation. Since $(\mathcal{L}_1)_\rho \hat{\sigma} \mathcal{L}_2 = (\mathcal{L}_1)_\rho \hat{\sigma} \hat{\mathcal{F}}_r(\mathcal{L}_2)$, we may as well assume that \mathcal{L}_2 is an AFL and so closed under concatenation. Hence any \mathcal{L}_2 -substitution into a homomorphic replication of a language L in \mathcal{L}_1 can be expressed as the homomorphic replication of a language obtained by some \mathcal{L}_2 -substitution into L , whence the desired result. \square

THEOREM 4.2: *Let \mathcal{L} , \mathcal{L}_1 and \mathcal{L}_2 be full semiAFLs such that $\mathcal{L}_1 \hat{\sigma} \mathcal{L}_2 \subseteq \hat{\mathcal{F}}_r(\mathcal{L})$. Then $\hat{\mathcal{F}}_r(\mathcal{L}_1) \hat{\sigma} \mathcal{L}_2 \subseteq \hat{\mathcal{F}}_r(\mathcal{L})$. In particular, $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution into $\hat{\mathcal{F}}_r(\text{REGL})$.*

Proof: Let $\mathcal{A}_0 = \mathcal{L}_1$ and for $i \geq 0$, let $\mathcal{A}_{i+1} = \hat{\mathcal{F}}_r(\hat{\mathcal{M}}_r(\mathcal{L})) = \text{REGL} \hat{\sigma} \hat{\mathcal{M}}_r(\mathcal{L}_i)$.

Then $\hat{\mathcal{F}}_r(\mathcal{L}_1) = \bigcup_{i \geq 0} \mathcal{L}_i$.

We use lemma 4.1 to establish by induction on i , that $\mathcal{A}_i \hat{\circ} \mathcal{L}_2 \subseteq \hat{\mathcal{F}}_r(\mathcal{L})$ and hence $\hat{\mathcal{F}}_r(\mathcal{L}_1) \hat{\circ} \mathcal{L}_2 \subseteq \hat{\mathcal{F}}_r(\mathcal{L})$. The basis step, $i=0$, is true by hypothesis. Now assume that we have shown the inclusion to hold for $i \geq 0$. Then

$$\begin{aligned} \mathcal{A}_{i+1} \hat{\circ} \mathcal{L}_2 &= (\text{REGL} \hat{\circ} \hat{\mathcal{M}}_r(\mathcal{A}_i)) \hat{\circ} \mathcal{L}_2 = \text{REGL} \hat{\circ} (\hat{\mathcal{M}}_r(\mathcal{A}_i) \hat{\circ} \mathcal{L}_2), \\ &\subseteq \text{REGL} \hat{\circ} (\hat{\mathcal{M}}_r(\mathcal{A}_i \hat{\circ} \mathcal{L}_2)) \text{ by lemma 4.1,} \\ &\subseteq \text{REGL} \hat{\circ} \hat{\mathcal{F}}_r(\mathcal{L}) \text{ by the induction hypothesis,} \\ &\subseteq \hat{\mathcal{F}}_r(\mathcal{L}). \end{aligned}$$

If we take $\mathcal{L}_1 = \text{REGL}$ and $\mathcal{L}_2 = \hat{\mathcal{F}}_r(\mathcal{L})$, we can conclude that $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution into $\hat{\mathcal{F}}_r(\text{REGL})$. \square

COROLLARY: $\hat{\mathcal{F}}_r(\text{REGL})$ is closed under substitution.

Now we wish to show that theorem 4.1 is the best result of its kind possible in the sense that, for full semiAFLs, $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution into \mathcal{L}_1 if and only if either \mathcal{L}_1 is contained in $\hat{\mathcal{F}}_r(\text{REGL})$ or \mathcal{L} is itself already an AFL closed under homomorphic replication and under substitution into \mathcal{L}_1 . Our approach is to establish a dichotomizing syntactic lemma which says that, if a particular type of substitution is in $\hat{\mathcal{F}}_r(\mathcal{L})$, either one language is in $\hat{\mathcal{F}}_r(\text{REGL})$ or another is in $\mathcal{L} \vee \mathcal{L}^R$. This requires some way of explicitly expressing members of $\hat{\mathcal{F}}_r(\mathcal{L})$ in terms of members of \mathcal{L} .

Unfortunately, we do not know of an attractive characterization of $\hat{\mathcal{F}}_r(\mathcal{L})$. One can use a complex expression scheme generalizing the $\langle \rho, h_1, \dots, h_n \rangle(L)$ formulation for $\hat{\mathcal{F}}_r(\mathcal{L})$ or specialized machine formulations – for example, a nested restriction on the finite visit \mathcal{L} -based automata [19] or adding to the finite reversal \mathcal{L} -based automata a finite number of single reversal tapes in a nested fashion. We shall define a family of special language operators $v_{k,s}$ such that all members of $\hat{\mathcal{F}}_r(\mathcal{L})$ can be obtained as a -transducer images of $v_{k,s}(L)$ for L in \mathcal{L} . The idea is to use brackets to nest alternate replications and applications of Kleene +.

DEFINITION: Let $L \subseteq \Sigma^+$ be a language and let $[1, 1], \dots, [k, k]$ be $2k$ reserved symbols. For $s \geq 1$, let $v_{0,s}(L) = L$ and for $i \geq 0$, let

$$v'_{i+1,s}(L) = \{ [i+1] ([i+1] w_{i+1}) [i+1] w^R_{i+1}]_{i+1}^s \mid w \text{ in } v_{i,s}(L) \}$$

and

$$v_{i+1,s}(L) = (v'_{i+1,s}(L))^+.$$

LEMMA 4.3: Let \mathcal{L} be a full semiAFL. Let $\mathcal{L}_0 = \mathcal{L} = \mathcal{G}_0$ and for $i \geq 0$ let

$$\mathcal{L}_{i+1} = \hat{\mathcal{F}}_r(\hat{\mathcal{M}}_r(\mathcal{L}_i))$$

and

$$\mathcal{G}_{i+1} = \hat{\mathcal{M}}(\{v_{i+1,s}(L) \mid s \geq 1, L \text{ in } \mathcal{L}\}).$$

The for all $i \geq 0$,

$$\mathcal{L}_i = \mathcal{G}_i.$$

Proof: The proof proceeds by induction on i . The basis step $i=0$ is obvious. Assume that the lemma is true for some $i \geq 0$. Clearly $v'_{i+1,s}(L)$ is in $\hat{\mathcal{M}}_r(v_{i,s}(L))$ and $v_{i+1,s}(L)$ is in $\hat{\mathcal{F}}(v'_{i+1,s}(L))$. By the induction hypothesis, $v_{i,s}(L)$ is in \mathcal{L}_i for L in \mathcal{L} , and hence $v_{i+1,s}(L)$ is in \mathcal{L}_{i+1} , whence $\mathcal{G}_{i+1} \subseteq \mathcal{L}_{i+1}$.

The outer brackets in $v'_{i+1,s}(L)$ serve as endmarkers, so $\hat{\mathcal{F}}(v'_{i+1,s}(L)) = \hat{\mathcal{M}}(v_{i+1,s}(L))$ for any L [12]. Hence to show that $\mathcal{L}_{i+1} \subseteq \mathcal{G}_{i+1}$, it suffices to show that any language in $\mathcal{M}_r(\mathcal{L}_i)$ can be expressed as an a -transducer mapping of $v'_{i+1,s}(L)$ for some L in \mathcal{L} and $s \geq 1$.

Consider $\mathcal{M}_r(\mathcal{L}_i)$. It suffices to consider a language of the form $L_1 = \langle \rho, h_1, \dots, h_{2n} \rangle (L_2)$ for L_2 in \mathcal{L}_i , the h_j non-length-increasing homomorphisms and $\rho(j) = 1$ if and only if j is odd [17, 19, 21]. By the induction hypothesis, $L_2 = M_1(v_{i,s}(L_3))$ for L_3 in \mathcal{L} , $s \geq 1$ and an a -transducer $M_1 = (K_1, \Sigma_i, \Delta, H_1, q_0, F_1)$ where $\Sigma_i = \Sigma \cup \{[j, j] \mid 1 \leq j \leq i\}$ for a finite vocabulary Σ not including any bracket symbol. We can assume that $n \geq s$ (by adding, if needed, new ww^R and new homomorphisms which only erase) and that M_1 is 1-bounded [7, 12], and $\#F_1 = 1$. Further, since words in $v_{i,s}(L_3)$ contain at most $4(2^i - 1)$ bracket symbols in a row, by using the states to rearrange transitions we can assume that M_1 gives empty output on bracket symbols (i. e., H_1 contains no transitions $(p, [j, w, q)$ or $(p, j], w, q)$ with $w \neq e$), has no e input rules $((p, e, w, q))$ initially, terminally or in between bracket symbols and is deterministic on bracket symbols in the sense that for b in $\Sigma_i - \Sigma$ and q in K_1 there is at most one p in K_1 with (q, b, e, p) in H_1 and at most one p' with (p', b, e, q) in H_1 . We shall alter L_3 and M_1 to obtain an a -transducer deterministic everywhere, so that we can be sure that each h_j is applied to the same w .

We want to construct an a -transducer M_2 such that

$$L_1 = \langle \rho, h_1, \dots, h_{2n} \rangle (M_1(v_{i,s}(L_3))) = M_2(v'_{i+1,n}(L))$$

for some language L in \mathcal{L} . We need some auxiliary definitions. Let π_i be the projection on the i th coordinate of a tuple (i. e., $\pi_i(a_1, \dots, a_m) = a_i$ for $m \geq i$). For a in $\Sigma \cup \{e\}$, let Γ_a be the collection of all possible $(2s)^i$ -tuples of members of $H_a = \{(q, a, u, q') \mid (q, a, u, q') \text{ in } H_1\}$; for $a = e$ we assume that H_1 contains all quadruples (q, e, e, q) for q in K_1 .

Let g be the homomorphism defined by $g(\gamma) = a$ for γ in Γ_a , a in $\Sigma \cup \{e\}$. Let $L = g^{-1}(L_3)$. Clearly L is in \mathcal{L} . What we have done is to encode each symbol of a

word in L_3 by some guess as to the quadruples applied by M_1 in each of its $(2s)^i$ visits to that symbol during a successful computation of M_1 on that word; some choices may be incorrect and will cause M_2 to block. Our special assumptions on M_1 ensure that we do not have to specify M_1 's actions on brackets and so brackets can remain brackets. The new a -transducer M_2 will act on $v'_{i+1, n}(L)$.

The a -transducer M_2 will have state set $K_1 \times (\{1, 2, \dots, 2n\})^{i+1}$. In a state $(q_1, j_1, \dots, j_i, j_{i+1})$, q indicates the state of M_1 being simulated and j_{i+1} the replicate $h_{j_{i+1}}(w)$ currently under construction. The integer $j_i - 1$ is, roughly speaking, the number of $[\dots]$ subwords read since the last occurrence of $_{i+1}$ or a reset $(_i)^2$ ($(_i)^2$ (corresponding to an application of Kleene +), and (j_1, \dots, j_i) gives base $2s$ the visit of M_1 to the symbol in L_3 currently simulated. Machine M_2 must reconstruct w $2n$ times and output $h_{j_{i+1}}(w)$ or $(h_{j_{i+1}}(w))^R$. Further complications occur because if $n \neq s$ there are "dummy" subwords to skip and for the reversed replicates computations of M_1 must be simulated backwards.

We define the transition set of M_2 , H_2 , as follows, for M_2 acting on input α :

(1) Transitions to simulate M_1 on subwords of α which are in L_3 .

Let $1 \leq j_t \leq 2s$ for $1 \leq t \leq i$, let d be in Γ_a for a in $\Sigma \cup \{e\}$ and let $r = 1 + \sum_{t=1}^i (j_t - 1)^{t-1}$, $\gamma = \pi_r(d)$ and $1 \leq j_{i+1} \leq 2n$. The integer r indicates the visit of M_1 to the symbol a encoded by d , while j_{i+1} indicates the current replicate simulated. For j_{i+1} odd, the replicate is of the form $h_{j_{i+1}}(w)$, so if M_2 is in the state indicated by the r th coordinate of d , it outputs $h_{j_{i+1}}(\pi_3(\gamma))$ and changes to the indicated next state. For j_{i+1} even, the replicate is of the form $h_{j_{i+1}}(w^R)$, so the actions of M_1 must be reversed. If $a = e$, M_2 simulates either a "real" action of M_1 on e input or else a dummy (q, e, e, q) transition:

(A) For j_{i+1} odd, H_2 contains

$$((\pi_1(\gamma), j_1, \dots, j_{i+1}), d, h_{j_{i+1}}(\pi_3(\gamma)), (\pi_4(\gamma), j_1, \dots, j_{i+1})).$$

(B) For j_{i+1} even, H_2 contains

$$((\pi_4(\gamma), j_1, \dots, j_{i+1}), d, h_{j_{i+1}}(\pi_3(\gamma)), (\pi_1(\gamma), j_1, \dots, j_{i+1})).$$

If $n \neq s$, there are extra replicates of w in L which do not correspond to actions of M_1 on $g(w)$ in L_3 . Hence M_2 must skip such occurrences, using the next set of rules.

(2) Transitions to skip extra subwords.

If j_t is greater than $2s$ for any $t \neq i+1$, then H_2 contains for all q in K , a in $\Sigma \cup \{e\}$, d in Γ_a , the transition

$$((q, j_1, \dots, j_{i+1}), d, e, (q, j_1, \dots, j_{i+1})).$$

Now we define the transitions for reading brackets. Initially, M_2 must read a bracket sequence from $[_{i+1}$ to $]_1$ and simulate the initial transitions of M_1 while, at the end, M_2 is reading a reversed word and so simulates the initial transitions of M_1 in reverse order.

(3) Initial and Final Transitions. Let

$$w = ([_i]^2 \dots ([_1])^2 \quad \text{and suppose } (q_0, w, e) \underset{M_1}{\vdash}^* (q, e, e).$$

Then H_2 contains the transitions

$$((q_0, 1, \dots, 1), [_{i+1} [_{i+1} w, e, (q, 1, \dots, 1))$$

and

$$((q, 2n, \dots, 2n), (i+1])^2 (i)^2 \dots (1)^2, e, (q_0, 2n, \dots, 2n)).$$

We must also reinitialize the simulation of M_1 whenever M_2 passes from the j_{i+1} -st replicate to the next one, marked by reading the bracket sequence $]_{i+1} [_{i+1}$. This is handled by the next sequence of rules.

(4) Transitions to reinitialize the simulation of M_1 . Let

$$w_1 = (1])^2 \dots (i])^2 \quad \text{and} \quad w_2 = ([_i]^2 \dots ([_1])^2$$

and suppose $(p, w_1, e) \underset{M_1}{\vdash}^* (f, e, e)$ for f in F and $(q_0, w_2, e) \underset{M_1}{\vdash}^* (q, e, e)$.

Let $1 \leq j_{i+1} \leq 2n - 1$:

(A) If j_{i+1} is odd, then H_2 contains

$$((p, 2n, \dots, 2n, j_{i+1}), w_1]_{i+1} [_{i+1} w_2, e, (p, 2n, \dots, 2n, j_{i+1} + 1)).$$

(B) If j_{i+1} is even, then H_2 contains

$$((q, 1, \dots, 1, j_{i+1}), w_1]_{i+1} [_{i+1} w_2, e, (q, 1, \dots, 1, j_{i+1} + 1)).$$

Finally, we need the transitions to simulate the action of M_1 on strings of bracket symbols which are not initial or final. These strings are of two kinds, marking transfer from one subword y of α in some $v'_{t,n}(L_3)$ either into another subword of α in $v'_{t,n}(L_3)$ (corresponding to an application of Kleene $+$) or into a replicate of y . The first kind [represented in rule set (5) below] resets all j_m for $m \leq t$ while the second kind [rule set (6) below] also increases j_{t+1} .

(5) Transitions to simulate M_1 on bracket strings within $v_{t,n}(L_3)$.

Let $1 \leq t \leq i$, and let $1 \leq j_m \leq 2n$ for $t + 1 \leq m \leq 2n$.

Let $w = (1])^2 \dots (i])^2 ([_i]^2 \dots ([_1])^2$ and suppose that

$$(p, w, e) \underset{M_1}{\vdash}^* (q, e, e).$$

(A) For j_{i+1} odd, H_2 contains

$$((p, 2n, \dots, 2n, j_{i+1}, \dots, j_{i+1}), w, e, (q, 1, \dots, 1, j_{i+1}, \dots, j_{i+1})).$$

and

(B) For j_{i+1} even, H_2 contains

$$((q, 1, \dots, 1, j_{i+1}, \dots, j_{i+1}), w, e, (p, 2n, \dots, 2n, j_{i+1}, \dots, j_{i+1}))$$

(6) Transitions to simulate M_1 on bracket strings within $v'_{t+1, n}(L_3)$.

Let $0 \leq t \leq i-1$ and $w = (1)^2 \dots (i)^2_{t+1} [_{t+1} ((i)^2 \dots (1)^2 \text{ (if } t=0, w = 1)] [1]$.

Suppose that

$$(p, w, e) \xrightarrow[M_1]{*} (q, e, e)$$

and $1 \leq j_{t+1} \leq 2n-1$:

(A) For j_{i+1} odd, H_2 contains

$$((p, 2n, \dots, 2n, j_{i+1}, \dots, j_{i+1}), w, e, (q, 1, \dots, 1, j_{i+1}+1, \dots, j_{i+1}))$$

(B) For j_{i+1} even, H_2 contains

$$((q, 1, \dots, 1, j_{i+1}+1, j_{i+2}, \dots, j_{i+1}), w, e, (p, 2n, \dots, 2n, j_{i+1}, \dots, j_{i+1})).$$

This completes the construction. It can be shown that $L_1 = M_2(v'_{i+1, s}(L))$. \square

REMARK: Theorem 3.2 pf [13] asserts that if \mathcal{L} is a full principal AFL, then $\mathcal{F}(\mathcal{L}_p)$ is a full principal AFL. Lemma 4.3 can be considered a generalization, saying that a family of generators of \mathcal{L} correspond in a uniform way to a family of generators of $\hat{\mathcal{F}}_r(\mathcal{L})$. The extra complication in the proof is needed to have one set of brackets suffice, and use languages in \mathcal{L} as building blocks.

Now we establish our lemma governing the existence of certain types of substitution languages in $\hat{\mathcal{F}}_r(\mathcal{L})$.

DEFINITION: For languages L_1 and L_2 , $L_1 \subseteq \Sigma_1^*$, let

$$\begin{aligned} \tau(L_1, L_2) = & \{ a_1 w_1 \dots a_n w_n \mid a_1 \dots a_n \text{ in } L_1, \text{ each } a_i \text{ in } \Sigma_1 \text{ and } w_i \text{ in } L_2 \} \\ & \cup \{ e \mid e \text{ is in } L_1 \}. \end{aligned}$$

Clearly, $\tau(L_1, L_2) = \bar{\tau}(L_1)$ for the substitution $\bar{\tau}(a) = aL_2$.

LEMMA 4.4: Let $L_1 \subseteq \Sigma_1^+$, $L_2 \subseteq \Sigma_2^+$, $\Sigma_1 \cap \Sigma_2 = \emptyset$. Let \mathcal{L} be a full semiAFL. If $\tau(L_1, L_2)$ is in $\hat{\mathcal{F}}_r(\mathcal{L})$, either L_1 is in $\hat{\mathcal{F}}_r(\text{REGL})$ or L_2 is in $\mathcal{L} \vee \mathcal{L}^R$.

Proof: Let $L = \tau(L_1, L_2)$ be in $\hat{\mathcal{F}}_r(\mathcal{L})$. By lemma 4.4, there are $k, s \geq 1$, an a -transducer M and a language \bar{L} in \mathcal{L} such that $L = M(v_{k,s}(\bar{L}))$.

We shall now define from M and \bar{L} two languages \bar{L}_1 and \bar{L}_2 such that \bar{L}_1 is in $\mathcal{F}_r(\text{REGL})$, \bar{L}_2 is in $\mathcal{L} \vee \mathcal{L}^R$, and $\bar{L}_i \subseteq L_i, i = 1, 2$. Then we shall show that for words of the form $\mu(a_1 \dots a_n, w) = a_1 w \dots a_n w$ in L with $n \geq 2$, each a_i in Σ_1 and w in L_2 , either $a_1 \dots a_n$ is in \bar{L}_1 or w is in \bar{L}_2 . Finally we shall be able to conclude that either $L_1 = \bar{L}_1$ or $L_2 = \bar{L}_2$.

Let $\bar{L} \subseteq \Delta^*$ and let $\Delta_k = \Delta \cup \{[j, j] \mid 1 \leq j \leq k\}$. There are non-length-increasing homomorphisms g and h and a regular set R such that $L = g(h^{-1}(v_{k,s}(\bar{L})) \cap R)$ [11, 12]. Let $h: \Gamma^* \rightarrow \Delta_k^*$. Let

$$B_1 = \{b \text{ in } \Gamma \mid h(b) = [1]\} \quad \text{and} \quad B_2 = \{b \text{ in } \Delta \mid h(b) = [1]\}.$$

We can assume that $B_1 \neq \emptyset \neq B_2$ or else L would be empty. Since R is regular, there is a congruence relation on Γ^* with congruence classes C_1, \dots, C_n such that R is the union of some of the C_i [28]. Let $A_{i,j} = C_i \cap (C_j)^R$. For t in $\{1, R\}$, say that (i, j, y) has property (\star, t) if y^t is in $h^{-1}(\bar{L}) \cap A_{i,j}$ and there are u, v in Γ^* , b_1 in B_1 , b_2 in B_2 such that $w = ub_1 y b_2 v$ is in R and $h(w) = h(u)[1 h(y)_1] h(v)$ is in $v_{k,s}(\bar{L})$. Suppose (i, j, y) has property (\star, t) and x^t is in $h^{-1}(\bar{L}) \cap A_{i,j}$. Form w' from w by replacing all occurrences of $b'_1 y^\alpha b'_2, \alpha$ in $\{1, R\}$ and b'_i in B_i , by $b'_1 x^\alpha b'_2$. Since h is non-length-increasing and $h(x^t)$ is in \bar{L} , $h(w')$ is still in $v_{k,s}(\bar{L})$; since x is congruent to y and x^R to y^R , w' is still in R . Thus, if (i, j, y) has property (\star, t) , (i, j, x) has property (\star, t) for all x^t in $h^{-1}(\bar{L}) \cap A_{i,j}$, so we say that (i, j) has (\star, t) if any (i, j, y) does. The crucial point is that if (i, j) has (\star, t) , all words in $g(h^{-1}(\bar{L}^t) \cap A_{i,j})$ actually appear as subwords of words in L . Let M_1 be the a -transducer acting on $(\Sigma_1 \cup \Sigma_2)^*$ such that $M_1(w) = \{y \text{ in } \Sigma_2^+ \mid w = uaybv, a, b \text{ in } \Sigma_1\}$. Let

$$\begin{aligned} \bar{L}_2 = & \bigcup_{(i,j) \text{ has } (\star, 1)} M_1(g(h^{-1}(\bar{L}) \cap A_{i,j})) \\ & \cup \bigcup_{(i,j) \text{ has } (\star, R)} M_1(g(h^{-1}(\bar{L}^R) \cap A_{i,j})). \end{aligned}$$

Since words in $g(h^{-1}(\bar{L}^t) \cap A_{i,j})$ are subwords of words of L when (i, j) has (\star, t) , $\bar{L}_2 \subseteq L_2$. Clearly \bar{L}_2 is in $\mathcal{L} \vee \mathcal{L}^R$. Now we must define \bar{L}_1 .

Let f be the homomorphism defined by $f(a) = a, a$ in Σ_1 and $f(a) = e, a$ in Σ_2 . For a in $\Sigma_1 \cup \{e\}$, let (a, i, j) be a new symbol if there is a y in $h^{-1}(\bar{L}) \cap A_{i,j}$ such that $f(g(y)) = a$, and let T be the set of all such symbols. Extend g and h by $h((a, i, j)) = (a, i, j)$ and $g((a, i, j)) = a$. Let M_1 be the a -transducer which nondeterministically either acts as the identity on Γ^* or replaces a word in $A_{i,j}$

by a symbol (a, i, j) in T , and let $R_1 = M_1(R)$. Let $\bar{L}_1 = f(g(h^{-1}(v_{k,s}(T)) \cap R_1))$. Clearly \bar{L}_1 is in $\hat{\mathcal{F}}_r(\text{REGL})$ since T is finite and so regular. If w is in $v_{k,s}(T)$, each symbol (a, i, j) in w can be replaced by a word $h(y)$ or $(h(y))^R$ in \bar{L} with y in $A_{i,j}$ and $f(g(y)) = f(g((a, i, j))) = a$ in such a way that the resulting word w' is in $v_{k,s}(\bar{L})$. If x is in $h^{-1}(w)$, we can replace (a, i, j) in x by y or y^R as above so that the resulting word x' is in $h^{-1}(w')$ and if x is in R_1 , x' is in R . Thus $g(x')$ is in L and $f(g(x)) = f(g(x'))$ is in $f(L) = L_1$, so $f(g(x))$ is in L_1 . Hence $\bar{L}_1 \subseteq L_1$.

Consider a word $\mu(a_1 \dots a_n, w)$ in L . For some x in $v_{k,s}(\bar{L})$, z in $h^{-1}(x) \cap R$, $\mu(a_1 \dots a_n, w) = g(z)$. We can decompose x as

$$x = u_1 [{}_1 y_1 {}_1] \dots u_m [{}_1 y_m {}_1] u_{m+1},$$

the y_i in $\bar{L} \cup \bar{L}^R$ and the u_i strings of brackets, and z as $z = v_1 z_1 \dots v_m z_m v_{m+1}$, $h(z_i) = y_i$. If $f(g(z_i)) \in \Sigma_1 \cup \{e\}$ for all $i, 1 \leq i \leq m$, we say that $a_1 \dots a_n$ splits in $\mu(a_1 \dots a_n, w)$. In that case, each y_i could be replaced by a symbol of T , so $a_1 \dots a_n$ is in \bar{L}_1 . If $n \geq 2$ and $a_1 \dots a_n$ does not split in $\mu(a_1 \dots a_n, w)$, then for some l , $a_l w a_{l+1}$ is a subword of some $g(z_k)$. For some i, j, t , (i, j, z_k^t) has property (\star, t) and $w \in M_1(g(z_k))$, so $w \in \bar{L}_2$.

If for each $a_1 \dots a_n$ in L_1 there is a w in L_2 such that $a_1 \dots a_n$ splits in $\mu(a_1 \dots a_n, w)$, $L_1 = \bar{L}_1$ so $L_1 \in \hat{\mathcal{F}}_r(\text{REGL})$. Otherwise there is a word $a_1 \dots a_n$ in L_1 with $n \geq 2$ (since a_1 necessarily splits in $\mu(a_1, w)$ if $a_1 \in L_1 \cap \Sigma_1$) such that for all w in L_2 , $a_1 \dots a_n$ does not split in $\mu(a_1 \dots a_n, w)$; hence $L_2 = \bar{L}_2 \in \mathcal{L} \vee \mathcal{L}^R$. \square

Lemma 4.4 has the following useful consequence.

LEMMA 4.5: *Let $\mathcal{L}, \mathcal{L}_1$ and \mathcal{L}_2 be full semiAFLs such that $\mathcal{L}_1 \hat{\sigma} \mathcal{L}_2$ is contained in $\hat{\mathcal{F}}_r(\mathcal{L})$. If \mathcal{L}_1 is not contained in $\hat{\mathcal{F}}_r(\text{REGL})$, then \mathcal{L}_2 is contained in $\mathcal{L} \vee \mathcal{L}^R$ and if \mathcal{L}_2 is not contained in $\mathcal{L} \vee \mathcal{L}^R$, \mathcal{L}_1 is contained in $\hat{\mathcal{F}}_r(\text{REGL})$.*

Proof: Suppose \mathcal{L}_1 is not contained in $\hat{\mathcal{F}}_r(\text{REGL})$. Then there is a language L_1 in \mathcal{L}_1 but not in $\hat{\mathcal{F}}_r(\text{REGL})$ such that $L_1 \subseteq \Sigma_1^+$ for some vocabulary Σ_1 . Consider an arbitrary language L_2 in \mathcal{L}_2 . Since full semiAFLs are closed under renaming and addition and deletion of the empty word, it suffices to consider $L_2 \subseteq \Sigma_2^+$ with $\Sigma_1 \cap \Sigma_2 = \emptyset$. Thus $\tau(L_1, L_2)$ is in $\hat{\mathcal{F}}_r(\mathcal{L})$ so by lemma 4.4, L_2 is in $\mathcal{L} \vee \mathcal{L}^R$. The argument if \mathcal{L}_2 is not contained in $\mathcal{L} \vee \mathcal{L}^R$ is similar. \square

Now we are ready to draw some conclusions about the substitution of $\hat{\mathcal{F}}_r(\mathcal{L})$ into or by another full semiAFL.

THEOREM 4.6: *Let \mathcal{L} and \mathcal{L}_1 be full semiAFLs:*

- (1) $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution into \mathcal{L}_1 if and only if either \mathcal{L}_1 is

contained in $\hat{\mathcal{F}}_r(\text{REGL})$ or $\mathcal{L} = \hat{\mathcal{F}}_r(\mathcal{L})$ and \mathcal{L} is closed under substitution into \mathcal{L}_1 .

(2) $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution by \mathcal{L}_1 if and only if either both \mathcal{L} and \mathcal{L}_1 are contained in $\hat{\mathcal{F}}_r(\text{REGL})$, or $\hat{\mathcal{M}}_\sigma(\mathcal{L}_1)$ is contained either in \mathcal{L} or in \mathcal{L}^R and $\mathcal{L} \hat{\circ} \mathcal{L}_1$ is contained in $\hat{\mathcal{F}}_r(\mathcal{L})$.

Proof: First observe that if \mathcal{L}_1 is contained in $\hat{\mathcal{F}}_r(\text{REGL})$, then $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution into \mathcal{L}_1 by theorem 4.2. Next, assume that $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution into \mathcal{L}_1 and that \mathcal{L}_1 is not contained in $\hat{\mathcal{F}}_r(\text{REGL})$. Now $\mathcal{L}_1 \hat{\circ} \hat{\mathcal{F}}_r(\mathcal{L}) \subseteq \hat{\mathcal{F}}_r(\mathcal{L})$. By lemma 4.5, $\hat{\mathcal{F}}_r(\mathcal{L})$ is contained in $\mathcal{L} \vee \mathcal{L}^R$ and hence $\hat{\mathcal{F}}_r(\mathcal{L}) = \mathcal{L} \vee \mathcal{L}^R$. If $\mathcal{L} \neq \mathcal{L}^R$, then $\mathcal{L} \vee \mathcal{L}^R$ is not an AFL [15]. Thus $\hat{\mathcal{F}}_r(\mathcal{L}) = \mathcal{L} = \mathcal{L}^R$ and \mathcal{L} is closed under substitution into \mathcal{L}_1 . This establishes (1).

If \mathcal{L} and \mathcal{L}_1 are both contained in $\hat{\mathcal{F}}_r(\text{REGL})$, then

$$\hat{\mathcal{F}}_r(\mathcal{L}) = \hat{\mathcal{F}}_r(\text{REGL}) = \hat{\mathcal{F}}_r(\mathcal{L}_1),$$

so $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution by \mathcal{L}_1 . On the other hand, if $\mathcal{L} \hat{\circ} \mathcal{L}_1 \subseteq \hat{\mathcal{F}}_r(\mathcal{L})$, theorem 4.2 tells us that $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution by \mathcal{L}_1 .

Now suppose that $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution by \mathcal{L}_1 . Then $\mathcal{L} \hat{\circ} \mathcal{L}_1 \subseteq \hat{\mathcal{F}}_r(\mathcal{L})$, and $\hat{\mathcal{F}}_r(\mathcal{L}) \hat{\circ} \hat{\mathcal{M}}_\sigma(\mathcal{L}_1) \subseteq \hat{\mathcal{F}}_r(\mathcal{L})$. If \mathcal{L} is not contained in $\hat{\mathcal{F}}_r(\text{REGL})$ then, by lemma 4.5, $\hat{\mathcal{M}}_\sigma(\mathcal{L}_1)$ is contained in $\mathcal{L} \vee \mathcal{L}^R$. Hence $\hat{\mathcal{M}}_\sigma(\mathcal{L}_1)$ must be contained in either \mathcal{L} or \mathcal{L}^R [15]. Finally, suppose that \mathcal{L}_1 is not contained in $\hat{\mathcal{F}}_r(\text{REGL})$. Since $\hat{\mathcal{M}}_\sigma(\mathcal{L}_1) = \hat{\mathcal{M}}_\sigma(\mathcal{L}_1) \hat{\circ} \hat{\mathcal{M}}_\sigma(\mathcal{L}_1)$ is contained in $\hat{\mathcal{F}}_r(\mathcal{L})$, by lemma 4.5, $\hat{\mathcal{M}}_\sigma(\mathcal{L}_1)$ is contained in $\mathcal{L} \vee \mathcal{L}^R$ and hence in either \mathcal{L} or \mathcal{L}^R . This establishes (2). \square

A consequence of theorem 4.6 worthy of special attention is the following. It is immediate from theorem 4.6(1), taking $\mathcal{L}_1 = \mathcal{L}$.

THEOREM 4.7: *Let \mathcal{L} be a full semiAFL. Then $\hat{\mathcal{F}}_r(\mathcal{L})$ is closed under substitution if and only if either \mathcal{L} is contained in $\hat{\mathcal{F}}_r(\text{REGL})$, so $\hat{\mathcal{F}}_r(\mathcal{L}) = \hat{\mathcal{F}}_r(\text{REGL})$ or \mathcal{L} is closed under both substitution and homomorphic replication.*

5. If \mathcal{L} is a full semiAFL, then the closure of \mathcal{L} under deterministic two-way finite state transductions is $\text{FINITE.VISIT}(\mathcal{L})$, the family of languages accepted by the one-way \mathcal{L} -based preset Turing machines of [19] restricted to a finite number of visits per working tape square. (A one-way \mathcal{L} -based preset Turing machine has a one-way input tape and one working tape preset to words in L for some L in \mathcal{L} .) If we restrict the number of reversals rather than visits of

\mathcal{L} -based preset Turing machines, we obtain FINITE-REVERSAL(\mathcal{L}) which is equal to $\hat{\mathcal{M}}_r(\mathcal{L})$ and also to the closure of \mathcal{L} under iterating controls on linear context-free grammars. For the special case $\mathcal{L} = \text{REGL}$, FINITE-REVERSAL(REGL) is the family of languages accepted by finite reversal checking automata, while FINITE-VISIT (REGL) is the family of languages accepted by finite visit checking automata. If \mathcal{L} is a full AFL not closed under homomorphic replication, FINITE-REVERSAL (\mathcal{L}) is not a full AFL while FINITE-VISIT (\mathcal{L}) is an AFL [19]. Thus “most of the time”, FINITE-VISIT(\mathcal{L}) property contains FINITE-REVERSAL(\mathcal{L}), so visits are more powerful than reversals.

For any full AFL \mathcal{L} , we have

$$\text{FINITE . REVERSAL } (\mathcal{L}) \subseteq \hat{\mathcal{F}}_r(\mathcal{L}) \subseteq \text{FINITE . VISIT } (\mathcal{L}).$$

We shall now show that, if \mathcal{L} is closed under substitution but not homomorphic replication, these containments are always proper. Since FINITE . VISIT (\mathcal{L}) is closed under substitution for any substitution closed full semiAFL \mathcal{L} , this follows from theorem 4.7 when \mathcal{L} is not contained in $\hat{\mathcal{F}}_r(\text{REGL})$.

It remains to show that $\hat{\mathcal{F}}_r(\text{REGL})$ is property contained in FINITE . VISIT (REGL). We do so by showing that, although FINITE . VISIT (REGL) does have stronger iterative properties than those established in [19], the iterative properties of $\hat{\mathcal{F}}_r(\text{REGL})$ are still stronger.

First we show that if a generalized Ogden’s lemma [29] holds for a full semiAFL \mathcal{L} , it does so for $\hat{\mathcal{F}}_r(\mathcal{L})$. We call a language for which a generalized Ogden’s Lemma holds a strongly iterative language.

DEFINITION: A language L is *strongly k -iterative* for $k \geq 1$, if for each $t \geq 1$ there are *iteration numbers* p, q such that if w is in L , $|w| \geq p$ and p or more positions of w are designated as distinguished, then we can factor w as

$$w = u_1 v_1 \dots u_k v_k u_{k+1},$$

where:

(1) For some i , v_i contains at least t distinguished positions, u_i and u_{i+1} each contain at least one distinguished position and $u_i v_i u_{i+1}$ contains at most q distinguished positions.

(2) For each $n \geq 0$, $u_1 v_1^n \dots u_k v_k^n u_{k+1}$ is in L .

We call L *strongly iterative* if it is strongly k -iterative for some $k \geq 1$. A family of languages is *strongly iterative* if each of its members is strongly iterative.

The property of being strongly iterative is preserved by homomorphic replication and by substitution. The next lemmas generalize results in [13, 17, 19, 24 and 25], and are similar to results in [35] for a slightly different property, “locally linear”

LEMMA 5.1: *If \mathcal{L} is a strongly iterative full semiAFL, then $\hat{\mathcal{M}}_r(\mathcal{L})$ is strongly iterative.*

Proof: If L_1 is in $\hat{\mathcal{M}}_r(\mathcal{L})$, we can express it as $L_1 = \langle \rho, h_1, \dots, h_m \rangle(L)$ for L in \mathcal{L} and the h_i non-length-increasing homomorphisms. Let L be strongly k -iterative. We claim that L_1 is strongly km -iterative. For $t \geq 1$, let p, q satisfy the definition of strongly k -iterative for L . Let $p_1 = mp$ and $q_1 = q$. Consider w in L_1 with $|w| \geq p_1$. Then $w = (h_1(x))^{\rho(1)} \dots (h_m(x))^{\rho(m)}$ for x in L . Designate any p_1 or more positions in w as distinguished. At least p distinguished positions must lie in some $(h_j(x))^{\rho(j)}$ and, since h_j is not length increasing, must correspond to at least p positions in x which we call distinguished positions in x . Thus there is a factorization $x = u_1 v_1 \dots u_k v_k u_{k+1}$ satisfying (1) and (2) of the definition of strongly k -iterative. Then the factorization

$$w = (h_1(u_1) h_1(v_1) \dots h_1(v_k) h_1(u_{k+1}))^{\rho(1)} \dots \\ (h_m(u_1) h_m(v_1) \dots h_m(v_k) h_m(u_{k+1}))^{\rho(m)},$$

satisfies (1), (2) of the definition of strongly km -iterative with the $(h_j(v_i))^{\rho(j)}$ providing the km iterative factors, since distinguished positions in x and $h_j(x)$ are in one-one correspondence. \square

LEMMA 5.2: *If \mathcal{L}_1 and \mathcal{L}_2 are strongly iterative full semiAFLs, then $\mathcal{L}_1 \hat{\sigma} \mathcal{L}_2$ is strongly iterative.*

Proof: Let $L_1 = \tau(L)$ for $L \in \Sigma^*$, L in \mathcal{L}_1 and each $\tau(a)$ in \mathcal{L}_2 , a in Σ . Let L be strongly k -iterative and each $\tau(a)$ strongly k_a -iterative. We claim that L_1 is strongly k_1 -iterative for $k_1 = \text{Max}(\{k\} \cup \{k_a \mid a \text{ in } \Sigma\})$. For $t \geq 1$, let p, q be the iteration numbers for L and p_a, q_a those for $\tau(a)$, a in Σ . Let

$$p_1 = p \text{Max}(\{p_a \mid a \text{ in } \Sigma\})$$

and

$$q_1 = \text{Max}(\{q(p_a - 1), q_a \mid a \text{ in } \Sigma\}).$$

Consider w in L_1 with $|w| \geq p_1$ and call any p_1 or more positions in w distinguished. We can write $w = w_1 \dots w_m$, each w_i in $\tau(a_i)$, a_i in Σ and $y = a_1 \dots a_m$ in L . There are two cases. If some w_i contains at least p_a distinguished positions, we can use the factorization for w_i inherited from $\tau(a_i)$ since we are free to substitute for a_i any member of $\tau(a_i)$; if k_{a_i} is less than k_1 , we let the other $k_1 - k_{a_i}$ factors be the empty word. Suppose no w_i contains p_a distinguished positions. Thus there are at least p subwords w_i which contain at least 1 but no more than p_a distinguished positions. If w_i contains distinguished positions, let a_i be distinguished in y . Hence we have a factorization

$y = u_1 v_1 \dots u_k v_k u_{k+1}$ satisfying (1) and (2) of the definition of strongly k -iterative. Let

$$w = \tau(u_1)\tau(v_1) \dots \tau(u_k)\tau(v_k)\tau(u_{k+1}) \dots \tau(v_k)\tau(u_{k+1})$$

where if $k_1 > k$, $u_i = e$ for $i \geq k + 2$ and $v_i = e$ for $i \geq k + 1$. Since $u_1 v_1^n \dots u_k v_k^n u_{k+1}$ is in L for each $n \geq 0$, $\tau(u_1)\tau(v_1)^n \dots \tau(u_k)\tau(v_k)^n \dots \tau(u_{k+1})$ is in L_1 for each $n \geq 0$. If $u_i v_i u_{i+1}$ satisfies (1) of the definition for L , then $\tau(v_i)$ contains at least t distinguished positions, $\tau(u_i)$ and $\tau(u_{i+1})$ contain distinguished positions and $\tau(u_i v_i u_{i+1})$ contains at most $q \text{Max}(\{p_a - 1 \mid a \text{ in } \Sigma\})$ distinguished positions. \square

Since $\hat{\mathcal{F}}_r(\mathcal{L})$ is obtained from \mathcal{L} by alternating closure under homomorphic replication and closure under substitution into regular sets and all regular sets are strongly 1-iterative, we have the next theorem.

THEOREM 5.3: *If \mathcal{L} is a strongly iterative full semiAFL, so are $\hat{\mathcal{F}}_r(\mathcal{L})$ and $\hat{\mathcal{M}}_\sigma(\mathcal{L})$.*

Since every context-free language is strongly 2-iterative [29], we have the following corollary.

COROLLARY: *$\hat{\mathcal{F}}_r(\text{REGL})$ and $\hat{\mathcal{F}}_r(\text{CF})$ are strongly iterative.*

Showing that a particular member of FINITE . VISIT (REGL) is not strongly iterative is long and tedious.

LEMMA 5.4: *Let $L_1 = \{ a^n b^n \mid n \geq 1 \}$ and let L consist of all and only words of the form $w_1 \mathcal{S} w_2$ where*

$$w_1 = dx_1 y_1 cy_1^R x_1^R d \dots dx_m y_m cy_m^R x_m^R d$$

and

$$w_2 = dy_1 x_2 cx_2^R y_1^R d \dots dy_{m-1} x_m cx_m^R y_{m-1}^R d$$

for $m \geq 2$, x_i, y_i in L_1 for $1 \leq i \leq m$. Then L is not strongly iterative.

Proof: The idea is to find w for which the desired iterative factors exist but not as subwords of w . Suppose that L were strongly k -iterative. Let $t = 2$ and let p, q be the required iteration numbers for L .

Let $w = w_1 \mathcal{S} w_2$ be in L where w_1 and w_2 are expressed as above, $m = p + 1$ and $x_i \neq x_j$ and $y_i \neq y_j$ for $i \neq j$ and $x_i \neq y_j$ for all i, j . Call the second through the $(p + 1)$ st d 's in w_1 distinguished. Then we have $w_1 \mathcal{S} w_2 = u_1 v_1 \dots u_k v_k u_{k+1}$ and s , $1 \leq s \leq k$ such that v_s contains at least 2 distinguished positions. u_s and u_{s+1} each contain at least 1 distinguished position and $w(n) = u_1 v_1^n \dots u_k v_k^n u_{k+1}$ is in L for all $n \geq 0$. Thus v_s lies wholly in w_1 and does not contain the first two or the last two d 's in w_1 .

Now v_s cannot start and end with d or else $w(2)$ would contain two d 's in a row. Thus $v_s = \beta d \gamma d \alpha$ for $\alpha \beta$ in $\{a, b, c\}^+$ and w_1 has a subword

$$\dots d\alpha_1 \beta d \gamma d \alpha \beta_1 d \dots$$

with $\alpha_1 \beta$ and $\alpha \beta_1$ in

$$L_2 = \{ a^{n_1} b^{n_1} a^{n_2} b^{n_2} c b^{n_2} a^{n_2} b^{n_1} a^{n_1} \mid n_1, n_2 \geq 1 \}.$$

Further, if n_1, n_2 are the exponents for $\alpha_1 \beta$ and n_3, n_4 for $\alpha \beta_1$, no two of the four integers are the same. Now L contains $w(0) = u_1 u_2 \dots u_s u_{s+1} \dots u_{k+1}$ which has a subword

$$\dots d\alpha_1 \beta_1 d \dots$$

and

$$w(2) = u_1 v_1^2 \dots u_s v_s^2 u_{s+1} \dots u_{k+1}$$

which has a subword

$$\dots d\alpha_1 \beta d \gamma d \alpha \beta d \gamma d \alpha \beta_1 d \dots$$

so $\alpha_1 \beta_1$ and $\alpha \beta$ are also in L_2 . Either α or β contains c ; suppose it is β , then $\alpha = \alpha_1$. If α contains any b 's, this determines n_1 and so $n_3 = n_1$, a contradiction. Hence α is in a^* and β is a member of L_2 with $|\alpha|$ a 's removed from the left end. A similar argument applies if α contains c .

Let $y \setminus z = y'$ if $z = yy'$. We can write

$$v_s = (a^t \setminus x_i) y_i c y_i^R x_i^R d x_{i+1} y_{i+1} c y_{i+1}^R x_{i+1}^R d \dots d x_{i+t} y_{i+t} c y_{i+t}^R x_{i+t}^R d a^t$$

with $t \geq 1$. Write $w(n) = w_1(n) \S w_2(n)$.

Now w_2 has a subword

$$\dots d y_{i-1} x_i c x_{i-1}^R y_{i-1}^R d y_i x_{i+1} c x_{i+1}^R y_i d \dots d y_{i+t} x_{i+t+1} c x_{i+t+1}^R y_{i+t}^R d \dots$$

For $n \geq 2$, $w_1(n)$ has a subword

$$d(x_i y_i c y_i^R x_i^R d x_{i+1} y_{i+1} c y_{i+1}^R x_{i+1}^R d \dots d x_{i+t} y_{i+t} c y_{i+t}^R x_{i+t}^R d)^n$$

which by the definition of L means that $w_2(n)$ has a subword

$$d(y_i x_{i+1} c x_{i+1}^R y_i^R d \dots d y_{i+t-1} x_{i+t} c x_{i+t}^R y_{i+t-1}^R d y_{i+t} x_i c x_{i+t}^R y_{i+t}^R d)^{n-1}.$$

But since all the x_i and y_i in w are distinct, $d y_{i+t} x_i c x_{i+t}^R y_{i+t}^R d$ cannot be a subword of w and so this iterative factor in $w_2(n)$ cannot be obtained as a subword of w . This is a contradiction. Hence L is not strongly iterative. \square

COROLLARY: *There is a language L in FINITE . VISIT (REGL) which is not strongly iterative.*

Proof: It remains to show that the language L in lemma 5.4 is in FINITE . VISIT (REGL). Clearly L_1 and so $\$ (L_1 L_1 d)^+ \$$ are in FINITE . VISIT (REGL). A two-way deterministic finite state transducer acting on $\$ (L_1 L_1 d)^- \$$ can interpret an input string $\$ x_1 y_1 d \dots dx_m y_m d \$$ as giving the x_i and y_i for words in L and first go right and then left over $x_i y_i$ to produce $dx_i y_i c y_i^R x_i^R d$, on reading the right hand $\$$ out put $\$$ and then return to the first $\$$, now scanning $y_i dx_{i+1}$ to give $dy_i x_{i+1} c x_{i+1}^R y_{i+1}^R d$, and finally finish on again scanning the right hand $\$$. Since FINITE . VISIT (REGL) is closed under two-way deterministic finite state transducer mapping, it contains L . \square

We can state the main result of this section.

THEOREM 5.5: *If \mathcal{L} is a full AFL closed under substitution but not homomorphic replication*

$$\mathcal{L} \underset{\neq}{\subseteq} \text{FINITE . REVERSAL} (\mathcal{L}) \underset{\neq}{\subseteq} \mathcal{F}_r(\mathcal{L}) \underset{\neq}{\subseteq} \text{FINITE . VISIT} (\mathcal{L}).$$

REFERENCES

1. A. V. AHO, *Indexed Grammars. An Extension of Context-Free Grammars*, J. Assoc. Comp. Mach., Vol. 15, 1968, pp. 647-671.
2. P. ASVELD and J. ENGELFRIET, *Iterated Deterministic Substitution*, Acta Informatica, Vol. 8, 1977, pp. 285-302.
3. R. BOOK, *Simple Representations of Certain Classes of Languages*, J. Assoc. Comp. Mach., Vol. 25, 1978, pp. 23-31.
4. R. BOOK and M. NIVAT, *Linear Languages and the Intersection Closures of Classes of Languages*, S.I.A.M. J. Computing, Vol. 7, 1978, pp. 167-177.
5. R. BOOK and C. WRATHALL, *On Languages Specified by Relative Acceptance*, Theoret. Comp. Science (to appear).
6. R. W. EHRICH and S. S. YAU, *Two-Way Sequential Transductions and Stack Automata*, Inform. and Control, Vol. 18, 1971, pp. 404-446.
7. C. C. ELGOT and J. E. MEZEI, *On Relations Defined by Generalized Finite Automata*, I.B.M. J. Res. and Devel., Vol. 9, 1975, pp. 47-68.
8. J. ENGELFRIET and E. M. SCHMIDT, *IO and OI*, J. Comput. Syst. Sc. (to appear).
9. J. ENGELFRIET and S. SKYUM, *Copying Theorems*, Inform. Processing Letts., Vol. 4, 1976, pp. 157-161.
10. S. GINSBURG, *Algebraic and Automata-Theoretic Properties of Formal Languages*, North-Holland, Amsterdam, 1975.
11. S. GINSBURG and S. A. GREIBACH, *Abstract Families of Languages*, in *Studies in Abstract Families of Languages* GINSBURG, GREIBACH and HOPCROFT, eds., Memoirs Amer. Math. Soc., Vol. 87, 1969, pp. 1-32.

12. S. GINSBURG and S. A. GREIBACH, *Principal AFL*, J. Comput. Syst. Sc., Vol. 4, 1970, pp. 308-338.
13. S. GINSBURG and E. H. SPANIER, *AFL with the Semilinear Property*, J. Comput. Syst. Sc., Vol. 5, 1971, pp. 365-396.
14. S. GINSBURG and E. H. SPANIER, *Derivation-Bounded Languages*, J. Comput. Syst. Sc., Vol. 2, 1968, pp. 228-250.
15. S. GREIBACH, *Chains of Full AFLs*, Math. Syst. Theory, Vol. 4, 1970, pp. 231-242.
16. S. GREIBACH, *Checking Automata and One-Way Stack Languages*, J. Comput. Syst. Sc., Vol. 3, 1969, pp. 196-217.
17. S. GREIBACH, *Control Sets on Context-Free Grammar Forms*, J. Comput. Syst. Sc., Vol. 15, 1977, pp. 35-98.
18. S. GREIBACH, *Erasing in Context-Free AFLs*, Inform. and Control, Vol. 21, 1972, pp. 436-465.
19. S. GREIBACH, *One-Way Finite Visit Automata*, Theoret. Comput. Sc., Vol. 6, 1978, pp. 175-222.
20. S. GREIBACH, *Full AFLs and Nested Iterated Substitution*, Inform. and Control, Vol. 16, 1970, pp. 7-35.
21. S. GREIBACH, *Syntactic Operators on Full SemiAFLs*, J. Comput. Syst. Sc., Vol. 6, 1972, pp. 30-76.
22. S. GREIBACH and S. GINSBURG, *Multitape AFA*, J. Assoc. Comput. Mach., Vol. 19, 1972, pp. 193-221.
23. O. H. IBARRA, *Controlled Pushdown Automata*, Inform. Sc., Vol. 6, 1973, pp. 327-342.
24. N. A. KHABBAZ, *Control Sets on Linear Grammars*, Inform. and Control, Vol. 25, 1974, pp. 206-221.
25. N. A. KHABBAZ, *A Geometrical Hierarchy of Languages*, J. Comput. Syst. Sc., Vol. 8, 1974, pp. 142-157.
26. K. KLINGENSTEIN, *Structures of Bounded Languages in Certain Classes of Languages*, Inform. and Control (to appear).
27. K. KLINGENSTEIN, *ρ -Matrix Languages*, Theoret. Comput. Sc. (to appear).
28. A. NERODE, *Linear Automata Transformations*, Proc. Amer. Math. Soc., Vol. 9, 1958, pp. 541-544.
29. W. OGDEN, *A Helpful Result in Proving Inherent Ambiguity*, Math. Syst. Theory, Vol. 2, 1968, pp. 191-194.
30. A. SALOMAA, *Formal Languages*, Academic Press, New York, 1973.
31. A. SALOMAA, *Macros, Iterated Substitution and Lindenmayer AFLs*, DAIMI, PB-18, University of Aarhus, Aarhus, Denmark.
32. I. H. SUDBOROUGH, *On the Complexity of the Membership Problem for some Extensions of Context-Free Languages*, Intl. J. Computer Math. (to appear).
33. M. LATTEUX, *Cônes rationnels commutativement clos*, R.A.I.R.O. Informatique théorique, Vol. 11, 1977, pp. 29-51.
34. J. ENGELFRIET, *Three Hierarchies of Transducers*, Memo 217, Dept. Appl. Math., Twente Univ. Technology, 1978.
35. A. VAN DER WALT, *Locally Linear Families of Languages*, Inform. and Control, Vol. 32, 1976, pp. 27-32.