RAIRO. INFORMATIQUE THÉORIQUE

A. EHRENFEUCHT
G. ROZENBERG

Strong iterative pairs and the regularity of context-free languages

RAIRO. Informatique théorique, tome 19, nº 1 (1985), p. 43-56
<http://www.numdam.org/item?id=ITA_1985__19_1_43_0>

R.A.I.R.O. Informatique théorique/Théoretical Informatics
(vol. 19, n° 1, 1985, p. 43 à 56)

STRONG ITERATIVE PAIRS AND THE REGULARITY OF CONTEXT-FREE LANGUAGES (*)

by A. EHRENFEUCHT (1) and G. ROZENBERG (2)

Communicated by J. BERSTEL

Abstract. — *The notion of an* iterative pair *introduced by Boasson, formalizes pumping properties of (long enough) words in languages as e. g., expressed by the celebrated pumping lemma for context-free languages. Such an iterative pair* (x, y, z, u, t) *of a language K must be such that* $xyzut \in K$, $yu \neq \Lambda$, *and for every* $n \geq 1$, $xy^n zu^n t \in K$. *Since* $n \geq 1$, *an iterative pair allows pumping upwards. A* strong iterative pair *is like an iterative pair except that we allow every* $n \geq 0$; *thus also pumping downwards is permitted. A (strong) iterative pair* (x, y, z, u, t) *is said to be* very degenerate *if, for every* $n, m \geq 0$, $xy^n zu^m t \in K$. *It is proved that if K is a context-free language such that each of the strong iterative pairs of it is very degenerate then K is regular; this result generalizes an analogous result for iterative pairs proved by Boasson.*

Résumé. — *La notion de* paire itérante *introduite par Boasson formalise des propriétés d'itération de mots (assez longs) de langages, comme formulé par le célèbre lemme de la double étoile pour les langages algébriques. Une telle paire itérante* (x, y, z, u, t) *d'un langage K doit vérifier* $xyzut \in K$, $yu \neq \Lambda$, *et par tout* $n \geqslant 1$, $xy^n zu^n t \in K$. *Comme* $n \geqslant 1$, *une paire itérante permet une itération croissante. Une* paire itérante forte *est comme une paire itérante, sauf que tout* $n \geqslant 0$ *est autorisé; ceci permet également une itération décroissante. Une paire itérante (forte)* (x, y, z, u, t) *est dite* très dégénérée *si pour tout* $n, m \geqslant 0$, *on a* $xy^n zu^m t \in K$. *On montre que si K est un langage algébrique dont toute paire itérante forte est très dégénérée, alors K est un langage rationnel. Ce résultat généralise un résultat analogue prouvé par Boasson pour les paires itérantes.*

INTRODUCTION

The class of context-free languages ($\mathscr{L}_{CF}$) and the class of regular languages ($\mathscr{L}_{REG}$), where $\mathscr{L}_{REG} \subsetneqq \mathscr{L}_{CF}$, are important classes of languages within formal language theory, *see*, e. g., [4] and [5]. A way to understand the structure of context-free *grammars* is to impose restrictions on them which will

(*) Received in September 1983.
(1) Dept. of Computer Science, University of Colorado, Boulder, Boulder, Colorado 80309.
(2) Institute of Applied Math. and Computer Science, University of Leiden, Leiden, The Netherlands.

guarantee that the languages generated will be regular. Several restrictions of this kind are known, *see*, e. g., [4] and [5].

On the other hand, in order to understand the combinatorial structure of context-free *languages*, one can attempt to formulate conditions (combinatorial in nature) on the interrelationship of words in a context-free language which would force such a language to be context-free, *see*, e. g., [1]. A starting point can be the celebrated pumping lemma for context-free languages. Based on it, the notion of an *iterative pair* was introduced in [2], *see* also [1]. If K is a language, $K \subseteq \Sigma^*$, then $p=(x, y, z, u, t)$ is an iterative pair in K if, for every $n \geq 1$, $xy^n zu^n t \in K$ where yu is a nonempty word. Such a *synchronized* pumping of subwords (y and u) in a word ($xyzut$) in K gives one a possibility (using one iterative pair only) to generate context-free but not regular languages (e. g., $\{ a^n b^n : n \geq 1 \}$). However, if one *desynchronizes* such a pumping, that is, one requires that, for all r, $s \geq 0$, $xy^r zu^s t \in K$, then an iterative pair yields a regular language. This observation leads one to a conjecture that if each iterative pair $p=(x, y, z, u, t)$ of a context-free language K is *very degenerate* (that is, for all r, $s \geq 0$, $xy^r zu^s t \in K$) then K must be regular. This conjecture was shown to be true in [2]. An iterative pairs allows only "upward pumping", expressed by the fact that $n \geq 1$ and in this sense it does not fully formalize the idea from the pumping lemma for context-free languages where also pumping "downward" (i. e., $n=0$) is allowed. If in the definition of an iterative pair we require $n \geq 0$ rather than $n \geq 1$, then we get a strong iterative pair.

In this paper we prove that if every strong iterative pair of a context-free language K is very degenerate then K is a regular language. This result generalizes the result from [2] in the sense that we can obtain the latter directly from our result. It provides a positive solution of a conjecture stated in [1].

0. PRELIMINARIES

We assume the reader to be familiar with the theory of context-free and regular languages, e. g., in the scope of [4] or [5]. We will use rather standard formal language theoretic notation and terminology. Perhaps only the following points require an additional explanation. For a finite set A, $\# A$ denotes its cardinality. N denotes the set of natural numbers (including 0) while Z^+ denotes the set of positive integers. We consider finite alphabets only. Λ denotes the empty word. For a word w, *alph*(w) denotes the set of all letters appearing in w and $|w|$ denotes the length of w; if a is a letter, then $\#_a(w)$ denotes the number of occurrences of a in w. If $w \neq \Lambda$ then *last* (w) denotes the last letter of w and $w/last(w)$ denotes the word obtained from w by removing the last letter of it. For a language K, Pref(K) denotes the set of all prefixes of all words in K.

For an equivalence relation R, $index(R)$ denotes its index.

For an alphabet Σ, HOM (Σ, Σ) denotes the set of all homomorphisms from Σ^* into Σ^*.

We recall now the basic characterization of regular languages.

DEFINITION 0.1: Let K be a language, $K \subseteq \Sigma^*$. The *Myhil-Nerode relation induced by* K, denoted by $\sim_K$, is defined as follows. For $x, y \in \Sigma^*$, $x \sim_K y$ if and only if, for every $u \in \Sigma^*$, $xu \in K$ if and only if $yu \in K$. □

It is easily seen that $\sim_K$ is an equivalence relation. The following theorem (*see*, e. g., [5]) provides the fundamental characterization of regular languages.

THEOREM 0.1: *Let K be a language, $K \subseteq \Sigma^*$. K is regular if and only if $\sim_K$ is of finite index.* □

In the sequel we will need a somewhat modified version of this result.

Let K be a language, $K \subseteq \Sigma^*$. Let $M_K = \{ u \in \Sigma^+ : ut \in K \text{ for some } t \in \Sigma^* \}$. Let $(\sim_K)_{M_K}$ be the relation $\sim K$ restricted by M_K, hence

$$(\sim_K)_{M_K} = \{ (x, y) : (x, y) \in \sim_K, x \in M_K \text{ and } y \in M_K \}.$$

THEOREM 0.2: *Let K be a language, $K \subseteq \Sigma^*$. If $(\sim_K)_{M_K}$ is of finite index then K is regular.*

Proof: Let $w \in \Sigma^*$. Then either $w = \Lambda$ or $w \in M_K$ or $w \notin Pref(K)$. Consequently $index(\sim_K) \leq index((\sim_K)_{M_K}) + 2$ and so $\sim_K$ is of finite index. Thus, by Theorem 0.1, K is regular. □

1. BASIC NOTIONS

In this section several notions very basic to this paper are introduced and their rudimentary properties investigated.

We start by introducing the notion of a strong iterative pair which directly generalizes the notion of an iterative pair as introduced in [2], *see* also [1]. (This generalization was suggested by [3]). The difference is that we allow also shortening of a word and so we can consider the iteration starting from 0.

DEFINITION 1.1: Let K be a language, $K \subseteq \Sigma^*$. A *strong iterative pair*, abbreviated SIP, of K is a 5-tuple $p = (x, y, z, u, t)$ where $x, y, z, u, t \in \Sigma^*$, $yu \neq \Lambda$ and, for every $n \in N$, $xy^nzu^nt \in K$. We say that p is a *very degenerate strong iterative pair*, abbreviated VDSIP, *of K if*, *for every $n, m \in N$, $xy^nzu^mt \in K$.* □

For a language K, SIP(K) will denote the set of strong iterative pairs of K and VDSIP(K) will denote the set of very degenerate strong iterative pairs of K.

The following generalization of the notion of a strong iterative pair will be a very useful technical tool in our investigation.

DEFINITION 1.2: Let K be a language, $K \subseteq \Sigma^*$. A *generalized strong iterative pair*, abbreviated GSIP, of K is a $(4l+1)$-tuple

$$p=(x_1, \ldots, x_l, y_1, \ldots, y_l, z, u_l, \ldots, u_1, t_l, \ldots, t_1)$$

where $l \in Z^+, x_1, \ldots, x_l, y_1, \ldots, y_l, z, u_l, \ldots, u_1, t_l, \ldots, t_1 \in \Sigma^*$ and, for all $n_1, \ldots, n_l \in N$, $x_1 y_1^{n_1} x_2 y_2^{n_2} \ldots x_l y_l^{n_l} z u_l^{n_l} t_l u_{l-1}^{n_{l-1}} \ldots u_1^{n_1} t_1 \in K$. We say that p is a *very degenerate generalized strong iterative pair*, abbreviated VDGSIP, of K if, for every $n_1, \ldots, n_l, m_1, \ldots, m_l \in N$, $x_1 y_1^{n_1} \ldots x_l y_l^{n_l} z u_l^{m_l} t_l \ldots u_1^{m_1} t_1 \in K$. □

For a language K, GSIP(K) will denote the set of generalized strong iterative pairs of K and VDGSIP(K) will denote the set of very degenerate generalized strong iterative pairs of K. Also, in the above definition we refer to l as the *length* of p. Clearly SIP$(K) \subseteq$ GSIP(K).

The following result makes a useful connection between SIP(K) and GSIP(K).

THEOREM 1.1: *Let K be a language. If $SIP(K) \subseteq VDSIP(K)$ then $GSIP(K) \subseteq VDGSIP(K)$.* □

Proof: Let $p \in$ GSIP(K); we have to prove that, under the assumption of the theorem, $p \in$ VDGSIP(K). We will prove this by the induction on the length of p.

If the length of p equals one then $p \in$ SIP(K), hence $p \in$ VDSIP(K) and consequently $p \in$ VDGSIP(K).

Assume that the theorem holds for every GSIP p of K that is of length not exceeding $l-1$ where $l \geq 2$.

Consider now a GSIP p of length l; let

$$p=(x_1, \ldots, x_l, y_1, \ldots, y_l, z, u_l, \ldots, u_1, t_l, \ldots, t_1).$$

Let $n_1, \ldots, n_l \in N$ and let us consider the word

$$w=x_1 y_1^{n_1} \ldots x_l y_l^{n_l} z u_l^{n_l} t_l \ldots u_1^{n_1} t_1;$$

since p is a GSIP, $w \in K$. Let $x=x_1 y_1^{n_1} \ldots y_{l-1}^{n_{l-1}} x_l$, $y=y_l$, $u=u_l$ and $t=t_l u_{l-1}^{n_{l-1}} \ldots u_1^{n_1} t_1$. Clearly $(x, y, z, u, t) \in$ SIP(K).

Thus, by the assumption of the theorem, for all $m_1, m_2 \in N$ and for all $n_1, \ldots, n_{l-1} \in N$ we have

$$x_1 y_1^{n_1} \ldots y_{l-1}^{n_{l-1}} x_l y_l^{m_1} z u_l^{m_2} t_l u_{l-1}^{n_{l-1}} \ldots u_1^{n_1} t_1 \in K.$$

Consequently for all $m_1, m_2 \in N$

$$q(m_1, m_2) = (x_1, y_1, \ldots, x_{l-1}, y_{l-1}, x_l y_l^{m_1} z u_l^{m_2} t_l, u_{l-1}, t_{l-1}, \ldots, u_1, t_1)$$

is an element of GSIP(K) and the length of $q(m_1, m_2)$ equals $l-1$. Thus, by the inductive assumption, $q(m_1, m_2) \in$ VDGSIP(K). Hence for all $m_1, m_2 \in N$, for all $n_1, \ldots, n_{l-1} \in N$ and for all $r_1, \ldots, r_{l-1} \in N$ we have

$$x_1 y_1^{n_1} \ldots x_{l-1} y_{l-1}^{n_{l-1}} x_l y_l^{m_1} z u_l^{m_2} t_l u_{l-1}^{r_{l-1}} \ldots u_1^{r_1} t_1 \in K$$

and consequently $p \in$ VDGSIP(K).

Hence the theorem holds. □

Another important notion of this paper is that of a type of a word. It is defined as follows.

DEFINITION 1.3: Let Σ be an alphabet and let $u, w \in \Sigma^*$. We say that w *is of type* u or that u *is a type of* w (denoted $\tau(u, w)$) if

(i) for every $a \in \Sigma$, $\#_a(u) \leq 1$, and

(ii) there exists a homomorphism $h \in$ HOM(Σ, Σ) such that

(ii.1) for every $a \in \Sigma$, $h(a) \in \{a\} \cup \{a\} \Sigma^* \{a\}$, and

(ii.2) $h(u) = w$.

If u satisfies the above, we also say that u *is a type in* Σ^*. □

Example 1.1: (1) Let $\Sigma = \{a, b, c, d\}$, $u = abcd$ and $w = abcabccd$. Then $\tau(u, w)$ where we use the homomorphism h is defined by $h(a) = abca$, $h(b) = b$, $h(c) = cc$ and $h(d) = d$. It is instructive to notice that also the homomorphism $\bar{h}$ defined by $\bar{h}(a) = a$, $\bar{h}(b) = bcab$, $\bar{h}(c) = cc$ and $\bar{h}(d) = d$ will yield $\tau(u, w)$.

(2) Let $\Sigma = \{a, b, c\}$, $u_1 = acb$, $u_2 = ab$ and $w = acbabcb$. Then $\tau(u_1, w)$ if we use the homomorphism h_1 defined by $h_1(a) = a$, $h_1(c) = cbabc$ and $h_1(b) = b$. Also $\tau(u_2, w)$ if we use the homomorphism h_2 defined by $h_2(a) = acba$, $h_2(b) = bcb$ and $h_2(c) = c$. □

LEMMA 1.1: *Let* Σ *be an alphabet. Then*

(i) for every $w \in \Sigma^*$ there exists a $u \in \Sigma^*$ such that $\tau(u, w)$, and

(ii) the number of types in Σ^* is finite.

Proof: (i) Let $w \in \Sigma^*$. We will prove part (i) of the lemma by induction on $\# alph(w)$.

If $\# alph(w) = 0$ then clearly $\tau(\Lambda, w)$.

If $\# alph(w)=1$ then, for some $a\in\Sigma$ and $n\in Z^+ w=a^n$. Hence $\tau(a, w)$. Assume that the lemma holds whenever $\# alph(w)<m$ where $m\in N$, $m\geq 2$. Let now $\# alph(w)=m$.

If no letter from Σ occurs twice in w then $\tau(w, w)$.

Otherwise write w in the form $w=w_1aw_2aw_3$ where $w_1, w_2, w_3\in\Sigma^*$, $a\in\Sigma$, $a\notin alph(w_1)$, $a\notin alph(w_3)$ and if $w_1\neq\Lambda$ then every letter from $alph(w_1)$ occurs exactly once in w.

By the inductive assumption, there exists a $u_3\in\Sigma^*$ such that $\tau(u_3, w_3)$; let h_3 be a homomorphism involved. Now we define the homomorphism h of Σ^* as follows: for $b\in\Sigma$, $h(b)=b$ if $b\in alph(w_1)$, $h(b)=aw_2a$ if $b=a$ and $h(b)=h_3(b)$ if $b\in alph(w_3)$. Clearly h satisfies condition (ii) of Definition 1.3 and so it is easily seen that $\tau(w_1au_3, w)$.

This completes the inductive step and consequently part (i) of the lemma holds.

(ii) Obviously the number of types in Σ^* equals $\sum_{r=0}^{n} r!$ where $n=\#\Sigma$. □

In the sequel of this paper we will consider an arbitrary but fixed context-free grammar G in Chomsky Normal Form, $G=(\Sigma, \Delta, P, S)$ such that $L(G)$ is infinite (here Σ is the total alphabet of G, Δ its terminal alphabet, P its set of productions and S its axiom). We will use D_G to denote the set of all derivation trees in G. The following construction is very essential for our paper.

CONSTRUCTION 1.1: Let $T\in D_G$ and let $\rho=v_0v_1\ldots v_s$ be a path in T where $s\geq 1$, v_0 is the root of T, v_s is a leaf of T and $l(v_0), l(v_1), \ldots, l(v_s)$ are the node labels corresponding to nodes of ρ. Let $Q_\rho=((v_{i_{11}}, v_{i_{12}}), \ldots, (v_{i_{r1}}, v_{i_{r2}}))$ be a sequence of pairs of nodes from ρ such that $r\geq 0$, $i_{j1}<i_{j2}$ for $1\leq j\leq r$, $i_{j2}\leq i_{(j+1)1}$ for $1\leq j\leq r-1$ if $r\geq 2$ and $l(v_{i_{j1}})=l(v_{i_{j2}})$ for $1\leq j\leq r$. Let f be a function from $\{1, \ldots, r\}$ into $\{L, R\}$; for $1\leq j\leq r$, $f(j)$ is the *label* of $(v_{i_{j1}}, v_{i_{j2}})$.

Let $T(\rho, Q_\rho, f)$ be a tree obtained from T as follows. Successively for each $j=1, \ldots, r$ perform the following:

— if $f(j)=L$ delete from T every subtree U such that its root, *root*(U), is to the left of ρ and the direct ancestor of *root*(U) in T is among the nodes $\{v_{i_{j1}}, v_{i_{j1}+1}, \ldots, v_{i_{j2}-1}\}$;

— if $f(j)=R$ delete from T every subtree U such that *root*(U) is to the right of ρ and the direct ancestor of *root*(U) in T is among the nodes $\{v_{i_{j1}}, v_{i_{j1}+1}, \ldots, v_{i_{j2}-1}\}$. □

Example 1.2: A derivation tree $T\in D_G$ looks a follows (Fig. 1) where ρ is the path consisting of nodes 1 through 10. Clearly *yield*(T) $=a(bc)^2bab^2cb$.

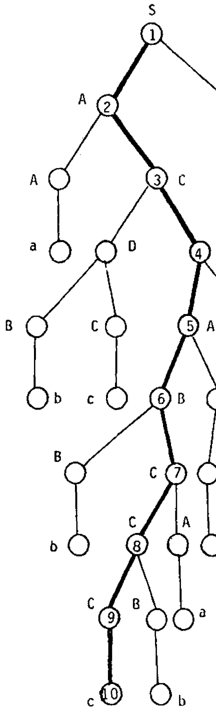

FIG. 1

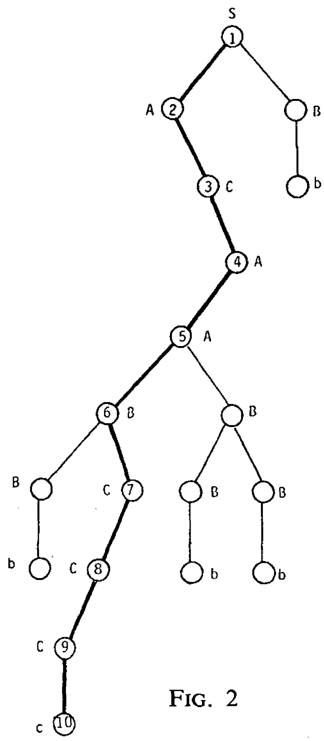

FIG. 2

Let $Q_\rho=((2,4),(4,5),(7,9))$ and $f((2,4))=L$, $f((4,5))=R$ and $f((7,9))=R$. Then $T(\rho, Q_\rho, f)$ looks as in Fig. 2. Note that $yield(T(\rho, Q_\rho, f))=bcb^2b$.

Note that, in general $T(\rho, Q_\rho, f)$ does not have to be a derivation tree in G. However, $T(\rho, Q_\rho, f)$ has a frontier and so its word, $yield\ (T(\rho, Q_\rho, f))$ is well defined. If T' is a tree such that $T'=T(\rho, Q_\rho, f)$ for some ρ, Q_ρ and f then we say that the *prune relation* holds between T and T' and we write $prune(T, T')$. Then we define

$$\mathrm{PR}(D_G)=\{\, T' : \text{there exists a } T\in D_G \text{ such that } prune(T, T')\,\}.$$

The usefulness of "pruned versions" of derivation trees in G stems from the following result.

LEMMA 1.2: *Assume that $SIP(L(G))\subseteq VDSIP(L(G))$. Then $yield(T')\in L(G)$ for every $T'\in PR(D_G)$.*

Proof: Let $T'\in \mathrm{PR}(D_G)$ and let $T\in D_G$ be such that $prune(T, T')$; let ρ, Q_ρ and f be such that $T'=T(\rho, Q_\rho, f)$. Let $yield(T)=w$.

Let $Q_\rho=((v_{i_{11}}, v_{i_{12}}), \ldots, (v_{i_{r1}}, v_{i_{r2}}))$ where $\rho=v_0v_1 \ldots v_s$, $s\geq 1$. If $r=0$ then obviously $T=T'$ and the lemma holds.

Assume then that $r \geq 1$.

Let $w = w_1 z w_2$ where the depicted occurence of a subword $z \in \Delta^+$ is the contribution of $v_{i_{r2}}$ to w.

Let

x_1 be the contribution to w_1 of the sequence of nodes $v_0, \ldots, v_{i_{11}-1}$ (if this sequence is empty then $x_1 = \Lambda$) through nodes to the left of ρ,

y_1 be the contribution to w_1 of the sequence of nodes $v_{i_{11}}, \ldots, v_{i_{12}-1}$ through nodes to the left of ρ,

and, for $2 \leq j \leq r$,

y_j be the contribution to w_1 of the sequence of nodes $v_{i_{j1}}, \ldots, v_{i_{j2}-1}$ through nodes to the left to ρ,

if $i_{j1} = i_{(j-1)2}$ then $x_j = \Lambda$,

otherwise x_j is the contribution to w_1 of the sequence of nodes $v_{i_{(j-1)2}}, v_{i_{(j-1)2}+1}, \ldots, v_{i_{j1}-1}$ through nodes to the left of ρ.

Analogously to the sequence $x_1, y_1, \ldots, x_r, y_r$, we define the sequence $t_1, u_1, \ldots, t_r, u_r$ where the only difference is that we consider the contributions of the appropriate sequences of nodes on ρ to w_2 (through nodes to the right of ρ) rather than to w_1.

From the way that the sequence $x_1, y_1, \ldots, x_r, y_r, z, u_r, t_r, \ldots, u_1, t_1$ was constructed it immediately follows that

$$p = (x_1, \ldots, x_r, y_1, \ldots, y_r, z, u_r, \ldots, u_1, t_r, \ldots, t_1) \in \mathrm{GSIP}(L(G)).$$

Hence by Theorem 1.1 and the assumption of the lemma it follows that $p \in \mathrm{VDGSIP}(L(G))$.

We notice now that

$$yield(T') = x_1 y_1^{n_1} \ldots x_r y_r^{n_r} z u_r^{m_r} t_r \ldots u_1^{m_1} t_1$$

where, for $1 \leq j \leq r$, $n_j = 0$ and $m_j = 1$ if $f(j) = L$, while $n_j = 1$ and $m_j = 0$ if $f(j) = R$.

Since $p \in \mathrm{VDGSIP}(L(G))$, $yield(T') \in L(G)$ and so the lemma holds. □

The following construction marking a fixed path in a derivation tree allows one to retain enough information in specially marked (labelled) nodes of the path to be able to produce derivation trees (with special properties) starting with such a marked path only.

Let $\overline{\Sigma} = \{(A, B, C, k) : k \in \{1, 2\}$ and $A \to BC \in P\} \cup \{(A, a) : A \to a \in P\} \cup \Delta$; we refer to $\overline{\Sigma}$ as the *marking alphabet* (of G).

CONSTRUCTION 1.2: Let $T \in D_G$ and let $\rho = v_0 v_1 \ldots v_s$ be a path in T where $s \geq 1$, v_0 is the root of T, v_s is a leaf of T and $l(v_0), l(v_1), \ldots, l(v_s)$ are the labels corresponding to nodes of ρ. Now for each node v_j, $0 \leq j \leq s$, change its label to $\bar{l}(v_j)$ as follows:

(1) if $A \to BC$ is the production used to rewrite the node j (hence $l(v_j)=A$) and v_j has a direct descendant to the left of ρ, then $l(v_j)$ is changed to $\bar{l}(v_j)=(A, B, C, 1)$,

(2) if $A \to BC$ is the production used to rewrite the node j and v_j has a direct descendant to the right of ρ, then $l(v_j)$ is changed to $\bar{l}(v_j)=(A, B, C, 2)$,

(3) if $A \to a$ is the production used to rewrite the node j then $l(v_j)$ is changed to $\bar{l}(v_j)=(A, a)$, and

(4) $\bar{l}(v_s)=l(v_s)$.

The resulting tree is called the *marked* ρ-*version of* T and denoted by $\overline{T}(\rho)$. The word $\bar{l}(v_0) \ldots \bar{l}(v_s)$ is referred to as the *spine of* $\overline{T}(\rho)$ and denoted by $spine(\overline{T}(\rho))$. □

Example 1.3: Let $T \in D_G$ be as follows (Fig. 3) where ρ consists of nodes 1 through 7.

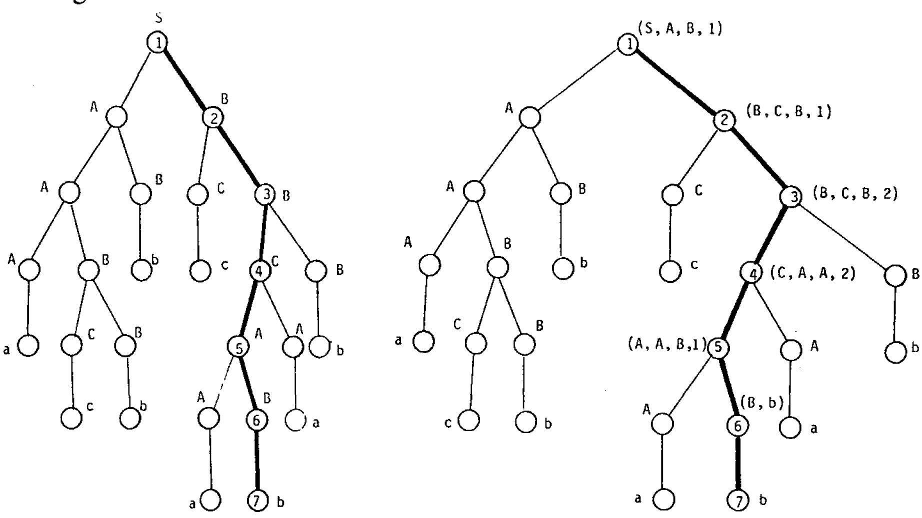

FIG. 3 FIG. 4

Then $\overline{T}(\rho)$ looks as in Fig. 4 and
$spine(\overline{T}(\rho))=(S, A, B, 1)(B, C, B, 1)(B, C, B, 2)(C, A, A, 2)(A, A, B, 1)(B, b)b.$

2. THE MAIN RESULT

In this section we prove the main result of this paper which states that if every strong iterative pair of a context-free language K is very degenerate, then K is a regular language.

We start by defining a ternary relation $\mu \subseteq \Sigma^+ \times \bar{\Sigma}^+ \times \Sigma^*$, a binary relation $\delta \subseteq \Sigma^+ \times \bar{\Sigma}^+$ and a function Θ from $M_{L(G)}$, the set of nonempty prefixes of $L(G)$, into the set of types in $\bar{\Sigma}^*$ as follows:

(i) for $w \in \Sigma^+$, $z \in \bar{\Sigma}^+$ and $u \in \Sigma^*$, $\mu(w, z, u)$ if and only if $wu \in L(G)$ and there exists a derivation tree T of wu in G and there exists a path ρ in T ending on the last (occurrence of a) letter of w such that *spine* $\bar{T}(\rho)) = z$,

(ii) for $w \in \Sigma^+$, $z \in \bar{\Sigma}^+$, $\delta(w, z)$ if and only if there exists a $u \in \Sigma^*$ such that $\mu(w, z, u)$,

(iii) for $w \in M_{L(G)}$, $\Theta(w) = \{ x \in \bar{\Sigma}^+ : \tau(x, z)$ and $\delta(w, z)$ for some $z \in \bar{\Sigma}^+ \}$.

The following lemma forms the major step in proving our main result.

LEMMA 2.1: *Let $w, w' \in M_{L(G)}$. If $\Theta(w) = \Theta(w')$ then $w \sim_{L(G)} w'$.*

Proof: Clearly, to prove the lemma it suffices to show that for every $u \in \Sigma^*$ if $\Theta(w) = \Theta(w')$ and $wu \in L(G)$ then $w'u \in L(G)$. . . . . (*).

To this aim we proceed as follows.

Let $u \in \Sigma^*$ be such that $wu \in L(G)$. Consider a derivation tree T of wu in G. Let ρ be a path in T beginning in the root of T and ending on the last (occurrence of a) letter of w. Consider $\bar{T}(\rho)$ and let $z = spine(\bar{T}(\rho))$.

Let $x \in \bar{\Sigma}^+$ be such that $\tau(x, z)$, say $x = X_1 \ldots X_s$, $s \geq 1$, where $X_j \in \bar{\Sigma}$ for $1 \leq j \leq s$. Let h be a homomorphism satisfying condition (ii.1) of Definition 1.3 (with Σ replaced by $\bar{\Sigma}$) such that $h(x) = z$. Let $z = z_1 \ldots z_s$ where $z_j = h(X_j)$ for $1 \leq j \leq s$.

Since $\Theta(w) = \Theta(w')$, $x \in \Theta(w')$. Thus there exist $u' \in \Sigma^*$, a derivation tree T' of $w'u'$ in G, a path ρ' in T' beginning in the root of T' and ending on the last (occurrence of a) letter of w' such that $spine(\bar{T}'(\rho')) = z'$ where $\tau(x, z')$.

Let h' be a homomorphism satisfying condition (ii.1) of Definition 1.3 (with h replaced by h' and Σ replaced by $\bar{\Sigma}$) such that $h'(x) = z'$.

Let $z' = z'_1 \ldots z'_s$ where $z'_j = h'(X_j)$ for $1 \leqslant j \leqslant s$.

Let $t \in \bar{\Sigma}^+$ be such that $t = t_1 \ldots t_s$ where, for $1 \leq j \leq s$, $t_j = (z_j/last(z_j))z'_j$; each t_j is referred to as the j'th *block* of t.

Note that such a j'th block t_j must be of one of the following four categories.

CATEGORY 1: If $|z_j| \geq 2$ and $|z'_j| \geq 2$ then $t_j = ay_1ay_2a$ where $a \in \bar{\Sigma}$, $y_1, y_2 \in \bar{\Sigma}^*$, $z_j = ay_1a$ and $z'_j = ay_2a$. We will refer to the three depicted occurences of a in t_j as the *first*, the *middle* and the *last pointer* of t_j respectively; y_1 and y_2 are referred as the *first* and the *last* bridge of t_j respectively.

CATEGORY 2: If $|z_j| \geq 2$ and $|z'_j| = 1$ then $t_j = ay_1a$ where $a \in \bar{\Sigma}$, $y_1 \in \bar{\Sigma}^*$,

$z_j = ay_1a$ and $z'_j = a$. We will refer to the two depicted occurrences of a in t_j as the *first* and the *last pointer* of t_j respectively; y_1 is referred as the *bridge* of t_j.

CATEGORY 3: If $|z_j| = 1$ and $|z'_j| \geq 2$ then $t_j = ay_2a$ where $a \in \overline{\Sigma}$, $y_2 \in \overline{\Sigma}^*$, $z_j = a$ and $z'_j = ay_2a$. We will refer to the two depicted occurences of a in t_j as the *first* and the *last pointer* of t_j respectively; y_2 is referred as the *bridge* of t_j.

CATEGORY 4: If $|z_j| = |z'_j| = 1$ then $t_j = a$ where $a \in \overline{\Sigma}$ and $z_j = z'_j = a$.

CLAIM 2.1: There exists a derivation tree U in G and a path γ in U such that $t = spine(\overline{U}(\gamma))$.

Proof of the claim: This follows easily from the observation that every two consecutive letters in t are either two consecutive letters in z or two consecutive letters in z'. □

Note that, clearly, such a path γ together with direct descendant nodes attached to it is (up to node isomorphism) uniquely determined by t. The so formed tree will be denoted by $sur(t)$. The word t induces the obvious division of path γ in $sur(t)$ into consecutive *segments* $\gamma_1, \ldots, \gamma_s$ corresponding to blocks $t_1, \ldots, t_s$ respectively. In this way we can talk about the nodes of γ_j, $1 \leq j \leq s$, which are the (first, middle or last) pointers of γ_j or which are nodes of the (first or last) bridge of γ_j. We say that γ_j, $1 \leq j \leq s$, is of Category i, $1 \leq i \leq s$, if t_j is of Category i.

Also the nodes in $sur(t)$ which are not on γ are called the *outside nodes* (of $sur(t)$); the outside nodes to the right of γ are called *right outside nodes*, similarly we get *left outside nodes*. By construction of t, these outside nodes correspond uniquely either to nodes of T or to nodes of T'; to simplify terminology we will say that *they are* from T or from T'.

We will extend now $sur(t)$ into a derivation tree in G as follows. Consider one by one each segment γ_j of γ, $1 \leq j \leq s$.

Assume that γ_j is of Category 1. From the definition of t_j it follows immediately that either for each pointer of γ_j its outside direct descendant is a right outside node (Case 1) or for each pointer of γ_j its outside direct descendant is a left outside node (Case 2). If Case 1 holds then we replace the outside direct descendant node e_1 of the first pointer by the subtree of T rooted at e_1 (remember that, according to our terminology, e_1 is also a node of T). The tree isomorphic to this one (with corresponding labels being the same) replaces also the outside direct descendant node e_m of the middle pointer of γ_j. The outside direct descendant node e_l of the last pointer of γ_j is replaced by the subtree of T' rooted at e_l.

If Case 2 holds then we replace the outside direct descendant node e_1 of the first pointer of γ_j by the subtree of T' rooted at e_1. The tree isomorphic to this one (with corresponding labels being the same) replaces also the outside direct descendant node e_m of the middle pointer of γ_j. The outside direct descendant node e_l of the last pointer of γ_j is replaced by the subtree of T' rooted at e_l.

In both cases each outside direct descendant e of a node on the first bridge is replaced by the subtree of T rooted at e and each outside direct descendant e of a node on the second bridge is replaced by the subtree of T' rooted at e.

If γ_j is either of Category 2 or of Category 3 then the process is quite analogous except that we do not have (outside direct descendants of) middle pointers to process. If γ_j is of Category 2 then outside direct descendants of nodes on the bridge are replaced by appropriate subtrees from T while if γ_j is of Category 3 then outside direct descendants of nodes on the bridge are replaced by appropriate subtrees from T'.

If γ_j is of Category 4 then nodes in γ_j do not leave direct descendants.

In this way we have extended $sur(t)$ into a derivation tree in G; this tree will be denoted by $\mathrm{SUR}(t)$.

The last step of our construction needed to prove (*), and hence to prove Lemma 2.1, is to construct the tree $\mathrm{SUR}'(t)$ such that $prune(\mathrm{SUR}(t), \mathrm{SUR}'(t))$ holds.

Consider γ. For each block γ_j of γ, $1 \leq j \leq s$, we do the following. If γ_j is of Category 1 then it yields two pairs of nodes: (p_{j1}, p_{jm}) followed by (p_{jm}, p_{jl}) where p_{j1}, p_{jm} and p_{jl} are the first, the middle and the last pointer of γ_j respectively. Then (p_{j1}, p_{jm}) is referred to as the *first pair* of γ_j and (p_{jm}, p_{jl}) is referred to as the *second pair* of γ_j.

If γ_j is of Category 2 or 3 then it yields one pair of nodes: (p_{j1}, p_{jl}) where p_{j1} is the first and p_{jl} is the last pointer of γ_j.

If going from $j=1$ to $j=s$ we select each block γ_j of γ that is of Category 1, 2 or 3 and form the sequence of pairs of nodes described above in this order (where for γ_j of Category 1 the first pair comes before the second), then we get the sequence Q_γ of pairs of nodes from γ.

Now to each pair from Q_γ the function f assigns either L or R as follows.

If γ_j is of Category 1 then f assigns L to its first pair and R to its second pair.

If γ_j is of Category 2 then f assigns L to its pair.
If γ_j is of Category 3 then f assigns R to its pair.
By the above construction we have obtained the tree

$$(\mathrm{SUR}(t))(\gamma, Q_\gamma, f) = \mathrm{SUR}'(t).$$

It follows directly from the construction of $\mathrm{SUR}(t)$ and $\mathrm{SUR}'(t)$ that $yield(\mathrm{SUR}'(t)) = w'u$. Hence by Lemma 1.2 it follows that $w'u \in L(G)$ and consequently (*) holds. Clearly (*) implies the lemma. □

We are ready now to prove the main result of this paper.

THEOREM 2.1: *Let K be a context-free language such that $SIP(K) \subseteq VDSIP(K)$. Then K is regular.*

Proof: Let $G = (\Sigma, \Delta, P, S)$ be a Λ-free context-free grammar in Chomsky Normal Form generating K. Consider two arbitrary words $w, w' \in M_K$. By Lemma 2.1, if $\Theta(w) = \Theta(w')$ then $w \sim_K w'$. But by Lemma 1.1 (ii) the number of types in $\bar{\Sigma}^*$ is finite and consequently $(\sim_K)_{M_K}$ is of finite index. Thus by Theorem 0.2, K is regular. □

We recall now the notion of an iterative pair as originally defined in [2], *see* also [1].

DEFINITION 2.1: Let K be a language, $K \subseteq \Sigma^*$. An *iterative pair*, abbreviated IP, of K is a 5-tuple $p = (x, y, z, u, t)$ where $x, y, z, u, t \in \Sigma^*$, $yu \neq \Lambda$ and, for every $n \in Z^+$, $xy^n zu^n t \in K$. We say that p is a *very degenerate iterative pair*, abbreviated VDIP, of K if, for every $n, m \in \mathrm{N}$, $xy^n zu^m t \in K$. □

For a language K, $\mathrm{IP}(K)$ will denote the set of iterative pairs of K and $\mathrm{VDIP}(K)$ will denote the set of very degenerate iterative pairs of K.

Thus the difference between the strong iterative pair and an iterative pair is that erasing of (the second and the fourth) components of a pair is allowed if it is a SIP but not allowed if it is an IP.

The following result is from [2]; we demonstrate now how it can be easily obtained from our result.

COROLLARY 2.1: *Let K be a context-free language such that $IP(K) \subseteq VDIP(K)$. Then K is regular.*

Proof: Since $\mathrm{SIP}(K) \subseteq \mathrm{IP}(K)$, $\mathrm{SIP}(K) \subseteq \mathrm{VDIP}(K)$ and consequently $\mathrm{SIP}(K) \subseteq \mathrm{VDSIP}(K)$. Thus, by Theorem 2.1, K is regular. □

ACKNOWLEDGMENTS

The authors are indebted to R. Verraedt for useful comments concerning the first draft of this paper. They also gratefully acknowledge the financial support of NSF grant MCS 79-03838.

REFERENCES

1. J.-M. AUTEBERT, J. BEAUQUIER, L. BOASSON and M. LATTEUX, Very small families of algebraic nonrational languages, in: *Formal Language Theory*, Book, R. (ed.), Academic Press, London-New York, 1981.
2. L. BOASSON, Un critère de rationalité des langages algébriques, in: *Automata, Languages and Programming*, Nivat, M. (ed.), North-Holland Publ. Comp., Amsterdam, 1973.
3. L. BOASSON, Private communication.
4. M. A. HARRISON, *Introduction to formal language theory*, Addison-Wesley, Reading, Mass., 1978.
5. A. SALOMAA, *Formal languages*, Academic Press, London-New York, 1973.