Pure languages of regulated rewriting and their codings[1)]

By J. Dassow

Dedicated to Prof. Herbert Goering in the occasion of his 60th birthday

0. Introduction

Since context-free grammars are not able to cover all aspects which are of interest (e.g. in the theory of programming languages), a lot of regulating mechanisms for the derivation process have been introduced. However, mostly the generative capacity of these mechanisms has been studied with respect to the generated set of words over a terminal alphabet. Thus all the intermediate steps of the derivation are not contained in the language, and therefore it often happens that the differences between the mechanisms disappear. Hence — in order to contribute to a comparison of the mechanisms — we shall investigate languages which contain also the words of the intermediate steps, the so-called pure languages.

Let us also mention that these languages are of interest for themselves by the following two facts:

— they form a sequential counterpart to the L systems (with regulation),

— the intermediate steps are important for the syntax analysis.

The first results on pure versions of grammars with regulated rewriting are presented in (1), (6), (2). However, besides (6) (where a different definition of the pure language is used) the appearance checking mode is used in the derivation. One of the purposes of this paper is to complete the hierarchy by the relations concerning pure languages of regulated rewriting without appearance checking.

Usual languages (containing only terminal words) can be obtained from a pure language by intersecting with the set of all words over the terminal alphabet. By the results by Ehrenfeucht and Rozenberg it is known that for L systems there is a second way, namely the application of a coding (letter-to-letter homomorphisms), if one is

[1)] Paper presented at the 4th Hungarian Computer Science Conference, Győr, July, 8—10, 1985.

only interested in the generated family of languages (see (7)). This does not hold for sequential rewriting. But in (3) it is shown that the hierarchy of families of languages obtained by codings lies between that of pure languages and that of usual languages. Again, here we shall add some results by consideration of grammars without appearance checking.

In this paper we shall restrict to the following three types of regulated devices: matrix grammars, programmed grammars, and random context grammars.

Throughout the paper we assume that the reader is familiar with the rudiments of formal language theory and has some information on regulated rewriting (e.g. see (8), (5)).

1. Definitions

For the sake of completeness we give the formal definitions for the pure versions of the above mentioned grammars.

In the following definitions, let V be an alphabet, and let S be a finite subset of V^+. (Usually in the theory of pure languages one uses a set of starting words, however, as one can see by our proofs our results do not change if S consists of only one word.)

A *pure random context* grammar is a triple $G=(V, P, S)$ where P is a finite set of productions of the form

$$(a \to w, R, Q),\ a\in V,\ w\in V^*,\ R\subseteq V,\ Q\subseteq V.$$

We say that $x\in V^+$ directly derives $y\in V^*$ (written as $x \Rightarrow y$) iff $x=z_1az_2$, $y=z_1wz_2$, $(a\to w, R, Q)\in P$, z_1z_2 contains all letters of R, and z_1z_2 contains no letter of Q. The language $L(G)$ generated by G is defined as

$$L(G) = \{y\colon\ z \overset{*}{\Rightarrow} y \quad \text{for some} \quad z\in S\}$$

where $\overset{*}{\Rightarrow}$ denotes the reflexive and transitive closure of $\Rightarrow$.

A *pure programmed* grammar is a triple $G=(V, P, S)$ where P is a finite set of rules of the form

$$(b,\ a \to w,\ E(b),\ F(b))$$

where b is a label of the production, $a\in V$, $w\in V^*$, and $E(b)$ and $F(b)$ are subsets of the set of labels. The language $L(G)$ consists of all words y such that there is a derivation

$$z = y_0 \underset{b_1}{\Longrightarrow} y_1 \underset{b_2}{\Longrightarrow} y_2 \underset{b_3}{\Longrightarrow} \dots \underset{b_{n-1}}{\Longrightarrow} y_{n-1} \underset{b_n}{\Longrightarrow} y_n = y$$

where $z\in S$, $(b_i, a_i\to w_i, E_i, F_i)$ are rules of P, $1\leq i\leq n$, and, for $1\leq i\leq n$,

$$y_{i-1} = z_{i1}a_iz_{i2},\ y_i = z_{i1}w_iz_{i2} \quad \text{for some} \quad z_{i1}, z_{i2}\in V^*,$$
$$\text{and} \quad b_{i+1}\in E_i \quad (\text{if} \quad i < n)$$

or

a_i does not occur in y_{i-1}, $y_i = y_{i-1}$, and $b_{i+1}\in F_i$ (again, if $i < n$).

A *pure matrix* grammar is a quadruple $G=(V, M, S, F)$ where M is a finite set of finite sequences of productions,

$$M = \{m_1, m_2, \dots m_r\},$$

$$m_i = (a_{i1} \to w_{i1},\ a_{i2} \to w_{i2}, \dots, a_{ir(i)} \to w_{ir(i)}),$$

$a_{ij} \in V$, $w_{ij} \in V^*$, $1 \leqq i \leqq r$, $1 \leqq j \leqq r(i)$, and F is a subset of occurrences of rules in M. Then, for $1 \leqq i \leqq r$, we say that $x \underset{m_i}{\Longrightarrow} y$ iff

$$x = y_0 \Rightarrow y_1 \Rightarrow y_2 \Rightarrow \dots \Rightarrow y_{r(i)} = y$$

where

$$y_{j-1} = z_{j1} a_{ij} z_{j2},\ y_j = z_{j1} w_{ij} z_{j2} \quad \text{for some} \quad z_{j1}, z_{j2} \in V^*$$

or

a_{ij} does not occur in y_{j-1}, $a_{ij} \to w_{ij} \in F$, and $y_j = y_{j-1}$.

The language $L(G)$ generated by G is defined as the set of all words y which are obtained by iterated applications of matrices (elements of M) to words of S and all intermediate words (y_j in the above notation) of these applications of matrices.

These definitions are the most general ones, i.e. rules of the form $a \to \lambda$ are allowed and the appearance checking mode is used in the derivation process.

By $\mathscr{L}(pRC^{\lambda}_{ac})$, $\mathscr{L}(pPR^{\lambda}_{ac})$, $\mathscr{L}(pM^{\lambda}_{ac})$ we denote the families of languages obtained by pure random context, pure programmed, and pure matrix grammars, respectively. We omit the upper or lower index or both indices if we consider the families of languages generated by grammars without erasing rules or without appearance checking (i.e. $Q=\emptyset$ and $F(b)=\emptyset$ for all productions, or $F=\emptyset$) or without both these features.

By $\mathscr{L}(pCF)$ and $\mathscr{L}(pCS)$ we denote the families of pure context-free and pure context-sensitive languages, respectively (the definitions can be given in an obvious way, e.g. see (4)), and we add the upper index λ if λ-rules are allowed.

$\#_a(w)$ denotes the number of occurrences of the letter a in the word w.

2. The hierarchy of pure language families

Let us consider the pure programmed grammar

$$G_1 = (\{a, b\}, \{(1, a \to b^3, \{1\}, \{2\}), (2, b \to a^3, \{2\}, \{1\})\}, \{a\}).$$

The language L_1 generated by G_1 contains only words of $\{a, b\}^+$ which satisfy one of the following conditions:

$$\#_a(w) + \frac{\#_b(w)}{3} = 3^{2n}, \qquad \frac{\#_a(w)}{3} + \#_b(w) = 3^{2n+1}, \tag{1}$$

where $n \in N$.

Lemma 1. $L_1 \in \mathscr{L}(pPR_{ac})$, $L_1 \notin \mathscr{L}(pPR^{\lambda})$.

Proof. By the construction of L_1 we have to prove only the second statement. Let us assume the contrary, i.e. $L_1 = L(G)$ for some pure programmed grammar G without appearance checking. First we note that, for w, $w' \in L_1$, $w \neq w'$, $|l(w) - l(w')| \geqq 2$ holds. Hence without loss of generality we can assume that G is λ-free. If there is a rule whose core production is of the form $a \to b$ or $b \to a$ then its application to a word of L_1 produces a word which do not belong to L_1.

For a production $p: (h, x \to w, E(h), \emptyset)$, we set

$$l(p) = l(w) - 1,$$

and we also define

$$l(G) = \max_{p \in P} l(p).$$

Obviously, $l(G) \geqq 2$. Further, let t_1 be the number of productions in G, let t_2 be the maximal length of a word in S, and let n be an integer such that $3^n > t_1 \cdot t_2 \cdot l(G)$. Now we consider a derivation D of $a^{3^{2n}} b^3 \in L_1$, especially the last $(t_1 + 2)$ steps of this derivation which increases the length, i.e.

$$D: s \overset{*}{\Longrightarrow} y_0 \underset{l_0}{\Longrightarrow} y_1 \overset{*}{\Longrightarrow} y_1 \underset{l_1}{\Longrightarrow} y_2 \overset{*}{\Longrightarrow} y_2 \Rightarrow \ldots \underset{l_{t_1}}{\Longrightarrow} y_{t_1+1} \overset{*}{\Longrightarrow} y_{t_1+1} \underset{l_{t_1+1}}{\Longrightarrow} a^{3^{2n}} b^3$$

where the derivation steps $y_i \Rightarrow y_{i+1}$ are obtained by application of the production $(l_i, x_i \to w_i, E_i, \emptyset)$ of G with $l(w_i) \geqq 2$ and the phases $y_i \overset{*}{\Rightarrow} y_i$ contain only applications of rules with core productions $a \to a$ or $b \to b$. By the definition of n and t_1, the following facts are valid:

— $\dfrac{\#_a(y_i)}{3} + \#_b(y_i) = 3^{2n-1}$ for $1 \leqq i \leqq t_1$,

— there are two integers k, j with $0 \leqq k < j \leqq t_1$ such that $l_k = l_j$.

Let

$$\#_a(y_k) = 3^{2n} - 3r_1, \quad \#_b(y_k) = r_1,$$
$$\#_a(y_j) = 3^{2n} - 3r_2, \quad \#_b(y_j) = r_2.$$

Then $3^n > r_1 - r_2 > 0$. Further we have also the correct derivation

$$s \overset{*}{\Longrightarrow} y_0 \underset{l_0}{\Longrightarrow} y_1 \overset{*}{\Longrightarrow} \ldots \Rightarrow y_k \underset{l_j}{\Longrightarrow} z_j \overset{*}{\Longrightarrow} z_j \underset{l_{j+1}}{\Longrightarrow} z_{j+1} \Rightarrow \ldots \underset{l_{t_1+1}}{\Longrightarrow} z_{t_1+1}$$

where $z_j, \ldots, z_{t_1+1}$ are appropriate strings with

$$\#_a(z_{t_1+1}) = 3^{2n} - 3(r_1 - r_2), \quad \#_b(z_{t_1+1}) = 3 + (r_1 - r_2).$$

This contradicts (1) and thus $z_{t_1+1} \notin L_1$. Since $z_{t_1+1} \in L(G)$ we obtain the desired contradiction to $L(G) = L_1$. □

Now we consider the pure matrix grammar

$$G_2 = (\{a, b, c, d\}, \{(a \to a^3, b \to b^3, c \to c^3), (a \to a^3, b \to d^3, c \to c^3)\},$$
$$\{abc\}, \{b \to b^3, b \to d^3\}).$$

Then the words of its generated language L_2 satisfy the following conditons:

If $\#_b(w) = 0$, then $w = a^{2n+1} d^{3m} c^{2n-1}$ or $w = a^{2n+1} d^{3m} c^{2n+1}$ with $2n+1 > 3m$, $n, m \in N$.

If $\#_b(w) \geqq 1$, then $w = a^{2n+1}w'c^{2n+1}$ or $w = a^{2n+1}w'c^{2n-1}$ where

$$w' \in \{b, d^3\}^+, \; l(w') = 2n+1 \quad \text{or} \quad l(w') = 2n-1.$$

Lemma 2. $L_2 \in \mathscr{L}(pM_{ac})$, $L_2 \notin \mathscr{L}(pM^\lambda)$.

Proof. Again, we have to prove only the second statement. Let us assume that $L_2 = L(G)$ for some pure matrix grammar G without appearance checking. As in the preceding proof we can show that all core productions (besides $x \to x$) have the form $x \to w$ with $l(w) \geqq 3$. Let n be a sufficiently large number. We consider a derivation of $w = a^{2n+1}d^3c^{2n+1}$, say

$$D: \; s \overset{*}{\Longrightarrow} v_1 \Rightarrow v_2 \Rightarrow w$$

where (without loss of generality) $l(v_1) < l(v_2) < l(w)$. By the structure of the words in L_2 it is easy to prove that $v_1 = a^{2n-1}d^3c^{2n-1}$, $v_2 = a^{2n+1}d^3c^{2n-1}$. Iterating this argument and taking into consideration that we can omit length preserving matrices we obtain

$$D: \; s \overset{*}{\Longrightarrow} u_1 \underset{m}{\Longrightarrow} u_2 \overset{*}{\Longrightarrow} w$$

where the derivation $u_2 \overset{*}{\Longrightarrow} w$ corresponds to the application of a proper initial part of a matrix or $u_2 = w$, and the application of m increases the number of a's and/or c's only. If it increases only the number of a's, then by its iterated application to u_2 we can generate a word y with $\#_a(y) - \#_c(y) > 2$ which contradicts the structure of the words in L_2. Analogously, the matrix m cannot only increase the number of c's. Hence it has to increase both numbers. Now we consider a derivation D' of $w' = a^{2n+1}b^{2n+1}c^{2n+1}$. Again,

$$D': \; s' \overset{*}{\Longrightarrow} z \overset{*}{\Longrightarrow} w'$$

where $z \overset{*}{\Longrightarrow} w'$ is the initial part of a matrix application or $z = w'$ and z is generated by iterated applications of matrices. Clearly, $\#_b(z) \geqq 1$. Then it is easy to show that the correct derivation

$$s' \overset{*}{\Longrightarrow} z \underset{m}{\Longrightarrow} z_1 \underset{m}{\Longrightarrow} z_2 \underset{m}{\Longrightarrow} z_3$$

produces a word z_3 which is not in L_2.

Lemma 3. *Let*

$$L_3 = \{a^2b^2c, ab^5c, b^3ab^2c, b^8c, b^{11}\} \cup \{a^{2n}b^5 \colon n \geqq 1\} \cup$$
$$\cup \{b^3a^{2n+1}b^2 \colon n \geqq 1\} \cup \{a^{2n+1}b^8 \colon n \geqq 1\}.$$

Then $L_3 \in \mathscr{L}(pRC_{ac})$, $L_3 \notin \mathscr{L}(pRC^\lambda)$.

Proof. The pure random context grammar $G_3 = (\{a, b, c\}, \{(c \to b^3, \emptyset, \emptyset), (a \to a^3, \emptyset, \{c\}), (a \to b^3, \{c\}, \emptyset)\}, \{a^2b^2c\})$ generates L_3.

Assume that $L_3 = L(G)$ for some pure random context grammar without appearance checking. We consider $a^{2n}b^5$ where n is sufficiently large. This word can be derived only from a word $a^{2m}b^5$ (without loss of generality we can assume that $m < n$), and thus we have a production $(a \to a^{2(n-m)+1}, R, \emptyset)$ or $(b \to a^{2(n-m)}b, R, \emptyset)$

with $R \subseteq \{a, b, c\}$. This rule is applicable to a^2b^2c producing a word not contained in L_3.

Lemma 4. *Let*

$$L_4 = \{ca^n b^n\colon n \geqq 2\} \cup \{ca^{n+1} b^n\colon n \geqq 2\} \cup \{a^n b^n\colon n \geqq 2\}.$$

Then $L_4 \in \mathscr{L}(pM^\lambda)$, $L_4 \in \mathscr{L}(pM_{ac})$.

Proof. Clearly, L_4 is generated by the pure matrix grammar

$$G_4 = (\{a, b, c\},\ \{(c \to c,\ a \to a^2, b \to b^2), (c \to \lambda)\},\ \{ca^2 b^2\}).$$

Thus $L_4 \in \mathscr{L}(pM^\lambda)$.

Now assume $L_4 = L(G)$ for some pure matrix grammar G with appearance checking but without λ-rules. We consider the word $a^n b^n$ where n is chosen such that $a^n b^n$ is not an axiom. Then there is a word z with $z \Rightarrow a^n b^n$ and $z \neq a^n b^n$. If $z = a^r b^r$ for some $r < n$ ($r > n$ is impossible by the λ-freeness and $r = n$ is already excluded), then we have applied a rule of the form $a \to a^{s+1} b^s$ to the last a in z or $b \to a^s b^{s+1}$ to the first b in z where $s \geqq 1$. Since $r \geqq 2$ we can apply this rule to the first a or last b, too, and then we derive a word which is not in L_4. Hence z is of the form $ca^r b^r$ or $ca^{r+1} b^r$, and we have to apply a rule of the form $c \to z'$ which yields $z'a^r b^r$ or $z'a^{r+1} b^r$. In the first case $z' = \lambda$ and $r = n$ have to hold and this contradicts the λ-freeness of G. In the second case we do not obtain $a^n b^n$. Therefore $L_4 = L(G)$ do not hold for all λ-free pure matrix grammars G.

Lemma 5. *Let* $L_4' = \{d\} \cup L_4$. *Then* $L_4' \in \mathscr{L}(pPR^\lambda)$, $L_4' \in \mathscr{L}(pPR_{ac})$.

Proof. It is easy to see that $L_4' \in \mathscr{L}(pPR^\lambda)$, and $L_4' \notin \mathscr{L}(pPR_{ac})$ can be proved analogously to the proof of Lemma 4.

Lemma 6. *Let*

$$L_5 = \{ca^n b^n d\colon n \geqq 2\} \cup \{c'a^{n+1} b^n d\colon n \geqq 2\} \cup \{c'a^n b^n d'\colon n \geqq 3\} \cup$$
$$\cup \{ca^n b^n d'\colon n \geqq 3\} \cup \{a^n b^n d\colon n \geqq 2\}.$$

Then $L_5 \in \mathscr{L}(pRC^\lambda)$, $L_5 \notin \mathscr{L}(pRC_{ac})$.

Proof. The pure random context grammar

$$G_5 = (\{a, b, c, d, c', d'\}, \{(c \to c'a, \{d\}, \emptyset), (d \to bd', \{c'\}, \emptyset)$$
$$(c' \to c, \{d'\}, \emptyset),\ (d' \to d, \{c\}, \emptyset),\ (c \to \lambda, \{d\}, \emptyset)\},\ \{ca^2 b^2 d\})$$

generates L_5. Hence $L_5 \in \mathscr{L}(pRC^\lambda)$.

Now let $L_5 = L(G)$ for some pure random context grammar G (with appearance checking) without erasing rules. We consider $w = a^n b^n d$ with sufficiently large n. As in the proof of Lemma 4 it can be shown that w cannot be generated from a word of the form $a^r b^r d$. Hence z with $z \Rightarrow w$, $z \neq w$ has to be of the form $ca^r b^r d$ or $c'a^{r+1} b^r d$. It is easy to see that $a^n b^n d$ is obtained iff $c \to \lambda$ is applied to z and $r = n$ holds, i.e. we get a contradiction to the λ-freeness.

Lemma 7. *Let*

$$L_6 = \{b^2a\} \cup \{ba^{5+3n}\colon\ n \geqq 0\} \cup \{a^{3m+4}ba^{3n+1}\colon\ m, n \geqq 0\}.$$

Then $L_6 \in \mathscr{L}(pRC)$, $L_6 \notin \mathscr{L}(pPR^\lambda_{ac})$, $L_6 \notin \mathscr{L}(pM^\lambda_{ac})$.

Proof. i) L_6 is generated by the pure random context grammar

$$G_6 = \big(\{a, b\},\ \{(b \to a^4, \{b\}, \emptyset),\ (a \to a^4, \{a\}, \emptyset)\},\ \{b^2a\}\big).$$

Thus $L_6 \in \mathscr{L}(pRC)$.

ii) Assume that $L_6 = L(G)$ for some pure programmed grammar G. Again, G is λ-free, and hence b^2a has to be an axiom. Further, in order to generate ba^{5+3n} for a sufficiently large integer n we need a production with the core rule $a \to a^{3m+1}$ for some $m > 0$. Clearly, it can be applied to the axiom, and this gives $b^2a \Rightarrow b^2a^{3m+1}$. Therefore $L(G)$ contains the word b^2a^{3m+1} which is not in L_6.

iii) can be proved analogously to ii).

Fact 1. *Let*

$$L_7 = \{a^nb^nc^n\colon\ n \geqq 1\} \cup \{a^{n+1}b^nc^n\colon\ n \geqq 1\} \cup \{a^{n+1}b^{n+1}c\colon\ n \geqq 1\}.$$

Then $L_7 \in \mathscr{L}(pM)$, $L_7 \notin \mathscr{L}(pPR^\lambda_{ac})$, $L_7 \notin \mathscr{L}(pRC^\lambda_{ac})$.

Fact 2. *If* $L_8 = \{a, a^5\} \cup \{a^{7+10n}\colon n \geqq 0\} \cup \{a^{11+10n}\colon n \geqq 0\}$, then $L_8 \in \mathscr{L}(pPR)$, $L_8 \notin \mathscr{L}(pRC^\lambda_{ac})$, $L_8 \notin \mathscr{L}(pM^\lambda_{ac})$.

Summarizing all these results and taking into consideration the relations to pure versions of the grammars of the Chomsky hierarchy which are already given in (1) we obtain the following diagram. Instead of $A \subsetneqq B$ we write $A \to B$, and if two families are not connected then they are incomparable.

Theorem 8.

$$\begin{tikzcd}[column sep=tiny]
& \mathscr{L}(pRC^\lambda_{ac}) & & & \mathscr{L}(pM^\lambda_{ac}) & & & \mathscr{L}(pPR^\lambda_{ac}) & & \\
\mathscr{L}(pRC_{ac}) \arrow[ur] & & \mathscr{L}(pRC^\lambda) \arrow[ul] & \mathscr{L}(pM_{ac}) \arrow[ur] & & \mathscr{L}(pM^\lambda) \arrow[ul] & \mathscr{L}(pPR_{ac}) \arrow[ur] & & \mathscr{L}(pPR^\lambda) \arrow[ul] & \mathscr{L}(pCS^\lambda) \\
& \mathscr{L}(pRC) \arrow[ul] \arrow[ur] & & & \mathscr{L}(pM) \arrow[ul] \arrow[ur] & & & \mathscr{L}(pPR) \arrow[ul] \arrow[ur] \arrow[rr] & & \mathscr{L}(pCS) \arrow[u] \\
& & & & & \mathscr{L}(pCF) = \mathscr{L}(pCF^\lambda) \arrow[ullll] \arrow[ul] \arrow[urr] & & & &
\end{tikzcd}$$

3. Codings of pure languages

Let X be a family of grammars. Then we set $\mathscr{C}(X) = \{L\colon L = h(L')$ for some $L' \in \mathscr{L}(X)$ and some coding $h\}$ (i.e. $h(a)$ has the length 1 for all letters a).

Lemma 9. $\mathscr{C}(pM) = \mathscr{C}(pPR)$.

Proof. $\mathscr{C}(pM) \subseteq \mathscr{C}(pPR)$ can be proved analogously to (3). Now let $G = (V, P, S)$ be a pure programmed grammar. We set $V' = \{(a, b)\colon a \in V, b$ is a label of a production in $P\}$. With the production $(b, a \to w, E(b), \emptyset)$ we associate the matrices

$$((a, b) \to w,\ x \to (x, b'))$$

where $x \in V$, $b' \in E(b)$. The set S' is defined as the set of all words $w_1(x, b)w_2$ where $w_1xw_2 \in S$, $(x, b) \in V'$. We consider the pure matrix grammar $G' = (V \cup V', M, S', \emptyset)$ where M is the set of all matrices of the above introduced type. Then the correct derivations have the following form (in the second row we note the applied rule):

$$s = w_{11}(x_1, b_1)w_{21} \xRightarrow[(x_1, b_1) \to z_1]{} w_{11}z_1w_{21} \xRightarrow[x_2 \to (x_2, b_2)]{} w_{12}(x_2, b_2)w_{22} \ldots \Rightarrow$$

$$\ldots \Rightarrow w_{1n}(x_n, b_n)w_{2n} \xRightarrow[(x_n, b_n) \to z_n]{} w_{1n}z_nw_{2n}$$

$$\xRightarrow[x_{n+1} \to (x_{n+1}, b_{n+1})]{} w_{1,n+1}(x_{n+1}, b_{n+1})w_{2,n+1} \Rightarrow \ldots.$$

Further we consider the coding h given by $h(x) = x$ for $x \in V$ and $h((x, b)) = x$ for $(x, b) \in V'$. Then the image of the above derivation is

$$w_{11}x_1w_{21} \xRightarrow[b_1]{} w_{11}z_1w_{21} = w_{12}x_2w_{22} \xRightarrow[b_2]{} \ldots$$

$$\ldots \Rightarrow w_{n1}x_nw_{2n} \xRightarrow[b_n]{} w_{1n}z_nw_{2n} = w_{1,n+1}x_{n+1}w_{2,n+1} \xRightarrow[b_{n+1}]{} \ldots$$

where, by the definition of the matrices, b_{i+1} is in the success field $E(b_i)$ of b_i. Thus we have proved that

$$L(G) = h(L(G')).$$

Since the composition of codings is a coding again,

$$h'(L(G)) = (h \circ h')(L(G')),$$

and thus

$$\mathscr{C}(pPR) \subseteqq \mathscr{C}(pM).$$

Combining Lemma 9 with the results of (3) we obtain

Theorem 10.

$$\mathscr{C}(pRC) \subseteqq \mathscr{C}(pPR) = \mathscr{C}(pm) \subseteqq \mathscr{C}(pM_{ac}) \subsetneqq \mathscr{C}(pPR_{ac}) = \mathscr{C}(pRC_{ac}) \subsetneqq \mathscr{C}(pCS).$$

It is known (see (8), (5)) that for usual languages (i.e. sets of words over a terminal alphabet) the following hierarchy holds:

$$\mathscr{L}(RC) \subseteqq \mathscr{L}(M) = \mathscr{L}(PR) \subseteqq \mathscr{L}(M_{ac}) = \mathscr{L}(PR_{ac}) = \mathscr{L}(RC_{ac}) \subsetneqq \mathscr{L}(CS)$$

(and that for $X \in \{M, PR, RC\}$, $\alpha = ac$ or empty, all the families $\mathscr{L}(X_\alpha^\lambda)$ coincide with the family of all recursively enumerable languages). Thus one sees that the hierarchy of families obtained by codings of pure languages is situated between the hierarchy of language families obtained by the use of nonterminals and terminals and that of pure language families.

Acknowledgement. The author is grateful to the referee and Henning Bordihn for their comments which improved the earlier version.

TECHNISCHE UNIVERSITÄT
OTTO VON GUERICKE
POSTFACH 124
MAGDEBURG 3010/DDR

References

[1] DASSOW, J., Pure grammars with regulated rewriting. Revue Roumaine Math. Pures Appl. 31 (1986) 657—666.
[2] DASSOW, J. & GH. PAUN, Further remarks on pure grammars with regulated rewriting. Revue Roumaine Math. Pures Appl. 31 (1986) 855—864.
[3] DASSOW, J. & GH. PAUN, Codings of pure languages obtained by regulated rewriting. Foundat. Control Engineer. 9 (1984) 3—14.
[4] MAURER, H., A. SALOMAA & D. WOOD, Pure grammars. Inform. Control 49 (1980) 47—72.
[5] MAYER, O., Some restrictive devices for context free grammars. Inform. Control 20 (1972) 69—92.
[6] PAUN, GH. A note on pure matrix grammars. To appear in Revue Roumaine Math. Pures Appl.
[7] ROZENBERG, G. & A. SALOMAA, The Mathematical Theory of L Systems. Academic Press, 1980.
[8] SALOMAA, A. Formal Languages. Academic Press, 1973.

(Received Oct. 15, 1985)