stichting
mathematisch
centrum

MC

AFDELING INFORMATICA
(DEPARTMENT OF COMPUTER SCIENCE)

IW 114/79 AUGUSTUS

P.R.J. ASVELD

ON CONTROLLED ITERATED GSM MAPPINGS AND RELATED OPERATIONS

Preprint

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

1980 Mathematics subject classification: 68F05

ACM-Computing Reviews-categories: 5.23

On Controlled Iterated GSM Mappings and Related Operations *)

by

Peter R.J. Asveld

ABSTRACT

In [17] G. PĂUN studied families of languages generated by iterated gsm mappings, iterated finite substitutions, and iterated homomorphisms. In this note we generalize some results in [17], and we discuss the relation between iterated finite substitutions (homomorphisms) and (deterministic) tabled context-independent Lindenmayer systems.

KEY WORDS & PHRASES: *generalized sequential machines, tabled Lindenmayer systems, controlled rewriting.*

---

*) This report will be submitted for publication elsewhere.

1. INTRODUCTION

Since the class of generalized sequential machine (or *gsm*) mappings is closed under the (associative) operation of composition [11], it is natural to describe the iteration of gsm mappings in terms of formal languages [17].

For each finite set U of gsm mappings and each language L, let $U^*(L)$ be the language obtained by applying on L all possible finite sequences of gsm mappings composed of elements of U. Interpreting U as an alphabet we may define a language M over U and consider languages of the form M(L) where we now restrict the iteration process to those sequences over U that belong to M. We will refer to M as the *control language,* since M controls the iteration on L of elements from U.

In [17] G. PĂUN investigated families of languages M(L) where M is a regular language and L is taken from one of the families in the Chomsky hierarchy (cf. [26]). Moreover, in this framework Păun studied the iteration of finite substitutions and homomorphisms by means of generalized sequential machines having a single state only.

The aim of the present note is twofold. Firstly, we consider the more general case in which L and M are taken from arbitrary language families $\mathcal{L}$ and $\mathcal{M}$ respectively (satisfying some simple conditions only) rather than from a privileged family in the Chomsky hierarchy (Section 2). Secondly, we will refer to another area in which iterated finite substitutions and homomorphisms have been investigated, viz. the theory of (extended tabled) Lindenmayer systems. So certain results obtained in Lindenmayer systems theory may be used to solve problems left open in [17] (cf. Section 3).

This note may be considered as a continuation of PĂUN's paper [17]. Therefore we assume the reader to be familiar with the notational conventions established in [17]. For other unexplained terminology and additional background material we refer to [14, 15, 16, 19].

As usual we consider two languages equal if they differ by at most the empty word λ. Similarly, two language families are defined to be equal if they contain the same languages modulo λ.

2. CONTROLLED ITERATED GSM MAPPINGS

We first introduce some definitions and notation concerning the family of languages generated by controlled iterated (non)deterministic gsm mappings.

DEFINITION. Let L and M be families of languages. Then $S^{\lambda D}(L,M) = \{M(L) \mid M \subseteq U^*$, U is a finite set of deterministic gsm mappings; $L \in L$; $M \in M\}$, $S^{\lambda}(L,M) = \{M(L) \mid M \subseteq U^*$, U is a finite set of nondeterministic gsm mappings; $L \in L$; $M \in M\}$. The corresponding language families based on the controlled iteration of λ-free (non)deterministic gsm mappings are denoted by $S(L,M)$ and $S^D(L,M)$. □

In order to avoid trivialities we always assume that the initial language L is λ-free.

Note that $S^{\lambda}(L,L_3)$ and $S(L,L_3)$ coincide with the families $S^{\lambda}(L)$ and $S(L)$ respectively, as defined in [17].

We show that the role played by the family L is rather unimportant (cf. Lemma 1 below) as soon as $M \supseteq L$ and M possesses some simple properties.

An *isomorphism* (or "renaming of symbols") is a one-to-one homomorphism. Let ρ be the *reversal operation* i.e. the mapping satisfying $\rho(\lambda) = \lambda$, $\rho(\alpha_1\alpha_2\ldots\alpha_{n-1}\alpha_n) = \alpha_n\alpha_{n-1}\ldots\alpha_2\alpha_1$, and $\rho(L) = \{\rho(w) \mid w \in L\}$ for each language L. Let L_{SYMBOL} be the language family defined by $L_{\text{SYMBOL}} = \{\{\sigma\} \mid \sigma$ is a symbol$\}$.

LEMMA 1. *Let L and M be families of languages such that $L_{\text{SYMBOL}} \subseteq L \subseteq M$. If M is closed under isomorphism, reversal and concatenation, then*

$$S^{\lambda D}(L,M) = S^{\lambda D}(L_{\text{SYMBOL}},M), \qquad S^{\lambda}(L,M) = S^{\lambda}(L_{\text{SYMBOL}},M),$$

$$S^{D}(L,M) = S^{D}(L_{\text{SYMBOL}},M), \qquad S(L,M) = S(L_{\text{SYMBOL}},M).$$

PROOF. Let $L_0 = M(L)$ for some $M \subseteq U^*$, $M \in M$ and $L \subseteq \Sigma^+$, $L \in L$. Let $\bar{L} \subseteq \bar{\Sigma}^+$ be an isomorphic copy of L such that Σ, $\bar{\Sigma}$ and U are mutually disjoint. Furthermore, let σ be a symbol not in $\Sigma \cup \bar{\Sigma} \cup U$. Define the new control language M_0 by $M_0 = \rho(\bar{L})M$, while U is extended to $U \cup \bar{\Sigma}$. For each $\bar{\alpha} \in \bar{\Sigma}$ we define a new gsm mapping $\bar{\alpha}$ by

$\bar{\alpha}(\sigma) = \{\alpha\}$,

$\bar{\alpha}(w) = \{\alpha w\}$, for each $w \in \Sigma^+$, $\alpha \in \Sigma$.

It will be clear that $M_0(\{\sigma\}) = M(L)$, and hence we have $S^\lambda(L,M) \subseteq S^\lambda(L_{SYMBOL},M)$. Since the gsm mappings $\bar{\alpha}$ may be chosen λ-free and deterministic, similar inclusions hold in the three other cases, whereas the inverse inclusions are obvious. □

A family of languages is called *natural* [24] if it is closed under intersection with Σ^* for each finite alphabet Σ. Let $N(L)$ be the smallest natural family which includes the family L. Clearly, we have

$$N(L) = \{L \cap \Sigma^* \mid L \in L;\ \Sigma \text{ is a finite alphabet}\}.$$

Next we show that regular control on iterated gsm mappings is equivalent to applying the operator N to the corresponding uncontrolled family (cf. Lemma 2. Note that $NS^{\lambda D}(L,L_3) = S^{\lambda D}(L,L_3)$ and similarly for S^λ, S^D and S.)

But we first introduce some notation.

<u>DEFINITION</u>. Let L be a family of languages. Then

$$S_0^\lambda(L) = \{U^*(L) \mid U \text{ is a finite set of nondeterministic gsm mappings; } L \in L\},$$

$$S_0^{\lambda D}(L) = \{U^*(L) \mid U \text{ is a finite set of determinstic gsm mappings; } L \in L\}.$$

The language families based on the iteration of λ-free (non)deterministic gsm mappings are denoted by $S_0(L)$ and $S_0^D(L)$. □

<u>LEMMA 2</u>. *If L is closed under isomorphism, then*
(i) $NS_0^{\lambda D}(L) = S^{\lambda D}(L,L_3)$, $NS_0^\lambda(L) = S^\lambda(L,L_3)$,
(ii) $NS_0^D(L) = S^D(L,L_3)$, $NS_0(L) = S(L,L_3)$.

PROOF.

(i) The proof is similar to the proof of Theorem 2.1 in [1].

(ii) We only show $NS_0^D(L) = S^D(L,L_3)$ since the other case is completely analogous.

Firstly, the inclusion $NS_0^D(L) \subseteq S^D(L,L_3)$ is obvious. So it remains to establish the converse inclusion.

Let $L = M(L_0)$ be in $S^D(L,L_3)$, i.e. $L_0 \in L$, $M \subseteq U^*$ is a regular language, and $U = \{g_1,\ldots,g_k\}$ is a finite set of λ-free deterministic gsm mappings. Let M be accepted by a deterministic finite automaton (Q,U,q_0,Q_F,δ) where Q is the set of states, U is the input alphabet, q_0 is the initial state, Q_F is the set of final states, and δ: $Q \times U \to Q$ is the transition function.

If Σ is the union of the input and output alphabets of all gsm's in U, then we introduce a new alphabet $\Sigma_1 = \Sigma \cup \Sigma \times Q$, a new set of gsm mappings $U_1 = \{h_0,h_1,\ldots,h_k\}$ such that $L = U_1^*(L_1) \cap \Sigma^*$ with $L_1 = \iota(L_0)$ where the isomorphism ι is defined by $\iota(\sigma) = [\sigma,q_0]$ for each $\sigma \in \Sigma$. Clearly, we have $L_1 \in L$.

The gsm mappings $h_1,\ldots,h_k$ are obtained from $g_1,\ldots,g_k$ in the following way. For each i $(1 \le i \le k)$ if $sa \to b_1\ldots b_n s'$ is a λ-free production in g_i, then h_i possesses a finite set of corresponding productions $s[a,q] \to [b_1,q']\ldots[b_n,q']s'$ for each $q,q' \in Q$ such that $\delta(q,g_i) = q'$. Note that h_i is a λ-free deterministic gsm mapping. The gsm mapping h_0 is a partial homomorphism defined by $h_0([\sigma,q]) = \sigma$ iff $q \in Q_F$. (In all other cases h_0 is undefined.)

It is straightforward to show that $U_1^*(L_1) \cap \Sigma^* = L$. Hence we have $S^D(L,L_3) \subseteq NS_0^D(L)$. □

Lemmas 1 and 2 enable us to extend Theorem 2 in [17] and Theorems 2 and 23 in [26] in the following way (cf. Corollary 2 in [17]).

THEOREM 3. *Let L and M be families of languages such that $L_{SYMBOL} \subseteq L \subseteq L_0$, and $L_3 \subseteq M \subseteq L_0$. Then*

(1) $$S^\lambda(L,M) = NS_0^\lambda(L) = NS_0^\lambda(L_{SYMBOL}) = L_0,$$

(2) $$S^{\lambda D}(L,M) = NS_0^{\lambda D}(L) = NS_0^{\lambda D}(L_{SYMBOL}) = L_0.$$

PROOF.

(1) From Lemmas 1 and 2 it follows that $S^{\lambda}(L_3,L_3) = S^{\lambda}(L_{\text{SYMBOL}},L_3) = NS_0^{\lambda}(L_{\text{SYMBOL}})$. The monotonicity of S^{λ} and N implies that $S^{\lambda}(L_3,L_3) \subseteq S^{\lambda}(L,M)$ and $S^{\lambda}(L_3,L_3) \subseteq NS_0^{\lambda}(L)$. By Church's thesis (or an argument similar to the proof of Theorem 1 in [17]) we have $S^{\lambda}(L,M) \subseteq L_0$ and $NS_0^{\lambda}(L) \subseteq L$.

According to Theorem 2 in [17] we have $L_0 = S^{\lambda}(L_3,L_3)$, which completes the proof.

(2) Using Theorem 23 in [26] we obtain in a similar way $L_0 = S_0^{\lambda D}(L_{\text{SYMBOL}}) \subseteq NS_0^{\lambda D}(L_{\text{SYMBOL}}) \subseteq NS_0^{\lambda D}(L) \subseteq S^{\lambda D}(L,M) \subseteq L_0$. □

In Theorem 3 it was shown that all type-0 languages are obtained by means of controlled iterated (possibly erasing) gsm mappings. We now show that for some families of control languages the family L_0 can also be achieved by controlled iterated λ-free gsm mappings (cf. Theorem 2.2 in [1] or Proposition 4.3 in [3]).

THEOREM 4. *Let* M *be a family of languages closed under reversal. If* $\{h(L) \mid L \in M;\ h$ *is a (possibly erasing) homomorphism*$\} = L_0$, *then* $S^D(L_{\text{SYMBOL}},M) = S(L_{\text{SYMBOL}},M) = L_0$.

PROOF. Remark that $M \subseteq L_0$, and therefore by Church's thesis $S^D(L_{\text{SYMBOL}},M) \subseteq S(L_{\text{SYMBOL}},M) \subseteq L_0$. So it remains to show that $L_0 \subseteq S^D(L_{\text{SYMBOL}},M)$.

Let $L \subseteq \Sigma^+$ be a λ-free language in L_0. Then there exist a language $M \subseteq U^+$ in M and a homomorphism $h: U^+ \to \Sigma^+$ such that $h(M) = L$. For each g in U we will define a λ-free deterministic gsm mapping such that $(\rho(M))(\{\$\}) = L$ for some $\$ \notin \Sigma$, i.e. $L \in S^D(L_{\text{SYMBOL}},M)$. In defining g we distinguish the following cases:

(1) $h(g) = \lambda$. Then the gsm mapping g is defined to be the identity mapping.

(2) $h(g) \neq \lambda$. In this case g is defined by

$g(\$) = h(g)$

$g(x) = h(g)x$ for each $x \in \Sigma^+$ (or equivalently, $x \neq \$$),

which can be easily realized by a λ-free deterministic gsm. □.

Note that Theorem 4 applies to the family L_1, the intersections of linear context-free languages [6], and each complexity class which includes DSPACE(log n) (i.e. the family of languages accepted by two-way deterministic multi-tape Turing machines which scan at most log n tape squares at each auxiliary tape during a computation on an input of length n) [3]. On the other hand it is still open whether e.g. $S^D(L_{SYMBOL}, L_2)$ and $S(L_{SYMBOL}, L_2)$ are properly included in L_0.

3. RELATED OPERATIONS

In this section we first give some definitions and notation concerning tabled Lindenmayer (or L) systems. Then we relate the families $H(L_3)$ and $F(L_3)$, as defined in [17], to certain classes of tabled L systems. This enables us to solve the open problems in [17] (p. 932) because the corresponding problems in L systems theory have already been solved.

DEFINITION. Let M and A be families of languages. An M-*controlled extended tabled Lindenmayer system with* A-*axioms* or an (M)*ETOL*(A) *system* G = (V,Σ,U,M,A) consists of
- an alphabet V
- a terminal alphabet $\Sigma \subseteq V$
- a finite set U of finite substitution over V
- a language $M \subseteq U^*$ in M (M is the *control language* of G)
- a language $A \subseteq V^*$ in A (A is the *initial language* of G).

The language generated by G is defined by $L(G) = M(A) \cap \Sigma^*$.

If U contains only homomorphisms over V, then G is called *deterministic* or an (M)*EDTOL*(A) *system*. If each finite substitution [or homomorphism] in U is λ-free then G is called *propagating* or an (M)*EPTOL*(A) [or (M)*EPDTOL*(A)] system. For X = ETOL, EDTOL, etc. the family of languages generated by (M)X(A) systems is denoted by (M)X(A). □

Moreover, in order to comply with standard terminology in L systems theory we use the following notational conventions:

(1) We drop the "E" in the family name when V = Σ; e.g. (M)TOL(A), (M)DTOL(A), etc.

(2) Whenever $A = L_{\text{SYMBOL}}$ we omit the reference to A; e.g. (M)ETOL, (M)EDTOL, etc.

(3) In case $M = \{U^* \mid U \text{ is finite}\}$ we delete M in the family name; e.g. ETOL(A), EDTOL(A), etc.

Combinations of these conventions are of course possible; so families like ETOL, EDTOL, (M)TOL are defined in an implicit way.

Note that for each M and each A we have $(M)\text{ETOL}(A) = N((M)\text{TOL}(A))$ and $(M)\text{EDTOL}(A) = N((M)\text{DTOL}(A))$.

From the definitions we immediately obtain the following equivalences.

PROPOSITION 5. *Let L be a family of languages. Then*

$$H(L) = (L_3)\text{PDTOL}(L), \qquad H^\lambda(L) = (L_3)\text{DTOL}(L),$$

$$F(L) = (L_3)\text{PTOL}(L), \qquad F^\lambda(L) = (L_3)\text{TOL}(L). \qquad \square$$

Here $H(L)$ is defined by $H(L) = \{R(L) \mid L \in L;\ R \in L_3,\ R \subseteq U^*;\ U$ is a finite set of λ-free homomorphisms$\}$, and similarly for $H^\lambda(L)$, $F(L)$, and $F^\lambda(L)$; cf. [17].

THEOREM 6.

(1) *For each family L with $L_{\text{SYMBOL}} \subseteq L \subseteq$ EDTOL, $H(L) = H^\lambda(L) =$ EDTOL = EPDTOL (modulo λ).*

(2) *For each family L with $L_{\text{SYMBOL}} \subseteq L \subseteq$ ETOL, $F(L) = F^\lambda(L) =$ ETOL = EPTOL (modulo λ).*

PROOF. In both cases the former and the latter equalities are known (cf. [17] and respectively [18, 4]). So it remains to show the middle equality.

(1) From Proposition 5 and the assumption on L it follows that $(L_3)\text{DTOL} \subseteq (L_3)\text{DTOL}(L) = H^\lambda(L) \subseteq (L_3)\text{DTOL(EDTOL)}$. In the proof of Lemma 3.3 in [5] it was shown that $(L_3)\text{DTOL} = (L_3)\text{EDTOL}$ which in turn equals EDTOL [4]. In the same way we obtain $(L_3)\text{DTOL(EDTOL)} = (L_3)\text{EDTOL(EDTOL)} =$ EDTOL(EDTOL). Since EDTOL is the smallest (nontrivial) natural family closed under iterated homomorphism [4], we have EDTOL(EDTOL) = EDTOL. Hence EDTOL $\subseteq H^\lambda(L) \subseteq$ EDTOL.

(2) As under (1) we have $(L_3)\mathrm{TOL} \subseteq (L_3)\mathrm{TOL}(L) = F^{\lambda}(L) \subseteq (L_3)\mathrm{TOL}(\mathrm{ETOL})$. From Theorem 1 in [9] it follows that $(L_3)\mathrm{TOL} = (L_3)\mathrm{ETOL}$ and hence $(L_3)\mathrm{TOL} = \mathrm{ETOL}$ by Theorem 2.1 in [1]. Similarly, we have $(L_3)\mathrm{TOL}(\mathrm{ETOL}) = (L_3)\mathrm{ETOL}(\mathrm{ETOL}) = \mathrm{ETOL}(\mathrm{ETOL})$. As ETOL is the smallest (nontrivial) natural family closed under iterated finite substitution [7], it follows that ETOL(ETOL) = ETOL, which finally yields $F^{\lambda}(L) = \mathrm{ETOL}$. □

<u>COROLLARY 7</u>. *$H^{\lambda}(L_3) = \mathrm{EDTOL}$, and $F^{\lambda}(L_3) = \mathrm{ETOL}$.* □

<u>COROLLARY 8</u>.

(1) *$H^{\lambda}(L_3)$ is properly included in $F^{\lambda}(L_3)$.*

(2) *$H^{\lambda}(L_3)$ and L_2 are incomparable, i.e. neither family includes the other one.*

<u>PROOF</u>. (1) follows from Corollary 7 above and Corollary 4 in [8]. Corollary 7 together with the main result in [10] imply (2). □

Clearly, Corollary 8 solves the open problems in [17] (p. 932).

By Corollary 7 many results obtained for the families ETOL and EDTOL (cf. e.g. [4, 7, 18]) now become applicable to $F^{\lambda}(L_3)$ and to $H^{\lambda}(L_3)$ respectively. In particular Theorem 9 and Proposition 2 in [17] extend to the main result in [7] and to Corollary 4.5 in [4] respectively.

For further generalizations to M-controlled tabled L systems, the (M-controlled) iteration of K-substitutions where K is an arbitrary language family, etc. we refer to the literature [1 - 5, 12, 13, 20 - 26].

ACKNOWLEDGEMENT

The author is indebted to Dr. Gheorghe Păun and to Professor Solomon Marcus for suggesting to write this note.

REFERENCES

[1] ASVELD, P.R.J., *Controlled iteration grammars and full hyper-AFL's,* Information and Control 34 (1977) 248-269.

[2] ASVELD, P.R.J., *"Iterated Context-Independent Rewriting - An Algebraic Approach to Families of Languages"*, (Doctoral dissertation, 1978), Twente University of Technology, Enschede, The Netherlands.

[3] ASVELD, P.R.J., *Space-bounded complexity classes and iterated deterministic substitution,* Report No. IW 108/79, Mathematical Centre, Amsterdam, The Netherlands (1979).

[4] ASVELD, P.R.J. & J. ENGELFRIET, *Iterated deterministic substitution,* Acta Informatica 8 (1977) 285-302.

[5] ASVELD, P.R.J. & J. ENGELFRIET, *Extended linear macro grammars, iteration grammars, and register programs,* Acta Informatica 11 (1979) 259-285.

[6] BAKER, B.S. & R.V. BOOK, *Reversal-bounded multipushdown machines,* J. Computer and System Sciences 8 (1974) 315-332.

[7] CHRISTENSEN, P.A., *Hyper-AFL's and ETOL systems,* DAIMI PB-35, University of Aarhus, Denmark (1974). Cf. pp. 254-257. *in* G. Rozenberg, A. Salomaa (Eds.), *"L Systems"*, Lecture Notes in Computer Science 15 (1974) Springer, Berlin-Heidelberg-New York.

[8] EHRENFEUCHT, A. & G. ROZENBERG, *Three useful results concerning L languages without interactions,* pp. 72-77 *in* G. Rozenberg, A. Salomaa (Eds.), *"L Systems"*, Lecture Notes in Computer Science 15 (1974), Springer, Berlin-Heidelberg-New York.

[9] EHRENFEUCHT, A. & G. ROZENBERG, *Nonterminals versus homomorphisms in defining languages for some classes of rewriting systems,* Acta Informatica 3 (1974) 265-283.

[10] EHRENFEUCHT, A. & G. ROZENBERG, *On some context-free languages that are not deterministic ETOL languages,* Rev. Française Automat. Informat. Rech. Opérat. IT 11 (1977) 273-291.

[11] ELGOT, C.C. & J.E. MEZEI, *On relations defined by generalized finite automata*, IBM J. Research & Development 9 (1965) 47-65.

[12] ENGELFRIET, J., *Iterating iterated substitution*, Theoretical Computer Science 5 (1977) 85-100.

[13] ENGELFRIET, J., *Three hierarchies of transducers*, TW-memorandum No. 217 (1978), Twente University of Technology, Enschede, The Netherlands.

[14] HARRISON, M.A., *"Introduction to Formal Language Theory"*, Addison-Wesley (1978), Reading, Massachusetts.

[15] HERMAN, G.T. & G. ROZENBERG, *"Developmental Systems and Languages"*, North-Holland (1975), Amsterdam.

[16] HOPCROFT, J.E. & J.D. ULLMAN, *"Formal Languages and Their Relation to Automata"*, Addison-Wesley (1969), Reading, Massachusetts.

[17] PĂUN, G., *On the iteration of gsm mappings*, Rev. Roum. Math. Pures et Appl. 23 (1978) 921-937.

[18] ROZENBERG, G., *Extensions of tabled OL-systems and languages*, Internat. J. Comp. Inform. Sci. 2 (1973) 311-336.

[19] SALOMAA, A., *"Formal Languages"*, Academic Press (1973), New York.

[20] SALOMAA, A., *Macros, iterated substitution and Lindenmayer AFL's*, DAIMI PB-18, University of Aarhus, Denmark. Cf. pp. 250-253 *in* G. Rozenberg, A. Salomaa (Eds.), *"L systems"*, Lecture Notes in Computer Science 15 (1974), Springer, Berlin-Heidelberg-New York.

[21] VAN LEEUWEN, J., *A generalization of Parikh's theorem in formal language theory*, pp. 17-26 *in* J. Loeckx (Ed.), *"Automata, Languages, and Programming* 2nd *Colloquium"*, Lecture Notes in Computer Science 14 (1974), Springer, Berlin-Heidelberg-New York.

[22] VAN LEEUWEN, J., *A study of complexity in hyper-algebraic families*, pp. 323-333 *in* A. Lindenmayer, G. Rozenberg (Eds.), *"Automata, Languages, Development"*, North-Holland (1976), Amsterdam.

[23] VAN LEEUWEN, J., *Effective constructions in well-partially-ordered free monoids*, Discrete Math. 21 (1978) 237-252.

[24] VAN LEEUWEN, J. & D. WOOD, *A decomposition theorem for hyper-algebraic extensions of language families,* Theoretical Computer Science 1 (1976) 199-214.

[25] WOOD, D., *A note on Lindenmayer systems, Szilard languages, spectra, and equivalence,* Internat. J. Comp. Inform. Sci. 4 (1975) 53-62.

[26] WOOD, D., *Iterated a-NGSM maps and Γ-systems,* Information and Control 32 (1976) 1-26.