INFORMATIQUE THÉORIQUE ET APPLICATIONS

MANFRED KUDLEK

Comparison of basic language generating devices (non-deterministic systems)

Informatique théorique et applications, tome 24, n° 5 (1990), p. 489-508.

<http://www.numdam.org/item?id=ITA_1990__24_5_489_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 24, n° 5, 1990, p. 489 à 508)

COMPARISON OF BASIC LANGUAGE GENERATING DEVICES (NON-DETERMINISTIC SYSTEMS) (*)

by Manfred KUDLEK (1)

Communicated by W. BRAUER

Abstract. – *This paper gives an overview and a comparison of the language families defined by simple rewriting systems and generative devices. Such rewriting systems are Regular, Semi-Thue, Normal, Indian Parallel, and Lindenmayer systems with context-independent and context-dependent productions, non-erasing and erasing productions, at most one or more axioms. Generative devices are sentential form languages, adult and terminal word languages, the application of different non-erasing or erasing homomorphisms on them, and the intersection with a terminal subalphabet.*

Résumé. – *Cet article expose les principaux résultats relatifs aux langages engendrés par des systèmes de réécriture simples et les systèmes de génération. On étudie en particulier les systèmes réguliers, de Semi-Thue, normaux, « parallèles indiens » et les systèmes de Lindenmayer avec des productions dépendant ou non du contexte, des productions effaçantes ou non, et ayant un ou plusieurs axiomes. Parmi les systèmes de génération on considère les langages de formes sententiels, les langages « adultes » à mots terminaux et différentes opérations sur ces langages : morphismes effaçant ou pas, intersection avec un alphabet terminal.*

0. INTRODUCTION

Usually formal languages are defined as languages generated by rewriting systems, or as languages accepted by automata. Another possibility is to consider the algebraic closure of a basic class of sets under some operations on them. E. g. taking as basic sets the empty set and those consisting only of one symbol, as operations union, catenation, and catenation closure, one obtains the class of Regular languages.

This paper concentrates on the generative aspect, and the main feature will be in the study of simple devices for defining Formal languages by different rewriting systems, and in the comparison of the generative power.

(*) Received May 1985, revised November 1989.
(1) Fachbereich Informatik, Universität Hamburg, R.F.A.

Informatique théorique et Applications/Theoretical Informatics and Applications
0988-3754/90/05 489 20/$4.00/

The first step is to consider the sentential form languages of some simple rewriting systems. Such systems are Regular, Semi-Thue, and Normal systems as sequential ones, and Indian Parallel and Lindenmayer systems as parallel ones.

Until recently an extensive research on the effect of context-independent and context-dependent productions, non-erasing or erasing productions, of at most one or more axioms, determinism or non-determinism in the productions, has been done only for Lindenmayer systems.

The next step is to consider the effect of applying different kinds of non-erasing or erasing homomorphisms, such as arbitrary or letter-to-letter homomorphisms, on sentential form languages, or of the intersection with a terminal subalphabet.

Finally, also the sets of terminal and adult words of some rewriting systems may be studied.

To have a simple and compact notation for all the systems and language families, the framework introduced for Lindenmayer systems is used here.

Since this is only an overview, only the most important theorems are presented, without giving the proofs. Details may be found in the literature cited in the references. The results obtained so far, are summarized in a number of diagrams which are complete only for Regular, Semi-Thue and Lindenmayer systems.

1. DEFINITIONS

Sequential rewriting

In sequential rewriting systems rewriting occurs in a bounded part of a word only. Three kinds of simple sequential rewriting systems are introduced, together with two other 'mirror systems'. Such systems are triples $G=(V, A, P)$ where V is a finite set of symbols (alphabet), $A \subseteq V^*$ a finite set of starting words (axioms), and $P \subseteq V^* \times V^*$ a finite set of productions. The different systems are distinguished by the place of the rewriting within a word. For $(a, b) \in P$, usually written $a \to b \in P$, the one-step derivation relations $\to$ are defined by $aw \to bw$ for (Right $-$) Regular systems, $w_1 aw_2 \to w_1 bw_2$ for Semi-Thue systems, and $aw \to wb$ for (Post) Normal systems, respectively. The two other kinds are defined by $wa \to wb$ for Left-Regular systems, and by $wa \to bw$ for Antinormal systems. These are merely mirror images of the corresponding Regular or Normal systems.

As usual, the reflexive and transitive closure of $\rightarrow$ will be denoted by $\overset{*}{\rightarrow}$.

If $P \subseteq V^* \times V^*$ is arbitrary (but finite), also productions $\lambda \rightarrow u \in P$ are possible. In this case the symbols R', S', N' are used to denote Regular, Semi-Thue, and Normal systems, respectively, as well as the symbols $\bar{R}'$, $\bar{N}'$ to denote Left-Regular and Antinormal systems. If productions $\lambda \rightarrow u \in P$ are forbidden, however, *i.e.* $P \subseteq V^+ \times V^*$, then the symbols R, S, N, as well as $\bar{R}$, $\bar{N}$ are used.

In this paper, however, only R', S', N', R, S, N will be considered.

If the productions are context-independent (context-free), *i.e.* $P \subseteq (\{\lambda\} \cup V) \times V^*$, the symbol O for no interaction (context) is used, otherwise the symbol I for interaction. Thus, e.g. *FOR*, *FIS'*, *FON* systems are systems with finite sets of axioms which the symbol F is used for.

If $\text{card}(A) \leqq 1$ the letter F will be omitted giving e.g. *IR*, *OS'*, *IN* systems.

If $lg(a) \leqq lg(b)$ for $a \rightarrow b \in P$ such a production is called monotone or propagating. If all productions of a system are propagating this will be denoted by the letter P, giving e.g. *PFIR*, *POS'*, *PIN* systems.

Let $u \leqq v$ stand for the fact that u is a prefix (in the case of R, R', N, N'), subword (in the case of S, S'), suffix (in the case of $\bar{R}$, $\bar{R}'$, $\bar{N}$, $\bar{N}'$) of v.

If $(a_1 \rightarrow b_1 \in P \wedge a_2 \rightarrow b_2 \in P \wedge (a_1 \leqq a_2 \vee a_2 \leqq a_1)) \Rightarrow (a_1 = a_2 \wedge b_1 = b_2)$ holds for all productions, the system is called deterministic, and this will be denoted by the symbol D, giving e.g. *PDOR*, *PDFIS*, *DON* systems.

Parallel Rewriting

In parallel rewriting systems rewriting occurs at an unbounded number of places within a word.

Two kinds of parallel rewriting systems are introduced. Again, such systems are triples $G = (V, A, P)$. The difference to the sequential systems is in the way how productions are applied.

In Indian Parallel systems one symbol is rewritten in one step, but at every place of occurrence in the word whereas in Lindenmayer systems all symbols of the word are rewritten in one step, if possible. For Indian Parallel systems the letter B for Bharat, the Sanskrit name of India, will be used, and for Lindenmayer systems the letter L.

In the context-independent case $P \subseteq V \times V^*$ holds, and the one-step derivation relations are defined in the following way: for Indian Parallel systems

$$(u_0 x u_1 x \ldots u_{k-1} x u_k \to u_0 b_1 u_1 \ldots u_{k-1} b_k u_k) \Leftrightarrow$$
$$(\neg\, x \leqq u_0 u_1 \ldots u_{k-1} u_k \wedge \forall i \in \{1, \ldots, k\} : x \to b_i \in P)$$

for Lindenmayer systems

$$(x_1 \ldots x_k \to b_1 \ldots b_k) \Leftrightarrow \forall i \in \{1, \ldots, k\} : x_i \to b_i \in P$$

In the context-dependent case

$$P \subseteq ((\{\lambda\} \cup \{\$\})\, V^* \times V \times V^*\, (\{\lambda\} \cup \{\$\})) \times V^*$$

holds, where $\$ \notin V$ is a dummy symbol denoting the fact that the context may occur at the ends of the word. In this case the one step derivation relations are defined in the following way: for Indian Parallel systems

$$(u_0 x u_1 \ldots u_{k-1} x u_k \to u_0 b_1 u_1 \ldots u_{k-1} b_k u_k) \Leftrightarrow$$
$$(\neg\, x \leqq u_0 u_1 \ldots u_{k-1} u_k \wedge \forall i \in \{1, \ldots, k\} : (1_i, x, r_i, b_i) \in P$$

where $l_i (r_i)$ is the left (right) context of the i-th x in $\$ u_0 x u_1 \ldots u_{k-1} x u_k \$$) for Lindenmayer systems

$$(x_1 \ldots x_k \to b_1 \ldots b_k) \Leftrightarrow$$

($\forall i \in \{1, \ldots, k\} : (l_1, x_i, r_i, b_i) \in P$ where l_i (r_i) is the left (right) context of x_i in $\$ x_1 \ldots x_k \$$)

As for sequential systems, the letters *O*, *I*, *F*, *P* are used to denote context-independent, context-dependent systems, those with more than one axiom, and propagating systems, giving e. g. *PFOB*, *PIL* systems.

Deterministic systems are also defined in a similar way, namely by $((l_1, x, r_1, b_1) \in P \wedge (l_2, x, r_2, b_2) \in P \wedge (l_1 \text{ suff } l_2 \vee r_1 \text{ pref } r_2 \vee l_2 \text{ suff } l_1 \vee r_2 \text{ pref } r_1)) \Rightarrow (l_1 = l_2 \wedge r_1 = r_2 \wedge b_1 = b_2)$ where u pref v (u suff v) means that u is a prefix (suffix) of v. For systems with such a property the letter *D* will be used, giving e. g. *PDFOB*, *PDIL* systems.

In contrast to sequential systems, in parallel rewriting systems it is also possible to change the set of productions from one derivation step to another. In other words, triples $G = (V, A, \underline{T})$ with $\underline{T} = \{P_1, \ldots, P_m\}$ and the P_i not necessarily disjoint, may be considered, where each P_i is a set of productions. Such sets are also called tables, and therefore the letter *T* will be used to

denote such systems. In each derivation step only productions of one table may be used.

If the productions of all tables are propagating, the letter P will be used again.

If each table $P \in T$ is deterministic, this will be denoted by the letter D again. Thus e. g. *PDTFOB*, *DTOL*, *TFIB*, *TIL* systems are obtained.

Languages

The simplest way to define a language by a rewriting system G is just to take the sentential form language generated by G which is defined by

$$S(G) := \{ w \in V^* \mid \exists u \in A : u \overset{*}{\rightarrow} w \}.$$

Another possibility is to take terminal or dead words which are defined by w dead $\Leftrightarrow \neg \exists w' \in V^* : w \rightarrow w'$. The set of all dead words in $S(G)$ is denoted by $M(G)$ (M for Latin mortuus = dead).

A third possibility is to consider adult words which are defined by w adult $\Leftrightarrow (w \overset{*}{\rightarrow} w' \Rightarrow w = w')$. The set of all adult words in $S(G)$ is denoted by $A(G)$ (A for adult).

If a subalphabet $V_T \subseteq V$ of terminal symbols is specified the language of a system G is defined in the well known way by $L(G) := S(G) \cap V_T^*$. In Lindenmayer systems this specification of a subalphabet is called extension, usually written as $G = (V, V_T, A, P)$. On all languages defined so far homomorphisms h may be applied. Important are arbitrary homomorphisms, nonerasing homomorphisms ($lg(h(x)) \geqq 1$), letter-to-letter homomorphisms with possible erasing ($lg(h(x)) \leqq 1$) or without erasing ($lg(h(x)) = 1$). The last two usually are called weak codings or codings, respectively.

Language Families

To have also a short and compact notation of corresponding language families the notations of systems are just underlined giving the various classes of sentential form languages, e. g. *PFIR*, *FOS*, *OL*.

For families of languages of dead words the letter M (for the Latin word mortuus = dead) is attached in front and underlined, giving e. g. *MOS*, *MPIN*.

Similarly the letter A is used to denote families of adult languages, giving e. g. *AOL*, *APOB*.

For language families defined by using terminal subalphabets the letter E for extension is used, to give e. g. $\underline{EON}$, $\underline{ETOL}$.

To denote language families defined by an application of some kind of homomorphism, the letters $\hat{H}$, H, $\hat{C}$, C are used denoting arbitrary homomorphisms, non-erasing homomorphsims, weak codings, and codings, respectively. Thus, e. g. $\underline{\hat{H}IR}$, $\underline{HON}$, $\underline{\hat{C}EOB}$, $\underline{CIL}$ are obtained.

The order of these various letters denoting systems and language families is given in the following schema

$\hat{H}$	E	M	P	D	T	F	O	R'
H	–	A	–	–	–	–	I	R
$\hat{C}$		–						S'
C								S
–								N'
								N
								B
								L

where T is used only if B or L is present, and – denotes the possibility of omitting this position.

It is easy to show that E on the one hand, and $\hat{H}, H$, $\hat{C}$, C on the other hand commute, *i. e.* $\underline{E\hat{H}X} = \underline{\hat{H}EX}$ etc. for any language family $\underline{X}$.

For any language L let $L^{\Delta} := L - \{\lambda\}$, and for any language family $\underline{X}$ define $X^{\Delta} := \{L^{\Delta} \mid L \in \underline{X}\}$, and $\underline{X}^{\blacktriangle} := \underline{X}^{\Delta} \cup \{\{\lambda\}\}$.

The classical language families of Regular, Context-free, Context-sensitive, and Recursively enumerable languages are denoted by $\underline{REG}$, $\underline{CF}$, $\underline{CS}$, and $\underline{RE}$, respectively.

2. RESULTS

Regular systems

Such systems have been studied in [1] and [19]. The effect of applying various kinds of homomorphisms on sentential form languages defined by Regular systems and a complete investigation with detailed proofs is given in [12]. The results obtained are summarized in the complete diagrams given in figure 1 for the context-independent case, and in figure 2 for the context-dependent case.

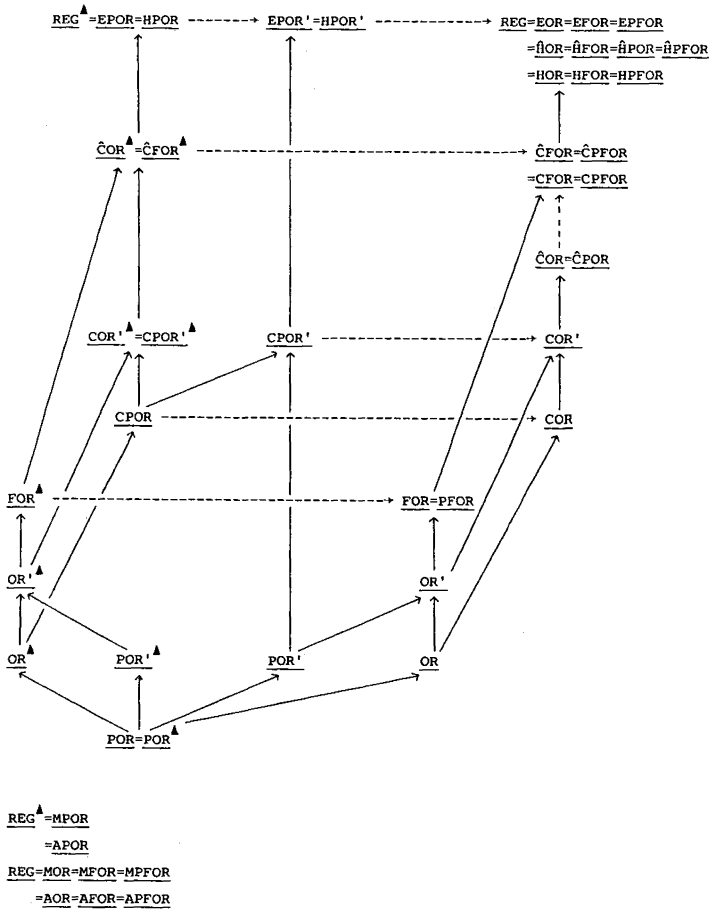

Figure 1.

All proofs are straight-forward using e. g. the closure under union, catenation, and catenation closure.

The family *REG* is closed under union, catenation, catenation closure, homomorphism, inverse homomorphsim, intersection, and mirror image.

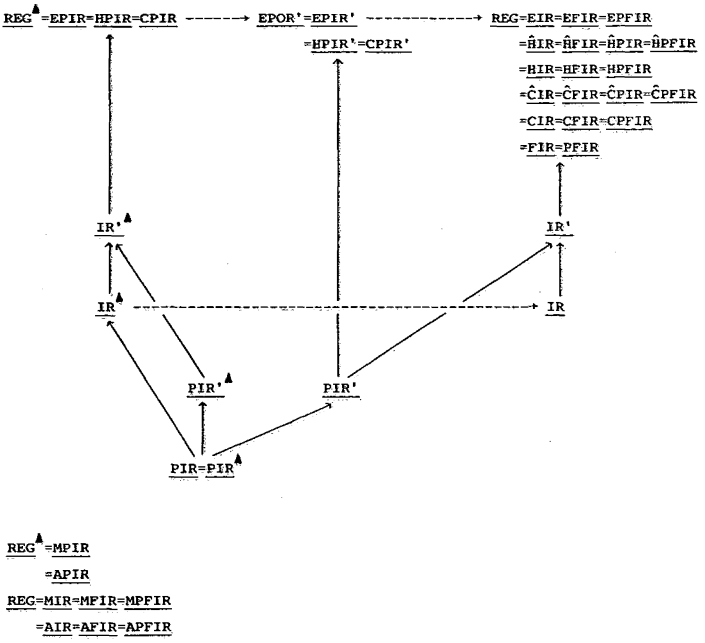

Figure 2.

Semi-Thue systems

Chomsky grammars just are Semi-Thue systems with a terminal subalphabet and special productions. A detailed study of Chomsky type languages may be found in almost any textbook on theoretical computer science. Sentential form languages have been studied in [4], [14], [25] and [26], and the application of homomorphisms in [3], a complete investigation of that effect in [6], where also all detailed proofs may be found. The results are summarized in the complete diagrams given in figure 3 for the context-independent case, and in figure 4 for the context-dependent case.

The family *CF* is closed under union, catenation, catenation closure, homomorphism, inverse homomorphism, intersection with regular sets, and mirror image, but not under intersection.

The family *CS* is closed under the same operations as *CF* except for arbitrary homomorphism, which has to be restricted to non-erasing homomorphism. It is also closed under intersection.

The family *RE* is closed under the same operations as *CF*, and also under intersection.

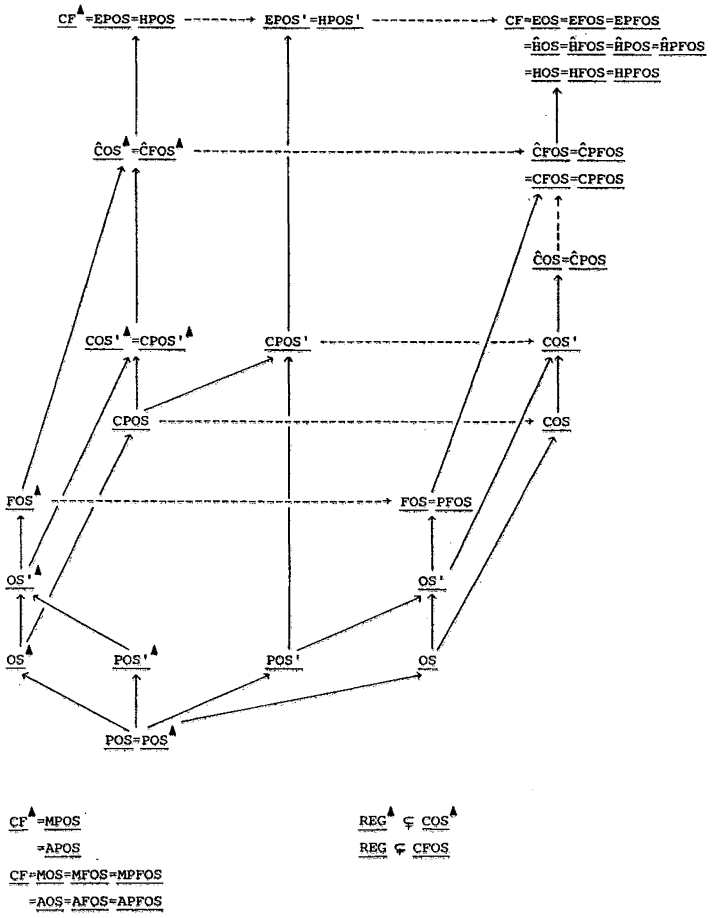

Figure 3.

Non-trivial lemmas and theorems in the context-independent case are:

LEMMA 1: *For each* $L \in \hat{C}POS$ *there exist a* $G' = (V', \{u_0'\}, P') \in POS$ *and a weak coding* g, *such that*

(1) $L = g(S(G'))$

(2) $g(u) \neq \lambda$ *for each production* $x \to u \in P'$

(3) $g(u_0') = \lambda \Leftrightarrow \lambda \in L$

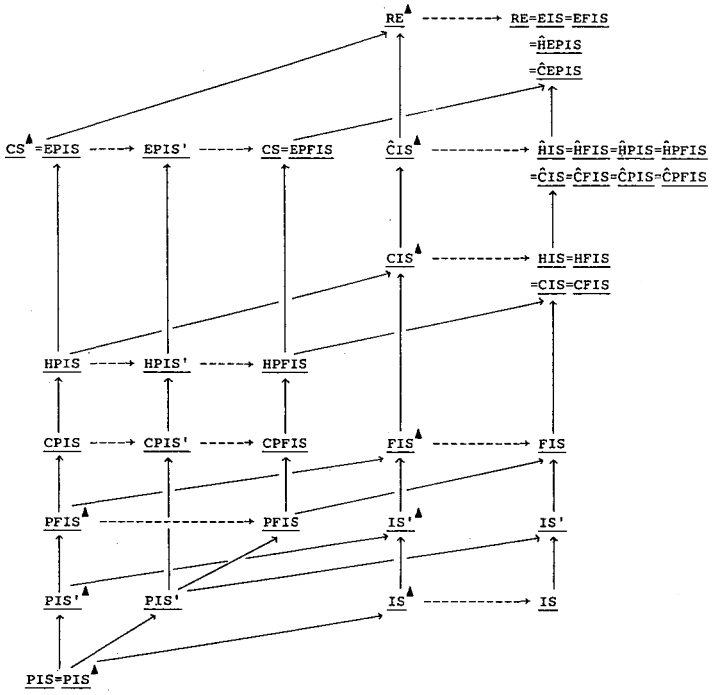

Figure 4.

LEMMA 2: *If* $G=(V,\{u_0\},P)\in POS$ *and* h *is a weak coding with* $h(u)\neq\lambda$ *for each production* $x\to u\in P$ *and* $h(u_0)\neq\lambda$, *then there exist a* $G'=(V',\{u_0'\},P')\in POS$ *and a coding* g, *such that* $h(S(G))=g(S(G'))$.

THEOREM 3: $\hat{\mathrm{C}}\mathrm{FOS}=CFOS$.

LEMMA 4: $\mathrm{REG}\subsetneq CPFOS$.

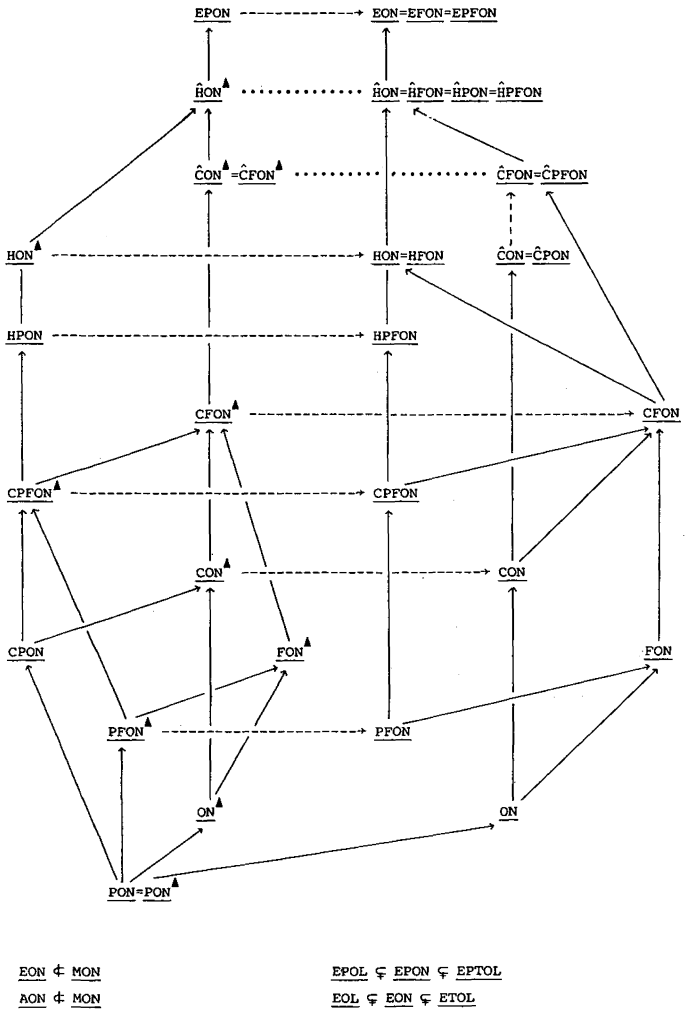

Figure 5.

In the context-dependent case the following non-trivial lemmas and theorems should be mentioned:

THEOREM 5: $\underline{HIS} = \underline{CIS}$, $\underline{\hat{H}IS} = \underline{\hat{C}IS}$.

THEOREM 6: $\underline{CPFIS} \subsetneq \underline{HPFIS}$.

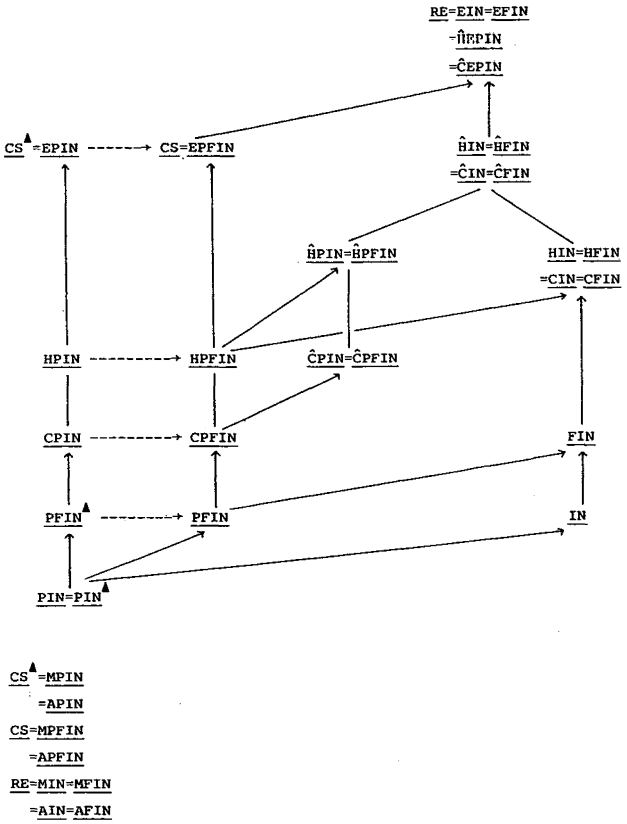

Figure 6.

LEMMA 7: *If $L \in V^*$ is any recursively enumerable set, then both,*

$$pref(L) := \{u \in V^* \mid \exists v \in V^* : uv \in L\}$$

and

$$\text{sub}(L) := \{u \in V^* \mid \exists v \in V^* \exists v' \in V^* : vuv' \in L\}$$

are elements of $\underline{\hat{C}IS}$.

LEMMA 8: *It is decidable for any $G \in IS$, any coding h, and any finite set F, whether $h(S(G)) = F$.*

LEMMA 9: *It is undecidable for any $G \in IS$, any weak coding h, and any finite set F, whether $h(S(G)) = F$.*

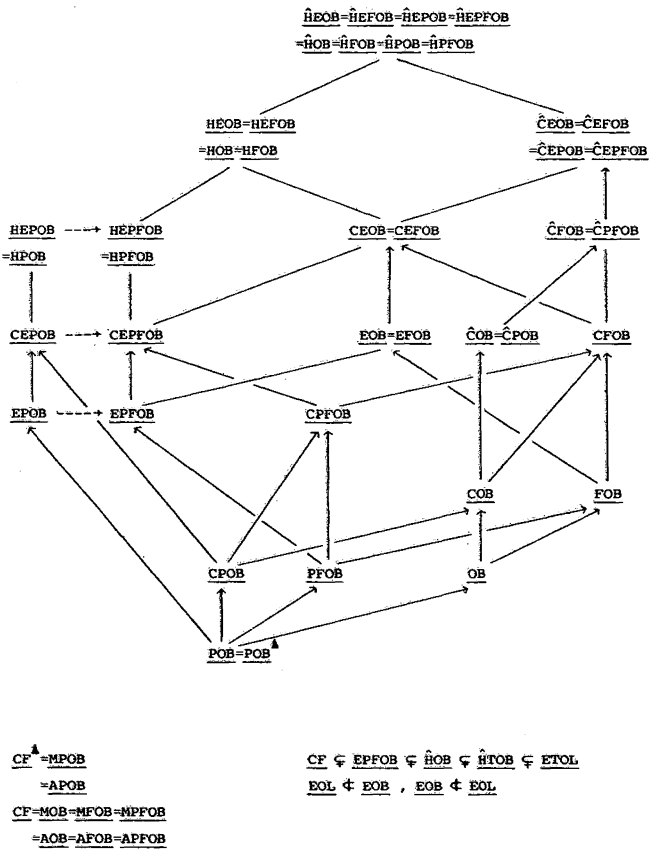

Figure 7.

THEOREM 10: $CIS \subsetneq \hat{C}IS$.

THEOREM 11: $CF^{\blacktriangle} \subsetneq CPIS$.

Normal systems

Such systems have at first been studied in [17]. A characterization in the context-independent case and the relations to Lindenmayer systems have been considered in [2] and [10]. The effect of the application of homomorphisms in the same case is investigated in [13].

The results known so far are summarized in the nearly complete diagram of figure 5.

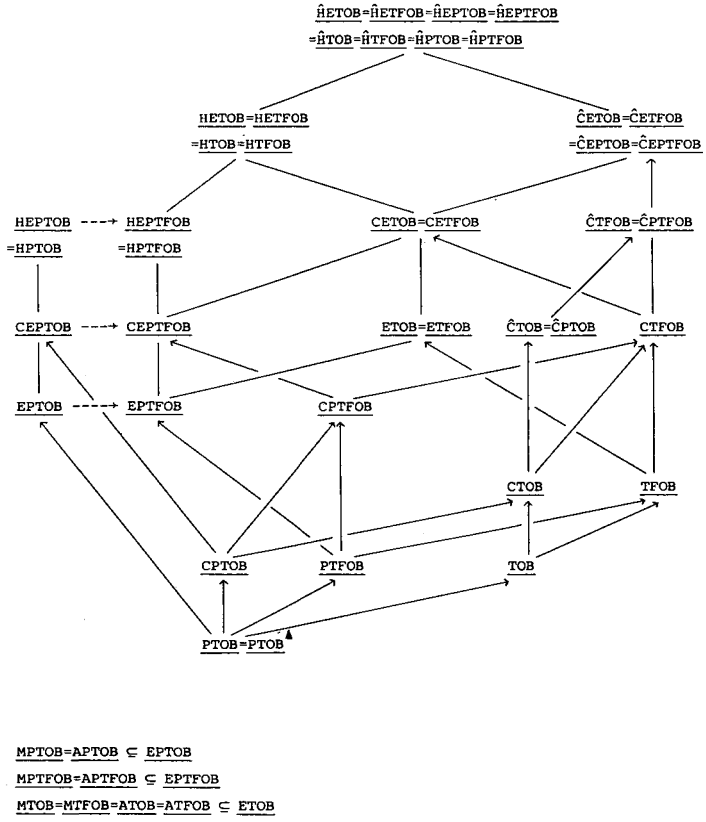

Figure 8.

In the context-independent case the results known so far are given in figure 6.

EON, being the largest family in the context-independent case, is closed under union, homomorphism, intersection with regular sets, and mirror image, but not under inverse homomorphism, catenation, and catenation closure.

Non-trivial lemmas and theorems are:

THEOREM 12: *EPON is the closure of the cyclic permutations of languages from POL under monotone and deterministic general sequential machine mappings*, i.e. $EPON = PDGSM(CYC(POL))$.

LEMMA 13: $\hat{H}ON \subsetneq EON$.

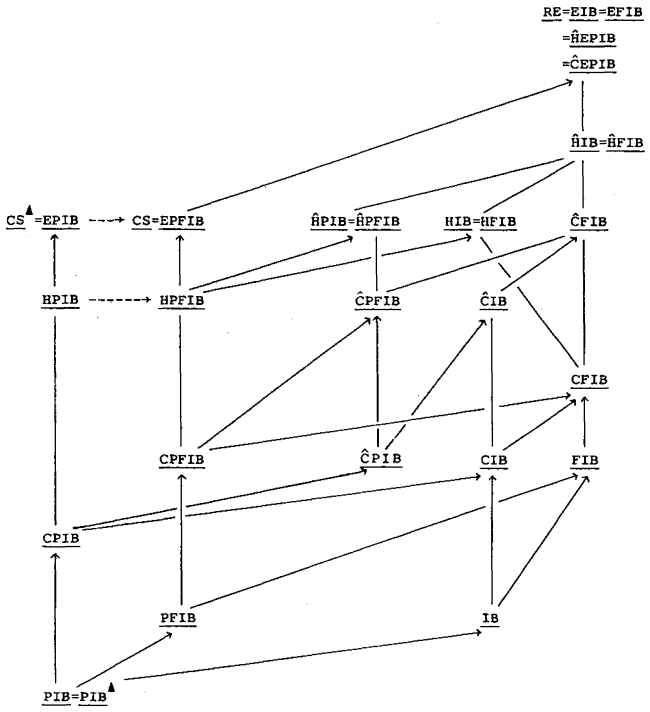

Figure 9.

LEMMA 14: $ON \not\subset CPFON$.

LEMMA 15: $\hat{C}ON \not\subset HFON$.

LEMMA 16: $REG^{\blacktriangle} \not\subset \hat{H}IN, PIN' \not\subset \hat{H}IN$.

LEMMA 17: $EON \not\subset MON, AON \not\subset EON$.

Open problems to mention are: $HON \subsetneq HPFON$ and $\hat{C}PIN \subsetneq \hat{C}IN$.

Indian parallel systems

Such systems, in the context-independent, deterministic table case, have been introduced in [21]. They have been studied also in [20], [22], and [9]. More research has been done on them in [7], [11], and [27]. Very little, so

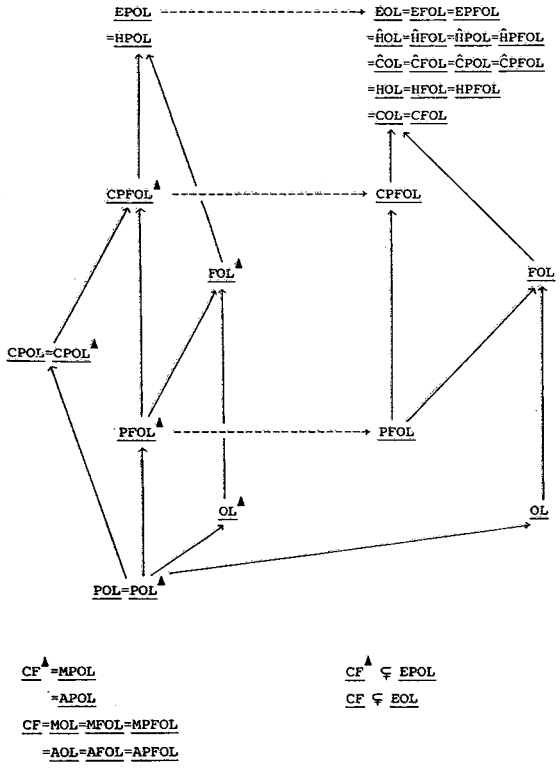

Figure 10.

far, has been published for the context-dependent case. It should be noted that the definitions in [9] and [11] are more general than in the other publications.

The results, known so far, are given in the diagrams of figures 7 and 8. In contrast to all the other systems, in the context-independent case of Indian parallel systems, $\underline{EOB}$ is not the largest class, but $\underline{\hat{H}OB}$. $\underline{EOB}$ has quite poor closure properties, as it is not closed under union, catenation, inverse homomorphism, intersection with Regular sets, and non-erasing homomorphism. Important lemmas and theorems are:

THEOREM 18: *If*

$$G=(V,\{u\},P)\in OB,$$

$$V_T\subseteq V, \quad \text{and} \quad \forall x\in V_T:(x\overset{*}{\rightarrow} w\in V_T^*\Rightarrow w=x),$$

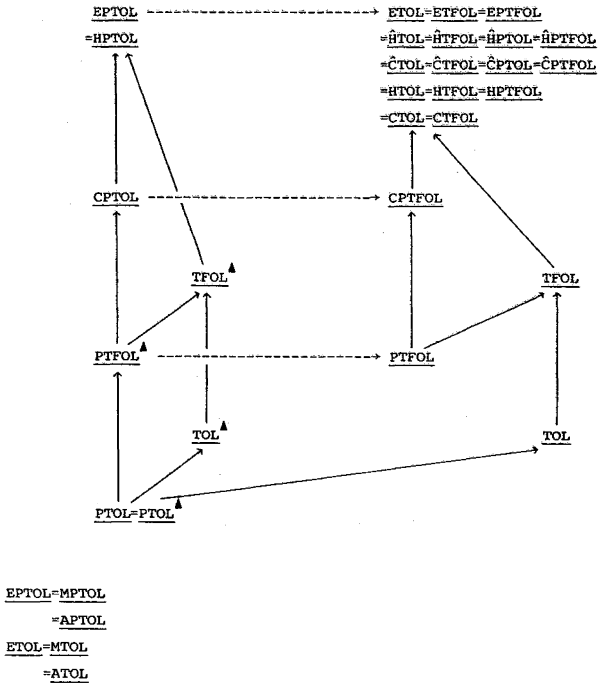

Figure 11.

then $L(G) = S(G) \cap V_T^* \in \underline{CF}$.

LEMMA 19: *$\underline{EOL}$ is incomparable with $\underline{EOB}$.*

THEOREM 20: $\underline{EOB} \subsetneq \underline{HOB}$.

THEOREM 21: $\underline{AOB} = \underline{CF}$.

Few results are known in the context-dependent case. They are given in the diagram of figure 9.

Open problems to mention are: $\underline{HOB} \subsetneq \underline{\hat{H}OB}$ and $\underline{EPFOB} \subsetneq \underline{EFOB}$.

Lindenmayer systems

Lindenmayer systems have been the first systems for which a systematic research concerning all the simple language definition devices has been done.

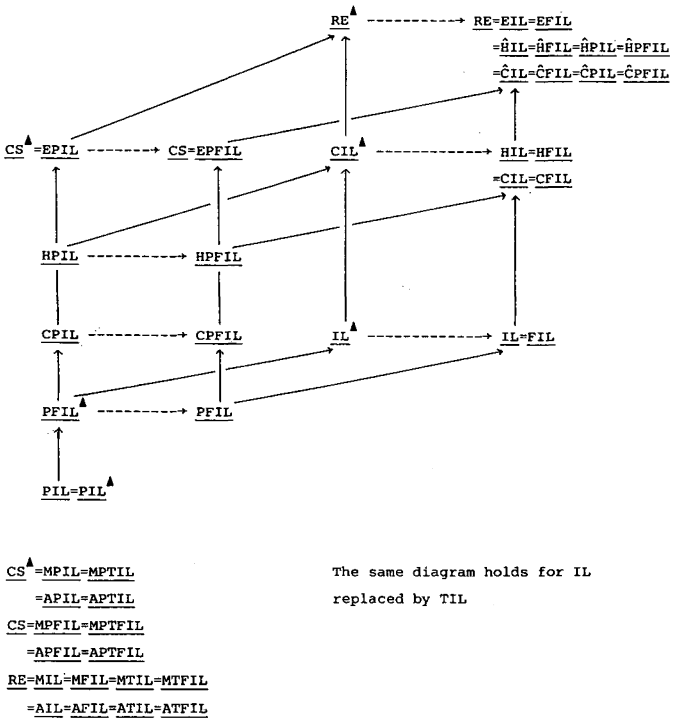

Figure 12.

It should be noted that the definition of such systems given in this paper is slightly different from the usual one which assumes that there exists a production for every symbol $x \in V$, whereas this is not assumed here. Thus, the one step derivation relation is defined by applying productions on all symbols within a word if possible. This difference does not affect, however, the generative properties. Furthermore, it is assumed in most of the papers on Lindenmayer systems, that if $L \in \underline{X}$ for some language family $\underline{X}$, then also $L^{\Delta} \in \underline{X}$, which too is not assumed here.

General information on Lindenmayer systems may be found in [5] and [18].

The results in the context-independent case may be found in [15] and [16], those in the context-dependent case in [23].

Adult languages are investigated in [24] and [23].

All results are summarized in the complete diagrams of figures 10, 11, and the nearly complete diagram of figure 12.

3. DIAGRAMS

In the diagrams of figures 1 to 11 the arrows have the following meanings:

$$\underline{X} \longrightarrow \underline{Y} \Leftrightarrow (\underline{X} \subsetneq \underline{Y} \wedge \underline{X}^{\blacktriangle} \neq \underline{Y}^{\blacktriangle})$$

$$\underline{X} \dashrightarrow \underline{Y} \Leftrightarrow (\underline{X} \subsetneq \underline{Y} \wedge \underline{X}^{\blacktriangle} = \underline{Y}^{\blacktriangle})$$

$$\underline{X} \text{ —— } \underline{Y} \Leftrightarrow \underline{X} \subseteq \underline{Y} \text{(vertical)}$$

$\underline{X} \ldots \underline{Y} \Leftrightarrow$ relation unknown (given for some cases only)

The diagrams of figures 1, 2, 3, 4, 9 and 10 are complete in the sense that language families are incomparable if there is no directed path, using both types of arrows, connecting them.

In the diagram classes with R' or S' are mentioned only if they are not identical to the corresponding ones with R or S.

4. OUTLOOK

Only non-deterministic systems have been considered in this article, giving even incomplete diagrams in some cases for lack of more results. The deterministic systems will be dealt with in a forthcoming paper since there are results enough for R, S, and L systems.

REFERENCES

1. J. R. Büchi, Regular Canonical Systems, *Arch. Math. Logik Grundlagenforsch.*, 1964, *6*, pp. 91-111.
2. A. Ehrenfeucht, J. Engelfriet, G. Rozenberg, Context Free Normal Systems and ETOL Systems, *J.C.S.S.*, 1983, *26*, pp. 34-46.
3. A. Ehrenfeucht, G. Rozenberg, Nondeterminals Versus Homomorphisms in Defining Languages for some Classes of Rewriting Systems, *A.I.*, 1974, *3*, pp. 265-283.
4. A. Gabrielian, Pure Grammars and Pure Languages Research, Report C.S.R.R., 2027, 1970 and I.J.C.M., 1981, *9*, pp. 3-16.
5. G. T. Herman, G. Rozenberg, Developmental Systems and Languages, North-Holland, 1975.
6. M. Jantzen, M. Kudlek, Homomorphic Images of Sentential Form Languages Defined by Semi-Thue Systems, Research Report FBI-HH-89/83, *Univ. Hamburg*, 1983, Record of 2nd Conference on FST & TCS, 1982, pp. 126-135, (short), TCS, 1984, *33*, pp. 13-43.
7. H. C. M. Kleijn, G. Rozenberg, A Study in Parallel Rewriting Systems, *I.C.*, 1980, *44*, pp. 134-163.

8. M. KUDLEK, Characterization of Derivation Sets of Formal Systems, *L.N.C.S.*, 1973, *2*, pp. 156-165.

9. M. KUDLEK, Comparing Several Ways of Context-independent Parallel Rewriting, *L.N.C.S.*, 1975, *28*, pp. 122-130.

10. M. KUDLEK, Context Free Normal Systems, *L.N.C.S.*, 1979, *74*, pp. 346-352.

11. M. KUDLEK, Indian Parallel Systems, Record of 2nd Conference on FST & TCS, 1982, pp. 283-289.

12. M. KUDLEK, Homomorphic Images of Sentential Form Languages Defined by Regular Systems, Research Report FBI-HH-72/86, *Univ. Hamburg*, 1986.

13. M. KUDLEK, Languages Defined by Context-free Normal Systems, Record of 3rd Conference on FST& TCS, 1983, pp. 539-549.

14. H. A. MAURER, A. SALOMAA, D. WOOD, Pure Grammars, *I.C.*, 1980, *44*, pp. 47-72.

15. M. NIELSEN, G. ROZENBERG, A. SALOMAA, S. SKYUM, Nondeterminals, Homomorphisms and Codings in Different Variations of OL-Systems. I. Deterministic Systems, *A.I.*, 1974, *4*, pp. 87-106.

16. M. NIELSEN, G. ROZENBERG, S. SALOMAA, S. SKYUM, Nondeterminals, Homomorphisms and Codings in Different Variations of OL-Systems. II. Nondeterministic Systems, *A.I.*, 1974, *3*, pp. 357-364.

17. E. POST, Formal Reduction of the General Combinatorial Decision Problem, *A.J.M.*, 1943, *65*, pp. 197-215.

18. G. ROZENBERG, A. SALOMAA, The Mathematical Theory of L-Systems, *Academic Press*, 1980.

19. A. SALOMAA, Theory of Automata, *Pergamon Press*, 1969.

20. A. SALOMAA, Parallelism in Rewriting Systems, *L.N.C.S.*, 1974, *14*, pp. 523-533.

21. R. SIROMONEY, K. KRITHIVASAN, Parallel Context-free Languages, *I.C.*, 1974, *24*, pp. 155-162.

22. S. SKYUM, Parallel Context-free Languages, *I.C.*, 1974, *26*, pp. 280-285.

23. P. M. B. VITÁNYI, Lindenmayer Systems: Structure, Languages, and Growth Functions, *Mathematisch Centrum, Amsterdam*, 1978.

24. A. WALKER, Adult Languages of L Systems and the Chomsky Hierarchy, *L.N.C.S.*, 1974, *15*, pp. 201-215.

25. A. V. GLADKIJ, Konfiguracinnye charakteristiki jazykov, *Problemy Kibernetiki*, 1963, *10*, pp. 251-260.

26. M. NOVOTNÝ, Bemerkung über ableitbare Sprachen, *Publ. Fac. Sci. Univ.*, J. E. Purkyně, Brno, ČSSR, 1965, *468*, pp. 503-507.

27. M. KUDLEK, Languages Defined by Indian Parallel Systems, in G. ROZENBERG, A. SALOMAA, Eds., *The Book of L*, 1986, pp. 233-243, Springer.