NONTERMINALS, HOMOMORPHISMS AND CODINGS IN DIFFERENT VARIATIONS OF OL-SYSTEMS

Part I & Part II

by

Mogens Nielsen
G. Rozenberg
Arto Salomaa
Sven Skyum

DAIMI PB-21
January 1974

Institute of Mathematics University of Aarhus
DEPARTMENT OF COMPUTER SCIENCE
Ny Munkegade - 8000 Aarhus C - Denmark
Phone 06-12 83 55

NONTERMINALS, HOMOMORPHISMS AND CODINGS IN DIFFERENT VARIATIONS OF OL-SYSTEMS

by

Mogens Nielsen,
G. Rozenberg,
Arto Salomaa,
Sven Skyum

I. DETERMINISTIC SYSTEMS

Summary

The use of nonterminals versus the use of homomorphisms of different kinds in the basic types of deterministic OL-systems is studied. A rather surprising result is that in some cases the use of nonterminals produces a comparatively low generative capacity, whereas in some other cases the use of nonterminals gives a very high generative capacity. General results are obtained concerning the use of erasing productions versus the use of erasing homomorphisms. The paper contains a systematic classification of the effect of nonterminals, codings, weak codings, nonerasing homomorphisms and homomorphisms for all basic types of deterministic OL-languages, including table languages.

CONTENTS

I. DETERMINISTIC SYSTEMS

II. NONDETERMINISTIC SYSTEMS

1. Introduction

The study of developmental languages has been one of the major trends in automata and formal language theory during the past few years. Developmental languages are generated by string manipulating systems (L-systems, Lindenmayer-systems) which are applied in such a way that at every step of a derivation a production is applied to every letter in the string. L-systems have been investigated in a large number of papers both from formal language theory and theoretical biology point of view (See, e.g., [4], [5], [7], [8], and their references.)

The purpose of this paper is to make a systematic study on the effect of two essentially different defining mechanisms for various types of informationless L-systems (OL-systems): the use of nonterminals and the use of homomorphic mappings of different kinds. In general, consider a rewriting system $G = <\Sigma, P, \omega>$ with alphabet Σ, set P of productions and axiom ω. Then the language $L(G)$ generated by G can be defined in two essentially different ways:

i) One specifies a subalphabet Δ of Σ and defines $L(G)$ to consist of all words over Δ which can be derived from ω using productions in P.

ii) One introduces a homomorphism h and defines $L(G)$ to consist of the images under h of all words derivable from ω by productions in P.

We refer to the first of these definitional mechanisms as "defining languages by the use of nonterminals" and, to the second, as "defining languages by the use of homomorphisms". Both mechanisms are frequently used in formal language theory, cf. [8]. The distinction between terminals and nonterminals is well motivated from the linguistic point of view because nonterminals correspond to the syntactic classes of the language. This distinction is not so well motivated in the theory of developmental languages in which the set of all strings generated by a rewriting system is of primary interest. From the point of view of developmental languages, also the homomorphic mappings (especially those in which a letter is mapped to a letter, the so-called codings) are of considerable importance. The reasons for this are as follows. When we make observations of a particular organism, and wish to describe it by strings of symbols, we first associate a symbol to each particular cell. We divide the cells into a number of types and associate the same symbol to all the cells of the same type. It is possible that the development of the organism can be described by a developmental system, but the actual system describing it uses a finer subdivision into types than what we could observe. This is often experimentally unavoidable. In this case, the set of strings generated by a given developmental system is a coding of the "real" language of the organism which the given developmental system describes.

One of the basic facts which have made the use of nonterminals interesting within the theory of developmental languages is that it was established in [2] and [3] that, for the basic families of OL and TOL languages, the use of nonterminals is equivalent to the use of codings, as far as the generative capacity is concerned. Thus, the trade-off between the two language definition mechanisms (i) and (ii) has become a very interesting and well-motivated problem for L-systems. Codings of OL languages have also been considered in [1].

This paper treats the trade-off between (i) and (ii) for all of the basic families of OL languages. In fact, a finer subdivision of (ii) is introduced by considering arbitrary homomorphisms (H), nonerasing homomorphisms (N), codings (C), and weak codings (W), the latter being homomorphisms where each letter is mapped either to a letter or to the empty word. In the present first part, the attention is restricted to deterministic systems. We hope to return to the nondeterministic case in a forthcoming paper.

After some definitions and basic lemmas, we study the effect of the different operators E (allowing the use of nonterminals), H, N, C, and W for the family of DOL languages. A similar study is made for the families PDOL, DFOL, and PDFOL obtained from DOL by the propagating restriction and/or allowing a finite set of axioms. The corresponding families obtained by using systems with tables are studied in the same way. Whenever possible, the study is extended from language families to the corresponding families of sequences. The paper ends with a discussion of a couple of open problems.

2. Definitions

The reader is assumed to be familiar with the basic notions and facts concerning developmental languages, cf. the introductory chapters of [4] whose notation and terminology we shall mostly follow. Therefore, rather than giving the definitions of the different systems in full formal detail, we will use a somewhat more descriptive way of defining them.

We use the customary Kleene operators $*$ and $+$. Thus, for an alphabet Σ, we denote by Σ^+ (resp. Σ^*) the set of all nonempty words over Σ (resp. the set of all words over Σ, including the empty word λ). For a word x, $\min(x)$ denotes the set of letters occurring in x, and $|x|$ denotes the length of x. Consider words x over some fixed alphabet $\Sigma = \{a_1, \ldots, a_n\}$. By the Parikh vector associated with x we mean the n-dimensional row vector whose i'th component equals the number of occurrences of a_i in x, for $i = 1, \ldots, n$. The partial order $\leqq$ for Parikh vectors is defined componentwise:

$$(b_1, \ldots, b_n) \leqq (b_1', \ldots, b_n')$$

iff $b_i \leqq b_i'$, for $i = 1, \ldots, n$. The relation $v_1 < v_2$ holds iff $v_1 \leqq v_2$ and $v_1 \neq v_2$.

By definition, an EOL-system is a quadruple $G = \langle \Sigma, P, \omega, \Delta \rangle$, where Σ and Δ are alphabets with $\Delta \subseteq \Sigma$, P is a finite set of context-free productions containing at least one production for every letter of Σ, and $\omega \in \Sigma^+$. The direct yield relation $\Rightarrow$ on the set Σ^* is defined as follows: $x \Rightarrow y$ holds iff there is an integer $k \geqq 1$, letters a_i and words α_i, $1 \leqq i \leqq n$, such that

$$x = a_1 \ldots a_n, \quad y = \alpha_1 \ldots \alpha_n,$$

and $a_i \to \alpha_i$ is a production in P, for each $i = 1, \ldots, n$. The relation $\Rightarrow^*$ is the reflexive transitive closure of $\Rightarrow$. The language $L(G)$ generated by G is defined by

$$L(G) = \{ w \in \Delta^* \mid \omega \Rightarrow^* w \}.$$

The EOL-system is an OL-system iff $\Delta = \Sigma$. It is deterministic (abbreviated: D) iff there is exactly one production for every letter of Σ. It is propagating (abbreviated: P) iff the right side of every production is distinct from the empty word λ. We may also combine these notions and speak, for instance, of PDOL- or EPOL-systems.

By definition, an ETOL-system is a quadruple $G = <\Sigma, \mathcal{P}, \omega, \Delta>$, where Σ, ω, and Δ are as in the definition of an EOL-system and $\mathcal{P}$ is a finite set (whose elements are called tables) such that for every $P \in \mathcal{P}$, $<\Sigma, P, \omega, \Delta>$ is an EOL-system. The direct yield relation $\Rightarrow$ means now that in the transition $x \Rightarrow y$ only productions belonging to the same table are used. The generated language is defined exactly as before, using the reflexive transitive closure of the relation $\Rightarrow$. Again, a TOL-system means that $\Delta = \Sigma$. An ETOL- or TOL-system is deterministic (resp. propagating) iff each of the underlying EOL-systems is deterministic (resp. propagating). Thus, we may speak of PDTOL- or EPTOL-systems.

We also consider generalizations of the systems defined above obtained by replacing the axiom ω by a finite set Ω of axioms. The language generated by such a system consists of the (finite) union of the languages generated by the systems obtained by choosing each element $\omega \in \Omega$ to be the axiom. This generalization is denoted by the letter F. Thus, we may speak of EPDTFOL-systems.

For any class of systems, we use the same notation for the family of languages generated by these systems. E.g., EPDTOL denotes the family of languages generated by EPDTOL-systems. (In [4], the notation $\mathfrak{F}$(EPDTOL) is used. We have chosen the simpler notation because there is no danger of confusion and the simpler notation is more convenient in the diagrams and chains of inclusions.)

By a coding we mean a length-preserving homomorphism (often also called a literal homomorphism). A weak coding is a non-length-increasing homomorphism (i.e., a homomorphism mapping every letter to a letter or to the empty word). The prefix W, C, H, or N attached to the name of a language family indicates that we are considering weak codings, codings, homomorphic images, or homomorphic images under nonerasing homomorphisms of the languages in the family, respectively.

The purpose of this paper is a systematic study of the effect of the operators E, C, N, W, H on the families DOL, PDOL, DFOL, and PDFOL, as well as with the same families with tables. More specifically, we consider the families

DOL	EDOL	CDOL	NDOL	WDOL	HDOL
PDOL	EPDOL	CPDOL	NPDOL	WPDOL	HPDOL
DFOL	EDFOL	CDFOL	NDFOL	WDFOL	HDFOL
PDFOL	EPDFOL	CPDFOL	NPDFOL	WPDFOL	HPDFOL,

as well as the same families with T added. Thus, all families we are investigating in this first part are deterministic (D is present).

We make the following definitional convention: Whenever a language L belongs to one of our families, then also $L \cup \{\lambda\}$ belongs to the same family. The convention is made in order to avoid trivial exceptions in the statements of many theorems.

Systems listed on the first two lines above define a unique sequence of words. (This is not the case for deterministic systems where F or T is present.) For them, we also consider the family of generated sequences which is denoted by S. Thus, S(EPDOL) is the family of sequences generated by EPDOL-systems.

3. Basic lemmas

We shall in this section establish some results which will be used frequently in the sequel. The results are called "lemmas" although they might have some interest in their own right. Whenever reference is made to a collection of language families, this collection is understood to be a subset of the collection of the 48 families listed in Section 2. By pure families we mean families whose name contains none of the operators E, C, N, W, H. The first two lemmas deal with the effect of erasing.

Lemma 3.1.

Assume that X is not a propagating pure family (i.e., the name of X does not contain the letter P). Then CX = NX and WX = HX.

Proof.

We prove the first equation, the proof of the second one being quite similar. The inclusion $CX \subseteq NX$ follows by definition. To prove the reverse inclusion $NX \subseteq CX$, we consider an arbitrary language $L \in NX$. Hence, there is a language L_1 generated by an X-system G_1 and a nonerasing homomorphism h_1 such that $L = h_1(L_1)$. A new

X-system G_2 with $L(G_2) = L_2$ and a coding h_2 satisfying $h_1(L_1) = h_2(L_2)$ will now be constructed. The alphabet of G_2 will consist of all letters obtained in the following fashion. Assume that a is a letter of G_1 and $h_1(a) = b_1 \dots b_k$, where each b_i is a letter and $k \geqq 1$. Then we include $a, [b_2], \dots, [b_k]$ to the alphabet of G_2. (Note that there is no distinction between terminals and non-terminals in G_1, since we are dealing with a pure family X.) The coding h_2 is now defined by

$$h_2(a) = b_1, \quad h_2([b_i]) = b_i, \quad i = 2, \dots, k.$$

The axiom (resp. axioms) of G_2 is obtained from the axiom of G_1 (resp. are obtained from the axioms of G_1) by replacing every occurrence of every letter a with the word $a[b_2] \dots [b_k]$. (Thus, if for some letter a we have $k = 1$, the occurrences of this letter remain unchanged.) The same change is made to the right side of every production of G_1. The left sides of the productions remain unaltered. The definition of G_2 is completed by adding the productions $[b_i] \to \lambda$, for each bracketed letter. (These productions are added to every table if we are dealing with a system with tables.) It is easily verified that G_2 is an X-system and that

$$L = h_1(L_1) = h_2(L_2).$$

Lemma 3.2.

Assume that we are dealing with a family $X = HY$ or $X = WY$. Then:

(i) The family X is not altered if the letter P is added to or removed from the name Y.

(ii) $HY = WY$.

(iii) The family X is closed under union.

Proof

To prove statement (i), we note that the family X_1 with P is included in the family X_2 without P, by definition. To establish the reverse inclusion $X_2 \subseteq X_1$, we consider a language L_1 generated by a Y-system G_1 and a homomorphism (resp. weak coding) h_1. A propagating Y-system G_2 is now defined by adding a new letter $\#$, replacing every production $a \to \lambda$ in G_1 by $a \to \#$ and adding the production $\# \to \#$ (to every table if we are dealing with systems with tables). The homomorphism (resp. weak coding) h_1 is extended to a homomorphism (resp. weak coding) h_2 over the new alphabet by defining

$$h_2(\#) = \lambda,\ h_2(a) = h_1(a) \text{ for } a \neq \# .$$

It is immediate that $h_2(L(G_2)) = h_1(L_1)$.

The statement (ii) is an immediate consequence of the statement (i) and Lemma 3.1.

To establish (iii), we will prove that the language $h_1(L(G_1)) \cup h_2(L(G_2))$ is in WY, for arbitrary two Y-systems G_1 and G_2 and weak codings h_1 and h_2. This is obvious if F occurs in Y. Hence by (i), it suffices to consider the cases Y = PDOL and Y = PDTOL. We give the proof for the former case, since the latter case can be established by repeating the same construction for every table.

Thus, assume that G_1 and G_2 are PDOL-systems with alphabets $\{a_1, \ldots, a_m\}$ and $\{b_1, \ldots, b_n\}$, respectively. Without loss of generality, we assume that these alphabets are disjoint. (This situation can be reached by a suitable renaming because we are interested only in the language $h_1(L(G_1)) \cup h_2(L(G_2))$.) A new PDOL-system G_3 and a weak coding h will now be defined. The alphabet of G_3 equals

$$\{a_i, a_i' \mid 1 \leqq i \leqq m\} \cup \{b_i, b_i' \mid 1 \leqq i \leqq n\} .$$

The axiom of G_3 equals $w_1 w_2'$, where w_1 is the axiom of G_1 and w_2' is obtained from the axiom of G_2 by replacing every letter b_i with b_i'. For any production $a_i \to x_i$ of G_1 and $b_i \to y_i$ of G_2, $a_i \to x_i'$ and $b_i \to y_i'$ are productions of G_3. (Here x_i' and y_i' are obtained from x_i and y_i by replacing every letter with the corresponding primed one.) Furthermore, G_3 contains the productions

$$a_i' \to a_i, \text{ for } 1 \leqq i \leqq m, \text{ and } b_i' \to b_i, \text{ for } 1 \leqq i \leqq n.$$

The weak coding h is defined by

$$h(a_i) = h_1(a_i), \ h(b_i) = h_2(b_i), \ h(a_i') = h(b_j') = \lambda \ .$$

Clearly,

$$h(L(G_3)) = h_1(L(G_1)) \cup h_2(L(G_2)).$$

This completes the proof of Lemma 3.2.

Remark

We have stated Lemmas 3.1 and 3.2 for D-families only, i.e., in the form needed in this paper. However, the lemmas are valid (and the proofs the same) for the corresponding nondeterministic families.

Lemma 3.3

HDTOL $\subseteq$ EDTOL and NPDTOL $\subseteq$ EPDTOL.

Proof

We prove the first inclusion. The second inclusion is established by exactly the same method.

Assume that G is a DTOL-system, h is an arbitrary homomorphism and $L = h(L(G))$. We construct an EDTOL-system G_1 such that $L = L(G_1)$. Without loss of generality, we assume that the alphabet Σ of G is disjoint from the target alphabet Σ_1 of h. The alphabet of G_1 equals, by definition, $\Sigma \cup \Sigma_1$ and Σ_1 is the terminal alphabet. The axiom of G_1 is that of G. The tables of G_1 are obtained from those of G by adding to each table the production $a_1 \to a_1$, for every $a_1 \in \Sigma_1$, and by introducing one additional table

$$[\{a \to h(a) \mid a \in \Sigma\} \cup \{a_1 \to a_1 \mid a_1 \in \Sigma_1\}].$$

Clearly, $L = L(G_1)$ and Lemma 3.3 follows.

Our two last lemmas give examples of languages belonging (resp. not belonging) to some families.

Lemma 3.4

The language $L = \{a^n \mid n \geq 1\}$ belongs to the family CPDOL but it belongs to neither EDFOL nor DTFOL.

Proof

The first assertion follows by considering the PDOL-system

$$<\{a, a_1\}, \{a_1 \to aa_1, a \to a\} a_1>$$

and the coding h defined by

$$h(a_1) = h(a) = a.$$

To prove the second assertion, it suffices to consider any finite number of EDOL-systems $G_1, \ldots, G_k$ with terminal alphabet $\{a\}$ and let

$$L' = L(G_1) \cup \ldots \cup L(G_k).$$

Assume that a^{i_j} is the first terminal word in the sequence generated by G_j, for $j = 1, \ldots, k$, such that $i_j > 1$. Let p be a prime number greater than $\max\{i_j \mid j = 1, \ldots, k\}$. Then $a^p \notin L'$ and, consequently, $L' \neq L$. This shows that $L \notin$ EDFOL.

Consider next an arbitrary DTFOL-system $G = <\{a\}, \{[a \to a^{j_1}], \ldots, [a \to a^{j_m}]\}, \{a^{i_1}, \ldots, a^{i_k}\}>$. (Here the last item is the set of axioms.) Let p be a prime number greater than

$$\max\{i_n, j_t \mid 1 \leqq n \leqq k, 1 \leqq t \leqq m\}.$$

Then $a^p \notin L(G)$, which shows that $L \notin$ DTFOL.

Lemma 3.5

The language

$$L = \{a^{2^n} \mid n \geqq 1\} \cup \{b^n \mid n \geqq 1\}$$

belongs to the family WPDOL but it does not belong to the family NDOL.

Proof

The first assertion is an immediate consequence of Lemma 3.2 (iii), Lemma 3.4 and the obvious fact that $\{a^{2^n} \mid n \geqq 1\}$ is in WPDOL (and even in PDOL). To prove the second assertion, we assume the contrary: $L \in$ NDOL. Then, by Lemma 3.1, $L \in$ CDOL. Let G be a DOL-system and h a coding such that $L = h(L(G))$. Then the alphabet of G can be divided into two disjoint subalphabets Σ_a and Σ_b such that

$$h(c) = a, \text{ for } c \in \Sigma_a \text{ ; } h(c) = b, \text{ for } c \in \Sigma_b .$$

We now make use of the fact that any set of mutually incomparable Parikh vectors is finite. (This fact is well-known and also easy to establish directly. Cf. [6], where similar arguments are used.) Consider the sequence

$$w_1 = \omega, \ w_2, \ w_3, \ldots .$$

generated by G. Every word in this sequence is either over the alphabet Σ_a or over the alphabet Σ_b. There are infinitely many words over Σ_a in this sequence. Consequently, there are natural numbers m and n such that the Parikh vectors of w_m and w_{m+n} are comparable, and w_m and w_{m+n} are over Σ_a. Since L is an infinite language, the Parikh vector of w_{m+n} must be greater than that of w_m. Since w_m and w_{m+n} are over Σ_a,

$$|w_{m+n}| \geqq 2 \cdot |w_m| .$$

This implies that, for any $i \geqq 0$,

$$(*) \qquad |w_{m+in}| \geqq 2^i \cdot |w_m| .$$

(Note that the Parikh vectors of all words w_{m+in} must be comparable.)

We now obtain a contradiction by considering words over Σ_b in the sequence. Let t be the length of the longest right side among the productions of G. Denote $q = |w_m|/t^n$. Consider any word w_j in the sequence, where the index j satisfies

$$m + in < j < m + (i+1)n,$$

for some $i \geqq 0$. Clearly,

$$t^n \cdot |w_j| \geqq w_{m+(i+1)n}.$$

From this we obtain by (*)

$$|w_j| \geqq 2^{i+1} q.$$

If we now choose a value i_0 such that

$$2^{i_0+1} q > m + (i_0+1)n$$

it follows that $\{b^j \mid 1 \leqq j \leqq m + (i_0+1)n\}$ is not a subset of $h(L(G))$. This contradiction completes the proof of Lemma 3.5.

4. Systems without tables

In this section the effects of the operators E, C, N, W, and H on the families DOL, PDOL, DFOL, and PDFOL are studied.

Theorem 4.1 $\quad DOL \subsetneq EDOL$.

Proof

The inclusion follows from definition and that it is proper is seen from the finite language $L = \{abb, aacc\}$ which is in EDOL but not in DOL.

$L \in EDOL$. $L = L(G_1)$, where G_1 is the EDOL-system

$$G_1 = <\{a, b, c, d\}, \{a \to aa, b \to c, c \to d, d \to d\}, abb, \{a, b, c\}>$$

$L \notin DOL$. Assume the existence of a DOL-system $G_2 = <\{a, b, c\}, P, \omega, \{a, b, c\}>$ such that $L = L(G_2)$, then either 1) $\omega = abb$ or 2) $\omega = aacc$.

1) $\omega = abb$. The deterministic production set of G_2 must satisfy $abb \Rightarrow aacc$, otherwise $L \neq L(G_2)$. The only possible productions for the letter b are $b \to c$ or $b \to \lambda$, which implies that the only possible productions for the letter a are $a \to aa$ or $a \to aacc$. In both cases aacc will derive a word not belonging to L.

2) $\omega = aacc$. P must satisfy $aacc \Rightarrow abb$. But then either $a \to \lambda$ or $c \to \lambda$, and in both cases the production of the other letter is impossible to define in such a way that the above mentioned requirement to P is met. This completes the proof of theorem 4.1.

Theorem 4.2 $\quad EDOL \subsetneq CDOL$.

Proof

Let $G = <\Sigma, P, \omega, \Delta>$ be an EDOL-system. The following describes the construction of a DOL-system $G = <\Sigma', P', \omega', \Sigma'>$ and a coding h from Σ' into Δ such that $L(G) = h(L(G'))$.

Consider the sequence of words from Σ^* generated by G, $\omega = \omega_1, \omega_2, \omega_3, \ldots$. There exist natural numbers n and m such that $\min(\omega_m) = \min(\omega_{m+n})$, which implies that for any $i \geq 0$ and any j, $0 \leq j < n$:

$$(1) \qquad \min(\omega_{m+j}) = \min(\omega_{m+ni+j}).$$

Let d_k denote the cardinality of $\min(\omega_k)$, $1 \leq k < m+n$. Define

$$N_\Delta = \{k \in N \mid 1 \leq k < m+n, \ \min(\omega_k) \subseteq \Delta\} .$$

For any $k \in N_\Delta$ introduce new symbols not in Σ

$$\Sigma_k = \{a_{kj} \mid 1 \leq j \leq d_k\} ,$$

and define isomorphism f_k mapping $\min(\omega_k)$ onto Σ_k, where

$$f_k(a) = a_{kj} \quad \text{iff} \quad a \text{ is the } j\text{'th symbol of } \min(\omega_k), \ k \in N_\Delta$$

Note that the f_k's are defined for some fixed enumerations of the sets $\min(\omega_k)$. Σ' is going to be the union of the above defined Σ_k's:

$$\Sigma' = \bigcup_{k \in N_\Delta} \Sigma_k \quad .$$

Define k_1 and k_2 as the minimal and maximal elements of N_Δ.

For any of the letters a_{kj} where $k \neq k_2$ define production in P':

$$a_{kj} \to f_{k'}(\alpha)$$

where k' is the smallest element in N_Δ greater than k and α is the string derived from $f_k^{-1}(a_{kj})$ in $(k'-k)$ steps in G.

It follows from (1) that $L(G)$ is finite if $k_2 < m$. If this is the case then define for any j, $1 \leq j \leq d_{k_2}$ production in P':

$$a_{k_2 j} \to a_{k_2 j} \ .$$

Otherwise, let k_3 denote the minimal element in N_Δ greater than or equal to m, and define productions in P' for any j, $1 \leq j \leq d_{k_2}$, for which $f_{k_2}^{-1}(a_{k_2 j})$ derives some string $\alpha \in \Delta^*$ in $(n-k_2+k_3)$ steps in G:

$$a_{k_2 j} \to f_{k_3}(\alpha) \ .$$

Note that the use of f_{k_3} is well-defined since $\min(\omega_{k_2+(n-k_2+k_3)}) = \min(\omega_{k_3})$ (from the fact that $k_3 \geq m$ and (1) above). Finally define the coding h from Σ' into Δ in the way that for every $a_{kj} \in \Sigma'$: $h(a_{kj}) = f_k^{-1}(a_{kj})$. Then

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

where $G' = <\Sigma', P', f_{k_1}(\omega_{k_1}), \Sigma'>$, and this proves the inclusion of the theorem. That the inclusion is proper is seen from Lemma 3.4.

<u>Theorem 4.3</u> $\quad$ CDOL = NDOL .

<u>Proof</u>

Follows from Lemma 3.1.

<u>Theorem 4.4</u> $\quad$ NDOL $\subsetneq$ WDOL .

<u>Proof</u>

Follows from Lemma 3.5 and Theorem 4.3.

<u>Theorem 4.5</u> $\quad$ WDOL = HDOL .

<u>Proof</u>

Follows from Lemma 3.1.

The effects of the E-operator and the various homomorphisms on DOL-systems can now be summarized (Theorems 4.1-4.5).

(D1) $\quad$ DOL $\subsetneq$ EDOL $\subsetneq$ CDOL $=$ NDOL $\subsetneq$ WDOL $=$ HDOL .

Theorem 4.6 $\quad$ 1) PDOL $\subsetneq$ EPDOL .

2) EPDOL $\subsetneq$ CPDOL .

3) NPDOL $\subsetneq$ HPDOL .

4) WPDOL $=$ HPDOL .

Proof

1) Follows from the proof of Theorem 4.1 and the propagating example considered in this proof.

2) If the EDOL-system considered in the proof of Theorem 4.2 is propagating, then so is the constructed CDOL-system, i.e., the proof of Theorem 4.2 is also valid in the propagating case.

3) The inclusion follows from definition and it is proper by Lemma 3.5.

4) Follows from Lemma 3.2.

Theorem 4.6 can be illustrated in a diagram corresponding to (D1).

(D2) $\quad$ PDOL $\subsetneq$ EPDOL $\subsetneq$ CPDOL $\subseteq$ NPDOL $\subsetneq$ WPDOL $=$ HPDOL .

Before we continue considering systems with a finite set of axioms we shall briefly mention some results for the families of sequences generated by systems considered so far.

Corollary 4.7

Theorems 4.1-4.6 are also valid for the families of generated sequences.

Proof

Follows from slight modifications of the proofs in this and the previous section.

Theorem 4.8 $\quad$ S(CPDOL) $\subsetneq$ S(NPDOL).

Proof

The inclusion follows from definition. Obviously, the sequence of lengths generated by any CPDOL-system is non-decreasing, which is not always the case for a NPDOL-system. Consider as a trivial example the PDOL-system $G = \langle \{a, b\}, \{a \to b, b \to b\}, a, \{a, b\} \rangle$ and the nonerasing homomorphism h, for which $h(a) = aa$ and $h(b) = b$.

Corollary 4.7 and Theorem 4.8 are illustrated in the diagrams D3 and D4.

(D3) $S(DOL) \subsetneq S(EDOL) \subsetneq S(CDOL) = S(NDOL) \subsetneq S(WDOL) = S(HDOL)$.
(D4) $S(PDOL) \subsetneq S(EPDOL) \subsetneq S(CPDOL) \subsetneq S(NPDOL) \subsetneq S(WPDOL) = S(HPDOL)$

And now we continue the line of this section, investigating the behaviour of systems with a finite set of axioms.

Theorem 4.9 $\qquad$ $DFOL \subsetneq EDFOL$ and $PDFOL \subsetneq EPDFOL$.

Proof

Both of the inclusions follow from definition.
Consider the EPDOL-system

$$G = \langle \{a, b, c, d, e\}, P, cab, \{a, b, c, d\} \rangle$$

where $P = \{a \to aa, b \to bb, c \to d, d \to e, e \to d\}$.

$$\text{Clearly, } L(G) = \{cab\} \cup \{da^{2^{2n+1}} b^{2^{2n+1}} \mid n \geq 0\}$$

Assume that there exists a DFOL-system $G' = \langle \{a, b, c, d\}, P', \omega', \{a, b, c, d\} \rangle$, such that $L(G) = L(G')$. The word da^2b^2 occurs somewhere in one of the sequences generated by G' (from one of the axioms of G'). The assumptions that this word derives itself or the word cab lead both of them to contradictions, i.e., there exists an $n \geq 1$ such that

$$da^2b^2 \Rightarrow da^{2^{2n+1}} b^{2^{2n+1}},$$

which implies that $\{d \to d, a \to a^{2^{2n}}, b \to b^{2^{2n}}\} \subseteq P'$ (all other assumptions lead to contradictions). But then, no matter how the production of letter c, $c \to \omega_c$, is defined in P', the word cab derives a word of the form

$$\omega_c a^{2^{2n}} b^{2^{2n}}, \quad n \geq 1,$$

which is not a word in L(G).
This proves that both of the inclusions of the theorem are proper.

Theorem 4.10 EDFOL $\subsetneq$ CDFOL and EPDFOL $\subsetneq$ CPDFOL .

Proof

Let G be a given EDFOL-system. Construct for each of the generated sequences of G (one from each axiom) a CDOL-system generating the same sequence (according to the proof of Theorem 4.2), in such a way that the constructed CDOL-systems are working in mutually disjoint alphabets. Let G' be the CDFOL-system for which the alphabet is the union of the alphabets of these constructed systems, the production set is the union of the production sets of the systems, the axioms are the axioms of these systems, and the coding is the unique extension of all the constructed codings to the alphabet of G'. Then L(G) = L(G').

If G is a propagating EDFOL-system then so is the constructed system G', and this proves both of the inclusions of the theorem.

Lemma 3.4 implies that the inclusions are proper.

Theorem 4.11 CDFOL = NDFOL .

Proof

Follows from Lemma 3.1.

Theorem 4.12 CPDFOL = NPDFOL .

Proof

NPDFOL is included in NDFOL and by Theorem 4.11 it is then sufficient to prove CDFOL $\subseteq$ CPDFOL, which is true iff DOL $\subseteq$ CPDFOL (note that the composition of two codings is again a coding).

Let $G = <\Sigma, P, \omega, \Sigma>$ then be a DOL-system, and let ω_i ($\omega_i^{(a)}$) denote the i'th word generated from ω (from the letter a) using productions from P. Define

$$\Sigma^f = \{a \in \Sigma \mid \{\omega_i^{(a)} \mid i \geq 0\} \text{ is finite}\}.$$

Then for every $a \in \Sigma^f$ there exist constants $n_a, m_a \in N$, $n_a, m_a \geq 1$, such that $\omega_{m_a}^{(a)} = \omega_{m_a+n_a}^{(a)}$, which implies that

(2) $\forall i,\ 0 \leq i < n_a,\ \forall j \in N : \omega_{m_a+i}^{(a)} = \omega_{m_a+i+jn_a}^{(a)}$.

Define $p = \prod_{a\in\Sigma^f} (m_a \cdot n_a)$, then (2) above implies that

$$\forall a \in \Sigma^f,\ \forall j \geq 1 \ : \ \omega_p^{(a)} = \omega_{pj}^{(a)} \quad .$$

Let s denote the length of the longest word derived from any letter from Σ in p steps using productions from P, so

$$s = \max_{a\in\Sigma} \{|\omega_p^{(a)}|\} \quad .$$

Then clearly, there exists a constant $q \geq 1$, such that for any $a \in \Sigma\backslash\Sigma^f$ (a deriving an infinite language), $\omega_{p(q+1)}^{(a)}$ is of length greater than s.

The idea is now to simulate G in a system, where an occurrence of a letter a is replaced by information about the letter from which this a was derived in G in p steps, say b, and the position of this particular a in $\omega_p^{(b)}$. The productions of the system will simulate the behaviour of letters from G in steps of length pq.

Formally $L(G)$ will be generated by a CDFOL-system G' with coding h', where

$$G' = <\Sigma \cup (\Sigma \times \{1,2,\ldots,s\}),\ P',\ \{\omega_i \mid 0 \leq i < pq\},\Sigma \cup (\Sigma \times \{1,2,\ldots,s\})>.$$

The productions of P' are defined as follows. For each $a \in \Sigma$, the right-hand side of its P'-production is the string from $(\Sigma \times \{1,2,\ldots,s\})^*$ obtained from $\omega_{pq}^{(a)}$ by replacing any occurrence of a letter, say b, with the symbol $(c,i) \in \Sigma \times \{1,2,\ldots,s\}$, iff the particular b is the i'th letter (from left to right) generated (in p steps) from an occurrence of the letter c in $\omega_{p(q-1)}^{(a)}$. If $\omega_{pq}^{(a)} = \lambda$ then $a \to \lambda$ will be a production of P'.

For each $(a, i) \in \Sigma \times \{1, 2, \dots, s\}$ for which $i < |\omega_p^{(a)}|$, if the i'th symbol of $\omega_{p(q+1)}^{(a)}$ is derived as the j'th symbol from an occurrence of the letter b in $\omega_{pq}^{(a)}$, then $(a, i) \to (b, j)$ is a production in P'. For the symbol $(a, |\omega_p^{(a)}|)$, the right-hand side of its production is the string from $(\Sigma \times \{1, 2, \dots, s\})^+$ of length $(|\omega_{p(q+1)}^{(a)}| - |\omega_p^{(a)}| + 1)$, where the i'th symbol is defined as the symbol (b, j) for which the $(|\omega_p^{(a)}| - 1 + i)$'th letter of $\omega_{p(q+1)}^{(a)}$ is derived as the j'th letter from an occurrence of the letter b in $\omega_{pq}^{(a)}$. Note that by the construction of p and q all (a, i)-productions defined above are well-defined, and none of them are λ-productions.

For all symbols $(a, i) \in \Sigma \times \{1, 2, \dots, s\}$ for which the productions in P' have not been defined above, let $(a, i) \to (a, i)$ be in P'. (These symbols will all be useless symbols in the sense that they will never occur in L(G').)

Finally define the coding h' as follows:

$\forall a \in \Sigma$:	$h'(a) = a$
$\forall a \in \Sigma,\ 1 \le i \le |\omega_p^{(a)}|$:	$h'((a, i)) = b$ iff the i'th letter in $\omega_p^{(a)}$ is a b
$\forall a \in \Sigma,\ |\omega_p^{(a)}| < i \le s$:	$h'((a, i)) = a$ (a is a useless symbol).

From this construction it follows that $L(G) = h'(L(G'))$. Furthermore, only the axioms of G' are over the alphabet Σ, all other generated words are over the alphabet $\Sigma \times \{1, 2, \dots, s\}$, i.e., λ-productions are applicable only on the axioms of G'. But then define P'' as the production set obtained from P' by replacing all a-productions by identity-productions ($a \to a$ for every $a \in \Sigma$). Then $L(G) = h'(L(G''))$ where G'' is the following PDFOL-system:

$$G'' = <\Sigma \cup (\Sigma \times \{1, 2, \dots, s\}), P'', \{\omega_i \mid 0 \le i < pq\} \cup \{\omega_i' \mid 0 \le i < pq\},$$

$$\Sigma \cup (\Sigma \times \{1, 2, \dots, s\} >,$$

(ω_i' denotes the string generated in 1 step from ω_i in G'). Note that the original axioms from G' generate themselves in G'', and the words generated directly from the axioms in G' are now included as axioms in G''.

This completes the proof of theorem 4.12.

<u>Theorem 4.13</u> WPDFOL = NPDFOL and WDFOL = NDFOL .

<u>Proof</u>

It follows from the proof of Theorem 4.12 and Lemma 3.2 that it is sufficient to prove that WPDOL $\subseteq$ NPDFOL.

Let $G = <\Sigma, P, \omega, \Sigma>$ be a PDOL-system and h a weak coding, $h : \Sigma \to \Delta$.

For every $a \in \Sigma$, there exist constants $m_a, n_a \in N$, $m_a, n_a \geq 1$, such that

$$\min(\omega^{(a)}_{m_a}) = \min(\omega^{(a)}_{m_a+n_a})$$

where $\omega^{(a)}_i$ as in the proof of Theorem 4.12 denotes the i'th word from Σ^* derived from the letter a, using productions from P. This implies that

$$(3) \qquad \forall\, i,\ 0 \leq i < n_a,\ \forall\, j \in N : \min(\omega^{(a)}_{m_a+i}) = \min(\omega^{(a)}_{m_a+i+jn_a}).$$

Define $p = \prod_{a\in\Sigma} (n_a \cdot m_a)$. Then by (3) above the following is true for every letter $a \in \Sigma$:

$$\forall\, j \in N,\ j \geq 1 \ : \ h(\omega^{(a)}_{pj}) \neq \lambda$$
iff
$$h(\omega^{(a)}_p) \neq \lambda.$$

Define
$$\Sigma^1 = \{ a \in \Sigma \mid h(\omega^{(a)}_p) \neq \lambda \}$$

and a homomorphism $f : \Sigma \to \Sigma^1$, where

$$\forall\, a \in \Sigma :\ f(a) = \begin{cases} a & \text{if } a \in \Sigma^1 \\ \lambda & \text{if } a \notin \Sigma^1 \end{cases}$$

Consider now the PDFOL-system:

$$G' = <\Sigma^1, P', \{f(\omega_i) \mid 0 \leq i < p \wedge f(\omega_i) \neq \lambda\}, \Sigma^1>$$

where ω_i denotes the i'th word generated in G, and P' contains the productions $a \to f(\omega_p^{(a)})$ for every $a \in \Sigma$. If h' is defined as the non-erasing homomorphism $h' : \Sigma^1 \to \Delta$ for which $h'(a) = h(\omega_p^{(a)})$ for every $a \in \Sigma^1$, then

$$h'(L(G')) \supseteq h(L(G)) \setminus \{h(\omega_i) \mid 0 \leq i < p\}.$$

Assume that Σ^1 and Δ are disjoint, then consider

$$G'' = <\Sigma^1 \cup \Delta, P'', \{f(\omega_i) \mid 0 \leq i < p\} \cup \{h(\omega_i) \mid 0 \leq i < p\}, \Sigma^1 \cup \Delta>$$

where P'' contains all productions from P' and identity-productions for all letters $a \in \Delta$. If h'' is defined as the non-erasing homomorphism $h'' : \Sigma^1 \cup \Delta \to \Delta$ for which $h''(a) = h'(a)$ if $a \in \Sigma^1$ and $h''(a) = a$ if $a \in \Delta$, then $h''(L(G'')) = h(L(G))$, which proves the theorem.

<u>Theorem 4.14</u> WDFOL = HDFOL and WPDFOL = HPDFOL.

<u>Proof</u>

Follows from Lemma 3.1.

We are now able to summarize Theorems 4.9–4.14 in the following diagrams.

(D5) DFOL $\subsetneq$ EDFOL $\subsetneq$ CDFOL = NDFOL = WDFOL = HDFOL.

(D6) PDFOL $\subsetneq$ EPDFOL $\subsetneq$ CPDFOL = NPDFOL = WPDFOL = HPDFOL.

5. Mixed diagram and relations to the Chomsky hierarchy

In this section the effects of the P-restriction and the F-extension on DOL-systems are studied in relation to the effects of the operators E, C, N, W, and H, i.e., we investigate how diagrams D1, D2, and D5 relate to each other. At the end of the section we also relate all the families considered in the previous section to the Chomsky hierarchy.

Lemma 5.1

The language $L = \{(aba)^{2^n} \mid n \geq 0\}$ belongs to the family DOL, but not to the family EPDOL.

Proof

The first assertion follows by considering the DOL-system,

$$G = < \{a, b\}, \{a \to aba, b \to \lambda\}, aba, \{a, b\} >,$$

for which $L = L(G)$.
Assume that there exists an EPDOL-system G', such that $L = L(G')$. Since G' is propagating, the word aba must derive in some number of steps, say k, the word abaaba. But since G' is also deterministic, then either a or b must derive in k steps the empty word, which is a contradiction to the assumption that G' is propagating.

Lemma 5.2

The language

$$L = \{a^{2^n} \mid n \geq 1\} \cup \{cb^n \mid n \geq 1\}$$

belongs to the family DFOL, but it does not belong to the family NDOL.

Proof

That L belongs to the family DFOL is seen by considering the DFOL-system:

$G = < \{a,b,c\}, \{a \to aa, b \to b, c \to cb\}, \{aa, cb\}, \{a,b,c\} >$

for which $L = L(G)$.

The proof of the fact that L does not belong to the family NDOL follows the same lines as the second part of the proof of Lemma 3.5.

Theorem 5.3 DOL $\subsetneq$ NPDOL .

Proof

A constructive proof of this theorem, originally due to Jan van Leeuwen, can be found in [4].

Theorem 5.4 CDOL = NPDOL .

Proof

CDOL is included in NPDOL by Theorem 5. and the fact that the concatenation of a coding and a non-erasing homomorphism is again a non-erasing homomorphism. NPDOL is included in CDOL by Theorem 4.3.

Theorem 5.5 PDOL $\subsetneq$ DOL and EPDOL $\subsetneq$ EDOL .

Proof

The inclusions follow by definition and they are proper by Lemma 5.1.

Theorem 5.6

The families DOL and EPDOL are incomparable.

Proof

Follows by Lemma 5.1 and the proofs of Theorems 4.1 and 4.6.

Theorem 5.7 DOL $\subsetneq$ DFOL and EDOL $\subsetneq$ EDFOL .

Proof

The inclusions follow by definition and they are proper by Lemma 5.2.

Theorem 5.8

The family DFOL is incomparable to the families EDOL and EPDOL.

Proof

Follows by the proof of Theorem 4.9 and Lemma 5.2.

Theorem 5.9

The families CPDOL and CDOL are incomparable to the families EDFOL and DFOL.

Proof

Follows by Lemmas 3.4 and 5.2.

Finally we are able to summarize results from this and the previous sections in the following mixed diagram. The meaning of the diagram is the following. If two nodes labelled X and Y are connected by an edge (oriented edge), the node X being below the node Y, then $X \subseteq Y$ ($X \subsetneq Y$). If two nodes labelled X and Y are connected by a broken edge then X and Y are incomparable.

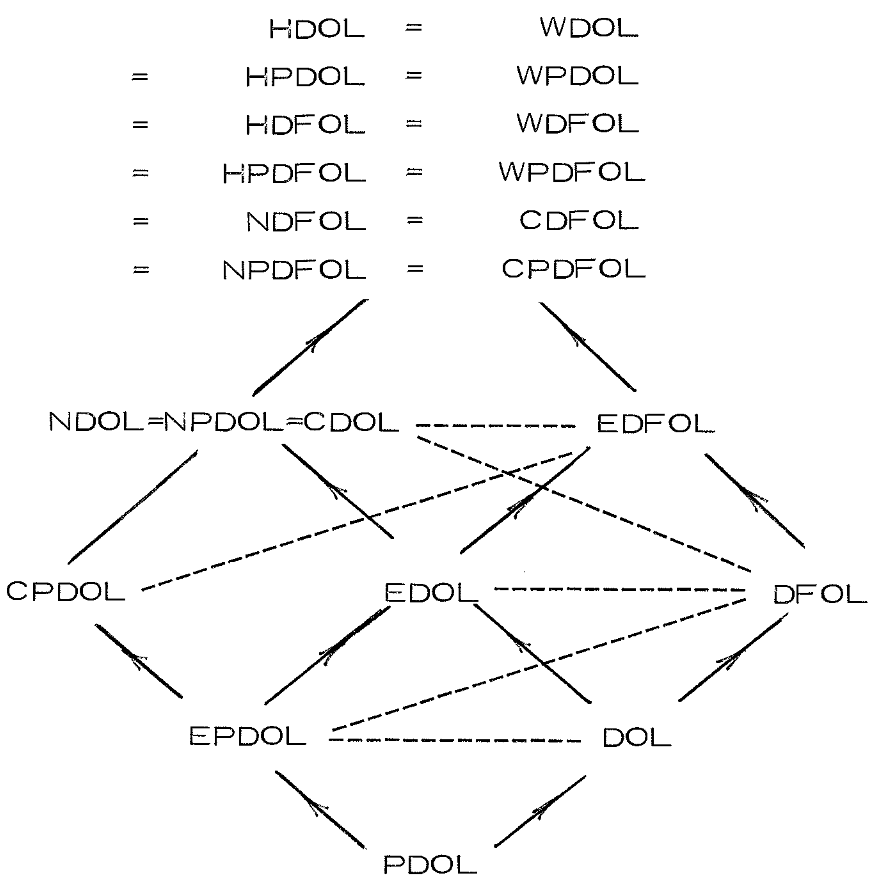

Figure 1. Mixed diagram

Before we start discussing systems with tables, we shall briefly relate all families without tables to the families of the Chomsky hierarchy.

Lemma 5.10

For any language L belonging to one of the variations of the OL-families considered in sections 4 and 5, there exists a constant $k \in N$, such that for any $n \in N$, the number of words from L of length n is less than k.

Proof

A proof of the lemma for the family PDOL can be found in [7] using the theory of growth-functions. Since a coding is length-preserving, this result carries over directly to the family CPDOL, and then, obviously, also to the family CPDFOL. This proves the lemma (see the mixed diagram in Figure 1).

Theorem 5.11

All families of languages considered in sections 4 and 5 are incomparable with the family of regular languages and the family of context-free languages.

Proof

The smallest of the families considered, the family PDOL, contains the non-context-free language

$$L = \{ a^{2^n} \mid n \geq 0 \}.$$

On the other hand, consider the languages

$$L_1 = \{ a, b\}^*$$
$$L_2 = \{ x \in L_1 \mid \#_a(x) = \#_b(x)\}.$$

L_1 is a regular language, L_2 is a context-free language, and none of the two languages is in any of the considered families by Lemma 5.10.

Theorem 5.12

All families of languages considered in sections 4 and 5 are properly included in the family of context-sensitive languages.

Proof

A proof of the theorem for the family OL (and thereby DOL) can be found in [8]. Since the family of context-sensitive languages is closed under union and non-erasing homomorphism, then this result carries over to the family NDFOL, and this proves the theorem (see the mixed diagram in Figure 1).

6. The table case

We first consider the effect of the operators E, C, N, W, H on DTOL-systems.

Theorem 6.1 DTOL $\subsetneq$ CDTOL .

Proof

The proper inclusion follows from Lemma 3.4.

Theorem 6.2 CDTOL = NDTOL .

Proof

Follows from Lemma 3.1 .

Theorem 6.3 NDTOL = EDTOL .

Proof

The inclusion NDTOL $\subseteq$ EDTOL follows from Lemma 3.3. The reverse inclusion EDTOL $\subseteq$ NDTOL is established by showing, using the method of [3] that EDTOL $\subseteq$ CDTOL. Indeed, this method is applicable as such because the sets Contr (c_i, τ) (cf. [3]) will be singletons and, consequently, the resulting tables will be deterministic.

Theorem 6.4 WDTOL = HDTOL .

Proof

Follows from Lemma 3.2.

Theorem 6.5 EDTOL = WDTOL .

Proof

EDTOL $\subseteq$ WDTOL follows from Theorems 6.3 and 6.4 and the fact that NDTOL $\subseteq$ HDTOL. WDTOL $\subseteq$ EDTOL follows from Lemma 3.3.

We have established the following diagram for DTOL-systems:

(D7) DTOL $\subsetneq$ CDTOL = NDTOL = EDTOL = WDTOL = HDTOL .

Theorem 6.6 PDTOL $\subsetneq$ CPDTOL and WPDTOL = HPDTOL.

Proof

The proofs for Theorems 6.1 and 6.4 are also valid for propagating table systems.

Theorem 6.7 EPDTOL = WPDTOL.

Proof

The inclusion EPDTOL $\subseteq$ WPDTOL is again established by the method of [3].

The reverse inclusion is established as follows.

Let $G = <\Sigma, \mathcal{P}, \omega, \Sigma>$ be an arbitrary PDTOL-system where $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ and $h : \Sigma^+ \to \Delta^*$ a weak coding.

First we will define some functions which we will use later in the proof. Define for all $S \in 2^\Sigma$ the function $g'_S : \Sigma^+ \to \Sigma^*$ to be the weak coding:

$$a \in \Sigma : g'_S(a) = \begin{cases} \lambda \text{ if } a \in S \\ a \text{ otherwise} \end{cases}$$

and define $g_S : \Sigma^+ \to \Sigma^+ \cup \{\#\}$ to be the following function:

$$\omega \in \Sigma^+ : g_S(\omega) = \begin{cases} g'_S(\omega) \text{ if } g'_S(\omega) \neq \lambda \\ \# \text{ otherwise} \end{cases}$$

and finally $f : (\Sigma^+ \cup \{\#\}) \times 2^\Sigma \to (\Sigma \times 2^\Sigma)^* \cup \{\#\}$ as follows:

$$S \in 2^\Sigma,\ a_1 a_2 \ldots a_n \in \Sigma^*;\ f((a_1 a_2 \ldots a_n), S) = (a_1, S)(a_2, S)\ldots(a_n, S)$$

$$f(\#, S) = \#.$$

We are now ready to define the equivalent EPDTOL-system

$$H = <\Sigma \times 2^\Sigma \cup \Delta \cup \{\#, X\},\ \mathcal{P}',\ X,\ \Delta>\ (X \notin \Sigma \cup \Delta).$$

$\mathcal{P}'$ is defined as follows:

$$\mathcal{P}' = P \cup \bigcup_{\substack{1 \le i \le n \\ S \in 2^\Sigma}} P_{i,S}, \text{ where}$$

1) P contains the following productions:

$$(a,S) \in \Sigma \times 2^{\Sigma} : (a,S) \to \begin{cases} h(a) \text{ if } h(a) \neq \lambda \text{ and } S \subseteq h^{-1}(\lambda) \\ \# \quad \text{otherwise} \end{cases}$$

Also,
$\# \to \#$, $X \to X$ and $\delta \to \delta$ for all $\delta \in \Delta$ are in P.

2) $P_{i,S}$ contains the following productions:

$(a,S') \in \Sigma \times 2^{\Sigma} : (a,S') \to f(g_S(\omega_a), S'')$ where $a \to \omega_a$ is a production in P_i and
$S'' = S \cup \bigcup_{b \in S'} \min(\omega_b)$ where $b \to \omega_b$ is a production in P_i.

Also,
$\# \to \#$, $X \to f(g_S(\omega), S \cap \min(\omega))$ and $\delta \to \delta$ for all $\delta \in \Delta$ are in $P_{i,S}$.

We now have that $h(L(G)) = L(H)$ because we can show the inclusions in both directions with the following argument.

Assume that $a_1 a_2 \ldots a_m \in L(G)$.

Then $\omega = \omega_0 \underset{P_{i_1}}{\Rightarrow} \omega_1 \underset{P_{i_2}}{\Rightarrow} \omega_2 \underset{P_{i_3}}{\Rightarrow} \ldots\ldots \underset{P_{i_k}}{\Rightarrow} \omega_k = a_1 a_2 \ldots a_m$

Define k+1 sets $A_0, \ldots, A_k$ where $A_i \subseteq \min(\omega_i)$ in the following way:

A_0 is the set of symbols in ω_0 which derives a subword of ω_k where at least one symbol belongs to $\Sigma \setminus h^{-1}(\lambda)$.

$a \in A_i$, $i \geq 1$, iff

1) There exists an occurrence of a in ω_i such that the ancestor in ω_{i-1} of that occurrence of a derives a subword of ω_k where at least one symbol belongs to $\Sigma \setminus h^{-1}(\lambda)$.

2) a derives a subword in ω_k (in k-i steps) which is included in $h^{-1}(\lambda)^*$.

Then $X \underset{P_1, A_0}{\Rightarrow} \omega'_0 \underset{P_{i_1}, A_1}{\Rightarrow} \omega'_1 \underset{P_{i_2}, A_2}{\Rightarrow} \ldots\ldots \underset{P_{i_k}, A_k}{\Rightarrow} \omega'_k \underset{P}{\Rightarrow} \omega' = h(\omega_k)$

which means that $h(L(G)) \subseteq L(H)$.

The other inclusion is shown in the same way.

Example:

Assume that $G = \langle \{a,b\}, \mathscr{P}, bab \rangle$ where $\mathscr{P}$ consists of the table

$$\left\{ \begin{array}{l} a \to ab \\ b \to a \end{array} \right\}$$

and that our weak coding is $h(a) = \lambda$, $h(b) = c$.

Then

$$H = \langle \{(a,\emptyset),(a,\{a\}),(a,\{b\}),(a,\{a,b\}),(b,\emptyset),(b,\{a\}),(b,\{b\}),(b,\{a,b\}), X, \#, c\}, \mathscr{P}, X, \{c\} \rangle$$

where

$$P: \left\{ \begin{array}{lcl} (a,\emptyset) & \to & \# \\ (a,\{a\}) & \to & \# \\ (a,\{b\}) & \to & \# \\ (a,\{a,b\}) & \to & \# \\ (b,\emptyset) & \to & c \\ (b,\{a\}) & \to & c \\ (b,\{b\}) & \to & \# \\ (b,\{a,b\}) & \to & \# \\ X & \to & X \\ \# & \to & \# \\ c & \to & c \end{array} \right\}$$

$$P_{1,\emptyset}: \left\{ \begin{array}{lcl} (a,\emptyset) & \to & (a,\emptyset)(b,\emptyset) \\ (a,\{a\}) & \to & (a,\{a,b\})(b,\{a,b\}) \\ (a,\{b\}) & \to & (a,\{a\})(b,\{a\}) \\ (a,\{a,b\}) & \to & (a,\{a,b\})(b,\{a,b\}) \\ (b,\emptyset) & \to & (a,\emptyset) \\ (b,\{a\}) & \to & (a,\{a,b\}) \\ (b,\{b\}) & \to & (b,\{a\}) \\ (b,\{a,b\}) & \to & (a,\{a,b\}) \\ X & \to & (b,\emptyset)(a,\emptyset)(b,\emptyset) \\ \# & \to & \# \\ c & \to & c \end{array} \right\}$$

$$P_{1,\{a\}}: \left\{ \begin{array}{lcl} (a,\emptyset) & \to & (b,\{a\}) \\ (a,\{a\}) & \to & (b,\{a,b\}) \\ (a,\{b\}) & \to & (b,\{a,b\}) \\ (a,\{a,b\}) & \to & (b,\{a,b\}) \\ (b,\emptyset) & \to & \# \\ (b,\{a\}) & \to & \# \\ (b,\{b\}) & \to & \# \\ (b,\{a,b\}) & \to & \# \\ X & \to & (b,\{a\})(b,\{a\}) \\ \# & \to & \# \\ c & \to & c \end{array} \right\}$$

$$P_{1,\{b\}}: \left\{ \begin{array}{lcl} (a,\emptyset) & \to & (a,\{b\}) \\ (a,\{a\}) & \to & (a,\{a,b\}) \\ (a,\{b\}) & \to & (a,\{a,b\}) \\ (a,\{a,b\}) & \to & (a,\{a,b\}) \\ (b,\emptyset) & \to & (a,\{b\}) \\ (b,\{a\}) & \to & (a,\{a,b\}) \\ (b,\{b\}) & \to & (a,\{a,b\}) \\ (b,\{a,b\}) & \to & (a,\{a,b\}) \\ X & \to & (a,\{b\}) \\ \# & \to & \# \\ c & \to & c \end{array} \right\}$$

$$P_{1,\{a,b\}}: \left\{ \begin{array}{l} (a,\emptyset) \rightarrow \# \\ (a,\{a\}) \rightarrow \# \\ (a,\{b\}) \rightarrow \# \\ (a,\{a,b\}) \rightarrow \# \\ (b,\emptyset) \rightarrow \# \\ (b,\{a\}) \rightarrow \# \\ (b,\{b\}) \rightarrow \# \\ (b,\{a,b\}) \rightarrow \# \\ X \rightarrow \# \\ \# \rightarrow \# \\ c \rightarrow c \end{array} \right\}$$

We have, thus, established the following diagram:

(D8) PDTOL $\subsetneq$ CPDTOL $\subseteq$ NPDTOL $\subseteq$ EPDTOL = WPDTOL = HPDTOL

It is an open problem whether or not the two inclusions in the middle are proper.

Since the F-extensions of the systems with tables are treated in exactly the same manner (in fact, F does not alter any table families except the pure ones), we only give the final diagrams:

(D9) DTFOL $\subsetneq$ CDTFOL = NDTFOL = EDTFOL = WDTFOL = HDTFOL,
(D10) PDTFOL $\subsetneq$ CPDTFOL $\subseteq$ NPDTFOL $\subseteq$ EPDTFOL = WPDTFOL = HPDTFOL.

The two open probelms in (D10) are equivalent to the corresponding problems in (D8). The results from section 5 concerning relations to the Chomsky hierarchy do not carry over to the table case, eg., the family NPDTOL contains all regular languages.

7. Mixed diagram for the table case. Open problems.

Rather than giving the full mixed diagram, we only give the diagram for pure families which together with the results of the previous section has all the information we are able to present:

$$\begin{tikzcd}
 & \mathrm{DTFOL} & \\
\mathrm{DTOL} \arrow[ur] \arrow[rr, dashed, no head] & & \mathrm{PDTFOL} \arrow[ul] \\
 & \mathrm{PDTOL} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

This diagram follows by the definitions and by the easily established facts that the language

$$L_1 = \{ bc(aba)^{2^n} \mid n \geqq 1 \}$$

belongs to the family DTOL (and, in fact, to the family DOL) but does not belong to the family PDTFOL, whereas the language

$$L_2 = \{ a^2, a^3 \}$$

belongs to the family PDTFOL but does not belong to the family DTOL.

Some open problems have already been mentioned above. The most significant open problem in Sections 4 and 5 is whether DOL $\subsetneq$ CPDOL or whether the two families are incomparable. A solution to this problem would also settle the two other open problems in the mixed diagram, namely, whether or not the inclusion CPDOL $\subseteq$ NPDOL is proper and whether or not EDOL $\subsetneq$ CPDOL. In fact, DOL $\subsetneq$ CPDOL iff EDOL $\subsetneq$ CPDOL iff CPDOL = NPDOL. A somewhat related result is that the language L_1 above is not a finite union of PDOL-languages (and not even a finite union of PDTOL-languages). However, L_1 is in CPDOL.

REFERENCES

[1] Culik, K. II and Opatrny, J.: Literal Homomorphisms of OL-languages. University of Waterloo, Research Report CS-73-25 (1973).

[2] Ehrenfeucht, A. and Rozenberg, G.: The equality of EOL languages and codings of OL-languages. To appear in Int. Journal of Computer Mathematics.

[3] Ehrenfeucht, A. and Rozenberg, G.: Nonterminals versus homomorphisms in defining languages for some classes of rewriting systems. Manuscript submitted for publication (1973).

[4] Herman, G. and Rozenberg, G.: Developmental Systems and Languages. North-Holland Publ. Co., to appear.

[5] Lindenmayer, A. and Rozenberg, G.: Developmental systems with locally catenative formulas. Acta Informatica 2 (1973) 214-248.

[6] Nielsen, M.: The decidability of some equivalence problems for DOL-systems. To appear in Information and Control.

[7] Paz, A. and Salomaa, A.: Integral sequential word functions and growth equivalence of Lindenmayer systems. Information and Control 23 (1973) 313-343.

[8] Salomaa, A.: Formal Languages, Academic Press (1973).

G. Rozenberg
Mathematics Department
University of Utrecht
The Netherlands

Mogens Nielsen,
Arto Salomaa, Sven Skyum
Computer Science Department
University of Aarhus
Denmark

NONTERMINALS, HOMOMORPHISMS AND CODINGS IN DIFFERENT VARIATIONS OF OL-SYSTEMS

by

Mogens Nielsen
G. Rozenberg
Arto Salomaa
Sven Skyum

II. NONDETERMINISTIC SYSTEMS

Summary

Continuing the work begun in Part I of this paper, we consider now variations of nondeterministic OL-systems. The present Part II of the paper contains a systematic classification of the effect of nonterminals, codings, weak codings, nonerasing homomorphisms and homomorphisms for all basic variations of nondeterministic OL-languages, including table languages.

8. Introduction and some lemmas

The outline of the present work corresponds to that of Part I, the difference being that we now investigate nondeterministic L-systems, i.e., the letter D is not present in the name of the system. Thus, the purpose of this Part II is a systematic study of the effect of the operators E, C, N, W, H on the families OL, POL, FOL, and PFOL, as well as the same families with tables. More specifically, we consider the families

OL	EOL	COL	NOL	WOL	HOL
POL	EPOL	CPOL	NPOL	WPOL	HPOL
FOL	EFOL	CFOL	NFOL	WFOL	HFOL
PFOL	EPFOL	CPFOL	NPFOL	WPFOL	HPFOL

as well as the same families with T added.

The reader is referred to Part I for basic definitions, motivation and background material. The definitional convention to the effect that whenever a language L belongs to one of our families, then also $L \cup \{\lambda\}$ belongs to the same family, is valid also in this Part II.

According to the remark made after Lemma 3.2, both Lemma 3.1 and Lemma 3.2 are valid also for nondeterministic families, i.e., families considered in this Part II. We now establish two additional lemmas.

Lemma 8.1

The families NY, EY, HY and WY are not altered if the letter F is added to or removed from the name Y.

Proof

If we are dealing with the families NY or EY, the lemma follows from the observation that we can choose a one-letter axiom in the Y-system if F is not present in the name Y. For the families HY and WY the statement follows by Lemma 3.2, (iii).

Lemma 8.2

The language

$$L = \{a^{2^n} \cup a^{3^n} \mid n \geq 0\}$$

belongs to the family CPOL but not to the family TFOL.

Proof

The first assertion is established by considering the POL-system with the axiom a_1 and productions

$$a_1 \to a_2a_2,\ a_1 \to a_3a_3a_3,\ a_2 \to a_2a_2,\ a_3 \to a_3a_3a_3$$

and the coding h defined by

$$h(a_1) = h(a_2) = h(a_3) = a\ .$$

To prove the second assertion, consider any TFOL-system G such that $a^2 \in L(G)$ and $a^3 \in L(G)$ and, furthermore, a production $a \to a^m$ with $m > 1$ appears in some of the tables. If m has a prime factor different from 2 (resp. different from 3), then a^2 (resp. a^3) yields directly according to G a word not in L. Consequently, $L \neq L(G)$, and the lemma follows.

9. Systems without tables

Theorem 9.1 $OL \subsetneq COL$.

Proof

Follows from Lemma 8.2.

Theorem 9.2 $COL = NOL$ and $WOL = HOL$.

Proof

Follows from Lemma 3.1.

Theorem 9.3 $COL = EOL = HOL$.

Proof

The proof can be found in [2].

Theorems 9.1-9.3 can be summarized in the following diagram:

(D1) $OL \subsetneq COL = NOL = EOL = WOL = HOL$.

Theorem 9.4 $POL \subsetneq CPOL$.

Proof

Follows from Lemma 8.2.

Theorem 9.5 $CPOL \subsetneq NPOL$.

Proof

The inclusion follows from definition and that it is proper is seen from the language

$$L = \{ a^n b^n c^n \cup d^{3^n} \mid n \geq 1 \}.$$

It is easy to see that L belongs to NPOL, but it does not belong to CPOL (see [1]).

Theorem 9.6 $\qquad$ NPOL = EPOL.

Proof

The inclusion EPOL $\subseteq$ NPOL can be established by using the method of [2] (the proof of EOL $\subseteq$ COL). The other inclusion is easily checked.

Theorem 9.7 $\qquad$ EPOL = WPOL = HPOL.

Proof

Follows from Lemma 3.2 and the fact that EPOL = EOL (see eg. [4]).

The following diagram is established from the theorems 9.4–9.7:

(D2) POL $\subsetneq$ CPOL $\subsetneq$ NPOL = EPOL = WPOL = HPOL.

Theorem 9.8 $\qquad$ 1) FOL $\subsetneq$ CFOL.
2) CFOL = NFOL.
3) NFOL = EFOL = WFOL = HFOL.

Proof

1) and 2) follow from Lemma 8.2 and Lemma 3.1 resp., 3) follows from Lemma 8.1 and theorems 9.2 and 9.3.

Theorem 9.9 $\qquad$ 1) PFOL $\subsetneq$ CPFOL.
2) CPFOL $\subseteq$ NPFOL = EPFOL = WPFOL = HPFOL.

Proof

1) follows from Lemma 8.2, 2) follows by definition, Lemma 8.1 and theorems 9.6 and 9.7.

Theorems 9.8 and 9.9 can be illustrated in the following diagrams:

(D3) FOL $\subsetneq$ CFOL = NFOL = EFOL = WFOL = HFOL.
(D4) PFOL $\subsetneq$ CPFOL $\subseteq$ NPFOL = EPFOL = WPFOL = HPFOL.

10. Mixed diagram and relations to the Chomsky hierarchy

Theorem 10.1
1) POL $\subsetneq$ OL.
2) OL $\subsetneq$ FOL.
3) POL $\subsetneq$ PFOL.
4) PFOL $\subsetneq$ FOL.
5) OL and PFOL are incomparable.

Proof

All inclusions in 1) to 4) are true by definition.

Define

$$L_1 = \{bc\} \cdot \{(aba)^{2^n} \mid n \geq 0\}$$

and

$$L_2 = \{a^{2^n} \cup a^{3 \cdot 2^n} \mid n \geq 0\}.$$

It is easy to see that $L_1 \in$ OL\PFOL and $L_2 \in$ PFOL\OL, which proves 5) and that the inclusions in 1) to 4) are proper.

Theorem 10.2
1) CPOL $\subsetneq$ CPFOL.
2) CPOL is incomparable with OL, PFOL and FOL.

Proof

1) The proof can be found in [1]. (The language L defined in the proof of theorem 9.5 belongs to CPFOL\CPOL.)

2) It follows from Lemma 8.2 that it is sufficient to show that OL and PFOL cannot be a proper subset of CPOL.
If OL $\subseteq$ CPOL then COL $\subseteq$ CPOL which is not true. PFOL cannot be a proper subset of CPOL because

$$\{a_1 a^n b_1 b^n c_1 c^n \mid n \geq 0\} \cup \{d^{3^n} \mid n \geq 1\} \in \text{PFOL} \setminus \text{CPOL}.$$

The following two theorems give some relations between the families considered in section 9 and the families of the Chomsky hierarchy.

Theorem 10.3

All the families considered in section 9 are properly included in the family of context-sensitive languages.

Proof

A proof of the theorem for the family OL can be found in [4] and because of the closure properties for the family of context-sensitive languages this holds true for NOL and thereby for all the other families considered as well.

Theorem 10.4

The family of context-free languages is properly included in the family CPFOL.

Proof

Let $G = <V, \Sigma, P, S>$ be a cf-grammar of a language not containing λ in Greibach-normal form (i.e., the productions are of the form $A \to a$ or $A \to aA_1 \ldots A_n$). Suppose there are no useless symbols in V.

For each $A \in V$ we choose

$$w_A \in \{w \in \Sigma^* \mid A \overset{*}{\Rightarrow} w, \ |w| \text{ minimal}\}.$$

w_A will, in the rest of the proof, be fixed for every letter $A \in V$.

Define $k : V \to N$ by $k(A) = |w_A|$, and furthermore

$$s(A) = \{xw_x \in \Sigma^{k(A)}V^* \mid A \underset{\text{left}}{\overset{*}{\Rightarrow}} xw_x, \ |x| = k(A)\} \text{ and}$$

$$m(A) = \{x \in \Sigma^{k(A)} \mid \exists \ w \in V^* : xw \in s(A)\}$$

Since the grammar was in Greibach normal form, $s(A)$ and $m(A)$ are finite sets of strings.

Let $n : V \to N$ be defined as $n(A) = \{\text{number of strings in } m(A)\}$.

We will use $m(A)$ as an ordered set.

Now we can construct a PFOL system H and a coding h such that $h(L(H)) = L(G)$:

$$H : \left\langle \Sigma \cup \bigcup_{\substack{A \in V \\ 1 \le i \le k(A) \\ 1 \le j \le n(A)}} A_i^j ,\ P',\ \{s_1^1 s_2^1 \ldots s_{k(S)}^1, s_1^2 s_2^2 \ldots s_{k(S)}^2, \ldots, s_1^{n(S)} s_2^{n(S)} \ldots s_{k(S)}^{n(S)}\} \right\rangle$$

P' is defined **as** follows:

1) For all $a \in \Sigma$, $a \to a$ is in P'.
2) For all $A \in V$, $1 \le j \le n(A)$, and $1 \le i \le k(A)-1$, $A_i^j \to a_i^j$ is in P', where a_i^j is the i'th terminal in the j'th string in $m(A)$.
3) For all $A \in V$ and $1 \le j \le n(A)$

$$A_{k(A)}^j \to a_{k(A)}^j B_{11}^{k_1} B_{12}^{k_1} \ldots B_{1k(B_1)}^{k_1} B_{21}^{k_2} B_{22}^{k_2} \ldots B_{2k(B_2)}^{k_2} \ldots$$

$$\ldots B_{q1}^{k_q} B_{q2}^{k_q} \ldots B_{qk(B_q)}^{k_q}$$

is in P' for all $B_1, B_2, \ldots, B_q$ and $1 \le k_i \le n(B_i)$ where $xB_1 B_2 \ldots B_q \in S(A)$ and x is the j'th string in $m(A)$.

The coding h is defined by $h(a) = a$ for all $a \in \Sigma$, and $h(A_1^j A_2^j \ldots A_{k(A)}^j) = w_A$, for all $A \in V$ and $1 \le j \le n(A)$.

We prove that $L(G) \subseteq h(L(H))$. The other inclusion is shown in the same way.

Let $w \in L(G)$.

There exists a derivation of w in G such that

$$S = A_1 \overset{*}{\underset{\text{left}}{\Rightarrow}} x_1' A_2 A_3 \ldots A_n$$

$$\overset{*}{\Rightarrow} x_1' x_2' B_{21} \ldots B_{2q_2} x_3' B_{31} \ldots B_{3q_3} \ldots x_n' B_{n1} \ldots B_{nq_n}$$

$$\overset{*}{\Rightarrow} x_1' x_2' x_{21}'' \ldots x_{2q_2} x_3' x_{31}'' \ldots x_{3q_3}'' \ldots x_n' x_{n1}'' \ldots x_{nq_n}'' = w$$

where $x_i' \in m(A_i)$ for all $1 \le i \le n(A)$ and $B_{ij} \overset{*}{\Rightarrow} x_{ij}''$ for $2 \le i \le n$ and $1 \le j \le q_i$.

It suffices then to show that there exists an axiom $S_1' S_2' \ldots S_{k(S)}'$ in H such that:

$$S_1' S_2' \ldots S_{k(S)}' \underset{H}{\Rightarrow}$$

$$x_1' A_{21}^{k_2} A_{22}^{k_2} \ldots A_{2k(A_2)}^{k_2} A_{31}^{k_3} A_{32}^{k_3} \ldots A_{3k(A_3)}^{k_3} \ldots A_{n1}^{k_n} A_{n2}^{k_n} \ldots A_{nk(A_n)}^{k_n}$$

and

$A_{j1}^{k_j} A_{j2}^{k_j} \ldots A_{jk(A_j)}^{k_j} \underset{H}{\Rightarrow} x_j' w_j$ for all $2 \le j \le n$ but that is exactly how H

is constructed.

Remark

Note that parallellism is used essentially in the proof.

Consider a somewhat related problem, namely whether or not it is true that every context-free or regular language can be obtained as a coding of sentential forms of a cf-grammar, where we allow ourselves to have a finite set of axioms instead of a single start symbol.

The answer is negative for cf-languages. E.g., the language $\{a^n bcd^n \mid n \ge 1\}$ cannot be obtained by a coding of sentential forms because the only way one can produce an equal number of a's and d's is from one non-terminal in the "middle" and this cannot be coded into bc.

The answer is positive for regular languages. (e.g., see the proof of RG $\subseteq$ CPFOL in [1]. In this proof the parallellism is completely avoided.) This later result is not true if you only allow one single axiom instead of a finite set of axioms, see [3]. In this case, there even exist finite sets, e.g. $\{ab, cd\}$, which are not codings of sentential-forms for any cf-grammar.

Theorem 10.5

PFOL is incomparable to the family of regular languages.

Proof

$$\{a^{2^n} \mid n \geq 0\} \in PFOL \setminus RG$$

$$(aba)^* cbc(aba)^* \in RG \setminus PFOL$$

Finally we will summarize the results from this and the two previous sections in the following mixed diagram. If two nodes labelled X and Y are connected by an edge (resp. oriented edge), the node X being below the node Y, then $X \subseteq Y$ (resp. $X \subsetneq Y$). If two nodes labelled X and Y are connected by a broken edge then X and Y are incomparable.

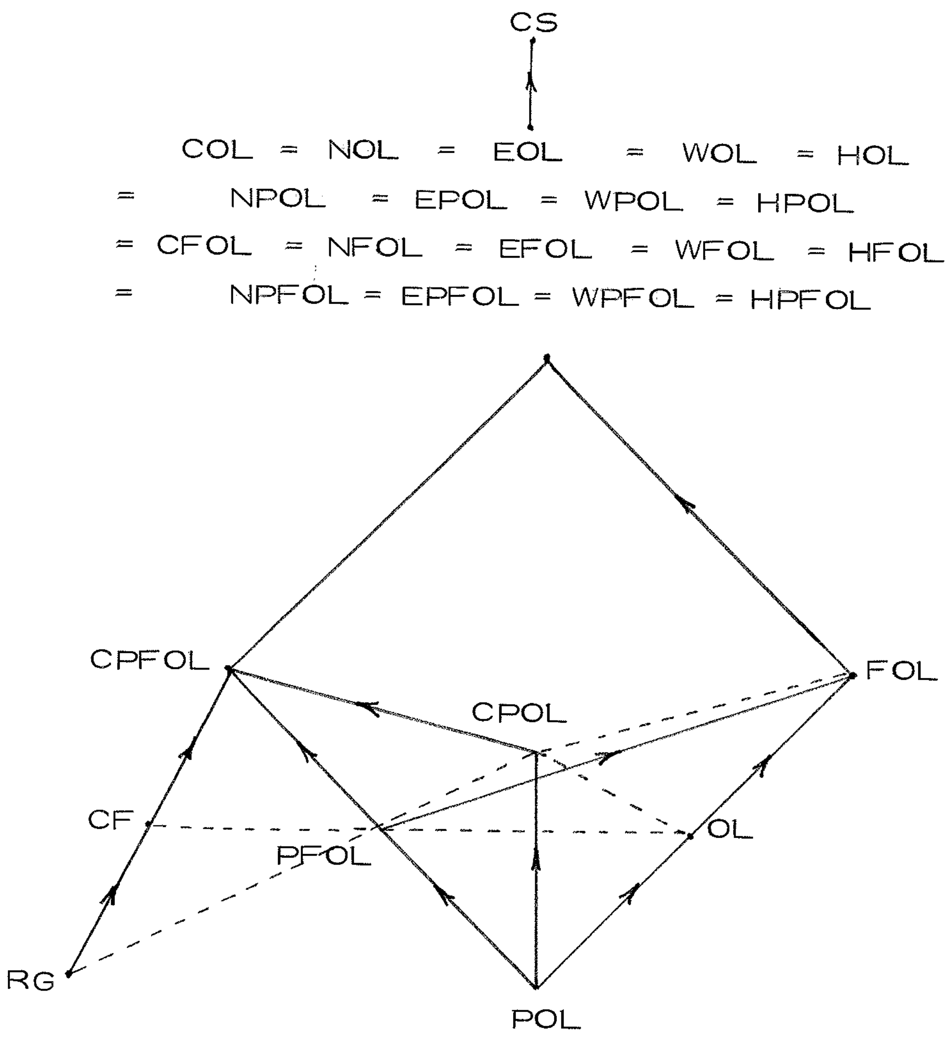

Figure 1.

The most significant open problems in Sections 9 and 10 are whether CPFOL = COL (= EOL) or CPFOL $\subsetneq$ COL, and whether or not the context-free languages (or even regular languages) are included in CPOL. It follows by theorem 10.4 that every context-free language which contains a one-letter word is in CPOL.

11. The table case

Theorem 11.1

1) TOL $\subsetneq$ CTOL.
2) CTOL = NTOL.
3) WTOL = HTOL.

Proof

1) follows from Lemma 8.2 and 2) and 3) from Lemma 3.1.

Theorem 11.2 CTOL = ETOL = HTOL.

Proof

The proof can be found in [3].

Theorems 11.1 and 11.2 give the following diagram:

(D5) TOL $\subsetneq$ CTOL = NTOL = ETOL = WTOL = HTOL.

Theorem 11.3 PTOL $\subsetneq$ CPTOL.

Proof

Follows from Lemma 8.2.

Theorem 11.4 NPTOL = EPTOL = HPTOL.

Proof

The statement can be established by using the method of [3].

Theorem 11.5 WPTOL = HPTOL.

Proof

Follows from Lemma 3.1.

Theorems 11.3 to 11.5 can be summarized in the following diagram:

(D6) PTOL $\subsetneq$ CPTOL $\subseteq$ NPTOL = EPTOL = WPTOL = HPTOL.

It is an open problem whether or not the inclusion CPTOL $\subseteq$ NPTOL is proper.

Since the F-extensions do not alter any table families except the pure ones, the following diagrams follow immediately from diagrams D5 and D6 and Lemma 8.2.

(D7) TFOL $\subsetneq$ CTFOL = NTFOL = ETFOL = WTFOL = HTFOL.

(D8) PTFOL $\subsetneq$ CPTFOL $\subseteq$ NPTFOL = EPTFOL = WPTFOL = HPTFOL.

12. Mixed diagram and relations to the Chomsky hierarchy for the table case

Theorem 12.1

1) PTOL $\subsetneq$ TOL.
2) TOL $\subsetneq$ TFOL.
3) PTOL $\subsetneq$ PTFOL.
4) PTFOL $\subsetneq$ TFOL.
5) TOL and PTFOL are incomparable.

Proof

The proof of theorem 10.1 is also valid in the table case, i.e., $L_1 \in$ TOL\PTFOL and $L_2 \in$ PTFOL\TOL.

It is an open problem whether TOL $\subsetneq$ CPTOL or whether TOL and CPTOL are incomparable.

Theorem 12.2

The family of context-free languages is properly contained in the family CPTOL.

Proof

We know from theorem 10.4 that CF $\subsetneq$ CPFOL and this together with CPFOL $\subseteq$ CPTFOL = CPTOL gives the result.

Theorem 12.3

All the families in the previous section are properly contained in the family of context-sensitive languages.

Proof

Analogous to the proof of theorem 10.3 (TOL $\subseteq$ CS).

Theorem 12.4

PTFOL is incomparable with the family of regular languages.

Proof

The proof for theorem 10.5 holds true for PTFOL as well as for PFOL.

We finally summarize the results in this and the previous section in the following mixed diagram for the table families.

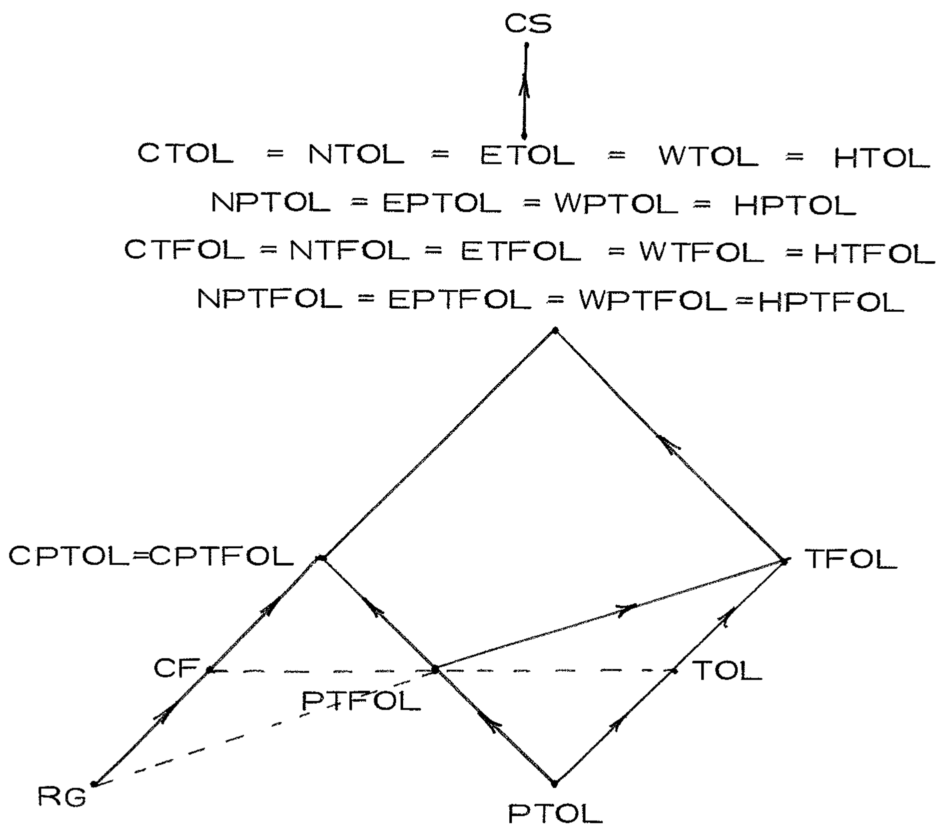

Figure 2.

The three open problems in the diagram are equivalent: CPTOL is properly contained in ETOL iff TFOL and CPTOL are incomparable iff TOL and CPTOL are incomparable.

13. Conclusion

An overview of the results obtained in both parts of this paper has already been given in the diagrams and in the mixed diagrams. A furhter discussion of the results is omitted – we only want to point out the rather surprising gact that the generative capacity caused by the operator E varies to a large extent depending on what types of pure systems we are dealing with. Apart from three open problems, we have completed the task set at the beginning: to characterize the role of the operators E, N, H, C and W. The open problems are whether or not the equations

$$CPXOL = CXOL, \text{ for } X = D, F, T,$$

hold true. Various equivalent formulations for these problems were also given. Although we have not shown that the problems are equivalent among themselves, it seems very likely that a break through in one of them would also solve the others. On the other hand, the techniques used so far in the study of L-systems seem not to be applicable. Eg., neither the method of [2] for establishing the equation COL = EOL nor the more recent method using recurrence systems (Gabor Herman, personal communication) for establishing the same equation seem not to be extendable for solving the problem whether or not CPFOL = CFOL (=COL).

REFERENCES

[1] Culik, K. II and Opatrny, J.: Literal Homomorphisms of OL-languages. University of Waterloo, Research Report, CS-73-25 (1973).

[2] Ehrenfeucht, A. and Rozenberg, G.: The equality of EOL-languages and codings of OL-languages. To appear in Int. Journal of Computer Mathematics.

[3] Ehrenfeucht, A. and Rozenberg, G.: Nonterminals versus homomorphisms in defining languages for some classes of rewriting systems. Manuscript submitted for publication (1973).

[4] Herman, G. and Rozenberg, G.: Developmental Systems and Languages. North-Holland Publ. Co., to appear.

G. Rozenberg
Mathematics Department
University of Utrecht
Utrecht
The Netherlands

Mogens Nielsen,
Arto Salomaa, Sven Skyum
Dept. of Computer Science
University of Aarhus
8000 Aarhus C
Denmark