# Macro OL-Systems

K. Čulík II and J. Opatrný

CS-73-06

Department of Applied Analysis &
Computer Science

University of Waterloo
Waterloo, Ontario, Canada

**Faculty of Mathematics**

**University of Waterloo**

**Waterloo, Ontario**

**Canada**

**Department of Applied Analysis**

**&**

**Computer Science**

# Macro OL-Systems

K. Čulík II and J. Opatrný

CS-73-06

Department of Applied Analysis &
Computer Science

University of Waterloo
Waterloo, Ontario, Canada

This work was supported by the National Research Council of Canada Grant No. A7403.

ABSTRACT

Macro OL-systems and two of their subclasses, called FMOL-systems and RMOL-systems are introduced. Macro OL-systems are motivated by theoretical models for the development of biological organism. Various properties of the families of languages generated by FMOL-systems and RMOL-systems are studied. It is shown that the family of languages generated by RMOL-systems forms the minimal full abstract family of languages containing the family of OL-languages.

1. L-systems, defined by Lindenmayer in [7], have been used for the modelling of the development of simple filamentous organisms. At discrete moments of time each cell of such an organism changes into zero, one or more new cells. In an L-system, each type of cell is represented by a symbol and the changes are described by a finite set of productions. OL-systems are L-systems which do not consider the possibility of interaction between cells, and so all changes are described by a finite set of context-free productions.

Properties of OL-languages, generated by OL-systems were studied in [4], [6], [8]-[13]. It has been shown that OL-languages are not closed under any of usual operations on languages as union, star, homomorphism and they even do not contain all finite sets. The family of OL-languages has been called an anti-AFL in [14].

We will consider the so called macro OL-systems, in particular two special cases, FMOL-system and RMOL-systems. These have quite natural biological motivation, and furthermore languages generated by them are closed under many operations and mappings on languages.

In macro OL-systems each type of cell is described as in OL-systems by a symbol called a terminal. In addition, there are non-terminal symbols which represent elements of the macro structure of an organism, namely groups of adjacent cells of the organism. The development of an organism is then described by an OL-system over nonterminals and by rules for transforming a macro structure representation into a cell-representation, i.e. replacement of nonterminals by terminal strings.

In an FMOL-system, a nonterminal symbol may be replaced by one of a finite number of terminal strings. In an RMOL-system a nonterminal may be replaced by a string from a regular language over the terminals. In macro OL-systems, only those developmental processes which lead from the starting nonterminal to a terminal string are valid.

In the next section, the definition of OL-systems is reviewed and some notation used throughout the paper is introduced.

We then continue with the definition of macro OL-systems and of their two versions, namely of FMOL-systems and RMOL-systems, together with the definitions of FMOL-languages and RMOL-languages. Furthermore, the relations between FMOL-languages and context-free and RMOL-languages are studied.

In section four the machine characterisations of FMOL-systems and RMOL-systems by FM-automata and RM-automata respectively are given.

In the last section the closure properties of the families of FMOL-languages, RMOL-languages are studied and it is shown that RMOL-languages form a full AFL (see [3]). Finally, it is proved that the family of RMOL-languages is the smallest full AFL containing OL-languages.

2. Preliminaries. We shall assume that the reader is familiar with the basic formal language theory from [5] and with the notion of AFL [3]. Now we will recall the definition of OL-systems and OL-languages. The definition given below is taken from [12] and only very slight modification in notation are made to be consistent with the notation used in the definitions of macro OL-systems in this paper.

Definition: An OL-system G is a 3-tuple $(\Sigma, P, \sigma)$ where

(a) $\Sigma$ is a finite nonempty set of symbols;

(b) P is the finite set of ordered pairs from $\Sigma \times \Sigma^*$ called the productions. A production $(a,\alpha)$ where $a \in \Sigma$, $\alpha \in \Sigma^*$ is usually written as $a \to \alpha$;

(c) $\sigma \in \Sigma^+$ is the initial string.

Any OL-system has to be complete which means that for every $a \in \Sigma$ there must exist a string $\alpha \in \Sigma^*$ such that $(a,\alpha) \in P$.

Let $\alpha = a_1 a_2 \ldots a_n$, $\beta \in \Sigma^*$. $\alpha$ is said to directly derive $\beta$ in an OL-system G, written $\alpha \underset{G}{\Rightarrow} \beta$ if there exist $\beta_1, \beta_2, \ldots, \beta_n \in \Sigma^*$ such that $\beta = \beta_1 \beta_2 \ldots \beta_n$ and $a_1 \to \beta_1$, $a_2 \to \beta_2, \ldots, a_n \to \beta_n$ are productions in P.

Let $\overset{*}{\underset{G}{\Rightarrow}}$ be the reflexive and transitive closure of the relation $\underset{G}{\Rightarrow}$ on $\Sigma^*$. Language L generated by an OL-system G is denoted L(G) and is defined to be the set $\{w \in \Sigma^* : \sigma \overset{*}{\underset{G}{\Rightarrow}} w\}$.

Notation: Throughout the paper if r is any binary relation then $\overset{*}{r}$ denotes the reflexive and transitive closure of r, without repeating it specifically in every case.

The family of $\lambda$ languages is denoted by $\mathcal{L}_\lambda$, e.g. $\mathcal{L}_{CF}$ is the family of CF languages and $\mathcal{L}_{OL}$ is the family of OL-languages.

Symbol $\varepsilon$ is used throughout the paper to denote empty word.

3. If we compare OL-systems and CF grammars we notice that in OL-systems:

(a) no difference is made between terminals and nonterminals;

(b) OL-systems require completeness;

(c) one step of a derivation consists of simultaneous replacement of all symbols of a string.

We will now define a generalisation of OL-systems, called macro-OL-systems, which still requires the simultaneous replacement of all symbols of a string in each step of a derivation, but which uses two kinds of symbols the nonterminals and the terminals in a similar way like context-free grammars. There are only productions for replacement of nonterminals and only terminal strings are in the language generated by the system.

A macro OL-system works like an OL-system except that the produced strings are only intermediate strings of nonterminals. For this purpose, it uses productions with nonterminal strings on the right-hand sides. There is another type of productions where the right-hand side denotes a language over terminals. A terminal string in the language generated by the system may be obtained by replacing each nonterminal, say A, in an intermediate nonterminal string by a terminal string from a language denoted by d where $A \to d$ is a production.

Definition: A macro OL-system G is a 4-tuple (N,T,P,S) where

(a) N is a finite, nonempty set called the nonterminals;

(b) T is a finite, nonempty set called the terminals, $T \cap N = \phi$;

(c) P is a finite set of ordered pairs of the form $(A,\alpha)$ called the productions with $A \in N$, where either $\alpha \in N^*$ or $\alpha$ is an effective description of a language over T. A production $(A,\alpha)$ in P is usually written as $A \to \alpha$;

(d) $S \in N$ is the starting symbol.

We say that $\alpha$ directly derives $\beta$, and write $\alpha \underset{G}{\Rightarrow} \beta$, if there exist $n \geq 1$, $A_1, A_2, \ldots, A_n$ in $N$ and $\beta_1, \beta_2, \ldots, \beta_n$ in $N^* \cup T^*$ so that $\alpha = A_1 A_2 \ldots A_n$, $\beta = \beta_1 \beta_2 \ldots \beta_n$ and for each $k = 1, \ldots, n$ either $A_i \to \beta_i$ is in $P$ or $A_i \to d$ is in $P$ where $d$ denotes a language over $T$ containing $\beta_i$.

We say that $\alpha$ derives $\beta$ in a macro OL-system $G$ if $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$. The language generated by a macro OL-system $G = (N,T,P,S)$, denoted $L(G)$ is called the macro OL-language and is defined to be the set $\{\alpha \in T^*: S \overset{*}{\underset{G}{\Rightarrow}} \alpha\}$.

Note that in a macro OL-system there is not necessarily a terminal replacement for every nonterminal.

By restricting the type of languages over terminals which may be used in productions we obtain a particular class of macro OL-systems. We will consider two such classes. First, we consider finite languages and get finite macro OL-systems (FMOL-systems). Clearly, we can write one production for every string in a finite language and therefore, without restriction of generality, we may only allow productions of the form $A \to \alpha$ with $\alpha \in T^*$ in an FMOL-system ($\alpha$ denotes the language $\{\alpha\}$).

Definition: FMOL-system is a macro OL-system

$$G = (N,T,P,S) \text{ with } P \subseteq N \times (N^* \cup T^*).$$

The language generated by an FMOL-system is called finite macro OL-language, abbreviated FMOL-language.

Note that an FMOL-system is a context-free grammar with a modified interpretation. A string is generated by an FMOL-system only if in its context-free derivation tree all the paths from the root to any node labelled by a terminal are of the same length.

<u>Theorem 1</u>: $\mathcal{L}_{CF} \subsetneq \mathcal{L}_{FMOL}$ .

<u>Proof</u>: Let L be a CF-language generated by the CF-grammar $G = (N,T,P,S)$. We may assume that the productions of P are of the form $A \to \alpha$, where $\alpha \in N^+ \cup T^*$. Construct the FMOL-system $G_1 = (N,T,P_1,S)$ where $P_1 = P \cup \{A \to A : A \in N\}$. Then clearly $L(G_1) = L(G)$. Thus $\mathcal{L}_{CF} \subseteq \mathcal{L}_{FMOL}$.

The proper inclusion of CF-languages in FMOL-languages follows from the fact that the language $L = \{a^n b^n c^n | n \geq 1\}$ is generated by the FMOL-system $G = (N,\{a,b,c\},P,S)$ where $N = \{S,A_1,A,B_1,B,C_1,C\}$ and $P = \{S \to ABC,\ A \to A_1A,\ B \to B_1B,\ C \to C_1C,\ A_1 \to A_1,\ B_1 \to B_1,\ C_1 \to C_1,$
$A_1 \to a,\ A \to a,\ B_1 \to b,\ B \to b,\ C_1 \to c,\ C \to c\}$ .

Now we will define the second class of macro OL-systems, called <u>regular macro OL-systems</u> (RMOL-systems). In an RMOL-system a nonterminal may be replaced by a string from a regular language over the terminals given by a regular expression.

<u>Definition</u>: Regular macro OL-system, abbreviated RMOL-system, is a macro OL-system $(N,T,P,S)$ with $P \subset N \times (N^* \cup E_T)$ where $E_T$ is the set of regular expressions over T. The language generated by an RMOL-system is called <u>regular macro OL-language</u>, (RMOL-language).

<u>Theorem 2</u>: $\mathcal{L}_{FMOL} \subsetneq \mathcal{L}_{RMOL}$.

<u>Proof</u>: Inclusion of $\mathcal{L}_{FMOL}$ in $\mathcal{L}_{RMOL}$ is trivial. From [4] follows that the language $L = \{w \in \{a,b\}^* :$ number of a's in w is $2^n\}$ cannot be generated by an FMOL-system but this language is generated by the RMOL-system $G = (\{A,B\},\{a,b\},P,A)$ where $P = \{A \to BABAB,\ B \to B,\ B \to b^*,\ A \to a\}$.

4. Now we give a machine characterisation of FMOL and RMOL-languages.

An RM-automaton is basically a push-down automaton which writes on its push-down tape pairs consisting of a symbol of the push-down tape alphabet and a natural number which indicates the "level" of the symbol. There are two basic kinds of moves of an RN-automaton:

(a) "shift" - an input symbol, or $\varepsilon$, is read and a push-down tape symbol with the level number 1, or $\varepsilon$, is pushed on the top of the push-down tape;

(b) "reduction" - a nonempty string of symbols at the top of the push-down tape, all of them having the same level number, is replaced by a single symbol with the level number increased by 1.

<u>Definition</u>: An RM-automaton M is a 7-tuple $(K,\Sigma,\Gamma,\delta_1,\delta_2,q_0,q_f)$ where

(a) K is a finite, nonempty set of <u>states</u>;

(b) $\Sigma$ is a finite, nonempty set of symbols called the <u>input alphabet</u>;

(c) $\Gamma$ is a finite, nonempty set of symbols called the <u>push-down tape alphabet</u>;

(d) $\delta_1$ is the <u>shift-function</u> $\delta_1: K \times (\Sigma \cup \varepsilon) \to$ finite subsets of $K \times (\Gamma \cup \varepsilon)$;

(e) $\delta_2$ is the <u>reduction function</u> $\delta_2: K \times \Gamma^+ \to$ finite subsets of $K \times \Gamma$;

(f) $q_0$ is the <u>initial state</u>;

(g) $q_f$ is the <u>final state.</u>

Note If RM automata were interpreted in the standard way using the acceptance with final state, they would be equivalent to push-down automata. However, we will interpret them in a different way.

A configuration of an RM-automaton M is a 3-tuple $(q,\alpha,\gamma)$ where $q \in K$, $\alpha \in \Sigma^*$ and $\gamma \in (\Gamma \times I)^*$. In this section I denotes the set of natural numbers. The intuitive meaning of a configuration is the following:

- $q$ is the current state;
- $\alpha$ is the unread portion of the input;
- $\gamma$ is the current content of push-down tape with the top on the right.

For given RM-automaton M define a relation $\vdash$ as follows.

$(q_i,\alpha,\gamma_1) \underset{M}{\vdash} (q_j,\beta,\gamma_2)$ if one of the following two conditions holds

(a) - shift move, $(q_j,B) \in \delta_1(q_i,a)$, $\alpha = a\beta$, $a \in \Sigma \cup \{\varepsilon\}$ and $\gamma_2 = \gamma_1 c$ where $c = (B,1)$ if $B \in \Gamma$ otherwise $c = \varepsilon$, i.e. either nothing or one symbol a is read from the input tape and if $B \neq \varepsilon$ then B is pushed down with the level number 1;

(b) - reduction move, $(q_j,A) \in \delta_2(q_i,B_1B_2\ldots B_s)$ where $A,B_1,B_2,\ldots,B_s \in \Gamma$ $\alpha = \beta$, and there exist $\gamma_3 \in (\Gamma \times I)^*$ and $n \in I$ such that $\gamma_1 = \gamma_3(B_1,n)(B_2,n)\ldots(B_s,n)$, $\gamma_2 = \gamma_3(A,n+1)$, i.e. a string of symbols all of them with the same level number from the top of the push-down tape is reduced to a single push-down symbol with the level number increased by 1.

The language accepted by an RM-automaton M is denoted by L(M) and is defined as the set

$$\{\alpha : (q_0, \alpha, \varepsilon) \mid\frac{*}{M} (q_f, \varepsilon, A) \text{ for any } A \in \Gamma\}.$$

Note that the automaton M accepts only if the length of push-down tape is 1.

Definition: An RM-automaton $(K, \Sigma, \Gamma, \delta_1, \delta_2, q_0, q_f)$ with the shift function $\delta_1$ restricted to $K \times (\Sigma \cup \{\varepsilon\}) \rightarrow$ finite subsets of $K \times \Gamma$ is called an FM-automaton, i.e. FM-automaton has to push one symbol on tape in every shift-move.

Now we will show the equivalence of FM-automata and FMOL-systems.

Theorem 3: The family of languages accepted by FM-automata is $\mathscr{L}_{FMOL}$.

Proof: In this proof the following notation will be used:

If r is a binary relation defined on a set U, then $r^m$, where $m \geq 0$ is the binary relation on U defined by

(a) $\alpha \; r^o \; \alpha$ for all $\alpha$ in U;

(b) $\alpha \; r^m \; \beta$, where $\alpha, \beta \in U$ and $m > 0$ if and only if there exists $\gamma \in U$ such that $\alpha \; r^{m-1} \; \gamma \; r \; \beta$.

Let L be an FMOL-language given by an FMOL-system $G = (N,T,P,S)$. We will first construct an equivalent FMOL-system $G_1$ of a special form. Let $T' = \{a' : a \in T\}$. Let X be a symbol not in $N \cup T \cup T'$. Define homomorphism h, by $h(A) = A$ if $A \in N$ and $h(a) = a'$ if $a \in T$. Construct FMOL-system $G_1 = (N_1, T, P_1, S)$ where $N_1 = N \cup T' \cup \{X\}$ and $P_1 = h(P) \cup \{a' \rightarrow a : a \in T\} \cup \{A \rightarrow X : A \rightarrow \varepsilon \in P\} \cup \{X \rightarrow X, X \rightarrow \varepsilon\}$. Clearly, if a string can be generated using $P_1$, it can also be generated using P only and therefore $L(G_1) = L(G)$. The FMOL-system $G_1$ has the property that if $\alpha \in L(G_1)$ then there exists $\beta \in (T' \cup X)^+$ such that $S \overset{*}{\underset{G_1}{\Rightarrow}} \beta \underset{G_1}{\Rightarrow} \alpha$ and no production of the form $A \to \varepsilon$ has been used to derive $\beta$. Now we will construct an FM-automaton which simulates exactly all such derivations. Construct the FM-automaton $M = (K,T,\Gamma,\delta_1,\delta_2,q_0,q_f)$ where $K = \{q_0,q_f\}$ $\Gamma = N_1$, and $\delta_1$ and $\delta_2$ are defined as follows:

(a) if $A \to \varepsilon \in P_1$ then $(q_0,A) \in \delta_1(q_0,\varepsilon)$;

(b) if $A \to a \in P_1$ where $A \in N_1$, $a \in T$ then $(q_0,A) \in \delta_1(q_0,a)$;

(c) if $A \to B_1B_2\ldots B_n \in P_1$ where $A,B_1,B_2,\ldots,B_n \in N_1$ then $(q_0,A) \in \delta_2(q_0,B_1B_2\ldots B_n)$;

(d) $(q_f,S) \in \delta_2(q_0,S)$.

Clearly, $L(G_1) \subseteq L(M)$ since M simulates all derivations mentioned above. The inclusion $L(M) \subseteq L(G_1)$ will be shown if we prove that $(q_0,a_1a_2\ldots a_n,\varepsilon) \mid\frac{*}{M} (q_0,a_{i+1}a_{i+2}\ldots a_n,\gamma_1\gamma_2\ldots\gamma_m)$ where $i \geq 0$, $m \geq 1$, $\gamma_j = (X_1^j,n_j)(X_2^j,n_j)\ldots(X_{k_j}^j,n_j)$ for some $n_j \in I$, $X_p^j \in \Gamma$, $P = 1,2,\ldots,k_j$ and $j = 1,2,\ldots,m$, implies there exist $\alpha_1,\alpha_2,\ldots,\alpha_m$ such that $X_1^jX_2^j\ldots X_{k_j}^j \overset{*}{\underset{G_1}{\Rightarrow}} \alpha_j$ for $j = 1,2,\ldots,m$ and $\alpha_1\alpha_2\ldots\alpha_m = a_1a_2\ldots a_i$. We will prove it by the induction on the number k of moves of automaton M.

Let $k = 1$, i.e. $(q_0,a_1a_2\ldots a_n,\varepsilon) \mid\frac{}{M} (q_0,a_{i+1}a_{i+2}\ldots a_n,\gamma_1\gamma_2\ldots\gamma_m)$. From the definition of an FM automaton follows that $i \leq 1$, $m = 1$, $\gamma = (X_1^1,1)$ and if $i = 1$ then $X_1^1 \to a_1 \in P_1$ else if $i = 0$ then $X_1^1 \to \varepsilon \in P_1$.

Suppose now that the induction hypothesis is true for up to k moves, $k \geq 1$. Let $(q_0,a_1a_2\ldots a_n,\varepsilon) \mid\frac{k+1}{M} (q_0,a_{i+1}a_{i+2}\ldots a_n,\gamma_1\gamma_2\ldots\gamma_m)$. Then there exist $v,s \in I$ and $\lambda_1,\lambda_2,\ldots,\lambda_s$ such that

$(q_0, a_1a_2\ldots a_n, \varepsilon) \overset{k}{\underset{M}{\vdash}} (q_0, a_{v+1}a_{v+2}\ldots a_n, \lambda_1\lambda_2\ldots\lambda_s) \underset{M}{\vdash} (q_0, a_{i+1}a_{i+2}\ldots a_n, \gamma_1\gamma_2\ldots\gamma_m)$

where $v \geq i$, $\lambda_j = (Y_1^j, r_j)(Y_2^j, r_j)\ldots(Y_{t_j}^j, r_j)$, $Y_p^j \in \Gamma$ for $j = 1,2,\ldots,s$, $p = 1,2,\ldots,t_j$, $r_j \in I$. Because the induction hypothesis is true for up to k moves, there exist $\beta_1, \beta_2, \ldots, \beta_s$ such that $Y_1^j\, Y_2^j \ldots Y_{t_j}^j \overset{*}{\underset{G_1}{\Rightarrow}} \beta_j$ for $j = 1,2,\ldots,s$ and $\beta_1\beta_2\ldots\beta_s = a_1a_2\ldots a_v$. From the construction of automaton M follows that either the last move of the automaton is a reduction and so $i = v$, $\lambda_1 = \gamma_1, \ldots, \lambda_{m-1} = \gamma_{m-1}$, $k_m = 1$, $r_m = r_{m+1} = \ldots = r_s$ and $(q_0, X_1^m) \in \delta_2(q_0, Y_1^mY_2^m\ldots Y_{t_m}^m\ldots Y_1^sY_2^s\ldots Y_{t_s}^s)$ and $X_1^m \to Y_1^mY_2^m\ldots Y_{t_m}^m\ldots Y_1^sY_2^s\ldots Y_{t_s}^s$ is a production in $G_1$. Then $X_1^m \overset{*}{\underset{G_1}{\Rightarrow}} \beta_m\beta_{m+1}\ldots\beta_s$ and the induction hypothesis holds for k+1 moves, or if the last move of the automaton M is a shift then $i-1 \leq v \leq i$, $m = s+1$, $\lambda_1 = \gamma_1, \ldots, \lambda_s = \gamma_s, \gamma_m = (X_1^m, 1)$ and $(q_0, X_1^m) \in \delta_1(q_0, b)$ where if $i > v$ then $b = a_i$ else $b = \varepsilon$, and $X_1^m \to b$ is a production in $G_1$. In this case the induction hypotehsis is satisfied with $\alpha_1 = \beta_1$, $\alpha_2 = \beta_2, \ldots, \alpha_{m-1} = \beta_{m-1}$ and $\alpha_m = b$.

Let M be an FM-automaton, $M = (K, \Sigma, \Gamma, \delta_1, \delta_2, q_0, q_f)$. Construct the FMOL-system $G = (N, \Sigma, P, S)$ where $N = \{(q_i, A, q_j) : q_i, q_j \in K, A \in \Gamma\} \cup \{S\}$ and P is constructed as follows:

(i) $S \to (q_0, A, q_f) \in P$ for all $A \in \Gamma$ for which either $(q_f, A) \in \delta_2(q_j, \gamma)$, or $(q_f, A) \in \delta_1(q_j, a)$ where $q_j \in K$, $\gamma \in \Gamma^+$ and $a \in \Sigma \cup \{\varepsilon\}$;

(ii) if $(q_j, A) \in \delta_1(q_i, a)$ where $a \in \Sigma \cup \{\varepsilon\}$ then $(q_i, A, q_j) \to a \in P$;

(iii) if $(q_j,A) \in \delta_2(q_i,B_1B_2...B_n)$ then $(q_s,A,q_j) \to (q_s,B_1,q_{i_1})(q_{i_1},B_2,q_{i_2})...(q_{i_{n-1}},B_n,q_i) \in P$ for all $q_s,q_{i_1},...,q_{i_{n-1}} \in K$.

We will show the equivalence of automaton M and of system G by proving that for any $a_1,a_2,...,a_n \in \Sigma$, $n \geq 1$ the statements (1) and (2) are equivalent.

(1) $(q_j,a_1a_2...a_n,\varepsilon)\,|\frac{*}{M}(q_i,\varepsilon,(A_1,m_1)(A_2,m_2)...(A_s,m_s)$ where $A_v \in \Gamma$, $m_v \in I$ for $v = 1,2,...,s$.

(2) There exist $q_{r_1},q_{r_2},...,q_{r_{s-1}} \in K$ and $\alpha_1,\alpha_2,...,\alpha_s \in \Sigma^*$ such that $(q_j,A_1,q_{r_1}) \overset{m_1}{\underset{G}{\Rightarrow}} \alpha_1$, $(q_{r_1},A_2,q_{r_2}) \overset{m_2}{\underset{G}{\Rightarrow}} \alpha_2,...,(q_{r_{s-1}},A_s,q_i) \overset{m_s}{\underset{G}{\Rightarrow}} \alpha_s$ and $\alpha_1\alpha_2...\alpha_s = a_1a_2...a_n$.

(a) First we prove by the induction on the number of steps in the derivation that (1) follows from (2).

(a1) Suppose that all derivations are direct. Then from the definition of the FMOL-system follows that $\alpha_i \in \Sigma \cup \{\varepsilon\}$ for all $i \in 1,2,...,s$ and $(q_{r_1},A_1) \in \delta_1(q_j,\alpha_1),(q_{r_2},A_2) \in \delta_1(q_{r_2},\alpha_2),...,(q_i,A_s) \in \delta_1(q_{r_{s-1}},\alpha_s)$. So $(q_j,a_1a_2...a_n,\varepsilon)\,|\frac{*}{M}\,(q_i,\varepsilon,(A_1,1)(A_2,1)...(A_s,1))$.

(a2) Suppose that the induction hypothesis is true for up to k steps in any derivation. We can write $(q_j,A_1,q_{r_1}) \underset{G}{\Rightarrow} (q_j,B_1,q_1^u)(q_1^u,B_2,q_2^u)...(q_{v_1}^u,B_{v_1},q_{v_1}) \overset{k}{\underset{G}{\Rightarrow}} \alpha_1$ where $B_1,B_2,...,B_{v_1} \in \Gamma$, $q_1^u,...,q_{v_1}^u \in K$ and using the induction hypothesis we have $(q_j,\alpha_1,\varepsilon)\,|\frac{*}{M}(q_{v_1},\varepsilon,(B_1,k)(B_2,k)...(B_{v_1},k)$.

From the definition of the system G follows that

$(q_{v_1},\varepsilon,(B_1,k)(B_2,k)\ldots(B_{v_1},k)) \vdash_{M} (q_{r_1},\varepsilon,(A_1,k+1))$ and therefore

$(q_j,\alpha_1,\varepsilon) \vdash^{*}_{M} (q_{r_1},\varepsilon,(A_1,k+1))$. Similarly, we can obtain that

$(q_{r_1},\alpha_2,\varepsilon) \vdash^{*}_{M} (q_{r_2},\varepsilon,(A_2,m_2)),\ldots,(q_{r_{s-1}},A_s,\varepsilon) \vdash^{*}_{M} (q_i,\varepsilon,(A_s,m_s))$. Clearly, the induction hypothesis holds for k+1 steps.

(b) We will show that (2) follows from (1) by the induction on the number of moves of the automaton.

(b1) Suppose that $(q_j,a_1a_2\ldots a_n,\varepsilon) \vdash_{M} (q_i,\varepsilon,(A_1,m_1)(A_2,m_2)\ldots(A_s,m_s))$. From the definition of an FM automaton follows that $n \leq 1$, $s = 1$, and from the construction of the system G follows that $(q_j,A_1,q_i) \to b \in P$ where $b = a_1$, if $n = 1$, otherwise $b = \varepsilon$.

(b2) Suppose that the induction hypothesis holds for up to k moves, $k \geq 1$. Let $(q_j,a_1a_2\ldots a_n,\varepsilon) \vdash^{k+1}_{M} (q_i,\varepsilon,(A_1,m_1)(A_2,m_2)\ldots(A_s,m_s)$. Considering three possible kinds of moves of the automaton M in k+1st step, using the validity of the induction hypothesis for up to k moves, and from the construction of system G follows that the induction hypothesis is true for k+1 moves.

The next theorem shows the equivalence of RMOL-systems and RM-automata.

Theorem 4: The family of languages accepted by RM-automata is $\mathcal{L}_{RMOL}$.

Proof: (a) Let L be an RMOL-language generated by a system G = (N,T,P,S). We suppose that G is modified similarly as in the proof of Theorem 3, i.e. if $A \to \varepsilon \in P$ then we add to P productions $A \to X$, $X \to X$, $X \to \varepsilon$ where X is a new nonterminal. Let $R_i$, $i = 1,2,\ldots,n$ be all regular expressions used in productions in P and let $M_i = (K_i,\Sigma,f_i,s_i,F_i)$ be a finite automaton (see [5], p.26) accepting the language $L(R_i)$. We may suppose that $\bigcap_{i=1}^{n} K_i = \phi$. Let $q_0,q_f$ be not in $\bigcup_{i=1}^{n} K_i$. Construct the RM-automaton $M = (K,\Sigma,\Gamma,\delta_1,\delta_2,q_0,q_f)$ where $K = (\bigcup_{i=1}^{n} K_i) \cup \{q_0,q_f\}$, $\Gamma = N$, and $\delta_1$ and $\delta_2$ are defined as follows.

(i) If $A \to R_i \in P$ then $(s_i,\varepsilon) \in \delta_1(q_0,\varepsilon)$ and $(q_0,A) \in \delta_1(q,\varepsilon)$ for each $q \in F_i$;

(ii) $\delta_1(q,a) = f_i(q,a) \times \{\varepsilon\}$ for every $a \in \Sigma$ and $q \in K_i$, $i = 1,2,\ldots,n$;

(iii) If $A \to B_1B_2\ldots B_m \in P$ where $A,B_1,B_2,\ldots,B_m \in N$ then $(q_0,A) \in \delta_2(q_0,B_1B_2\ldots B_n)$;

(iv) $(q_f,S) \in \delta_2(q_0,S)$.

From the construction of M follows that $L(M) = L$.

(b) Let M be an RM-automaton, $M = (K,\Sigma,\Gamma,\delta_1,\delta_2,q_0,q_f)$. Define $R_{ij} = \{\alpha : (q_i,\alpha,\varepsilon) \overset{*}{\underset{M}{\vdash}} (q_j,\varepsilon,\varepsilon)\}$. Clearly, $R_{ij}$ is a regular language.

Let $R_{ij}$ be denoted by a regular expression $E_{ij}$. Construct the RNOL-system $G = (N,\Sigma,P,S)$ where $N = \{(q_i,A,q_j) : q_i,\ q_j \in K,\ A \in \Gamma\} \cup \{S\}$ and P contains the following productions.

(i) $S \to (q_0,A,q_f) \in P$ for all $A \in \Gamma$ for which $(q_f,A) \in \delta_2(q_\gamma,\gamma)$ or $(q_f,A) \in \delta_1(q_j,a)$ where $q_j \in K$, $\gamma \in (\Gamma \times I)^+$, $a \in \Sigma \cup \{\varepsilon\}$;

(ii) if $(q_j,A) \in \delta_1(q_k,a)$ where $a \in \Sigma \cup \{\varepsilon\}$ then $(q_i,A,q_j) \to E_{ik} \cdot a \in P$;

(iii) if $(q_j,A) \in \delta_2(q_i,B_1B_2\ldots B_n)$ then $(q_s,A,q_j) \to (q_s,B_1,q_{i_1})(q_{i_1},B_2,q_{i_2}) \ldots (q_{i_{n-1}},B_n,q_i) \in P$ for all $q_s,q_{i_1},\ldots,q_{i_{n-1}} \in K$.

We can prove by the induction the equivalence of the automaton M and of the system G similarly as we did in the proof of Theorem 3.

5. In this section the closure properties of $\mathcal{L}_{FMOL}$ and $\mathcal{L}_{RMOL}$ are studied. The main result is that $\mathcal{L}_{RMOL}$ forms a full-AFL, which is the smallest full-AFL containing OL-languages.

Lemma 1. $\mathcal{L}_{FMOL}$ is closed under the operations of union, concatenation, star and homomorphism.

Proof: Consider two FMOL systems $G_1 = (N_1,T_1,P_1,S_1)$, $G_2 = (N_2,T_2,P_2,S_2)$. We can suppose without lost of generality that $N_1 \cap N_2 = \phi$. Let $S,S',S'' \notin N_1 \cup N_2$.

(a) Let $G_3 = (N_1 \cup N_2 \cup \{S\}, T_1 \cup T_2,P_3,S)$ where $P_3 = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}$. Clearly, FMOL-system $G_3$ generates $L(G_1) \cup L(G_2)$.

(b) Let $G_4$ be FMOL-system $(N_1 \cup N_2 \cup \{S,S',S''\}, T_1 \cup T_2,P_4,S)$ where $P_4 = P_1 \cup P_2 \cup \{S \to S'S'',S' \to S',S'' \to S'',S' \to S_1,S'' \to S_2\}$. Then $L(G_4) = L(G_1)\cdot L(G_2)$.

(c) Let $G_5$ be FMOL-system $(N_1 \cup \{S\},T_1,P_5,S)$ where $P_5 = P_1 \cup \{S \to S_1,S \to SS,S \to S,S \to \varepsilon\}$. Clearly, $L(G_5) = L(G_1)^*$.

(d) Let h be a homomorphism $h:T_1 \to \Gamma^*$. Define homomorphism f on $T_1 \cup N_1$ by $f(a) = h(a)$ for all $a \in T_1$ and $f(A) = A$ for all $A \in N_1$. Let $G_6 = (N_1,\Gamma,P_6,S_1)$ be an FMOL-system where $P_6 = \{A \to f(\alpha):A \to \alpha \in P_1\}$. Then $L(G_6) = h(L(G_1))$.

Lemma 2. $\mathcal{L}_{FMOL}$ is closed under intersection with a regular set.

Proof: Consider $L_1$ an FMOL-language and let R be a regular language. Let $M_1 = (K_1,\Sigma,\Gamma,\delta_1,\delta_2,q_0,q_f)$ be an FM-automaton accepting $L_1$ and let $A = (K_2,\Sigma,g,p_0,F)$ be a finite automaton accepting R. We can suppose that $g(p,\varepsilon) = \{\phi\}$ for all $p \in K_2$. Construct the FM-automaton $M_2 = (K_1 \times K_2 \cup \{q_{f_2}\},\Sigma,\Gamma,\delta_1^2,\delta_2^2,(q_0,p_0),q_{f_2})$, where $\delta_1$ and $\delta_2$ are defined as follows. If $(q_i,Z) \in \delta_2(q_j,B_1B_2\ldots B_n)$, where $Z,B_1,B_2,\ldots,B_n \in \Gamma$ then $((q_i,p_k),Z) \in \delta_2^2((q_j,p_k),B_1B_2\ldots B_n)$ for all $p_k \in K_2$. If $(q_i,Z) \in \delta_1(q_j,\varepsilon)$ then $((q_i,p_k),Z) \in \delta_1^2((q_j,p_k),\varepsilon)$. If $(q_i,Z) \in \delta_1(q_j,a)$ for $a \in T$ and $p_s \in g(p_r,a)$ then $((q_i,p_s),Z) \in \delta_1^2((q_j,p_r),a)$. It is easy to see that if $(q_0,a_1a_2\ldots a_n,\varepsilon) \mid\frac{*}{M_1} (q_j,\varepsilon,\gamma)$ and $(p_o,a_1a_2\ldots a_n) \mid\frac{*}{A} (p_r,\varepsilon)$ then $((q_0,p_0),a_1a_2\ldots a_n,\varepsilon) \mid\frac{*}{M_2} ((q_j,p_r),\varepsilon,\gamma)$. So $L(G_2) = L_1 \cap R$.

Theorem 5: $\mathcal{L}_{FMOL}$ is closed under all AFL operations with exception of inverse homomorphism.

Proof: That $\mathcal{L}_{FMOL}$ is not closed under inverse homomorphism follows from the following. Language $L_1 = \{a^{2^n} \mid n \geq 1\} \in \mathcal{L}_{FMOL}$ but the language $L_2 = h^{-1}(L_1)$ where $h(a) = a$, $h(b) = \varepsilon$ is not in $\mathcal{L}_{FMOL}$ as it follows from what has been proved about $L_2$ in [4]. Closure of $\mathcal{L}_{FMOL}$ under the remaining operations has been shown in previous two lemmas.

Lemma 3. $\mathcal{L}_{RMOL}$ is closed under the operations of union, concatenation and star.

Proof: Similar to proof of Lemma 1.

Lemma 4. $\mathcal{L}_{RMOL}$ is closed under finite transducer mappings (see [1], p.224).

Proof: Let L be an RMOL-language accepted by an RM-automaton $A = (K_1,\Sigma,\Gamma,\delta_1,\delta_2,q_0,q_f)$. Let T be a finite transducer ([1], p.224) $T = (K_2,\Delta,\Sigma,f,p_0,F)$. We can suppose that in one move automaton T does not read or output more than one symbol. Construct RM-automaton $A' = (K',\Delta,\Gamma,\delta_1',\delta_2',q_0',q_f')$ where $K' = (K_1 \times K_2) \cup q_f'$, $q_0' = (q_0,p_0)$ and $\delta_1'$ and $\delta_2'$ are defined as follows. If $(q_i,Z) \in \delta_1(q_k,b)$ and $(p_j,b) \in f(p_n,a)$, $a \in \Delta \cup \{\varepsilon\}$, $b \in \Sigma$ then $((q_i,p_j),Z) \in \delta_1'((q_k,p_n),a)$. If $(q_i,Z) \in \delta_1(q_k,\varepsilon)$ then $((q_i,p_n),Z) \in \delta_1'((q_k,p_n),\varepsilon)$ for all $p_n \in K_2$. If $(p_j,\varepsilon) \in f(p_n,a)$ then $((q_k,p_j),\varepsilon) \in \delta_1'((q_k,p_n),a)$ for all $q_k \in K_1$. If $(q_i,Z) \in \delta_2(q_k,Z_1Z_2\ldots Z_n)$ then $((q_i,p_j),Z) \in \delta_2'((q_k,p_j),Z_1Z_2\ldots Z_n)$ for all $q_j \in K_2$.
$(q_f',Z) \in \delta_2'((q_f,p_j),Z)$ for all $p_j \in F$ and $Z \in \Gamma$.

Clearly $(q_0',\alpha,\varepsilon) \overset{*}{\underset{A'}{\vdash}} (q_f',\varepsilon,Z)$ if and only if $T(\alpha) = \beta$ and $(q_0,\beta,\varepsilon) \overset{*}{\underset{A}{\vdash}} (q_f,\varepsilon,Z)$.

Theorem 6: $\mathcal{L}_{RMOL}$ is a full-AFL.

Proof: The closure of $\mathcal{L}_{RMOL}$ under the operations of union, concatenation and star has been proved in Lemma 3. The closure of $\mathcal{L}_{RMOL}$ under the operations of homomorphism, inverse homomorphism and intersection with regular set follows from the closure of $\mathcal{L}_{RMOL}$ under the finite transduction [1].

Theorem 7: $\mathcal{L}_{FMOL}$ is equal to the closure of $\mathcal{L}_{OL}$ under generalized sequential mappings (see [11], pp.128-132).

Proof: Let L be an FMOL-language generated by an FMOL-system $G = (N,T,P,S)$. Let X be a symbol not in $N \cup T$. Construct an FMOL-system $G' = (N',T,P',S)$ where $N' = N \cup X$, $P' = P \cup \{X \to X\} \cup \{A \to X : A \in N\}$. Clearly $L(G') = L(G)$. Construct the OL-system $G_1 = (N',P_1,S)$ where $P_1 = P' \cap (N' \times (N')^*)$. Clearly, $G_1$ is complete. Define g.s. machine $M = (\{q_0\}, N',T,\delta,\{q_0\},\{q_0\})$ where $(q_0,w) \in \delta(q_0,A)$ if $A \to w \in P$, $w \in T^*$. Then $M(L(G_1)) = L(G)$.

Theorem 8: $\mathcal{L}_{RMOL}$ is equal to the closure of $\mathcal{L}_{OL}$ under finite transducer mappings.

Proof: Let L be an RMOL-language generated by an RMOL-system $G = (N,T,P,S)$. Let X be a symbol not in $N \cup T$. Construct an RMOL-system $G' = (N',T,P',S)$ where $N' = N \cup X$, $P' = P \cup \{X \to X\} \cup \{A \to X : A \in N\}$. Clearly, $L(G') = L(G)$. Construct the OL-system $G_1 = (N',P_1,S)$ where $P_1 = P' \cap (N' \times (N')^*)$. Clearly, $G_1$ is complete. Let $R_i$, $i = 1,2,\ldots,n$ are the regular expressions used in productions in P and let $M_i = (K_i,\Sigma,f_i,s_i,F_i)$ be the finite automaton accepting the language $L(R_i)$. We may suppose that $K_i$, $i = 1,2,\ldots,n$ are pairwise disjoint. Define the finite transducer $T = (K,N',T,\delta,q_0,F)$ where $K = \{q_0\} \cup (\bigcup_{i=1}^{n} K_i)$, $F = \bigcup_{i=1}^{n} F_i$ and $\delta$ is defined as follows. $(s_i,\varepsilon) \in \delta(q_0,A)$ if $A \to R_i \in P'$. $(q_0,\varepsilon) \in \delta(q,\varepsilon)$ for each $q \in \bigcup_{i=1}^{n} F_i$.

If $q_i \in f_i(q_j,a)$ then $(q_i,a) \in \delta(q_j,\varepsilon)$. We can see that $T(A_1A_2\ldots A_n) = a_1a_2\ldots a_m$ is equivalent to $A_1A_2\ldots A_n \underset{G}{\Rightarrow} a_1a_2\ldots a_m$. So $T(L(G_1)) = L(G)$.

Corollary $\mathscr{L}_{RMOL}$ is the smallest full AFL containing $\mathscr{L}_{OL}$.

Proof: Any AFL is closed under finite transduction and so any AFL containing $\mathscr{L}_{OL}$ has to contain $\mathscr{L}_{RMOL}$.

Now we will study the relation of $\mathscr{L}_{RMOL}$ to the family of Index Languages [2] which will be denoted by $\mathscr{L}_{INDEX}$.

Theorem 9: $\mathscr{L}_{RMOL} \subsetneqq \mathscr{L}_{INDEX}$.

Proof: Let L be an RMOL-language given by an RMOL-system $G = (N,T,P,S)$. Let $R_i$, $i = 1,2,\ldots,n$ be all regular expressions used in the productions in P and let $G_i = (N_i,T_i,P_i,S_i)$ be a regular grammar generating language $L(R_i)$. We can suppose that $N,N_1,\ldots,N_n$ are pairwise disjoint. Define the index grammar $G' = (N',T,F,P',S')$ where $S'$ is a symbol not used in any of grammars $G,G_1,G_2,\ldots,G_n$.

$$N' = \bigcup_{i=1}^{n} N_i \cup N \cup S'; \qquad F = \{f,g\};$$

$$f \ : \{P \cap (N \times N^+)\};$$

$$g \ : \{A \to S_i \ : \ A \to R_i \in P\};$$

$$P' = \bigcup_{i=1}^{n} P_i \cup \{S' \to Sg,\ S \to Sf\}.$$

Clearly, $S' \overset{*}{\underset{G'}{\Rightarrow}} Sf^n g \overset{*}{\underset{G'}{\Rightarrow}} A_1gA_2g\ldots A_mg$ is equivalent to $S \overset{*}{\underset{G}{\Rightarrow}} A_1A_2\ldots A_m$, $A_1,A_2,\ldots,A_m \in N$, and $A_ig \overset{*}{\underset{G'}{\Rightarrow}} w$ where $w \in T^*$ is equivalent to $A_i \underset{G}{\Rightarrow} w$.

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

Consider now the language $L = \{(a^n b^n c^n)^m : n,m \geq 1\}$ to show the proper inclusion of $\mathcal{L}_{RMOL}$ in $\mathcal{L}_{INDEX}$. It is quite straightforward but rather tedious to show that L is not an RMOL-language and we will omit this proof. However, L is generated by the index grammar $G = (N,\{a,b,c\}, \{f_1,f_2,f_3\},P,S)$ where $N = \{S,S_1,S_2,A,B\}$;

$$P = \{S \to S_2f_2S_1f_3,\ S_1 \to f_2S_1f_3,\ S_1 \to f_2f_3,\ S_2 \to S_2f_1\};$$

$$f_1 = \{S_2 \to S_2A,\ A \to A,\ S_2 \to A\};$$

$$f_2 = \{A \to aAc,\ A \to aBc\};$$

and $$f_3 = \{B \to bB,\ B \to b\}.$$

Clearly, all the derivations in G are of the form

$$S \overset{*}{\underset{G}{\Rightarrow}} S_2f_1^mf_2^nf_3^n \overset{*}{\underset{G}{\Rightarrow}} Af_2^nf_3^nAf_1f_2^nAf_1^2f_2^nf_3^n \ldots Af_1^{n-1}f_2^nf_3^n \overset{*}{\underset{G}{\Rightarrow}}$$

$$\overset{*}{\underset{G}{\Rightarrow}} (a^nBf_3^nc^n)^m \overset{*}{\underset{G}{\Rightarrow}} (a^nb^nc^n)^m.$$

The summary of results on proper inclusions of the considered families of languages is given in figure 1. The meaning of the tree is the following. If two nodes labeled say $\mathcal{L}_A, \mathcal{L}_B$ are connected by an edge, the node $\mathcal{L}_B$ being below the node $\mathcal{L}_A$, then $\mathcal{L}_B \subsetneq \mathcal{L}_A$.

Proper inclusion of $\mathcal{L}_{INDEX}$ in $\mathcal{L}_{CS}$ has been shown in [2]. Incomparability of $\mathcal{L}_{OL}$ and the family of finite languages has been shown in [12].

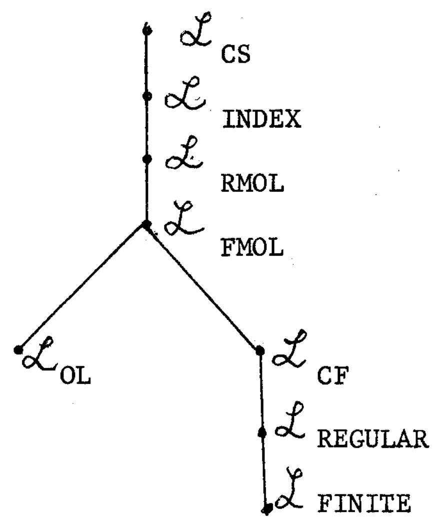
$\mathcal{L}_{CS}$
$\mathcal{L}_{INDEX}$
$\mathcal{L}_{RMOL}$
$\mathcal{L}_{FMOL}$
$\mathcal{L}_{OL}$
$\mathcal{L}_{CF}$
$\mathcal{L}_{REGULAR}$
$\mathcal{L}_{FINITE}$

Figure 1.

References


[1] Aho, A.V. and Ullman, J.D., The Theory of Parsing, Translation, and Compiling, Prentice-Hall, 1972.

[2] Aho, A.V., Index Grammar - An Extension of Context-free Grammar, JACM, 15, Oct. 1968, pp.647-671.

[3] Ginsburg, S., and Greibach, S., Abstract Families of Languages, Memoirs of the American Mathematical Society n. 87, 1969, pp.1-32.

[4] Herman, G.T., 1972a, Closure Properties of Some Families of Languages Associated with Biological Systems, Information and Control (submitted for publication).

[5] Hopcroft, J.E. and Ullman, J.D., Formal Languages and their Relation to Automata, 1969, Addison-Wesley Publ. Comp.

[6] Leeuwen, J. van, 1972, Canonical Restrictions of Lindenmayer-Languages, Abstracts of IVth Int. Congr. for Logic, Methodol. and Philos. of Sic., Bucharest, p.284.

[7] Lindenmayer, A., 1968, Mathematical Models for Cellular Interactions in Development, Parts I and II, J. Theoret. Biol. 18:280-315.

[8] Lindenmayer, A., 1971, Developmental Systems without Cellular Interactions, their Languages and Grammars, J. Theoret. Biol. 30:455-484.

[9] Lindenmayer, A. and Rozenberg, G., 1972, Developmental Systems and Languages, Proceedings of Fourth Annual ACM Symposium on Theory of Computing.

[10] Rozenberg, G., 1970, Some Results on OL-languages, Parts I and II, Elektr. Rekencentrum Utrecht, Publ. Nos.93 and 95, 30 pp.

[11] Rozenberg, G., 1971, On some Properties of Propagating D-O-L Systems, Elektr. Rekencentrum Utrecht, Publ. No.106, 30 pp.

[12] Rozenberg, G. and Doucet, P.G., 1971, On OL-Languages, Information and Control, 19:302-318.

[13] Rozenberg, G. and Lindenmayer, A., 1972, Developmental Systems with Locally Catenative Formulas, Acta Informatica (submitted for publication ).

[14] Salomaa, A.,(manuscript), Formal Languages (Academic Press, New York).