RAIRO-Theor. Inf. Appl. 48 (2014) 127–147
DOI: 10.1051/ita/2014007

Available online at:
www.rairo-ita.org

CONDITIONAL LINDENMAYER SYSTEMS WITH SUBREGULAR CONDITIONS: THE NON-EXTENDED CASE

Jürgen Dassow[1] and Stefan Rudolf[2]

Abstract. We consider conditional tabled Lindenmayer sytems without interaction, where each table is associated with a regular set and a table can only be applied to a sentential form which is contained in its associated regular set. We study the effect to the generative power, if we use instead of arbitrary regular languages only finite, nilpotent, monoidal, combinational, definite, ordered, union-free, star-free, strictly locally testable, commutative regular, circular regular, and suffix-closed regular languages. Essentially, we prove that the hierarchy of language families obtained from conditional Lindenmayer systems with subregular conditions is almost identical to the hierarchy of families of subregular languages.

Mathematics Subject Classification. 68Q42, 68Q45.

1. Introduction

In the theory of formal languages one imposes very often conditions to perform a step in the generation of words. By practical reasons – but also by theoretical considerations – it is very useful that one can check the condition by an efficient procedure. Thus one relates the condition to regular languages, for which the membership problem can be decided in linear time. We mention here as examples:

- regularly controlled context-free grammars, where a word only belongs to the generated languages if it can be derived by applying a sequence of rules which belongs to a given regular language (introduced in [19], see [13] for details),

Keywords and phrases. Lindenmayer systems, controlled derivations.

[1] Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, PSF 4120, 39016 Magdeburg, Germany. dassow@iws.cs.uni-magdeburg.de

[2] Fliederweg 7b, 65527 Niedernhausen, Germany

Article published by EDP Sciences

- conditional context-free grammars, where pairs of rules and regular sets are given, and the rule can be applied if and only if the current sentential form belongs to the regular set associated with the rule (introduced in [18], see [13] for details),
- tree controlled context-free grammars, where a word only belongs to the generated languages if it has a derivation tree such that all levels of the tree belong to a given regular language (introduced in [3]),
- networks of evolutionary processors, where sets of words are associated with nodes of a graph, in derivation steps local mutations are modelled, and in communication steps words associated with one node are sent to other nodes according filters, *i.e.*, the word can leave the node if it belongs to some regular language which corresponds to the node, and it can enter the other node, if it belongs to the regular set corresponding to that node (introduced in [1, 2], see [28] for details),
- contextual grammars with selection languages, where a context can only be wrapped around a word if and only if it belongs to a regular set associated with the context (introduced in [24, 27], see [31] for details).

In these cases the process of checking the condition given by a regular language or some regular languages is now very simple and efficient, however, the increase of generative power is considerable (for instance, for the first three devices, one has an increase from context-free languages to recursively enumerable languages). Since on the one hand practical requirement do not ask for arbitrary regular languages and on the other hand theoretical studies – for instance proofs – show that only special regular languages are used, it is very natural to study the devices with subregular languages for the control. Investigations on the change of the generative power, if subregular restrictions defined by combinatorial and algebraic properties are done in [4] for regularly controlled grammars, in [8, 10] for conditional grammars, in [16] for tree controlled grammars, in [11,25] for networks with evolutionary processors, and in [7,12] for contextual grammars. Results on the effect of subregular restrictions given by bounds on the number of states/nonterminals/productions necessary to accept/generate the regular language can be found in [6] for regularly controlled grammars, in [5] for conditional grammars, in [15] for tree controlled grammars, in [17] for networks with evolutionary processors, and in [12, 26] for contextual grammars.

In this paper we discuss conditional tabled Lindenmayer systems (conditional T0L systems, for short). The conditions given as regular sets are used as in the case of conditional context-free grammars, *i.e.*, a table can only be applied to a word, if the word is contained in the regular set associated with the table. In the papers [34] (for the extended case) and [9], conditional Lindenmayer systems were studied, where the conditions require that certain letters occur or some letters do not occur in their words.

In this paper we consider conditional T0L systems where the conditions are taken from the following subregular families: finite, nilpotent, monoidal, combinational, definite, ordered, union-free, star-free, strictly locally testable, commutative regular, circular regular, and suffix-closed regular languages.

We prove that the hierarchy of these families almost coincides with the hierarchy obtained by the conditional T0L systems.

2. Definitions

We assume that the reader is familiar with the basic concepts of the theory of formal languages and automata. In this section we only recall some notations and some definitions such that a reader can understand the results. We refer to [13, 32, 33].

For an alphabet V, *i.e.*, V is a finite non-empty set, the set of all words and all non-empty words over V are denoted by V^* and V^+, respectively. The empty word is denoted by λ. For a language L, let $\mathrm{alph}(L)$ be the minimal set V such that $L \subseteq V^*$. For a word $w \in V^*$ and a subset C of V, the number of occurrences of letters of C in w is denoted by $\#_C(w)$. If C only consists of the letter a, we write $\#_a(w)$ instead of $\#_{\{a\}}(w)$.

The families of finite and regular languages are denoted by *FIN* and *REG*, respectively.

2.1. Subregular families of languages

The aim of this section is the definition of the subregular families of languages considered in this paper and the relation between them.

For a language L over V, we set

$$\begin{aligned}
\mathrm{Comm}(L) &= \{a_{i_1} \dots a_{i_n} \mid a_1 \dots a_n \in L,\ n \geq 1,\ \{i_1, i_2, \dots, i_n\} = \{1, 2, \dots, n\}\},\\
\mathrm{Circ}(L) &= \{vu \mid uv \in L,\ u, v \in V^*\},\\
\mathrm{Suf}(L) &= \{v \mid uv \in L,\ u, v \in V^*\}
\end{aligned}$$

We consider the following restrictions for regular languages. For a language L with $V = \mathrm{alph}(L)$, we say that L is

- *combinational* iff it can be represented in the form $L = V^*A$ for some subset $A \subseteq V$,
- *definite* iff it can be represented in the form $L = A \cup V^*B$ where A and B are finite subsets of V^*,
- *nilpotent* iff L is finite or $V^* \setminus L$ is finite,
- *commutative* iff $L = \mathrm{Comm}(L)$,
- *circular* iff $L = \mathrm{Circ}(L)$,
- *suffix-closed* (or *fully initial* or *multiple-entry* language) iff $\mathrm{Suf}(L) = L$,
- *union-free* iff L can be described by a regular expression which is only built by product and star,

- *star-free* (or *non-counting*) iff L can be described by a regular expression which is built by union, product, and complementation,
- *monoidal* iff $L = V^*$,

Definite, star-free (or non-counting regular), and suffix-closed regular (or multiple-entry) were introduced in the papers [21, 29, 30], respectively. Some properties of the above mentioned types of languages can be found in [20, 36, 38].

It is obvious that combinational, definite, nilpotent, union-free and star-free languages are regular, whereas non-regular languages of the other types mentioned above exist.

We mention the characterization of star-free languages (as non-counting languages). A language L is k-non-counting iff, for all words $x, y, z \in V^*$, $xy^k z \in L$ if and only if $xy^{k+1}z \in L$. A regular language L is star-free if and only if there is a natural number $k \geq 1$ such that L is k-non-counting.

If L is an infinite star-free language over the unary alphabet $V = \{a\}$, then there is a word $a^p \in L$ with $p \geq k$. By the mentioned characterization, we get $a^{p-k}a^k \in L$ and hence $a^{p-k}a^{k+1} = a^{p+1} \in L$. Thus, there is a natural number $s \geq 0$ such that $L = F \cup \{a^n \mid n \geq s\}$ where F is a finite set of words of length $\leq s - 2$.

For a natural number $k \geq 1$, a language L is *strictly locally k-testable* iff there are three subsets A, B and C of V^k such that $a_1 a_2 \dots a_n$ with $n \geq k$ and $a_i \in V$, $1 \leq i \leq n$, belongs to L iff $a_1 a_2 \dots a_k \in A$, $a_{j+1}a_{j+2} \dots a_{j+k} \in B$ for $1 \leq j \leq n - k - 1$, and $a_{n-k+1}a_{n-k+2} \dots a_n \in C$. Moreover, a language L is called *strictly locally testable* iff it is strictly locally k-testable for some $k \geq 1$.

Obviously, strictly locally testable languages can be accepted by finite automata, and hence they are regular.

A set $R \subset V^*$ is strictly locally 1-testable if and only if there are sets $A \subseteq V$, $B \subseteq V$, and $C \subseteq V$ such that $R = AC^*B \cup (A \cap B)$ (see for instance [8]).

By $COMB$, DEF, NIL, $COMM$, $CIRC$, SUF, UF, SF, MON, LOC_k, $k \geq 1$, and LOC, we denote the families of all combinational, definite, nilpotent, regular commutative, regular circular, regular suffix-closed, union-free, star-free, monoidal, strictly locally k-testable, and strictly locally testable languages, respectively. We set

$$\mathcal{G} = \{FIN, MON, COMB, DEF, NIL, COMM, CIRC, SUF, UF, SF, LOC\} \\ \cup \{LOC_k \mid k \geq 1\}.$$

The relations between families of $\mathcal{G}$ are investigated *e.g.* in [23, 39] and their set-theoretic relations are given in Figure 1.

2.2. Conditional Lindenmayer systems

We start with some definitions concerning Lindenmayer systems and introduce then conditional Lindenmayer systems.

A *tabled Lindenmayer system without interaction* (*T0L system*, for short) is an $(r + 2)$-tuple $H = (V, P_1, P_2, \dots, P_r, w)$, where

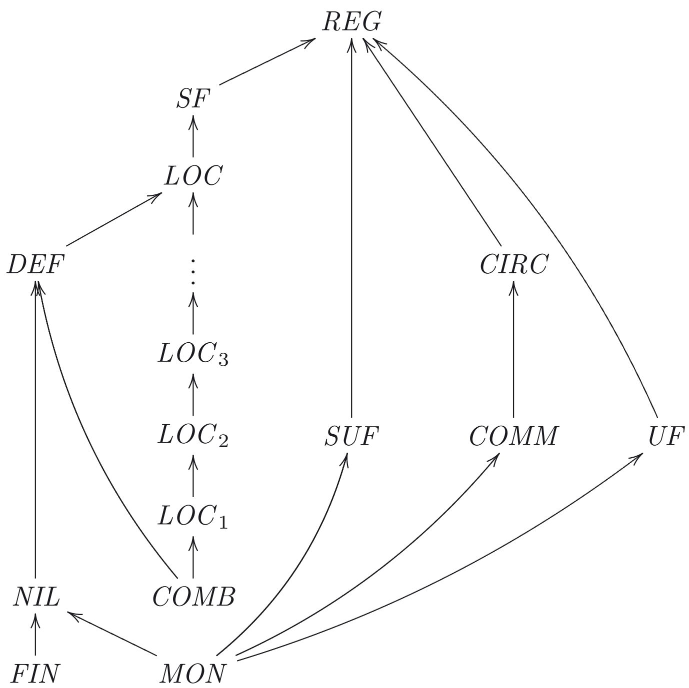

FIGURE 1. Hierarchy of subregular languages (an arrow from X to Y denotes $X \subset Y$, and if two families are not connected by a directed path then they are incomparable).

- V is an alphabet (called the underlying alphabet),
- for $1 \le i \le r$, P_i is a finite set of rules $a \to v$ with $a \in V$ and $v \in V^*$ such that, for any $b \in V$, there is a word v_b with $b \to v_b \in P_i$,
- $w \in V^+$.

The sets P_i, $1 \le i \le r$, are called tables. For simplicity, for a table, we shall give only the rules for the letters a for which a rule $a \to w$ with $w \ne a$ exists in the table, *i.e.*, for all letters b, for which no rules are mentioned, there is only the rule $b \to b$ in the table.

For $x \in V^+$ and $y \in V^*$, we say that x derives y in H, written as $x \Longrightarrow_H y$, iff

- $x = a_1 a_2 \dots a_n$ with $a_i \in V$ for $1 \le i \le n$,
- $y = y_1 y_2 \dots y_n$,
- $a_i \to y_i \in P_j$ for $1 \le i \le n$ and some j, $1 \le j \le r$.

The language $L(H)$ generated by H is defined as

$$L(H) = \{z \mid w \Longrightarrow_H^* z\}$$

where $\Longrightarrow_H^*$ is the reflexive and transitive closure of $\Longrightarrow_H$.

A T0L system is called *propagating* if no table contains a rule $a \to \lambda$.

By *T0L* and *PT0L*, we denote the families of all languages generated by T0L systems and propagating T0L systems, respectively.

Definition 2.1. A conditional T0L system is an $(n+2)$-tuple

$$H = (V, (P_1, R_1), (P_2, R_2), \dots, (P_n, R_n), w),$$

where

- $H' = (V, P_1, P_2, \dots, P_n, w)$ is a T0L system, and,
- for $1 \leq i \leq n$, R_i is a regular language over some alphabet $U \subseteq V$.

For $x \in V^+$ and $y \in V^*$, we say that x derives y in H, written as $x \Longrightarrow_H y$, if and only if there is a number j, $1 \leq j \leq n$

- $x = a_1 a_2 \dots a_t$ with $a_i \in V$ for $1 \leq i \leq t$,
- $y = y_1 y_2 \dots y_t$,
- $a_i \to y_i \in P_j$ for $1 \leq i \leq t$, and
- $x \in R_j$.

The language $L(H)$ generated by H is defined as

$$L(H) = \{z \mid w \Longrightarrow_H^* z\}$$

where $\Longrightarrow_H^*$ is the reflexive and transitive closure of $\Longrightarrow_H$.

By definition, in a conditional T0L system, a regular set R_j is associated with any table P_j, and a table P_j is only applicable to a sentential form x, if x belongs to the associated conditional language R_j.

In this paper, we study the generative power of conditional T0L systems, if one restricts to a class $X \in \mathcal{G}$ of regular languages. For $X \in \mathcal{G}$, we define $\mathcal{CL}(X)$ and $\mathcal{CPL}(X)$ as the families of all languages which can be generated by conditional T0L and conditional propagating T0L system $(V, (P_1, R_1), \dots, (P_n, R_n), w)$, where all languages R_i, $1 \leq i \leq n$, are in X.

The following relations follow immediately from the definitions.

Lemma 2.2. *For all $X, Y \in \mathcal{G}$ with $X \subseteq Y$,*

$$\mathcal{CL}(X) \subseteq \mathcal{CL}(Y), \ \mathcal{CPL}(X) \subseteq \mathcal{CPL}(Y), \ \text{and } \mathcal{CPL}(X) \subseteq \mathcal{CL}(X).$$

3. Some special languages

In this subsection we present some languages, which belong or do not belong to some language families.

Lemma 3.1. *Let*

$$L_1 = \{a^{2^n} \mid n \geq 0\} \cup \{a^{3^n} \mid n \geq 0\}.$$

Then $L_1 \in \mathcal{CL}(COMM)$, $L_1 \notin \mathcal{CL}(SF)$ and $L_1 \notin \mathcal{CL}(SUF)$.

Proof.
i) $L_1 \in \mathcal{CL}(COMM)$. The language L_1 is generated by the T0L system

$$\left(\{a\}, \left(\{a \to a^2\}, \{a\} \cup \{a^2\}^+\right), \left(\{a \to a^3\}, \{a\} \cup \{a^3\}^+\right), a\right)$$

with commutative conditions.

ii) $L_1 \notin \mathcal{CL}(SF)$. We start with some considerations which are independent of the type of the conditions.

Let $G = (V, (P_1, R_1), (P_2, R_2), \dots (P_n, R_n), \omega)$ be a conditional T0L system which generates L_1. Obviously, $V = \{a\}$ and that all rules occurring in the production sets have the form $a \to a^p$ for some $p \geq 1$ (the existence of $a \to \lambda$ would imply that the empty word $\lambda \notin L_1$ can be obtained if the rule is applicable, or the rule is not applicable and can be deleted). Let

$$h = \max\left\{|p| \mid a \to a^p \in P_i, 1 \leq i \leq n\right\}.$$

First, we note that, for any number r, there are a word $a^s \in L_1$ with $r < s$ and a component (P_i, R_i) which can be applied to a^s. If this would not hold, then we can apply components only to words of a length $\leq r$. Since we can generate from $a^{r'}$ with $r' \leq r$ only words of length $\leq r \cdot h$, the language $L(G)$ cannot be infinite.

Next we prove that any production set R_i, $1 \leq i \leq n$, which is applicable to a word a^t with $t > 2h$ only contains one rule. Assume that P_i contains two rules $a \to a^p$ and $a \to a^q$ with $p < q$. We consider the words w_1, w_2, and w_3 which are obtained by t, $t-1$ and $t-2$ applications of $a \to a^p$ and no, one or two applications of $a \to a^q$, respectively. Then

$$w_1 = a^{tp}, \;\; w_2 = a^{tp+q-p}, \text{ and } w_3 = a^{tp+2q-2p}.$$

Because the generated word belongs to the language, $w_1 \in L_1$ and hence $tp = 2^k$ or $tp = 3^k$ for some k. We only discuss the first case; the second one can be handled analogously.

Clearly,

$$2^k = tp > hp \geq h \tag{3.1}$$

because $p \geq 1$. If $tp + q - p$ is also a power of 2, say $tp + q - p = 2^l$, then we get $q - p = 2^l - 2^k$ This implies $h \geq q - p = 2^l - 2^k \geq 2^{l-1} \geq 2^k$ (because $k \leq l - 1$) which contradicts (3.1). If $tp + q - p$ is a power of 3, say $tp + q - p = 3^l$, we consider additionally $w_3 \in L_1$. If $tp + 2q - 2p = 2^m$ for some m, we get $q - p = (2^m - 2^k)/2$ and a contradiction as above. If $tp + 2q - 2p = 3^m$ for some m, then $q - p = 3^m - 3^l$ and we obtain analogously a contradiction, again.

We now prove that the only rule in P_i has the form $a \to a^{2^g}$ or $a \to a^{3^g}$ with $g \geq 1$. This can be seen as follows. First we note that we can assume without loss of generality that the only rule is different from $a \to a$ since a component with only $a \to a$ does not change the sentential forms and can be deleted. If $t = 2^x$ for some x, then the only word derivable from a^t using $a \to a^p$ is a^{p2^x}. If $p2^x = 2^y$ for some y, then $p = 2^{y-x}$. Setting $g = y - x$, we have that the only rule has the form

$a \to a^{2^g}$ for some $g \geq 1$. If $p2^x = 3^y$ for some y, we get $p = \frac{3^y}{2^x}$ which is impossible since the right hand side is not an integer. Analogously, for $t = 3^x$ for some x, we obtain that the rule has the form $a \to a^{3^g}$ with $g \geq 1$.

If R_i is a star-free condition, then there are a finite set $F \subset \{a\}^*$ and an integer $z \geq 1$ such that $R_i = F \cup \{a^j \mid j \geq z\}$. Without loss of generality we can assume that $t \leq z$. If $P_i = \{a \to a^{2^g}\}$, then we choose u such that $3^u \geq t$ and get $a^{3^u} \in R_i$. Hence we can apply $a \to a^{2^g}$ to a^{3^u} and obtain the word $a^{3^u 2^g}$ which is not in L_1. Thus we have a contradiction to $L(G) = L_1$. If $P_i = \{a \to a^{3^g}\}$, we choose u such that $2^u \geq t$ and derive a contradiction analogously.

Since we obtain a contradiction in all cases, L_1 cannot be generated by a T0L system with star-free conditions.

iii) $L_1 \notin \mathcal{CL}(SUF)$. As above we can show that there are components which are applicable to long words and that theses components only have the form $(\{a \to a^{2^g}\}, R)$ or $(\{a \to a^{3^g}\}, R)$ where $g \geq 1$ and R is a suffix-closed regular language. Let $a^t \in L_1$ be a sufficiently long word. If $t = 2^x$ and the only rule is $a \to a^{3^g}$, then we derive $a^{2^x 3^g} \notin L_1$ and thus a contradiction. If $t = 2^x$ and the only rule is $a \to a^{2^g}$, then we can apply the rule to a^{3^y} with $3^y \leq 2^x$, too, since $a^{2^x} \in R$ and R is suffix-closed, get $a^{3^y 2^g} \notin L_1$ and a contradiction, again. □

Lemma 3.2. *Let*

$$L_2 = \{b\} \cup \left\{a^{2^n} c^{2^n} a^{2^n} \mid n \geq 1\right\} \cup \left\{a^{2^n} c^{3^n} a^{2^n} \mid n \geq 1\right\}.$$

Then $L_2 \in \mathcal{CL}(SUF)$ and $L_2 \notin \mathcal{CL}(SF)$.

Proof.
i) $L_2 \in \mathcal{CL}(SUF)$. The T0L system

$$\big(\{a,b,c\}, \big(\{b \to a^2c^2a^2, a \to a^2, c \to c^2\}, \mathrm{Suf}\left(\{b\} \cup \{a\}^+\{c^2\}^+\{a\}^+\right)\big)$$
$$\big(\{b \to a^2c^3a^2, a \to a^2, c \to c^3\}, \mathrm{Suf}\left(\{b\} \cup \{a\}^+\{c^3\}^+\{a\}^+\right)\big), b\big)$$

generates L_2 (note that the only words of the control languages which occur as sentential forms are b, $a^{2^n} c^{2^n} a^{2^n}$ and $a^{2^n} c^{3^n} a^{2^n}$ with $n \geq 1$).

ii) $L_2 \notin \mathcal{CL}(SF)$. Let us assume that $L_2 \in \mathcal{CL}(SF)$. Then $L_2 = L(G)$ holds for some T0L system $G = (\{a,b,c\}, (P_1,R_1), (P_2,R_2), \ldots, (P_n,R_n), \omega)$ with star-free conditions. Analogously to the proof of Lemma 3.1 we can show that each table (P_i, R_i), $1 \leq i \leq n$, applicable to a sufficiently long word has only one rule with the left hand side a and c, and these rules are $a \to a^{2^p}$ and $c \to c^{2^p}$ or $c \to c^{3^p}$ where $p \geq 1$.

Let r be sufficiently large. If a table P_i with $c \to c^{2^p}$ is applicable to a word $a^{2^r} c^{3^r} a^{2^r}$, then we derive the word $a^{2^{r+p}} c^{3^r 2^p} a^{2^{r+p}}$ which is not in L_2. Hence such a table can only be applicable to words $a^{2^r} c^{2^r} a^{2^r}$. However, since R_i is star-free, there is a k such that $a^{2^r} c^k c^{2^r - k} a^{2^r} \in R_i$ implies $a^{2^r} c^{k+j} c^{2^r - k} a^{2^r} \in R_i$ for any $j \geq 0$. Because r is sufficiently large, we can assume that $2^r \geq k$. If we now choose

$j = 3^r - 2^r$, we obtain that $a^{2^r}c^{3^r}a^{2^r} \in R_i$. Then P_i is applicable to $a^{2^r}c^{3^r}a^{2^r}$ which leads to a contradiction as shown above. Therefore tables with $c \to c^{2^p}$ are not applicable to sufficiently long words.

By analogous arguments we can show that also tables with $c \to c^{3^p}$ are not applicable to sufficiently large words.

Thus no table is applicable to sufficiently long words which is impossible because we have to generate an infinite language. □

Lemma 3.3. *Let*

$$L_3 = \left\{a^2b^2, b^2a^2, a^4b^4\right\}.$$

Then $L_3 \in \mathcal{CL}(SUF)$, $L_3 \in \mathcal{CL}(FIN)$, $L_3 \in \mathcal{CL}(COMB)$, *and* $L_3 \notin \mathcal{CL}(CIRC)$.

Proof.
i) $L_3 \in \mathcal{CL}(SUF)$, $L_3 \in \mathcal{CL}(FIN)$, $L_3 \in \mathcal{CL}(COMB)$. The language L_3 is generated by the T0L systems

$$\left(\{a,b\}, (\{a \to b, b \to a\}, \mathrm{Suf}(\{a^2b^2\})), \left(\{a \to a^2, b \to b^2\}, \mathrm{Suf}(\{a^2b^2\})\right), a^2b^2\right)$$

with conditions in SUF (since the productions can only be applied to the axiom because the other words from $\mathrm{Suf}(\{a^2b^2\})$ do not occur),

$$\left(\{a,b\}, (\{a \to b, b \to a\}, \{a^2b^2\}), \left(\{a \to a^2, b \to b^2\}, \{a^2b^2\}\right), a^2b^2\right)$$

with finite conditions, and

$$\left(\{a,b\}, (\{a \to b, b \to a\}, \{a,b\}^*\{a\}), \left(\{a \to b^2, b \to a^2\}, \{a,b\}^*\{a\}\right), b^2a^2\right)$$

with conditions in $COMB$.

ii) $L_3 \notin \mathcal{CL}(CIRC)$. Assume that L_3 is in $\mathcal{CL}(CIRC)$. Then there is a T0L system $G = (\{a,b\}, (P_1, R_1), (P_2, R_2), \dots, (P_n, R_n), \omega)$ with circular conditions such that $L(G) = L_3$. We discuss the possibilities for the generation of b^4a^4.

Case 1. a^4b^4 is the axiom ω. In order to generate L_3 there is a table (P_i, R_i) which is applicable to the axiom and produces one of the other two words in L_3. Obviously, the rules of P_i delete at least four letters. If $a \to \lambda$ and $b \to \lambda$ are in P_i, then we can also derive the empty word, which is not in L_3. Let us assume that $a \to \lambda$ is in P_i and that the rules for b derive non-empty words. Then a^2b^2 or b^2a^2 has to be derived from b^4. This is only possible if $b \to a$ and $b \to b$ are in P_i. But then we can also generate $b^4 \notin L_3$. Analogously we get a contradiction if $b \to \lambda$ is in P_i. Hence b^4a^4 cannot be the axiom.

Case 2. a^4b^4 is derived in one step from a^2b^2. Let $a^2b^2 \Longrightarrow x_1x_2x_3x_4 = a^4b^4$ using a table (P_i, R_i) with $a \to x_1$, $a \to x_2$, $b \to x_3$, and $b \to x_4$. Then we can also derive the word $x_1x_1x_3x_3$. We discuss some subcases:

Subcase 2.1. $x_1x_1x_3x_3 = b^2a^2$. If $x_1 = \lambda$, then $x_3x_3 = b^2a^2$ which is impossible. Thus x_1 is non-empty and begins with a b. But then $x_1x_2x_3x_4$ starts with b, too, which contradicts $x_1x_2x_3x_4 = a^4b^4$.

Subcase 2.2. $x_1x_1x_3x_3 = a^2b^2$. Again, x_1 and x_3 are different from the empty word. Therefore, $x_1 = a$ and $x_3 = b$. This implies $x_2 = a^3b^z$ for some $z \geq 0$. Since we can generate $x_2x_2x_3x_3$, too, we can generate a word with at least six occurrences of a, which is not in L_3.

Subcase 2.3. $x_1x_1x_3x_3 = a^4b^4$. We get $a^4b^4 = x_1x_1x_3x_3 = x_1x_2x_3x_4$. Again, x_1 and x_3 are different from the empty word. It is now easy to see that $x_1 = x_2 = a^2$ and $x_3 = x_4 = b^2$ follows. Since $a^2b^2 \in R_i$, we also have $b^2a^2 \in R_i$ and can derive the word $x_3x_3x_1x_1 = b^4a^4$ which is not in L_3.

Case 3. a^4b^4 is derived in one step from b^2a^2. We get contradictions as in Case 2.

Thus $L_3 \notin \mathcal{CL}(CIRC)$. □

Lemma 3.4. *Let*

$$L_4 = \left\{aa, b^8, b^{10}, b^4c^3, b^5c^3, c^3b^4, c^3b^5, c^6, d^{15}e^9\right\}.$$

Then $L_4 \in \mathcal{CL}(FIN)$ *and* $L_4 \notin \mathcal{CL}(SUF)$.

Proof.
i) $L_4 \in \mathcal{CL}(NIL)$. The system

$$(\{a, b, c, d, e\}, (P_{1,1}, R_1), (P_{1,2}, R_1), (P_2, R_2), aa)$$

with

$$P_{1,1} = \left\{a \to b^4, a \to c^3\right\},\ R_1 = \{aa\},$$
$$P_{1,2} = \left\{a \to b^5, a \to c^3\right\},$$
$$P_2 = \left\{b \to d^3, c \to e^3\right\},\ R_2 = \left\{b^5c^3\right\},$$

has finite conditions and generates the language L_4.

ii) $L_4 \notin \mathcal{CL}(SUF)$. Suppose that there is a system G' with suffix-closed conditions which also generates this language. If the word aa is not the axiom of the system G', then it is derived from one of the other words. This can only be achieved if an erasing rule exists for every letter occurring in the word from which aa is derived. Then, however, also the empty word can be obtained which does not belong to the language L_4. Thus, the word aa is the axiom of the system G'. Since the word $d^{15}e^9$ is not the axiom, it is derived from one of the other words. It can be obtained from the word b^5c^3 by the rules $b \to d^3$ and $c \to e^3$. If it is derived from another word, then the corresponding table is not deterministic (it contains at least two rules for some letter) and hence another word consisting of the letters d and e could be generated which does not belong to the language L. Thus, the word $d^{15}e^9$ can only be obtained from the word b^5c^3. Hence, the system G' contains a pair (P, R) where the table P contains the rules $b \to d^3$ and $c \to e^3$ (other rules for b or c would yield a word which does not belong to the language L_4) and where the set R contains the word b^5c^3. Since the set R is suffix-closed, also the word $b^4c^3 \in L_4$ belongs to it and the table P can be applied to this word which yields the word $d^{12}e^9$. This word, however, does not belong to the language L_4 which proves that the language L_4 cannot be generated by a system with suffix-closed conditions. □

Lemma 3.5. *Let*

$$L_5 = \{d\} \cup \left\{ab^{2^n}c \mid n \geq 1\right\} \cup \left\{bc^{2^n}a \mid n \geq 1\right\} \cup \left\{ca^{2^n}b \mid n \geq 1\right\}.$$

Then $L_5 \in \mathcal{CL}(COMB)$ *and* $L \notin \mathcal{CL}(NIL)$.

Proof.
i) $L_5 \in \mathcal{CL}(COMB)$. The language L_5 is generated by the T0L system

$$\begin{aligned}(\{a,b,c,d\},\ &(\{a \to a, b \to b^2, c \to c, d \to abc\}, \{a,b,c,d\}^*\{c,d\}),\\ &(\{a \to a, b \to b, c \to c^2, d \to bca\}, \{a,b,c,d\}^*\{a,d\}),\\ &(\{a \to a^2, b \to b, c \to c, d \to cab\}, \{a,b,c,d\}^*\{b,d\}),\ d)\end{aligned}$$

with combinational conditions.

ii) $L_5 \notin \mathcal{CL}(NIL)$. Let us assume that L_5 is in $\mathcal{CL}(NIL)$. Then there is a T0L system $G = (\{a,b\}, (P_1,R_1), (P_2,R_2), \ldots, (P_n,R_n), \omega)$ with nilpotent conditions and $L_5 = L(G)$. If all sets R_i, $1 \leq i \leq n$, are finite, then the generated set is finite, too, in contrast to $L(G) = L_5$. Thus there is a number k, $1 \leq k \leq n$, such that R_i with $1 \leq i \leq k$ is infinite and R_j with $k+1 \leq j \leq n$ is finite. Obviously, R_i with $1 \leq i \leq k$ can be given as $R_i = F_i \cup \{w \mid |w| \geq r_i\}$ with some integer $r_i \geq 0$. Let r be a number such that $r > r_i$ for $1 \leq i \leq k$ and $r > |v|$ for all $v \in R_{k+1} \cup R_{k+2} \cup \cdots \cup R_n$.

Obviously, for any integer $m \geq 1$, there is a word w_m of length at least m and a table (P_{i_m}, R_{i_m}) such that the application of (P_{i_m}, R_{i_m}) to w_m yields a word which is longer than w_m since we cannot generate an infinite language, otherwise. Therefore P_{i_m} contains a rule $x \to y$ with $x \in \{a,b,c\}$, $y \in \{a,b,c\}^*$, and $|y| \geq 2$. We choose $m \geq r$ and discuss the possibilities for y.

Case 1. $y = w_1z_1w_2z_2w_3$ for two different letters z_1 and z_2 of $\{a,b,c\}$. Since $m \geq r$, R_{i_m} contains all words of length $\geq r$. Therefore we can also apply P_{i_m} to $x'x^{2^r}x''$ which yields a word $y'(w_1z_1w_2z_2w_3)^{2^r}y'' \notin L_5$ where $x' \to y' \in P_{i_m}$ and $x'' \to y'' \in P_{i_m}$.

Case 2. $y = z^p$ for some $z \in \{a,b,c\}$ and $p \geq 2$. Again, we can apply P_{i_m} to $x'(x'')^{2^r}x$ which yields a word ending with zz. Thus the generated word does not belong to L_5.

In both cases we were able to generate words not in L_5 which contradicts $L_5 = L(G)$. □

Lemma 3.6. *Let*

$$L_6 = \{d\} \cup \left\{ab^{2^n} \mid n \geq 0\right\} \cup \left\{b^{3^n} \mid n \geq 1\right\}.$$

Then $L_6 \in \mathcal{CL}(LOC_k)$ *for any* $k \geq 1$, $L_6 \notin \mathcal{CL}(DEF)$, *and* $L_6 \notin \mathcal{CL}(SUF)$.

Proof.
i) $L_6 \in \mathcal{CL}(LOC_k)$ for $k \geq 1$. It is sufficient to show that $L_6 \in LOC_1$. By the characterization mentioned in Subsection 2.1 and choosing $A = \{a,d\}$, $B = \{b,d\}$, $C = \{b\}$, the set $R_1 = AC^*B \cup (A \cap B) = \{d\} \cup \{a,d\}\{b\}^*\{b,d\}$ is strictly locally 1-testable. Analogously, we can obtain that $R_2 = \{b,d\} \cup \{b,d\}\{b\}^*\{b,d\}$ is strictly locally 1-testable. Therefore the T0L system

$$\left(\{a,b,d\}, \left(\left\{a \to a, b \to b^2, d \to ab\right\}, R_1\right), \left(\left\{a \to a, b \to b^3, d \to b^3\right\}, R_2\right), d\right)$$

has conditions in LOC_1. Moreover, it generates L_6.

ii) $L_6 \notin \mathcal{CL}(DEF)$. Let us assume that the language L_6 is generated by a T0L system $G = (\{a,b\}, (P_1,R_1), (P_2,R_2), \ldots, (P_n,R_n), \omega)$ with definite conditions. Then, for $1 \leq i \leq n$, $R_i = F_i \cup U_i^* A_i$ for some finite sets $F_i, A_i \subset U^*$. Let k and h be numbers such that $k \geq |u|$ for $u \in \bigcup_{i=1}^n (F_i \cup A_i)$ and h is larger than the length of all right hand sides in rules of $\bigcup_{i=1}^n P_i$. Now we consider the word ab^{2^r} for $r \geq k \cdot h \cdot (|\omega|+3)$. Then there is a derivation $w \Longrightarrow ab^{2^r}$ with $|w| \geq 3$, $w \neq ab^{2^r}$ using some table (P_i, R_i), $1 \leq i \leq n$. Thus w contains at least two occurrences of b.

Assume that $w = b^{3^j}$ for some $j \geq 1$. To generate ab^{2^r} there is a rule $b \to ax$ in P_i. Then $axaxaxz \in L(G)$ using this rule for the first three letters b. Because $axaxaxz \notin L_6$, we have a contradiction.

Assume that $w = ab^{2^j}$ for some $j \geq 1$. Note that $2^j \geq k$. Therefore $ab^f \in R_i$ and $b^f \in R_i$ for $f \geq 2^j$. Moreover, P_i contains rules $a \to ab^p$ and $b \to b^q$ with $p \geq 0$ and $q \geq 1$. We choose p and q maximal with respect to the rules in P_i. Then we also have the derivations

$$ab^{2^j} \Longrightarrow ab^{p+q2^j},\ ab^{2^{j+1}} \Longrightarrow ab^{p+q2^{j+1}},\ \text{and}\ ab^{2^{j+2}} \Longrightarrow ab^{p+q2^{j+2}}.$$

By our choices and the structure of L_6,

$$p + q2^j = 2^r,\ p + q2^{j+1} = 2^s,\ \text{and}\ p + q2^{j+2} = 2^t$$

for some $r < s < t$. Therefore

$$2^t - 2^s = q2^{j+1} = 2q2^j = 2(2^s - 2^r),$$

from which $2^t + 2^r = 32^s$ follows. Since $s < t$, we get $t = s+1$ and $2^t - 2^s = 2^s = q2^{j+1}$. Therefore $q = 2^u$ for some j. Now the application to b^{3^j} gives $b^{3^j 2^u} \in L(G)$. This is a contradiction since $b^{3^j 2^u} \notin L_6$.

iii) $L_6 \notin \mathcal{CL}(SUF)$ can be shown analogously, because ab^{2^j} has a suffix b^3 to which we can apply $b \to b^{2^u}$, too. □

Lemma 3.7. *Let k be an integer with $k \geq 2$ and*

$$S_k = \left\{a^k b^k c^k, a^k b^{2k} c^k\right\}.$$

Then $S_k \in \mathcal{CL}(FIN)$ and $S_k \notin \mathcal{CL}(LOC_k)$.

Proof.
i) $S_k \in \mathcal{CL}(FIN)$. The T0L system

$$G = \left(\{a, b, c\}, \left(\left\{a \to a, b \to b^2, c \to c\right\}, \left\{a^k b^k c^k\right\}\right), a^k b^k c^k\right)$$

generates T_k.

ii) $S_k \notin \mathcal{CL}(LOC_k)$. Let us assume that $S_k = L(G)$ is generated by a T0L system $G = (\{a, b, c\}, (P_1, R_1), (P_2, R_2), \ldots, (P_n, R_n), \omega)$ with strictly locally k-testable conditions. There has to be a table (P_i, R_i) which is applicable to the axiom. If there is a rule $x \to w$ such that $x \in \{a, b, c\}$ and w contains occurrences with two letters, then we can derive a word containing the subword ww which is impossible by the structure of S_k. Thus all rules have the form $x \to y^p$ for some $x, y \in \{a, b, c\}$ and $p \geq 0$. If $p = 0$ for some rule, then we can derive a word with occurrences of at most two letters which does not belong to S_k. Thus $p \geq 1$. Moreover, if $x \neq y$, then we can derive words which are not in $\{a\}^+\{b\}^+\{c\}^+$ and therefore not in S_k. This implies that $a^k b^k c^k$ is the axiom and the only possible rules are $a \to a$, $b \to b^2$, $c \to c$. Now let R_i be described by A_i, B_i and C_i. Then $a^k \in A_i$, $a^r b^s, b^s c^r \in B_i$ for $r + s = k$, $r \geq 0$, $s \geq 1$, and $c^k \in C_i$. This implies that $a^k b^{2k} c^k \in R_i$, too. Thus we can derive $a^k b^{4k} c^k \notin S_k$. □

Lemma 3.8. *Let k be an integer with $k \geq 1$ and*

$$T_k = \{c\} \cup \{ba^p \mid p \leq k\} \cup \left\{ba^{(k+2)5^n} \mid n \geq 0\right\}.$$

Then $T_k \notin \mathcal{CL}(LOC_k)$ and $T_k \in \mathcal{CL}(LOC_{k+1})$.

Proof.
i) $T_k \in \mathcal{CL}(LOC_{k+1})$. The finite language $U_1 = \{c\} \cup \{ab^r \mid r \leq k-1\}$ is strictly locally $(k+1)$-testable, because there is no requirement for words of length $< k+1$ by the definition of strictly locally $(k+1)$-testable languages. Moreover, if we choose $A = \{ba^k\}$, $B = C = \{b^{k+1}\}$, then $U_2 = \{ba^q \mid q \geq k+1\}$ is in LOC_{k+1}. Hence

$$\left(\{a, b, c\},\ \left(\left\{a \to a, b \to ba, c \to ba, c \to ba^{k+2}\right\}, U_1\right),\right.$$
$$\left.\left(\left\{a \to a^5, b \to b, c \to c\right\}, U_2\right),\ c\right)$$

is a T0L system with conditions in LOC_{k+1} and derives T_k.

ii) $T_k \notin \mathcal{CL}(LOC_k)$. Let us assume that $T_k = L(G)$ is generated by a T0L system $G = (\{a, b\}, (P_1, R_1), (P_2, R_2), \ldots, (P_n, R_n), \omega)$ with strictly locally k-testable conditions. Again, there is table (P_i, R_i), $1 \leq i \leq n$, that can be applied to a sufficiently long word w, that P_i contains at most one rule with left hand side a and that this rule has the form $a \to a^{5^p}$ for some $p \geq 1$, and that the only rule with left hand side b in P_i is $b \to b$. Since $w = ba^{(k+2)5^n} \in R_i$, we get $ba^{k-1} \in A_i$, $a^k \in B_i$, and $a^k \in C_i$, where A_i, B_i, and C_i are the sets describing R_i. Hence $ba^k \in R_i$ and the application of R_i gives $ba^{k\cdot 5^p}$. By the structure of L_6, $k \cdot 5^p = (k+2)5^q$ for some $q \geq 1$. Obviously, $q < p$. Hence

$$5^{p-q} = 1 + \frac{2}{k}\cdot \tag{3.2}$$

If $k \geq 3$, we get a contradiction since the right side of (3.2) is not an integer whereas the left side is an integer. If $k = 2$, the right hand side of (3.2) is 2, and thus not a divisor of the left side. If $k = 1$, the right hand side of (3.2) is 3, and hence not a divisor of the left side.

Therefore we get a contradiction in all cases which proves that our assumption is false and $T_k \notin \mathcal{CL}(LOC_k)$. □

Lemma 3.9. *Let*

$$L_7 = \{a^n cb^n \mid n \geq 1\} \cup \{a^n cb^n a^n cb^n : n \geq 1\} \cup \left\{a^n d^2 b^n a^n d^2 b^n : n \geq 1\right\}.$$

Then $L_7 \in \mathcal{CL}(SF)$ and $L_7 \notin \mathcal{CL}(LOC)$.

Proof.
i) $L_7 \in \mathcal{CL}(SF)$. Obviously, L_7 is generated by the conditional Lindenmayer system

$$\begin{aligned}(\{a,b,c,d\},\ &(\{a \to a, b \to b, c \to cbac, d \to d\},\ \{acb\}),\\ &\left(\{a \to a, b \to b, c \to acb, d \to d\}, \{a\}^+\{c\}\{b\}^+ \cup (\{a\}^+\{c\}\{b\}^+)^2\right),\\ &\left(\{a \to a, b \to b, c \to dd, d \to d\},\ \{a\}^+\{c\}\{b\}^+\{a\}^+\{c\}\{b\}^+\right), acb).\end{aligned}$$

The three conditions are 2-non-counting and, thus, they are star-free.

ii) $L_7 \notin \mathcal{CL}(LOC)$. Let us assume that L_7 is generated by some conditional Lindenmayer system $G = (\{a,b,c,d\}, (P_1, R_1), (P_2, R_2), \ldots, (P_n, R_n), \omega)$ with conditions in LOC. We look on the derivation of $a^n d^2 b^n a^n d^2 b^n$ for sufficiently large numbers n.

If $a^m d^2 b^m a^m d^2 b^m \Longrightarrow a^n d^2 b^n a^n d^2 b^n$ using some P_i, then – as above – we can show that there is exactly one rule with left side x for $x \in \{a,b,d\}$ and $a \to a^k, b \to b^k, d \to d$ are the unique rules for a, b, d, where k is a positive integer. From this it follows that from a finite set of words of the form $a^r d^2 b^r a^r d^2 b^r$ we cannot derive all words of this form.

A derivation $a^m cb^m \Longrightarrow a^n d^2 b^n a^n d^2 b^n$ is impossible for sufficiently large n since it requires a rule $c \to a^r d^2 a^n b^n d^2 b^s$ (because rules with left hand side a or b and d in the right side allow the derivation of words with more than four occurrences of d) which is impossible for large enough n.

Thus there is a derivation $a^m cb^m a^m cb^m \Longrightarrow a^n d^2 b^n a^n d^2 b^n$ with m, n sufficiently large by some set P_i. It is easy to see that $a \to a^t, b \to b^{t'}, c \to a^r d^2 b^s \in P_i$ for some numbers r, s, t, t' with $0 \leq r$, $0 \leq s$, $1 \leq t$, and $1 \leq t'$. Let R_i be the condition. Then R_i is a strictly locally k-testable language for some k. Clearly, we can assume that $k \leq m$. Then $a^m cb^m a^m cb^m \in R_i$ implies that $a^m cb^m$ is in R_i, too. Therefore we can derive the word $a^{mt+r} d^2 b^{mt'+s}$ which is not in L_7. Thus we get a contradiction to $L(G) = L_7$. □

Lemma 3.10. *Let* $L_8 = \{aabb, ababbb, abbabb, baabbb, bababb, baaaabbb\}$*. Then* $L_8 \in \mathcal{CL}(CIRC)$*,* $L_8 \in \mathcal{CL}(SUF)$*,* $L_8 \notin \mathcal{CL}(COMM)$*, and* $L_8 \notin \mathcal{CL}(COMB)$*.*

Proof.
i) $L_8 \in \mathcal{CL}(CIRC)$. The conditional Lindenmayer system

$$(\{a, b\},\ (\{a \to ab, a \to ba, b \to b\}, \mathrm{Circ}(\{aabb\})),$$
$$(\{a \to a^2, b \to b\}, \mathrm{Circ}(\{ba^2b^3\}),\ aabb)$$

with circular conditions generates L_8.

ii) $L_8 \in \mathcal{CL}(SUF)$. If we replace Circ by Suf in the system given in i), then we obtain that L_8 is generated by a Lindenmayer system with conditions in SUF.

iii) $L_8 \notin \mathcal{CL}(COMM)$. Let us assume that L_8 is generated by a conditional Lindenmayer system $G = (\{a, b\}, (P_1, R_1), (P_2, R_2), \ldots, (P_n, R_n), \omega)$ with commutative conditions. We now discuss the generation of $ba^4b^3 \in L_8$.

Case 1. ba^4b^3 is the axiom.

Since ba^4b^3 is the longest word in L_8, there is an i, $1 \le i \le n$, such that $ba^4b^3 \in R_i$, and there is an $x \in \{a, b\}$ such that $x \to \lambda \in P_i$. If $a \to \lambda$ and $b \to \lambda$ are both in P_i, we can generate the empty word which is not in L_8. If $a \to \lambda$ and $b \to w_b \neq \lambda$ or $b \to \lambda$ and $a \to w_a \neq \lambda$ are in P_i, we can generate w_b^4 or w_a^4, respectively. However, L_8 contains no word which is a fourth power of some word. Thus we have a contradiction, again.

Case 2. $a^2b^2 \Longrightarrow ba^4b^3$ by some P_i, and $a^2b^2 \in R_i$, $1 \le i \le n$.

Assume that $a \to \lambda \in P_i$. If $b \to \lambda$ is in P_i, too, we can generate the empty word which is not in L_8. Thus $b \to w_b \neq \lambda \in P_i$. Now we can generate w_b^2. Looking on the words in L_8, it follows that $w_b = abb$. If P_i contains only $a \to \lambda$ and $b \to abb$, then the assumed derivation $a^2b^2 \Longrightarrow ba^4b^3$ is impossible, and we have a contradiction. If there is a further rule $a \to w_a \neq \lambda$, then $w_a^2abbabb$ can be derived from a^2b^2, but $w_a^2abbabb$ does not belong to L_8. Thus $a \to \lambda$ is not in P_i.

Analogously, we can prove that $b \to \lambda$ is not in P_i.

Therefore, there are $a \to w_a$ and $b \to w_b$ with $w_a \neq \lambda$ and $w_b \neq \lambda$ in P_i. Then we can generate $z = w_a^2w_b^2$. Looking on the words in L_8 it follows that $w_b = b$ and $w_a \in \{a, ab, ba\}$. Hence we can only generate words of length ≤ 6 from a^2b^2 which contradicts the assumed derivation $a^2b^2 \Longrightarrow ba^4b^3$.

Case 3. $w \Longrightarrow ba^4b^3$ by some P_i, $1 \le i \le n$, and $w \in R_i$ is a word of length 6.

We mention that w contains exactly two occurrences of a and four occurrences of b.

If $b \to \lambda$ is the only rule with left hand side b in P_i and $a \to w_a \in P_i$, then $w_a = abb$ as above and the only derivation is $w \Longrightarrow ab^2ab^2$ and the assumed derivation $w \Longrightarrow ba^4b^3$ does not exist. Thus there is a rule $b \to w_b \neq \lambda$ in P_i.

If $a \to \lambda$ is in P_i, then we can generate w_b^4 which does not belong to L_8 as already mentioned above. Thus $a \to w_a \neq \lambda$ for any rule in P_i with left hand side

a. If $|w_b| \geq 2$, then we can generate a word of length $4|w_b|+2|w_a| \geq 10$ which gives a contradiction since all word in L_8 have a length ≤ 8. Thus we get $|w_b| = 1$. If we take into consideration that all words end with b, we get $w_b = b$. Starting from w of length 6, we have four derivations: $ababbb \Longrightarrow w_a b w_a bbb$ or $abbabb \Longrightarrow w_a bb w_a bb$ or $babab b \Longrightarrow b w_a b w_a bb$ or $baabbb \Longrightarrow b w_a w_a bbb$. In the first three cases we only obtain a word in L_8 if $w_a = a$, and the assumed derivation $w \Longrightarrow ba^4b^3$ is not possible. In the fourth case we get $w_a \in \{a, a^2\}$. If $a \to a$ and $a \to a^2$ are in P_i, we can also derive $baa^2b^3 = ba^3b^3 \notin L_8$. If $a \to a$ is the only rule, we do not have the assumed derivation. Thus we only have $a \to a^2$. Since $baabbb \in R_i$, we have $ababbb \in R_i$ by the commutativity. Thus we can also generate $a^2ba^2bbb \notin L_8$.

Since we got a contradiction in all cases, our assumption is false, *i.e.*, L_8 is not in $\mathcal{CL}(COMM)$.

iv) $L_8 \notin \mathcal{CL}(COMB)$. The proof can be given analogously to iii) (note that the only possible combinational conditions are $\{a,b\}^+\{b\}$ and $\{a,b\}^*\{a,b\}$ since all words in L_8 end with b). □

Lemma 3.11. *Let* $L_9 = \{aa, baac\}$. *Then* $L_9 \in \mathcal{CL}(FIN)$, $L_9 \in \mathcal{CL}(MON)$, *and* $L_9 \notin \mathcal{CPL}(REG)$.

Proof. The language L_9 is generated by the T0L system

$$(\{a,b,c\}, (\{a \to a, b \to \lambda, c \to \lambda\}, \{baac\}), baac)$$

with a finite condition, since the only possible derivation is $baac \Longrightarrow aa$. Thus $L_9 \in \mathcal{CL}(FIN)$.

If we replace the finite condition $\{baac\}$ by the monoidal condition $\{a,b,c\}^*$, we get L_9, too (note that in this case the only derivation is $baac \Longrightarrow aa \Longrightarrow aa \Longrightarrow aa \dots$). Therefore $L_9 \in \mathcal{CL}(MON)$.

If L_9 is generated by some propagating T0L system G with regular conditions, then aa is the axiom and we have a derivation $aa \Longrightarrow baac$. This derivation requires a rule $a \to w$ with $w \neq a$. Then $aa \Longrightarrow ww$ also holds, but $ww \notin L_9$ in contrast to our assumption $L_9 = L(G)$. □

4. Hierarchy of T0L systems with subregular conditions

We start with some relations between some language families generated by conditional T0L systems with subregular conditions.

Lemma 4.1. *We have* $\mathcal{CL}(REG) = \mathcal{CL}(UF)$ *and* $\mathcal{CPL}(REG) = \mathcal{CPL}(UF)$.

Proof. It is known that any regular language is a union of finitely many union-free languages. Let

$$G = (\ V,\ (P_1, R_1),\ (P_2, R_2), \dots,\ (P_n, R_n), \omega\ )$$

be a T0L system with regular conditions. Moreover, for $1 \leq i \leq n$, let

$$R_i = R_{i,1} \cup R_{i,2} \cup \cdots \cup R_{i,r_i},$$

where $R_{i,j}$ is union-free for $1 \leq j \leq n$. It is easy to prove that the T0L system

$$(V, (P_1, R_{1,1}), \ldots, (P_1, R_{1,r_1}), (P_2, R_{2,1}), \ldots, (P_n, R_{n,1}), \ldots, (P_n, R_{n,r_n}), \omega)$$

with union-free conditions generates $L(G)$. Hence, $\mathcal{CL}(REG) \subseteq \mathcal{CL}(UF)$.

The converse inclusion follows by Lemma 2.2 and the inclusions given in the diagram of Figure 1.

Thus $\mathcal{CL}(REG) = \mathcal{CL}(UF)$.

For propagating T0L systems, we have to repeat the proof. □

Lemma 4.2. *We have*

$$\mathcal{CL}(MON) \subseteq \mathcal{CL}(COMB) \subseteq \mathcal{CL}(SUF)$$

and

$$\mathcal{CPL}(MON) \subseteq \mathcal{CPL}(COMB) \subseteq \mathcal{CPL}(SUF).$$

Proof. For any language $L \in COMB$, we have $\mathrm{Suf}(L) = L \cup \{\lambda\}$.

Let $G = (V, (P_1, R_1), (P_2, R_2), \ldots, (P_n, R_n), \omega)$ be a T0L system with conditions in $COMB$. If $z \in L(G)$, then there is a derivation

$$\omega = z_0 \Longrightarrow_{P_{i_1}} z_1 \Longrightarrow_{P_{i_2}} z_2 \Longrightarrow_{P_{i_3}} \ldots \Longrightarrow_{P_{i_n}} z_n = z \tag{4.1}$$

such that $z_{j-1} \in R_{i_j}$ for $1 \leq j \leq n$. Since $z_{i-1} \in \mathrm{Suf}(R_{i_j})$, we get that (4.1) is a derivation according to the T0L system

$$G' = (V, (P_1, \mathrm{Suf}(R_1)), (P_2, \mathrm{Suf}(R_2)), \ldots, (P_n, \mathrm{Suf}(R_n)), \omega)$$

with conditions in SUF, too.

Conversely, if

$$\omega = w_0 \Longrightarrow_{P_{i_1}} w_1 \Longrightarrow_{P_{i_2}} w_2 \Longrightarrow_{P_{i_3}} \ldots \Longrightarrow_{P_{i_n}} w_n = w$$

with $w_j \in \mathrm{Suf}(R_{i_j})$ is a derivation in G', then it is also a derivation according to G since $w_j \neq \lambda$ and $\mathrm{Suf}(R_{i_j}) = R_{i_j} \cup \{\lambda\}$.

This proves $L(G) = L(G')$. Hence any language of $\mathcal{CL}(COMB)$ is in $\mathcal{CL}(SUF)$.

Since $X^* = X^*X \cup \{\lambda\}$ and $X^*X \in COMB$ for any alphabet X, we can prove the inclusion $\mathcal{CL}(MON) \subseteq \mathcal{CL}(COMB)$ by analogous arguments.

For propagating T0L systems, we can give the same proof. □

Theorem 4.3. *The diagram given in Figure 2 holds.*

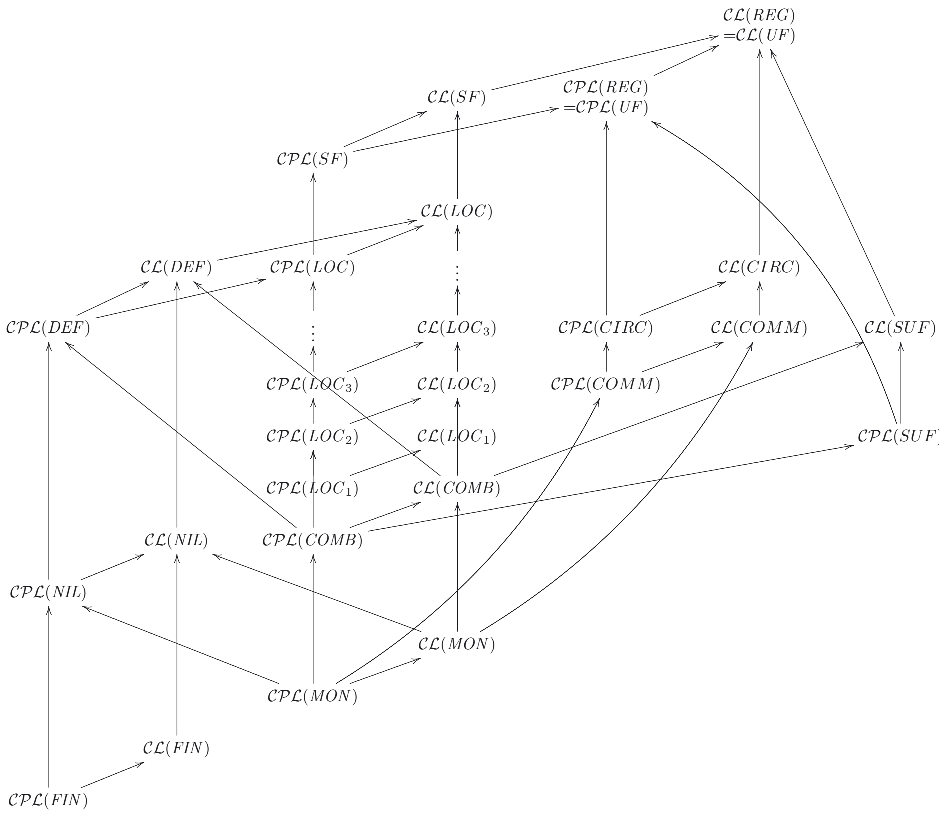

FIGURE 2. Hierarchy of language families $\mathcal{CL}(X)$ with $X \in \mathcal{G}$ (an arrow from $\mathcal{CL}(X)$ to $\mathcal{CL}(Y)$ denotes $\mathcal{CL}(X) \subset \mathcal{CL}(Y)$; and if two families are not connected by a directed path, then they are incomparable).

Proof. All inclusions of the diagram follow from Lemmas 2.2 and 4.2. By Lemma 4.1, the equalities of the diagram hold.

By Lemmas 3.2, 3.4, and 3.6, the families $\mathcal{CL}(X)$ above $\mathcal{CL}(FIN)$ or $\mathcal{CL}(LOC_k)$ and below $\mathcal{CL}(SF)$ are incomparable with $\mathcal{CL}(SUF)$.

By Lemmas 3.1 and 3.3, the families $\mathcal{CL}(X)$ above $\mathcal{CL}(FIN)$ or $\mathcal{CL}(COMB)$ and below $\mathcal{CL}(SF)$ or $\mathcal{CL}(SUF)$ are incomparable with $\mathcal{CL}(COMM)$ and $\mathcal{CL}(CIRC)$.

By Lemmas 3.6 and 3.7, $\mathcal{CL}(FIN)$, $\mathcal{CL}(NIL)$ and $\mathcal{CL}(DEF)$ are incomparable with $\mathcal{CL}(LOC_k)$, $k \geq 1$.

By Lemmas 3.5 and 3.7, $\mathcal{CL}(NIL)$ is incomparable with the families $\mathcal{CL}(COMB)$ and $\mathcal{CL}(LOC_k)$ for $k \geq 1$.

By these incomparabilities it follows that all inclusions $\mathcal{CL}(X) \subseteq \mathcal{CL}(Y)$ – except $\mathcal{CL}(FIN) \subseteq \mathcal{CL}(NIL)$, $\mathcal{CL}(LOC) \subseteq \mathcal{CL}(SF)$, $\mathcal{CL}(COMB) \subseteq \mathcal{CL}(SUF)$, and $\mathcal{CL}(COMM) \subseteq \mathcal{CL}(CIRC)$ – are proper. However, $\mathcal{CL}(FIN) \subseteq \mathcal{CL}(NIL)$ is also a proper inclusion because $\mathcal{CL}(FIN)$ contains only finite languages whereas $\mathcal{CL}(NIL)$ contains the infinite language $\{a^{2^n} | n \geq 1\}$ generated by the T0L system $(\{a\}, (\{a \to a^2\}, \{a\}^+), a^2)$ with a nilpotent condition, and the inclusions $\mathcal{CL}(LOC) \subseteq \mathcal{CL}(SF)$, $\mathcal{CL}(COMB) \subseteq \mathcal{CL}(SUF)$, and $\mathcal{CL}(COMM) \subseteq \mathcal{CL}(CIRC)$ are strict by Lemmas 3.9 and 3.10.

In the proofs of Lemmas 3.1–3.10, we have only used propagating systems to show the membership in a certain family. Thus, for all $X, Y \in \mathcal{G}$, $\mathcal{CL}(X) \subset \mathcal{CL}(Y)$ implies $\mathcal{CPL}(X) \subset \mathcal{CPL}(Y)$ and the incomparability of $\mathcal{CL}(X)$ and $\mathcal{CL}(Y)$ implies the incomparabilities of $\mathcal{CPL}(X)$ and $\mathcal{CPL}(Y)$ and of $\mathcal{CPL}(X)$ and $\mathcal{CL}(Y)$.

By Lemma 3.11, we get $\mathcal{CPL}(X) \subset \mathcal{CL}(X)$ for all $X \in \mathcal{G}$ and the incomparability of $\mathcal{CPL}(Y)$ and $\mathcal{CL}(X)$ for all $X, Y \in \mathcal{G}$ with $\mathcal{CL}(X) \subset \mathcal{CL}(Y)$. □

5. Conclusion

If we restrict to sets $\mathcal{CL}(X)$ (or $\mathcal{CPL}(X)$) with $X \in \mathcal{G}$, then we see that – except the change from the incomparabilities of *MON* and *COMB* and of *COMB* and *SUF* to the inclusions $\mathcal{CL}(MON) \subset \mathcal{CL}(COMB)$ and $\mathcal{CL}(COMB) \subset \mathcal{CL}(SUF)$ respectively, and from the inclusion $UF \subset REG$ to an equality $\mathcal{CL}(UF) = \mathcal{CL}(REG)$ – we have the same hierarchy for the subregular families (see Fig. 1) and the families obtained by T0L systems with subregular control (see Fig. 2). The first two changes come from the fact that we also have an inclusion of *MON* in *COMB* and of *COMB* in *SUF*, if we ignore the empty word, and that we can derive no word from the empty word.

Thus we have almost the same situation as for external contextual grammar where most of the relations in the hierarchy of subregular families also hold in the hierarchy of external contextual languages with subregular selection languages (see [7, 12]).

This is a strong contrast to conditional grammars, tree controlled grammars, and networks of evolutionary processors, where the corresponding hierarchies differ (for instance, in all cases, star-free, regular suffix-closed, and regular circular languages are as powerful as arbitrary regular languages; moreover, some further different subregular families lead to identical families, if used as regular restriction). In [14], we show that such a situation also holds for extended conditional T0L systems (where in the language are only words over a subset T of the underlying alphabet V; *i.e.*, the letters from $V \setminus T$ can be considered as nonterminals).

Thus it seems that the use of nonterminals leads to a change of the hierarchy whereas one gets almost the same hierarchy, if the devices do not use nonterminals.

Acknowledgements. The authors are grateful to Bianca Truthe for Lemma 3.4 by which an open problem in an earlier version could be solved. Moreover, we thank Bianca Truthe and an unknown referee for their comments.

References

[1] J. Castellanos, C. Martín-Vide, V. Mitrana and J.M. Sempere, Solving NP-Complete Problems With Networks of Evolutionary Processors. *IWANN'01: Proc. of the 6th International Work-Conference on Artificial and Natural Neural Networks*. Vol. 2084 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2001) 621–628.

[2] E. Csuhaj-Varjú and A. Salomaa, Networks of Parallel Language Processors. *New Trends in Formal Languages*. Vol. 1218 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (1997) 299–318.

[3] K. Čulik II and H.A. Maurer, Tree controlled grammars. *Comput.* **19** (1977) 129–139. *New Trends in Formal Languages – Control, Cooperation, and Combinatorics*. Vol. 1218 of *Lect. Notes Comput. Sci.* Springer-Verlag Berlin (1997) 299–318.

[4] J. Dassow, Subregularly controlled derivations: the context-free case. *Rostocker Mathematisches Kolloquium* **34** (1988) 61–70.

[5] J. Dassow, Conditional grammars with restrictions by syntactic parameters. *Words, Semigroups, Transductions,* edited by M. Ito, Gh. Păun and Sh. Yu. World Scientific, Singapore (2001) 59–68.

[6] J. Dassow, Subregularly controlled derivations: restrictions by syntactic parameters. *Where Math., Comput. Sci., Linguistics and Biology Meet*. Kluwer Academic Publishers (2001) 51–61.

[7] J. Dassow, Contextual grammars with subregular choice. *Fundamenta Informaticae* **64** (2005) 109–118.

[8] J. Dassow, Grammars with commutative, circular, and locally testable conditions. *Automata, Formal Languages, and Related Topics – Dedicated to Ferenc Gécseg on the occasion of his 70th birthday*. University of Szeged (2009) 27–37.

[9] J. Dassow and U. Fest, On regulated L systems. *Rostock. Math. Kolloq.* **25** (1984) 99–118.

[10] J. Dassow and H. Hornig, Conditional grammars with subregular conditions, in *Proc. Internat. Conf. Words, Languages and Combinatorics II*. World Scientific, Singapore (1994) 71–86.

[11] J. Dassow, F. Manea and B. Truthe, Networks of evolutionary processors with subregular filters, in *Languages and Automata Theory and Applications*. Vol. 6638 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2011) 262–273.

[12] J. Dassow, F. Manea and B. Truthe, On Contextual Grammars with Subregular Selection Languages, in *Descriptional Complexity of Formal Systems*. Vol. 6808 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2011) 135–146.

[13] J. Dassow and Gh. Păun. *Regulated Rrewriting in Formal Language Theory*. Springer-Verlag, Berlin (1989).

[14] J. Dassow and St. Rudolf, Conditional Lindenmayer systems with subregular conditions: the extended case (Submitted).

[15] J. Dassow, R. Stiebe and B. Truthe, Two collapsing hierarchies of subregularly tree controlled languages. *Theoretical Comput. Sci.* **410** (2009) 3261–3271.

[16] J. Dassow, R. Stiebe and B. Truthe, Generative capacity of subregularly tree controlled grammars. *Int. J. Foundations Comput. Sci.* **21** (2010) 723–740.

[17] J. Dassow and B. Truthe, On networks of evolutionary processors with filters accepted by two-state-automata. *Fundamenta Informaticae* **113** (2011) 1–14.

[18] I. Fris, Grammars with partial ordering. *Information and Control* **12** (1968) 415–425.

[19] S. Ginsburg and E.H. Spanier, Control sets on grammars. *Math. Syst. Theory* **2** (1968) 159–177.

[20] F. Gécseg and I. Peak, *Algebraic Theory of Automata*. Academiai kiado, Budapest (1972).

[21] A. Gill and L.T. Kou, Multiple-entry finite automata. *J. Comput. Syst. Sci.* **9** (1974) 1–19.

[22] Y.-S. Han, K. Salomaa and D. Wood, Nondeterministic state complexity of basic operations for prefix-suffix-free regular languages. *Fundamenta Informaticae* **90** (2009) 93–106.

[23] I.M. Havel, The theory of regular events II. *Kybernetika* **5** (1969) 520–544.

[24] S. Istrail, Gramatici contextuale cu selectiva regulata. *Stud. Cerc. Mat.* **30** (1978) 287–294.
[25] F. Manea and B. Truthe, Accepting Networks of Evolutionary Processors with Subregular Filters, in *Automata and Formal Languages – 13th International Conference AFL 2011*. College of Nyíregyháza (2011) 300–314.
[26] F. Manea and B. Truthe, On internal contextual grammars with subregular selection languages, in *Descriptional Complexity of Formal Systems*. Vol. 7386 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2012) 222–235.
[27] S. Marcus, Contextual grammars. *Revue Roum. Math. Pures Appl.* **14** (1969) 1525–1534.
[28] C. Martín-Vide and V. Mitrana, Networks of Evolutionary Processors: Results and Perspectives, in *Molecular Computational Models: Unconventional Approaches* (2005) 78–114.
[29] R. McNaughton and S. Papert, *Counter-Free Languages*. M.I.T. Press (1971).
[30] M. Perles, M.M. Rabin and E. Shamir, The theory of definite automata. *IEEE Trans. Electronic Comput.* **12** (1963) 233–243.
[31] G. Păun, *Marcus Contextual Grammars*. Kluwer Publ. House, Doordrecht (1998).
[32] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems*. Academic Press, New York (1980).
[33] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages*. Springer-Verlag, Berlin (1997).
[34] G. Rozenberg and S.H. von Solms, Priorities on context conditions in rewriting systems. *Inform. Sci.* **14** (1978) 15–51.
[35] A. Salomaa, *Formal Languages*. Academic Press, New York (1973).
[36] H.J. Shyr, *Free Monoids and Languages*. Hon Min Book Co., Taichung, Taiwan (1991).
[37] H.J. Shyr and G. Thierrin, Ordered automata and associated languages. *Tamkang J. Math.* **5** (1974) 9–20.
[38] P.H. Starke, *Abstrakte Automaten*. Deutscher Verlag der Wissenschaften, Berlin (1969).
[39] B. Wiedemann, *Vergleich der Leistungsfähigkeit endlicher determinierter Automaten*. Diplomarbeit, Universität Rostock (1978).

Communicated by M. Holzer.
Received February 2, 2013. Accepted March 14, 2014.