Acta Cybernetica **22** (2015) 313–334.

Conditional Lindenmayer Systems with Subregular Conditions: The Extended Case

Jürgen Dassow* and Stefan Rudolf†

In memorian Ferenc Gécseg, a pioneer in Theoretical Computer Science

Abstract

We study the generative power of extended conditional Lindenmayer systems where the conditions are finite, monoidal, combinational, definite, nilpotent, strictly locally (k)-testable, commutative, circular, suffix-closed, star-free, and union-free regular languages. The results correspond to those obtained for conditional context-free languages.

Keywords: Lindenmayer systems, controlled derivations, subregular conditions

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, conditional context-free grammars, tree controlled context-free grammars, networks of evolutionary processors with regular filters, and contextual grammars with selection languages (for details see [4], [16], [13], and [14]).

In these cases the process of checking the condition given by a regular language is now very simple and efficient, however, the increase of generative power is considerable (for instance, for the first four devices mentioned above, one has an increase from context-free languages to recursively enumerable languages). Since on the one hand practical requirements 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

*Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, PSF 4120, D-39016 Magdeburg, Germany, E-mail: dassow@iws.cs.uni-magdeburg.de

†Fliederweg 7b, D–65527 Niedernhausen, Germany

DOI: 10.14232/actacyb.22.2.2015.5

for the control. The effect of using subregular languages defined by combinatorial and algebraic properties to the generative power was investigated in the last two decades (see e. g. [1], [3], and [6]).

In 1968, A. Lindenmayer introduced a new type of formal grammars and languages in order to describe the development of organisms. We refer to [15] as a monograph on Lindenmayer systems and languages.

Also in this area it is necessary to restrict the applicability of tables by biological reasons (e.g. in order to model the change of the seasons, the different development if water is present or not etc.). In a conditional Lindenmayer system a table P can only be applied to a sentential form w if w belongs to a language associated to P. A first variant of such systems was studied in [17].

In [5] the systematic study of conditional Lindenmayer systems where the languages associated to the tables belong to some family of subregular languages was started. In [5], the case of non-extended Lindenmayer systems was investigated. In this paper we continue by the consideration of extended conditional Lindenmayer systems with subregular conditions. We prove that propagating extended conditional Lindenmayer systems with suffix-closed, union-free, star-free, circular, or strictly locally k-testable (for $k \geq 2$) conditions allow further characterizations of the family of context-sensitive languages, whereas the use of monoidal, combinational, definite, nilpotent and strictly-locally 1-testable languages as conditions does not lead to an increase of the power, i. e., one obtains the family of ET0L languages; systems with commutative conditions are as powerful as non-erasing matrix grammars (with appearance checking). For arbitrary Lindenmayer systems (with erasing rules) one gets characterizations of the family of recursively enumerable languages, if the conditions are suffix-closed, union-free, star-free, circular, strictly locally k-testable (for $k \geq 1$), or commutative; for the other families of subregular languages the place in the hierarchy is not determined completely.

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 [16], [15], and [4].

The inclusion of the set X in the set Y is denoted by $X \subseteq Y$. If the inclusion is strict, we write $X \subset Y$.

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 a letter a, we write $\#_a(w)$ instead of $\#_{\{a\}}(w)$.

Let $V = \{a_1, a_2, \ldots, a_n\}$ (with a fixed order of the letters of V). Then, for a word $w \in V^*$, we define the Parikh vector $\pi_V(w)$ of w by

$$\pi_V(w) = (\#_{a_1}(w), \#_{a_2}(w), \dots, \#_{a_n}(w)).$$

For a language L over V, we set

$$\pi_V(L) = \{\pi_V(w) \mid w \in L\}.$$

A language L over V is called semi-linear if $\pi_V(L)$ is a finite union of sets of the form

$$\{(a_1, a_2, \dots, a_n) + \sum_{j=1}^{p} \alpha_j (b_{1,j}, b_{2,j}, \dots, b_{n,j}) \mid \alpha_j \in \mathbb{N}\}.$$

If we consider a primed version $V' = \{a' \mid a \in V\}$ of some alphabet V, then, for a word $w = a_1 a_2 \dots a_m$ with $a_i \in V$ for $1 \leq i \leq m$, we set $w' = a'_1 a'_2 \dots a'_m$. Moreover, if U is a subset of V, then we set $U' = \{a' \mid a \in U\}$. Analogous notation we also use for double primed versions of V, etc.

In this paper two languages L_1 and L_2 are considered as equal if they differ at most in the empty word, i. e., $L_1 \setminus \{\lambda\} = L_2 \setminus \{\lambda\}$.

The families of finite, regular, context-free, context-sensitive, and recursively enumerable languages are denoted by *FIN*, *REG*, *CF*, *CS*, and *RE*, respectively.

2.1 Matrix Grammars and Languages

Matrix grammars are an important representant of grammars with controlled derivations. They are equivalent to many other such devices. We recall their definition since we shall show that also some extended conditional Lindenmayer systems are equivalent to matrix grammars.

A *matrix grammar* is a quintuple $G = (N, T, M, S, Q)$ where

- N and T are disjunct alphabets of nonterminals and terminals,
- $M = \{m_1, m_2, \dots, m_r\}$ is a finite set of finite sequences m_i of context-free rules, i. e.,

  $$m_i = (A_{i,1} \to v_{i,1}, A_{i,2} \to v_{i,2}, \dots A_{i,r_i} \to v_{i,r_i})$$

  for $1 \leq i \leq r$ (the elements of M are called matrices),
- S is an element of N, and
- Q is a subset of the productions occurring in the matrices of M

The application of a matrix m_i is defined as a sequential application of the rules of m_i in the given order where a rule of Q can be ignored if its left-hand side does not occur in the current sentential form, i.e., $x \Longrightarrow_{m_i} y$ holds iff there are words w_j, $1 \leq j \leq r_i + 1$, such that $x = w_1$, $y = w_{r_i+1}$ and, for $1 \leq j \leq r_i$,

$$w_j = x_j A_{i,j} y_j \text{ and } w_{j+1} = x_j v_{i,j} y_j$$

or

$$w_j = w_{j+1} \text{ and } A_{i,j} \text{ does not occur in } w_j \text{ and } A_{i,j} \to v_{i.j} \in Q.$$

The language $L(G)$ generated by G consists of all words $z \in T^+$ such that there is a derivation

$$S \Longrightarrow_{m_{i_1}} v_1 \Longrightarrow_{m_{i_2}} v_2 \Longrightarrow_{m_{i_3}} \ldots \Longrightarrow_{m_{i_t}} v_t = z$$

for some $t \geq 1$.

By MAT^λ and MAT we denote the families of languages generated by matrix grammars and matrix grammars without erasing rules, respectively.

It is well-known that

$$CF \subset MAT \subset CS \subset RE = MAT^\lambda.$$

2.2 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

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

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 $L = V^*$,

For more details on languages of the types defined above we refer to [19], [11], and [18].

In [2], it was shown that a regular language $R \subset V^*$ is commutative if and only if there is a semi-linear set M and $R = \pi_V^{-1}(M)$.

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.

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 $a_1 a_2 \ldots a_n$ with $n \geq k$ and $a_i \in V$,

$1 \leq i \leq n$, belongs to L iff $a_1a_2\ldots a_k \in A$, $a_{j+1}a_{j+2}\ldots a_{j+k} \in B$ for $1 \leq j \leq n-k-1$, and $a_{n-k+1}a_{n-k+2}\ldots a_n \in C$. 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 [2]).

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

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

The relations between families of $\mathcal{G}$ are investigated e. g. in [12] and [20] and their set-theoretic relations are given in Figure 1.

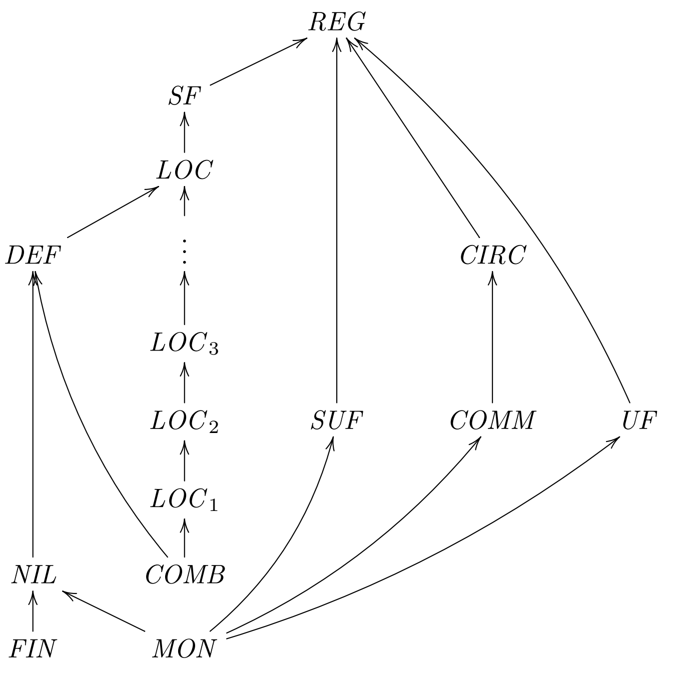

Figure 1: Hierarchy of subregular languages (an arrow from X to Y denotes $X \subset Y$, and if two families are not connected by a directed path then they are incomparable)

Representations of definite automata and definite and nilpotent tree automata and languages were studied by Ferenc Gécseg and coauthors in [7], [8], [9], and [10].

2.3 Extended Conditional Lindenmayer Systems

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

An *extended tabled Lindenmayer system without interaction* (*ET0L system*, for short) is an $(r+3)$-tuple $H = (V, T, P_1, P_2, \ldots, P_r, w)$, where
- V is an alphabet, T is a subset of V,
- for $1 \leq i \leq r$, 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, $1 \leq i \leq r$, 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 $w \neq a$ 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
- $x = a_1 a_2 \ldots a_n$ with $a_i \in V$ for $1 \leq i \leq n$,
- $y = y_1 y_2 \ldots y_n$,
- $a_i \to y_i \in P_j$ for $1 \leq i \leq n$ and some j, $1 \leq j \leq r$.

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

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

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

An ET0L system is called *propagating* if no table contains a rule $a \to \lambda$.

By *ET0L* and *EPT0L*, we denote the families of all languages generated by ET0L systems and propagating ET0L systems, respectively.

It is well-known that the following relation holds

$$CF \subset EPT0L = ET0L \subset MAT.$$

Definition 1. *A conditional ET0L system is an $(n+3)$-tuple*

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

where
- *$H' = (V, T, P_1, P_2, \ldots, P_n, w)$ is an ET0L 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, $1 \leq j \leq n$
- *$x = a_1 a_2 \ldots a_t$ with $a_i \in V$ for $1 \leq i \leq t$,*
- *$y = y_1 y_2 \ldots y_t$,*
- *$a_i \to y_i \in P_j$ for $1 \leq i \leq t$, and*
- *$x \in R_j$.*

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

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

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

By definition, in a condition ET0L system, a table P_j is only applicable to a sentential form x, if x belongs to the conditional language R_j associated with P_j.

Example 1. We consider the ET0L system

$$H = (V, \{a, b\}, (P_1, R_1), (P_2, R_2), (P_3, R_3), (P_4, R_4), (P_5, R_5), SD)$$

with

$$\begin{aligned} V &= \{S, A, B_1, B_2, C, D, a, b\}, \\ (P_1, R_1) &= (\{S \to ASC\}, V^*\{D\}), \\ (P_2, R_2) &= (\{S \to AC, D \to \lambda\}, V^*\{D\}), \\ (P_3, R_3) &= (\{A \to Ab, C \to B_1, C \to B_2\}, V^*\{C\}), \\ (P_4, R_4) &= (\{B_1 \to \lambda, B_2 \to C\}, V^*\{B_1\}), \\ (P_5, R_5) &= (\{A \to a\}, V^*\{b\}). \end{aligned}$$

We start with SD, have to apply sometimes P_1 and then once P_2 (the only rules where the words in the associated language end with D). This yields $A^n C^n$. Now we have to apply P_3 and get $(Ab)^n z$ where z is a word of length n over $\{B_1, B_2\}$. If z ends with B_2, then the derivation cannot be continued. If B_1 is the last letter of z, we can only apply P_4 and obtain $(Ab)^n C^r$ with $r < n$ (since we cancel at least the last letter of z). This process can be iterated, in each step we add a letter b after each A, and cancel at least one C. Finally, we get $(Ab^m)^n$ with $m \leq n$ (m gives the number of iterations, for which $1 \leq m \leq n$ holds). Now, by the use of P_5 we get $(ab^m)^n$ with $n \geq 1$ and $1 \leq m \leq n$. Thus

$$L(H) = \{(ab^m)^n \mid 1 \leq m \leq n\}.$$

We note that it is well-known that $L(H)$ cannot be generated by an ET0L system.

In this paper, we study the generative power of conditional ET0L systems, if one restricts to a class of subregular languages. For $X \in \mathcal{G}$, we define $\mathcal{CEL}(X)$ and $\mathcal{CEPL}(X)$ as the families of all languages which can be generated by conditional ET0L and conditional propagating ET0L system $(V, T, (P_1, R_1), \ldots, (P_n, R_n), w)$, where all languages R_i, $1 \leq i \leq n$, are in X.

By these definitions, the language from Example 1 is in $\mathcal{CEL}(COMB)$.

The following relations follow immediately from the definitions.

Lemma 1. *i) For all $X, Y \in \mathcal{G}$ with $X \subseteq Y$,*

$$\mathcal{CEL}(X) \subseteq \mathcal{CEL}(Y), \mathcal{CEPL}(X) \subseteq \mathcal{CEPL}(Y), \text{ and } \mathcal{CEPL}(X) \subseteq \mathcal{CEL}(X).$$

3 Some Equalities and Inclusions

In this section we prove inclusions $\mathcal{CEL}(X) \subseteq \mathcal{CEL}(Y)$ ($\mathcal{CEPL}(X) \subseteq \mathcal{CEPL}(Y)$) and equalities $\mathcal{CEL}(X') = \mathcal{CEL}(Y')$ ($\mathcal{CEPL}(X') = \mathcal{CEPL}(Y')$) for some families X, Y, X', and Y', respectively.

Lemma 2. *$\mathcal{CEL}(REG) = \mathcal{CEL}(UF)$ and $\mathcal{CEPL}(REG) = \mathcal{CEPL}(UF)$.*

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

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

be a conditional ET0L system with regular conditions. Moreover, for $1 \le i \le n$, 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 $1 \le j \le n$. It is easy to prove that the ET0L system

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

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

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

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

For propagating ET0L systems, we have to repeat the proof. □

Lemma 3. *$\mathcal{CEL}(REG) \subseteq RE$ and $\mathcal{CEPL}(REG) \subseteq CS$.*

Proof. Let $L \in \mathcal{CEL}(REG)$, and let $G = (V, T, (P_1, R_1)(P_2, R_2), \ldots, (P_n, R_n), \omega)$ be a conditional ET0L system with regular conditions generating L. Then we construct a Turing machine M which works as follows (the detailed description of M is left to the reader):

(1) M checks whether ω is the word on the tape. If this is the case, M accepts; otherwise, it continues with (2).

(2) M chooses an i, $1 \le i \le n$, remembers i in the state, and chooses a decomposition $w = w_1 w_2 \ldots w_m$ of the tape content w; this can be done by writing $w_1 \# w_2 \# \ldots w_{m-1} \# w_m$ at the tape.

(3) M replaces each w_j $1 \le j \le m$, by some a_j where $a_j \to w_j \in P_i$ (if w_j is the empty word, this means that a_j with $a_j \to \lambda \in P_i$ is inserted) and cancels all symbols $\#$. If M can perform this step (i. e., the tape content $w_1 w_2 \ldots w_m$ is changed to $a_1 a_2 \ldots a_m$), it continues with (4); otherwise, M stops without accepting.

(4) M checks whether $a_1 a_2 \ldots a_m \in R_i$. In the affirmative case, M continues with (1); otherwise, M stops without accepting.

It is easy to see that a word w is accepted by M if and only if

$$w \vdash_{i_1} w_1 \vdash_{i_2} w_2 \vdash_{i_3} \cdots \vdash_{i_{q-1}} w_{q-1} \vdash_{i_q} \omega,$$

where $\vdash_i$ stands for applying (1) to (4) with i chosen in (2), if and only if

$$\omega \Longrightarrow_{P_{i_q}} w_{q-1} \Longrightarrow_{P_{i_{q-1}}} w_{q-2} \Longrightarrow_{P_{i_{q-2}}} \cdots \Longrightarrow_{P_{i_2}} w_1 \Longrightarrow_{P_{i_1}} w$$

$\omega \in R_{i_q}$, $w_k \in R_{i_k}$ for $1 \le k \le q-1$, and $w \in T^*$ if and only if $w \in L(G)$. Therefore the language accepted by M is $L(G)$. Thus $\mathcal{CEL}(REG) \subseteq RE$.

If the conditional T0L system G is propagating, each w_j of the decomposition is non-empty. Then it follows easily that the maximal length of the tape contents is twice the length of the input word. Hence the Turing machine is a linearly bounded automaton, which implies $\mathcal{CEPL}(REG) \subseteq CS$. $\square$

Lemma 4. $RE \subseteq \mathcal{CEL}(LOC_1)$.

Proof. Let $L \in RE$. Then there is a grammar $G = (N, T, P, S)$ in Kuroda normal form (i. e., all rules have one of the following forms: $A \to BC$, $A \to a$, $A \to \lambda$, or $AB \to CD$ with $A, B, C, D \in N$ and $a \in T$) which generates L. Let P_1 be the set of all rules of P of the form $AB \to CD$ and $P_2 = P \setminus P_1$.

Let S'' and $\#$ be additional symbols not in $N \cup T$. We set

$$\begin{aligned}
V' &= \{a' \mid a \in N \cup T\} \cup \{\#\},\\
V_p &= \{a_p \mid a \in N \cup T\} \text{ for } p \in P_2,\\
V_p &= \{a_p \mid a \in N \cup T\} \cup \{a'_p \mid a \in N \cup T\} \text{ for } p \in P_1,\\
V_r &= \{a_r \mid a \in N \cup T \cup \{\#\}\} \cup \{a'_r \mid a \in N \cup T \cup \{\#\}\},\\
V &= \{S''\} \cup N \cup T \cup V' \cup V_r \cup \bigcup_{p \in P} V_p.
\end{aligned}$$

We now construct a conditional ET0L system H as follows: The basic and terminal alphabet are V and T, and the axiom is S''. Now we give all tables and conditions (if no rule is mentioned for some letter a, then $a \to a$ is the only rule for a in the production set):

$$(P_1, R_1) = (\{S'' \to \#S'\}, \{S''\}^+)$$

(we introduce from the axiom the word $\#S'$; the symbol $\#$ remembers the beginning of the word, because we shall use circular versions of a word; S' is the primed version of the start symbol of G),

$$\begin{aligned}
(P_2, R_2) = (&\bigcup_{a \in N \cup T} \{a' \to a', a' \to a_r, a' \to a'_r\} \cup \bigcup_{a \in N \cup T} \bigcup_{p \in P_1} \{a' \to a_p\}\\
&\cup \bigcup_{a \in N \cup T} \bigcup_{p \in P_2} \{a' \to a_p, a' \to a'_p\} \cup \{\# \to \#, \# \to \#_r, \# \to \#'_r\},\\
&V'(V')^+),
\end{aligned}$$

(given a word over V', we can change some letters a' to their versions a_r and a'_r or their versions associated with a rule p; looking at the conditions of the tables defined below, the obtained word can only be handled if the changes are only done for the last letter or last and first letters and the introduced versions have to fit; more precise, from $x'w'y'$ with $x', y' \in V'$ and $w' \in (V')^*$, we can derive only $x'w'y_p$ with $p \in P_1$, or $x_p w' y'_p$ with $p \in P_2$, or $x_r w' y'_r$),

$$(P_{r,a,b}, R_{r,a,b}) = (\{a_r \to b'a', b'_r \to \lambda\}, \{a_r\}(V')^*\{b'_r\}) \text{ for } a, b \in V'$$

(if we obtained $a_r w' b'_r$ from $a'w'b'$, we now derive $b'a'w'$, i. e., we have performed a rotation step from $a'wb'$ to $b'a'w'$),

$$(P_p, R_p) = (\{A_p \to B'C'\}, (V')^+\{A_p\}) \text{ for } p = A \to BC$$

(if we obtained $x'w'A_p$, then we obtain $x'w'B'C'$, i. e., we have simulated an application $A \to BC$)

$$(P_p, R_p) = (\{A_p \to a'\}, (V')^+\{A_p\}) \text{ for } p = A \to a$$

(if we obtained $x'w'A_p$, then we obtain $x'w'a'$, i. e., we have simulated an application $A \to a$)

$$(P_p, R_p) = (\{A_p \to \lambda\}, (V')^+\{A_p\}) \text{ for } p = A \to \lambda$$

(if we obtained $x'w'A_p$, then we obtain $x'w'$, i. e., we have simulated an application $A \to \lambda$)

$$(P_p, R_p) = (\{B_p \to D', A'_p \to C'\}, \{B_p\}(V')^+\{A'_p\}) \text{ for } p = AB \to CD$$

(if we obtained $B_p w' A_p$, then we obtain $D'w'C'$, i. e., we have simulated an application $AB \to CD$ up to some rotation),

$$(P_3, R_3) = (\{a' \to a \mid a \in T\} \cup \{\# \to \lambda\}, \{\#\}T^*)$$

(if we have a word $\#x'$ with $x \in T$, then we can derive x).

We now prove that $L(G) \subseteq L(H)$. The basic idea is to start with $\#S'$ (produced by one application of P_1 to the axiom S''), perform circular shifts on a sentential form getting words of the form $x'_{r+1}x'_{r+2}\dots x'_n\#x'_1x'_2\dots x'_r$ and simulate the application of a rule in G by applying some table which only changes the last (if the rule is in P_1) or first and last letter (which are neighbouring letters in the non-rotated word, if the rule is from P_2), and to finish by a cancellation of $\#$ and returning to non-primed letters. Thus we can generate in H any word $w \in T^*$ which can be generated by G.

The converse inclusion $L(H) \subseteq L(G)$ holds, since we can perform only the rotation steps, or simulations of rules of P, or a cancellation of the primes, if we have a terminal word.

Since all the conditions of H are in LOC_1, the statement follows. □

Lemma 5. *$CS \subseteq \mathcal{CEPL}(LOC_2)$.*

Proof. Let $L \in CS$. Then there is a context-sensitive grammar $G = (N, T, P, S)$ in Kuroda normal form, i. e., all rules have the form $A \to B$, $A \to BC$, $AB \to CD$, and $A \to a$ with $A, B, C, D \in N$ and $a \in T$, such that $L = L(G)$. Let $p_1, p_2, \dots, p_r$ be the rules of P which have the form third mentioned form. For each rule $p_i = A_iB_i \to C_iD_i$, we introduce new letters A'_i and B'_i such that $A'_i \neq B'_j$ for $1 \leq i, j \leq r$ and $A'_i \neq A'_j$ and $B'_i \neq B'_j$ for $1 \leq i, j \leq r$, $i \neq j$. Let

$$V' = \{A'_i \mid 1 \leq i \leq n\} \cup \{B'_i \mid 1 \leq i \leq n\}.$$

Then we define the conditional ET0L system

$$H = (N \cup T \cup V',\ T,\ (Q,R),\ (Q',R'),\ S)$$

with

$$\begin{aligned} Q =& \{X \to X \mid X \in N \cup T \cup V'\} \cup \{A \to w \mid A \to w \in P\} \\ & \cup \{A_i \to A_i' \mid p_i = A_iB_i \to C_iD_i, 1 \le i \le n\} \\ & \cup \{B_i \to B_i' \mid p_i = A_iB_i \to C_iD_i, 1 \le i \le n\} \\ R =& (N \cup T)^* \\ Q' =& \{X \to X \mid X \in N \cup T\} \cup \bigcup_{i=1}^{r} \{A_i' \to C_i, B_i' \to D_i\} \\ R' =& (N \cup T \cup \bigcup_{i=1}^{r} \{A_i'B_i'\})^*. \end{aligned}$$

It is easy to see that the conditions R and R' belong to LOC_2.

We now prove that $L(H) = L(G)$.

Let $u \in (N \cup T)^+$ be a sentential form of G. Let $u \Longrightarrow v$ using some rule r of P which is different from all p_i, $1 \le i \le r$. Then $u = u_1Au_2$, $y = u_1wu_2$, and $r = A \to w$. This derivation can be simulated by a derivation according to table Q using $X \to X$ for all letters in u_1 and u_2 and $A \to w$ for A in the special position. If a rule $r_i = A_iB_i \to C_iD_i$ is applied to u we get $u = v_1A_iB_iv_2 \Longrightarrow v_1C_iD_iv_2 = y$. This derivation can be simulated in a two-step derivation

$$u = v_1A_iB_iv_2 \Longrightarrow_Q v_1A_i'B_i'v_2 \Longrightarrow_{Q'} v_1C_iD_iv_2 = y$$

where $X \to X$ from Q and Q' are applied to the letters of v_1 and v_2. Since G as well as H start with the axiom S, it is clear that $L(G) \subseteq L(H)$.

Assume that $x \in (N \cup T)^+$ is a sentential form of H. Then the application of Q' does not change x. Thus we have to apply Q. Let $x \Longrightarrow_Q y$. If y contains a letter A_i, then its successor in y is B_i since we cannot continue the derivation, otherwise (by the definition of R'). Let us assume without loss of generality (only the positions of the letters X_i, $1 \le i \le n$, and the subwords $A_{i_j}B_{i_j}$, $1 \le j \le m$, can occur in another order) that

$$x = u_1X_1u_2X_2u_3X_3 \ldots u_nX_nv_1A_{i_1}B_{i_1}v_2A_{i_2}B_{i_2}v_3 \ldots v_mA_{i_m}B_{i_m}v_{m+1}.$$

If $X \to X$ is applied to all letters of the words u_i, $1 \le i \le n$, and v_j, $1 \le j \le m+1$, $X_i \to w_i \in P$ is applied to all letters X_i, $1 \le i \le n$, and $A_{i_j} \to A_{i_j}'$ and $B_{i_j} \to B_{i_j}'$ are applied to the letters A_{i_j} and B_{i_j}, $1 \le j \le m$, we get

$$x \Longrightarrow_Q u_1w_1u_2w_2u_3w_3 \ldots u_nw_nv_1A_{i_1}'B_{i_1}'v_2A_{i_2}'B_{i_2}'v_3 \ldots v_mA_{i_m}'B_{i_m}'v_{m+1} = \overline{y}.$$

If $n \ge 1$ and $m \ge 1$, we have to apply Q' and get

$$\overline{y} \Longrightarrow_{Q'} u_1w_1u_2w_2u_3w_3 \ldots u_nw_nv_1C_{i_1}D_{i_1}v_2C_{i_2}D_{i_2}v_3 \ldots v_mC_{i_m}D_{i_m}v_{m+1} = y.$$

Since we have the derivation

$$\begin{aligned} x &= u_1X_1u_2X_2u_3X_3\dots u_nX_nv_1A_{i_1}B_{i_1}v_2A_{i_2}B_{i_2}v_3A_{i_3}B_{i_3}v_4\dots v_mA_{i_m}B_{i_m}v_{m+1} \\ &\Longrightarrow u_1w_1u_2X_2u_3X_3\dots u_nX_nv_1A_{i_1}B_{i_1}v_2A_{i_2}B_{i_2}v_3A_{i_3}B_{i_3}v_4\dots v_mA_{i_m}B_{i_m}v_{m+1} \\ &\Longrightarrow u_1w_1u_2w_2u_3X_3\dots u_nX_nv_1A_{i_1}B_{i_1}v_2A_{i_2}B_{i_2}v_3A_{i_3}B_{i_3}v_4\dots v_mA_{i_m}B_{i_m}v_{m+1} \\ &\quad\dots \\ &\Longrightarrow u_1w_1u_2w_2u_3w_3\dots u_nw_nv_1A_{i_1}B_{i_1}v_2A_{i_2}B_{i_2}v_3A_{i_3}B_{i_3}v_4\dots v_mA_{i_m}B_{i_m}v_{m+1} \\ &\Longrightarrow u_1w_1u_2w_2u_3w_3\dots u_nw_nv_1C_{i_1}D_{i_1}v_2A_{i_2}B_{i_2}v_3A_{i_3}B_{i_3}v_4\dots v_mA_{i_m}B_{i_m}v_{m+1} \\ &\Longrightarrow u_1w_1u_2w_2u_3w_3\dots u_nw_nv_1C_{i_1}D_{i_1}v_2C_{i_2}D_{i_2}v_3A_{i_3}B_{i_3}v_4\dots v_mA_{i_m}B_{i_m}v_{m+1} \\ &\quad\dots \\ &\Longrightarrow u_1w_1u_2w_2u_3w_3\dots u_nw_nv_1C_{i_1}D_{i_1}v_2C_{i_2}D_{i_2}v_3\dots v_mC_{i_m}D_{i_m}v_{m+1} = y, \end{aligned}$$

the derivation $x \Longrightarrow^* y$ in H can be simulated in G. If $n = 0$ or $m = 0$, we get analogously simulations. Thus $L(H) \subseteq L(G)$, too. □

Corollary 1. *$\mathcal{CEL}(REG) = RE$ and $\mathcal{CEPL}(REG) = CS$.*

Proof. By Lemmas 1, 4, 5, and 3,

$$RE \subseteq \mathcal{CEL}(LOC_1) \subseteq \mathcal{CEL}(REG) \subseteq RE$$

and

$$CS \subseteq \mathcal{CEPL}(LOC_2) \subseteq \mathcal{CEPL}(REG) \subseteq CS,$$

from which the statement immediately follows. □

Lemma 6. *$RE = \mathcal{CEL}(SUF)$ and $CS = \mathcal{CEPL}(SUF)$.*

Proof. i) Let $L \in RE$. Then, by Corollary 1, $L = L(G)$ for some ET0L system

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

with regular conditions. Let $V' = \{a' \mid a \in V\}$, and let S, F, and $\#$ be additional symbols. Then we set

$$\begin{aligned} &P_{init} = \{S \to \#\omega'\} \cup \{a' \to a' \mid a' \in V' \cup \{\#, F\}\} \cup \{a \to F \mid a \in V\} \\ &\qquad \text{and } R_{init} = \{S, \lambda\}, \\ &\overline{P_i} = \{a' \to w' \mid a \to w \in P_i, a \in V\} \cup \{a \to a \mid a \in \{S, \#, F\}\} \cup \{a \to F \mid a \in V\} \\ &\qquad \text{and } \overline{R_i} = \mathrm{Suf}(\{\#z' \mid z \in R_i\}) \text{ for } 1 \le i \le n, \\ &P_{fin} = \{\# \to \lambda\} \cup \{a' \to a \mid a \in T\} \cup \{a' \to F \mid a' \in (V' \setminus T') \cup V \cup \{S, F\}\} \\ &\qquad \text{and } R_{fin} = \mathrm{Suf}(\{\#\}T^*) \end{aligned}$$

and consider the conditional ET0L system

$$H = (V \cup V' \cup \{S, F, \#\}, T, (P_{init}, R_{init}), (\overline{P_1}, \overline{R_1}), \dots (\overline{P_n}, \overline{R_n}), (P_{fin}, R_{fin}), S).$$

Any derivation in H starts with $S \Longrightarrow \#\omega$ and in the sequel P_{init} cannot be applied. Moreover, by the definition of the production sets of H, any derivable word – except S – has the form $\#z'$ for some $z \in V^*$ or z with $z \in T^*$ or it contains at least one letter F. A set $\overline{P_i}$ is applicable to $\#z'$ if and only if $z \in R_i$, and its application yields $\#u'$ if and only if $z \Longrightarrow_{P_i} u$ holds in G. Furthermore, $\#z' \Longrightarrow z$ if and only if $z \in T^*$ by application of R_{fin}. From elements of $z \in T^*$ we obtain a word consisting only of Fs. If a word x contains an occurrence of F, then all words derivable from x contain an F, too; hence we cannot terminate the derivation. Now it follows easily that $L = L(H)$. Thus we have $RE \subseteq \mathcal{CEL}(SUF)$.

The converse inclusion follows from the relation $\mathcal{CEL}(SUF) \subseteq \mathcal{CEL}(REG) = RE$ by Lemma 1 and Corollary 1.

ii) Let $L \in CS$ and $T = \mathrm{alph}(L)$. Moreover,

$$L = \bigcup_{a \in T} \{a\} L_a \text{ where } L_a = \{w \mid aw \in L_2\}.$$

Let

$$\begin{aligned} T_1 &= \{a \mid a \in T, L_a = \emptyset\}, \\ T_2 &= \{a \mid a \in T, L_a = \{\lambda\}\}, \\ T_3 &= \{a \mid a \in T, \lambda \in L_a, w \in L_a \text{ for some non-empty word}\}, \\ T_4 &= \{a \mid a \in T, \lambda \notin L_a, w \in L_a \text{ for some non-empty word}\}. \end{aligned}$$

If $a \in T_3$, then we set $L'_a = L_a \setminus \{\lambda\}$. Then we get

$$L = T_2 \cup T_3 \cup \bigcup_{a \in T_3} \{a\} L'_a \cup \bigcup_{a \in T_4} \{a\} L_a.$$

By the closure properties of CS, L_a for all $a \in T_4$ and L'_a for $a \in T_3$ are context-sensitive languages and only consist of non-empty words. Hence, by Corollary 1, for any $a \in T_4$, there is a propagating conditional ET0L system G_a such that $L(G_a) = L_a$.

Now, for each $a \in T_4$, we construct the ET0L system G'_a with suffix-closed conditions as in the proof of the first statement of this lemma where we only change $\# \to \lambda$ to $\# \to a$ in the set P_{fin}. Then it follows as above that $L(G_a) = \{a\} L_a$ and G_a is propagating. Analogously, we can construct a propagating ET0L system G'_a for $a \in T_3$ such that $L(G'_a) = \{a\} L'_a$.

Now we rename all nonterminals in the ET0L systems G'_a, $a \in T_3 \cup T_4$ such that no nonterminal occurs in two different systems. Moreover, we change the rules and regular sets according to the renaming and add to each table rules $A \to A$ for all nonterminals not occurring in this table. For $a \in T_3 \cup T_4$, let

$$G''_a = (V', T, (P''_{1,a}, R''_{1,a}), (P''_{2,a}, R''_{2,a}), \ldots, (P''_{n_a,a}, R''_{n_a,a}), S_a).$$

Now we construct the propagating conditional ETOL system G with the alphabets $V' \cup \{S\}$ and T, where S is an additional symbol, the axiom S, the tables

$(P''_{i,a} \cup \{S \to S\}, R_{i,a})$ for $a \in T_3 \cup T_4$ and $1 \leq i \leq n_a$ and the additional table

$$(\{S \to a \mid a \in T_2 \cup T_3\} \cup \{S \to S_a \mid a \in T_3 \cup T_4\}, \{S, \lambda\}).$$

Obviously, G is propagating, all conditions of G are suffix-closed, and

$$\begin{aligned} L(G) &= T_2 \cup T_3 \cup \bigcup_{a \in T_3 \cup T_4} L(G''_a) = T_2 \cup T_3 \cup \bigcup_{a \in T_3 \cup T_4} L(G'_a) \\ &= T_2 \cup T_3 \cup \bigcup_{a \in T_3} \{a\} L'_a \cup \bigcup_{a \in T_4} \{a\} L_a = L. \end{aligned}$$

□

Lemma 7. *$RE = \mathcal{CEL}(CIRC)$ and $CS = \mathcal{CEPL}(CIRC)$.*

Proof. Let $L \in RE$. Then, by Corollary 1, $L = L(G)$ for some ET0L system

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

with regular conditions. From G we construct the conditional ET0L system H as in the first part of the proof of Lemma 6, where we take Circ instead of Suf in all cases. Then the obtained system has circular conditions. Moreover, $L(H) = L(G) = L$ can be shown as in the proof of Lemma 6. Thus we have $RE \subseteq \mathcal{CEL}(CIRC)$.

The converse inclusion follows from Lemma 3.

The proof of the second statement of the Lemma can be given by modifications analogous to those in the proof of the second statement of Lemma 6. □

Lemma 8. *$\mathcal{CEPL}(LOC_1) \subseteq EPT0L$.*

Proof. Let L be a language in $\mathcal{CEPL}(LOC_1)$. Then L is generated by some conditional ET0L system $G = (V, T, (P_1, R_1), (P_2, R_2), \ldots, (P_n, R_n), \omega)$ with conditions in LOC_1. Then, for $1 \leq i \leq n$, $R_i = A_i B_i^* C_i \cup (A_i \cap C_i)$ for some sets $A_i, B_i, C_i \subseteq V$.

We first discuss the case that $\omega = azb$ for some $a, b \in V$ and $z \in V^*$.

Let

$$V' = \{a' \mid a \in V\},\ V'' = \{a'' \mid a \in V\} \text{ and } V''' = \{a''' \mid a \in V\}.$$

Moreover, for a set $U \subset V$, we set

$$U' = \{a' \mid a \in U\},\ U'' = \{a'' \mid a \in U\} \text{ and } U''' = \{a''' \mid a \in U\}.$$

For a word $w = a_1 a_2 \ldots a_m$ with $a_i \in V$, we set $w' = a'_1 a'_2 \ldots a'_m$. We define the EPT0L system

$$H = (V \cup V' \cup V'' \cup \{F\}, T, P'_1, P'_2, \ldots, P'_n, Q, a'' z' b''),$$

where

$$\begin{aligned} P'_i = \{x'' \to y'' v' \mid x \in A_i, x \to yv \in P_i\} &\cup \{x' \to v' \mid x \in B_i, x \to v \in P_i\} \\ \cup \{x''' \to v' y''' \mid x \in C_i, x \to vy \in P_i\}& \\ \cup \{x \to F \mid x \in (V' \cup V'' \cup V''' \cup \{F\}) \setminus (A''_i \cup B'_i \cup C'''_i)\}& \end{aligned}$$

for $1 \leq i \leq n$ and

$$Q = \{x' \to x \mid x \in T\} \cup \{x'' \to x \mid x \in T\} \cup \{x''' \to x \mid x \in T\}$$
$$\cup \{x \to F \mid x \in (V' \cup V'' \cup V''' \cup \{F\}) \setminus (T' \cup T'' \cup T''')\}.$$

By these settings, without introducing F in a sentential form of the system H, $x_1 v x_2 \Longrightarrow_{P_i} x_3 u x_4$ in G if and only if $x_1'' v' x_2''' \Longrightarrow_{P_i'} x_3'' u' x_4'''$ in H and, moreover, $x_1'' v' x_2''' \Longrightarrow_Q x_1 v x_2$ in H if and only if $x_1 v x_2 \in T^+$. Furthermore, if a letter F occurs in a sentential form w of H, then it also occurs in all sentential forms derivable from w in H. Thus it is obvious that

$$\omega = azb \Longrightarrow_{P_{i_1}} a_1 z_1 b_1 \Longrightarrow_{P_{i_2}} a_2 z_2 b_2 \Longrightarrow_{P_{i_3}} \ldots \Longrightarrow_{P_{i_k}} a_k z_k b_k$$

in G for some letters $a_i, b_i \in V$ and some words $z_i \in V^*$ for $1 \leq i \leq k$ if and only if

$$a'' z' b''' \Longrightarrow_{P'_{i_1}} a_1'' z_1' b_1''' \Longrightarrow_{P'_{i_2}} a_2'' z_2' b_2''' \Longrightarrow_{P_{i'_3}} \ldots \Longrightarrow_{P'_{i_k}} a_k'' z_k' b_k''' \Longrightarrow_Q a_k z_k b_k$$

in H. Therefore $L(G) = L(H)$ and it is shown that $L \in \mathit{EPT0L}$.

Now we discuss the case that ω is a letter. Then we define L_1 as the set of all letters, i. e., words of length 1, which can be derived in G, and $L_{2,i}$ with $1 \leq i \leq n$ as the set of all words of length ≥ 2, which can be obtained from $x \in L_1 \cap A_i \cap C_i$ by the application of a rule of P_i. Now we add a further letter S to the basic alphabet of H and a further table

$$Q' = \{S \to x \mid x \in L_1\} \cup \{S \to x \mid x \in L_{2,i}, 1 \leq i \leq n\}$$
$$\cup \{x \to x \mid x \in V \cup V' \cup V'' \cup V''' \cup \{F\}\}.$$

Now it follows analogously to the above considerations that $L(G) = L(H)$ holds. □

Lemma 9. *$\mathcal{CEPL}(DEF) \subseteq EPT0L$.*

Proof. Let $R = A \cup V^* B$ with finite sets $A \subseteq V^*$ and $B \subseteq V^*$. Let m be a number which is greater than the maximal length of words in A and B. Then we have

$$R = \{w \mid |w| \leq m, w \in L\} \cup V^*(\bigcup_{w \in B} V^{m-|v|}\{w\}),$$

i. e., R can be represented as $R = A' \cup V^* B'$ with $A' \subseteq \bigcup_{j=1}^{m} V^j$ and $B' \subseteq V^m$.

Let $L \in \mathcal{CEPL}(DEF)$. Then $L = L(G)$ for some propagating ET0L system

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

with definite conditions. By the above observation, without loss of generality, we can assume that there is a number $m \geq |\omega|$ such that, for $1 \leq i \leq n$, $R_i = A_i \cup V_i^* B_i$ with $A_i \subseteq \bigcup_{j=1}^{m} V_i^j$ and $B_i \subseteq V_i^m$ for some $V_i \subseteq V$.

Moreover, let $V' = \{a' \mid a \in V\}$ and $V'' = \{[w] \mid w \in V^*, |w| \leq m\}$. We construct the EPT0L system

$$H = (V \cup V' \cup V''), T, \overline{P_1}, \overline{P_2}, \ldots, \overline{P_n}, Q, [\omega])$$

with

$$\begin{aligned}\overline{P_i} = &\{a \to a \mid a \in V \cup \{F\}\} \cup \{a' \to z' \mid a \in V_i, a \to z \in P_i\}\\ &\cup \{[w] \to [z] \mid [w] \in V'', w \in A_i \cup B_i, w \Longrightarrow_{P_i} z, |z| \le m\}\\ &\cup \{[w] \to x'[z] \mid [w] \in V'', w \in A_i \cup B_i, w \Longrightarrow_{P_i} xz, |z| = m\}\\ &\cup \{a' \to F \mid a \in V \setminus V_i\} \cup \{[w] \to F \mid [w] \in V'', w \notin A_i \cup B_i\}\end{aligned}$$

for $1 \le i \le n$ and

$$Q = \{a \to a \mid a \in V \cup \{F\}\} \cup \{a' \to a \mid a \in V\} \cup \{[w] \to w \mid [w] \in V''\}.$$

By the construction, all sentential forms have the form $[w]$, $x'[w]$ or x with $[w] \in V''$ and $x \in V^+$. Furthermore, we have the derivations $[w] \Longrightarrow_{\overline{P_i}} [z]$ if and only if $w \Longrightarrow_{P_i} z$, $[w] \Longrightarrow_{\overline{P_i}} x'[z]$ if and only if $w \Longrightarrow_{P_i} xz$. Moreover, if a word $y \in V^*$ is obtained by using $x'[w] \Longrightarrow_Q y$ (if $xw = y$ and $|y| \ge m+1$) or $[y] \Longrightarrow_Q y$ (if $|y| \le m$), then it is not changed by further derivation steps, because we have only the rule $a \to a$ for $a \in V$ in all tables. Thus any derivation in H has the form

$$\begin{aligned}[\omega] &\Longrightarrow_{\overline{P_{i_1}}} [w_1] \Longrightarrow_{\overline{P_{i_2}}} \dots \Longrightarrow_{\overline{P_{i_r}}} [w_r]\\ &\Longrightarrow_{\overline{P_{i_{r+1}}}} x_1'[w_{r+1}] \Longrightarrow_{\overline{P_{i_{r+2}}}} x_2'[w_{r+2}] \Longrightarrow_{\overline{P_{i_{r+3}}}} \dots \Longrightarrow_{\overline{P_{i_{r+s}}}} x_s'[w_{r+s}]\\ &\Longrightarrow_Q x_s w_{r+s} \Longrightarrow x_s w_{r+s} \Longrightarrow \dots\end{aligned}$$

with $[w_i] \in V''$ for $1 \le i \le r+s$ and $x_j \in V^*$ for $1 \le j \le s$; and such a derivation exists if and only there is a derivation

$$\begin{aligned}\omega &\Longrightarrow_{P_{i_1}} w_1 \Longrightarrow_{P_{i_2}} \dots \Longrightarrow_{P_{i_r}} w_r \Longrightarrow_{P_{i_{r+1}}} x_1 w_{r+1}\\ &\Longrightarrow_{P_{i_{r+2}}} x_2 w_{r+2} \Longrightarrow_{P_{i_{r+3}}} \dots \Longrightarrow_{P_{i_{r+s}}} x_s w_{r+s}\end{aligned}$$

in G exists. Therefore, $L(H) = L(G)$ and $L \in EPT0L$. □

Lemma 10. *ET0L* $\subseteq \mathcal{CEPL}(MON)$.

Proof. Let L be a language in *ET0L*. Then there is a propagating ET0L system $G = (V, T, P_1, P_2, \dots, P_r, \omega)$ generating L. In an ET0L system, any table can be applied to any sentential form. Thus the conditional propagating ET0L system $(V, T, (P_1, V^*), (P_2, V^*), \dots, (P_r, V^*), \omega)$ with monoidal conditions generates L, too. Therefore *ET0L* $\subseteq \mathcal{CEPL}(MON)$. □

Lemma 11. $\mathcal{CEPL}(COMM) = MAT$ *and* $\mathcal{CEL}(COMM) = MAT^\lambda$

Proof. i) $MAT^\lambda \subseteq \mathcal{CEL}(COMM)$.

We recall that any recursively enumerable language can be generated by a matrix grammar G in 2-normal form (see Lemma 1.2.3 in [4]), i. e., by a matrix grammar $G = (N_1 \cup N_2 \cup \{S\},\ T,\ M,\ S,\ Q)$ where all matrices of M have one of the following forms

– $(S \to AX)$ with $A \in N_1$ and $X \in N_2$,

– $(A \to w,\ X \to Y)$ with $A \in N_1$, $w \in (N_1 \cup T)^*$, and $X, Y \in N_2$,
– $(A \to w,\ X \to \lambda)$ with $A \in N_1$, $w \in (N_1 \cup T)^*$, and $X \in N_2$,

and Q contains only rules of the form $A \to w$.

Let L be a language in MAT^λ. Then L is generated by a matrix grammar $G = (N_1 \cup N_2 \cup \{S\},\ T,\ M,\ S,\ Q)$ which satisfies the above mentioned normal form conditions. Let $m_1, m_2, \ldots m_r$ be the matrices $(A \to w, X \to z)$ of M, $z \in N_2 \cup \{\lambda\}$ with $A \to w \notin F$ and $m_{r+1}, m_{r+2}, \ldots m_s$ be the matrices $(A \to w, X \to z)$ of M, $z \in N_2 \cup \{\lambda\}$ with $A \to w \in F$ We set

$$V = N_1 \cup N_2 \cup \{S\} \cup T \cup \{B' \mid B \in N_1 \cup N_2 \cup T\} \cup \bigcup_{(A \to w, X \to z) \in M} \{A_m, X_m\}$$

and

$$(P, R) = (\{S \to AX \mid (S \to AX) \in M\}, \{S\}).$$

With a matrix $m = (A \to w, X \to z)$ with $z \in N_2$ or $z = \lambda$, we associate $(P_{m,1}, R_{m,1})$ and $(P_{m,2}, R_{m,2})$ defined by

$$\begin{aligned}
P_{m,1} &= \{A \to A, A \to A_m, X \to X_m\},\\
R_{m,1} &= (N_1 \cup N_2 \cup T)^+,\\
P_{m,2} &= \{A_m \to w, X_m \to Y\},\\
R_{m,2} &= \{w \mid w \in (N_1 \cup T \cup \{A_m, X_m\})^+, \#_{A_m} = \#_{X_m} = 1\},
\end{aligned}$$

and if $A \to w$ is an element of F, we add

$$(P'_{m,2}, R'_{m,2}) = (\{X_m \to Y\}, \{w \mid w \in ((N_1 \setminus \{A\}) \cup T \cup \{X_m\})^+, \#_{X_m} = 1\}).$$

We construct the conditional Lindenmayer system

$$\begin{aligned}
G' = \ & (V,\ T,\ (P, R),\ (P_{m_1,1}, R_{m_1,1}), (P_{m_1,2}, R_{m_1,2}),\\
& \ldots, (P_{m_r,1}, R_{m_r,1}), (P_{m_r,2}, R_{m_r,2}),\\
& (P_{m_{r+1},1}, R_{m_{r+1},1}), (P_{m_{r+1},2}, R_{m_{r+1},2}), (P'_{m_{r+1},2}, R'_{m_{r+1},2}),\\
& \ldots, (P_{m_s,1}, R_{m_s,1}), (P_{m_s,2}, R_{m_s,2}), (P'_{m_s,2}, R'_{m_s,2}),\ S).
\end{aligned}$$

Obviously, all conditions are commutative. We now prove that $L(G') = L(G)$.

In both devices any derivation starts with $S \Longrightarrow AX$.

Now let $z_1 A z_2 X$ be a sentential form of G, and let $m = (A \to w, X \to Y)$ be a matrix of M. Then in G we get $z_1 w z_2 Y$. In G', by application of $(P_{m,1}, R_{m,1})$, we obtain a word which differs from $z_1' A_m z_2' X_m$ where z_1' and z_2' are obtained from z_1 and z_2 by replacing some As by A_m. However, the derivation can only continued if there are no A_ms in z_1' and z_2', i.e., we obtained $z_1 A_m z_2 X_m$. Now only $(P_{m,2}, R_{m,2})$ can be applied which yields $z_1 w z_2 Y$. Therefore we have simulated a derivation step of G. If $A \to w$ is in F and the sentential form zX does not contain a letter A, then we get in G the word zY, and in G' we have the simulation $zX \Longrightarrow zX_m \Longrightarrow zY$.

Obviously, a successful derivation in G' consists only of the mentioned derivation steps.

Moreover, the derivation in G stops if and only if no table $(P_{m,2}, R_{m,2})$ and $(P'_{m,2}, R'_{m,2})$ changes the sentential form.

Thus $L(G') = L(G)$ follows.

ii) $MAT \subseteq \mathcal{CEPL}(COMM)$.

This can be shown analogously. We have only to start with the accurate normal form (see Definition 1.3.2 and Lemma 1.3.7 in [4]).

iii) $\mathcal{CEL}(COMM) \subseteq MAT^{\lambda}$

Let $L \in \mathcal{CEL}(COMM)$. Then there is a conditional Lindenmayer system $G = (V, T, (P_1, R_1), (P_2, R_2, \ldots, (P_n, R_n), w)$ such that, for any i, $1 \leq i \leq n$, R_i is a commutative and regular language.

Let
$$V = \{A_1, A_2, \ldots, A_m\} \text{ and } T = \{A_{p+1}, A_{p+2}, \ldots, A_m\}.$$
Obviously, for $1 \leq i \leq n$, R_i is a set over V and $R_i = \pi_V^{-1}(M_i)$ for some semi-linear set M_i. Let
$$M_i = \bigcup_{j=1}^{r_i} M_{i,j}$$
with
$$\begin{aligned} M_{i,j} &= \{(a_{1,i,j}, a_{2,i,j}, \ldots, a_{m,i,j}) \\ &\quad + \sum_{k=1}^{t_{i,j}} \alpha_k (b_{1,k,i,j}, b_{2,k,i,j}, \ldots, b_{m,k,i,j}) \mid \alpha_j \in \mathbb{N} \text{ for } 1 \leq k \leq t_{i,j}\} \end{aligned}$$
for some $a_{r,i,j}$ and $b_{r,k,i,j}$, $1 \leq r \leq m$, $1 \leq i \leq n$, $1 \leq j \leq r_i$, and $1 \leq k \leq t_{i_j}$.

We define the matrix grammar $G' = (N', T', M, S, Q)$ where
$$\begin{aligned} N' &= \{S, Z, \#\} \cup \bigcup_{i=1}^{m} \{A'_i, A''_i\} \cup \bigcup_{j=1}^{n} \{Z_i, Z'_i\} \cup \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq t_i}} \{Z_{i,j}\} \\ T' &= T \cup \{X\}, \\ Q &= \{A'_i \to \# \mid 1 \leq i \leq m\} \cup \{A''_i \to \# \mid 1 \leq i \leq m\}, \end{aligned}$$
and M consists of all matrices constructed as follows. As initial rules we take all rules
$$(S \to Z_i w') \text{ for } 1 \leq i \leq n$$
(we generate a primed version of the axiom w of G accompanied by some control symbol Z_i).

For any $1 \leq i \leq n$, $1 \leq i' \leq n$, and $1 \leq j \leq r_i$, we introduce the matrices
$$\begin{aligned} &(Z_i \to Z_{i,j}, (A'_1 \to A''_1)^{a_{1,i,j}}, (A'_2 \to A''_2)^{a_{2,i,j}}, \ldots, (A'_m \to A''_m)^{a_{m,i,j}}) \\ &(Z_{i,j} \to Z_{i,j}, (A'_1 \to A''_1)^{b_{1,k,i,j}}, (A'_2 \to A''_2)^{b_{2,k,i,j}}, \ldots, (A'_m \to A''_m)^{b_{m,k,i,j}}) \\ &\qquad \text{for } 1 \leq k \leq t_{i,j} \\ &(Z_{i,j} \to Z'_i, A'_1 \to \#, A'_2 \to \#, \ldots, A'_m \to \#) \text{ for } 1 \leq j \leq r_i, \end{aligned}$$

(applying these matrices to a sentential form $Z_i v'$ for some $v' \in (V')^*$ one checks whether the Parikh vector of v is contained in $M_{i,j}$; thus the $Z_i' v''$ can only be obtained if the sentential form is contained in R_i)

$$(Z_i' \to Z_i', A'' \to w') \text{ for } A \to w \in P_i$$

(after checking that the sentential form is in R_i we apply the rules of P_i),

$$(Z_i' \to Z_{i'}, A_1'' \to \#, A_2'' \to \#, \ldots, A_n'' \to \#)$$

(if all letters of v'' are replaced, i. e., we get z' where $v \Longrightarrow z$ holds in G and have simulated a derivation step in G, we can start the same process with i'),

$$\begin{aligned}
&(Z_i \to Z, A_1' \to \#, A_2' \to \#, \ldots, A_p' \to \#),\\
&(Z \to Z, A_q' \to A_q) \text{ for } p+1 \le q \le m,\\
&(Z \to X, A_{q+1}' \to \#, A_{q+2}' \to \#, \ldots, A_m' \to \#).
\end{aligned}$$

(if $Z_i v'$ does not contain the letters $A_1', A_2', \ldots A_p'$, i. e., v is a word over the terminal alphabet T, we replace all letters A_q' by A_q, and finally Z by X).

By the given explanations, it is easy to see that $L(G') = \{X\}L(G)$.

Thus $\{X\}L(G) \in MAT^\lambda$. By the closure properties of MAT (see [4], page 48), $L(G) \in MAT^\lambda$ which proves the statement.

iv) $\mathcal{CEPL}(COMM) \subseteq MAT$

Since the construction in iii) produces no erasing rules in the matrix grammar if the conditional Lindenmayer system contains no erasing rules, the statement follows by the same construction. □

Lemma 12. $\mathcal{CEPL}(FIN) = \mathcal{CEL}(FIN) = FIN$

Proof. Obviously, any language in $\mathcal{CEL}(FIN)$ is finite. Thus $\mathcal{CEL}(FIN) \subseteq FIN$.

Let $L \subset T^+$ be finite language (note that by our setting that languages are equal if they differ at most in the empty word, we can ignore the empty word, if it is in L). It is easy to see that the propagating ET0L system

$$(\{S\} \cup T, T, (\{S \to w \mid w \in L\}, \{S\}), S)$$

with a finite condition generates L. Thus $FIN \subseteq \mathcal{CEPL}(FIN)$.

By these inclusions and Lemma 1, we get the statement of the lemma. □

4 Summary and Conclusions

By a combination of the lemmas above and Example 1, we get the following theorem.

Theorem 1. *For all $s \ge 1$ and $r \ge 2$, the diagram given in Figure 2 holds.*

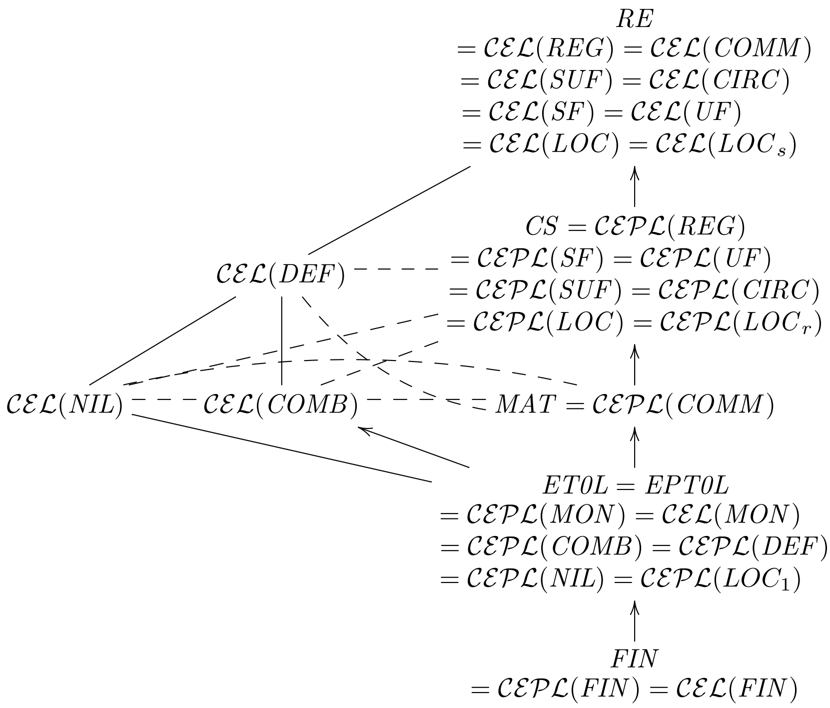

Figure 2: Hierarchy of language families $\mathcal{CEL}(X)$ and $\mathcal{CEPL}(X)$ with $X \in \mathcal{G}$ (an arrow from Z_1 to Z_2 denotes $Z_1 \subset Z_2$; a line from Z_1 to a higher positioned Z_2 stands for $Z_1 \subseteq Z_2$; the relation between families which are connected by a broken line is unknown; and if two families are not connected by a directed path or a broken line, then they are incomparable)

If one only considers the propagating families, then the hierarchy is completely determined. However, in the general case, there are some open problems related to the families $\mathcal{CEL}(NIL)$, $\mathcal{CEL}(COMB)$, and $\mathcal{CEL}(DEF)$; essentially we only have the relations which follow directly from the relation between the subregular families.

The obtained picture is very similar to that which was obtained for (sequential) context-free conditional grammars (for a definition see [4]). Especially,

$$\mathcal{CEL}(X) = Z \in \{RE, CS, MAT, ET0L\}$$

implies that the family of context-free conditional grammars with conditions from X coincides with Z, too; this implication also holds for systems/grammars with only non-erasing rules. However, the families of context-free conditional grammars with definite, nilpotent, and combinational conditions are also equal to $ET0L$. In contrast, for Lindenmayer systems $ET0L \subset \mathcal{CEL}(COMB) \subseteq \mathcal{CEL}(DEF)$.

References

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

[2] Dassow, J., Grammars with commutative, circular, and locally testable conditions. In Ésik, Z., and Fülöp, Z., editors, *Automata, Formal Languages, and Related Topics – Dedicated to Ferenc Gécseg on the Occasion of his 70th Birthday*, pages 27–37. University of Szeged, 2009.

[3] Dassow, J. Subregular restrictions for some language generating devices. In Freund, R., Holzer, M., Truthe, B., and Ultes-Nitsche, U., editors, *NCMA 2012, Proceedings*, pages 11–26. books@ocg.at 290, Österreichische Computer Gesellschaft, Austria, 2012.

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

[5] Dassow, J. and Rudolf, St. Conditional Lindenmayer systems with subregular conditions: the non-extended case. *RAIRO - Theor. Inf. Appl.*, **48**:127–147, 2014.

[6] Dassow, J., Stiebe, R., and Truthe, B. Generative capacity of subregularly tree controlled grammars. *International Journal of Foundations of Computer Science*, **21**:723–740, 2010.

[7] Gécseg, F. On some classes of tree automata and tree languages. *Annales Academiae Scientiarum Fennicae*, **25**: 325–336, 2000.

[8] Gécseg, F. and Gyurica, Gy. On the closedness of DR tree languages under Boolean operations. *Acta Cybernetica*, **17**:449–457, 2006.

[9] Gécseg, F. and Imreh, B. On isomorphic representations of generalized definite automata. *Acta Cybernetica*, **15**:33–43, 2001.

[10] Gécseg, F. and Imreh, B. on definite and nilpotent DR tree languages. *Journal of Automata, Languages, and Combinatorics*, **9**:55–60, 2004.

[11] Gécseg, F. and Peák, I. *Algebraic Theory of Automata.* Academiai kiado, Budapest, 1972.

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

[13] Martín-Vide, C. and Mitrana, V. Networks of Evolutionary Processors: Results and Perspectives. In: *Molecular Computational Models: Unconventional Approaches*, pages 78–114, 2005.

[14] Păun, Gh. *Marcus Contextual Grammars.* Kluwer Publ. House, Doordrecht, 1998.

[15] Rozenberg, G. and Salomaa, A. *The Mathematical Theory of L Systems.* Academic Press, New York, 1980.

[16] Rozenberg, G. and Salomaa, A., editors. *Handbook of Formal Languages.* Springer-Verlag, Berlin, 1997.

[17] Rozenberg, G. and von Solms, S. H. Priorities on context conditions in rewriting systems. *Inform. Sci.*, **14**:15–51, 1978.

[18] Shyr, H. J. *Free Monoids and Languages.* Hon Min Book Co., Taichung, Taiwan, 1991.

[19] Starke, P. H. *Abstrakte Automaten.* Deutscher Verlag der Wissenschaften, Berlin, 1969.

[20] Wiedemann, B. *Vergleich der Leistungsfähigkeit endlicher determinierter Automaten.* Diplomarbeit, Universität Rostock, 1978.

Received 13th May 2015