Journal of Universal Computer Science, vol. 2, no. 11 (1996), 745-755
submitted: 30/4/96, accepted: 28/8/96, appeared: 28/11/96 © Springer Pub. Co.

Remarks on Propagating Partition-Limited ET0L Systems

Henning Fernau
Wilhelm-Schickard-Institut für Informatik; Universität Tübingen
Sand 13; D-72076 Tübingen; Germany
email: fernau@informatik.uni-tuebingen.de

Abstract: In this paper, we sharpen the results of Gärtner on the universality of partition-limited ET0L systems by showing that such deterministic systems characterize the recursively enumerable sets, and, furthermore, the propagating deterministic partition-limited ET0L systems characterize the programmed languages with appearance checking disallowing erasing productions.
The main results of this paper have been announced in [10].
Key Words: Formal languages, parallel rewriting systems, k-limited systems.
Category: F.4.2, F.4.3

1 Introduction and definitions

Recently, interest emerged concerning limited parallel rewriting, which is also well-justified considering the original biological motivation of Lindenmayer systems [17, 18], since the idea of unlimited growth is only realistic at certain scales. Especially, realistic growth rates cannot be obtained by unrestricted Lindenmayer systems [21].

We assume the reader to be familiar with some basics of formal language theory. Our notational conventions are: the empty word is denoted by λ; the length of a word x is denoted by $|x|$. If $x \in V^*$, where V is some alphabet, and if $W \subseteq V$, then $|x|_W$ denotes the number of occurrences of letters from W in x. Inclusion is denoted by $\subseteq$, while strict inclusion is written $\subset$. We use bracket notations in order to say that the statement holds both in the case of excluding and in the case of including the bracket contents consistently.

A *partition-k-limited ET0L system* is a sixtuple $G = (V, V', \{P_1, \dots, P_r\}, \omega, \pi, k)$ where V' is a non-empty subset (terminal alphabet) of the alphabet V, $\omega \in V^+$ is the axiom, and each so-called table P_i is a finite subset of $V \times V^*$ which satisfies the *completeness condition*, i.e., for each $a \in V$, there is at least one word $w \in V^*$ such that $a \to w \in P_i$, hence, each P_i defines a finite substitution $\sigma_i : V^* \to 2^{V^*}$. π is a *partition* of V, i.e., $\pi = \{\pi_1, \dots, \pi_s\}$ with $\bigcup_{i=1}^{s} \pi_i = V$ and $1 \leq i < j \leq s$ implies $\pi_i \cap \pi_j = \emptyset$ (the sets π_i are called *parts* of π), and k is some natural number. G is called propagating if no table contains an erasing production $a \to \lambda$. Similarly, the notion of deterministic system is inherited from the theory of Lindenmayer systems, i.e., G is called deterministic if no table contains two different productions with the same left-hand side. According to G, $x \Rightarrow y$ (for $x, y \in V^*$) iff there is a table P_i and $x = x_0\alpha_1x_1 \cdots \alpha_nx_n$, $y = x_0\beta_1x_1 \cdots \beta_nx_n$ such that $\alpha_\nu \to \beta_\nu \in P_i$ for each $1 \leq \nu \leq n$, such that, for each part π_j of the partition of π, either (1) $|\alpha_1 \cdots \alpha_n|_{\pi_j} = k$ or (2) $|\alpha_1 \cdots \alpha_n|_{\pi_j} < k$ and $|x_0x_1 \cdots x_n|_{\pi_j} = 0$.

As usual, the language generated by G is given by $L(G) = \{w \in V'^* \mid \omega \stackrel{*}{\Rightarrow} w\}$, where $\stackrel{*}{\Rightarrow}$ denotes the reflexive transitive closure of $\Rightarrow$.

As special cases, we define *uniformly k-limited ET0L systems* (introduced by Wätjen and Unruh in [23, 26]) restricting ourselves to partitions of the form $\pi = \{V\}$, and *k-limited ET0L systems* (introduced by Wätjen in [22]) via the restriction to partitions of the form $\pi = \{\{a\} \mid a \in V\}$.

In order to state our results, we define programmed grammars. A *programmed grammar* ([6, 20]) is a construct $G = (V_N, V_T, P, S)$, where V_N, V_T, and S are the set of nonterminals, the set of terminals and the start symbol, respectively, and P is a finite set of productions of the form $(r : \alpha \to \beta,\ \sigma(r),\ \phi(r))$, where $r : \alpha \to \beta$ is a rewriting rule labelled by r and $\sigma(r)$ and $\phi(r)$ are two sets of labels of such core rules in P. By $\mathrm{Lab}(P)$ we denote the set of all labels of the productions appearing in P. For (x, r_1) and (y, r_2) in $V_G^* \times \mathrm{Lab}(P)$, we write $(x, r_1) \Rightarrow (y, r_2)$ iff either

1. $x = z_1\alpha z_2$, $y = z_1\beta z_2$, $(r_1 : \alpha \to \beta,\ \sigma(r_1),\ \phi(r_1)) \in P$, and $r_2 \in \sigma(r_1)$, or
2. $x = y$, the rule $r_1 : \alpha \to \beta$ for some production $(r_1 : \alpha \to \beta,\ \sigma(r_1),\ \phi(r_1)) \in P$ is not applicable to x, and $r_2 \in \phi(r_1)$.

In the latter case, the derivation step is done in appearance checking mode. The set $\sigma(r_1)$ is called *success field* and the set $\phi(r_1)$ *failure field* of r_1. The language generated by G is defined as

$$L(G) = \{w \in V_T^* \mid (S, r_1) \stackrel{*}{\Rightarrow} (w, r_2) \text{ for some } r_1, r_2 \in \mathrm{Lab}(P)\}\,.$$

In the literature, six language classes are considered, stemming from allowing/disallowing erasing productions and three different conditions on the success and failure fields:

with appearance checking no restrictions (notation: $\mathcal{L}(\mathrm{P}, \mathrm{CF}[-\lambda], \mathrm{ac})$);
without appearance checking $\phi(r) = \emptyset$ for each rule;
with unconditional transfer $\phi(r) = \sigma(r)$ for each rule.

Moreover, we need the notion of transducer.

A *one-input finite state transducer with accepting state*, or *1-a-transducer* for short, is a 6-tuple $M = (Q, X, Y, H, q_0, q_f)$, where Q is a finite set of states, X and Y are finite (input and output) alphabets, $q_0 \in Q$ is the initial state, and $q_f \in Q$ is an accepting or final state, and H is a finite subset of $Q \times (X \cup \{\lambda\}) \times Y^* \times Q$. M is called *λ-free* if $H \subseteq Q \times (X \cup \{\lambda\}) \times Y^+ \times Q$.

By a *computation* of such a 1-a-transducer a word $h = h_1 \cdots h_n \in H^+$ is understood such that

- $\mathrm{pr}_1(h_1) = q_0$, $\mathrm{pr}_4(h_n) = q_f$;
- $\forall 1 \le i \le n-1 (\mathrm{pr}_1(h_{i+1}) = \mathrm{pr}_4(h_i))$;

where pr_i are projection homomorphisms on H^* defined by

$$\mathrm{pr}_i((x_1, x_2, x_3, x_4)) = x_i \quad \text{for} \quad i = 1, 2, 3, 4.$$

The set of all computations of M is denoted by $C(M)$. A *1-a-transducer mapping* is defined for each language $L \subseteq X^*$ by $M(L) = \mathrm{pr}_3(\mathrm{pr}_2^{-1}(L) \cap C(L))$.

Remark. It is well-known [1] that a family of languages $\mathcal{L}$ is a trio if and only if it is closed under λ-free 1-a-transducer mappings. $\mathcal{L}$ is a full trio iff it is closed under 1-a-transducer mappings.

A *[full] abstract family of languages*, [F]AFL for short, is a [full] trio closed under union and iteration-$^+$.

2 Results

In the following, we give a proof of a sharpened variant of [13, Satz 6.3.3], which, in addition, answers some questions (raised by Gärtner in [13]) on the power of propagating and deterministic partition-limited ET0L systems.

First, we need a lemma whose proof is done by standard construction (see also [13, Satz 6.3.2]).

Lemma 1. *Let $k \geq 1$. The family of languages generated by [propagating] deterministic partition-k-limited ET0L systems is closed under union.* □

Theorem 2. *For every given [non-erasing] programmed grammar with appearance checking, there exists an equivalent [propagating] deterministic partition-1-limited ET0L system and vice versa.*

Proof. '$\Longrightarrow$': Let $L \in \mathcal{L}(\mathrm{P}, \mathrm{CF}[-\lambda], \mathrm{ac})$ be a language over the terminal alphabet V_T. Consider

$$L = \bigcup_{a,b \in V_T} \{ab\} L_{ab} \cup L',$$

where $L_{ab} = \{w \in V_T^+ \mid abw \in L\}$, and L' is a finite supplementary language, precisely, $L' = \{w \in L \mid |w| \leq 2\}$. Since $\mathcal{L}(\mathrm{P}, \mathrm{CF}[-\lambda], \mathrm{ac})$ is an AFL [6], it is closed under left derivatives and intersection with regular sets, hence each of the languages L_{ab} is contained in $\mathcal{L}(\mathrm{P}, \mathrm{CF}[-\lambda], \mathrm{ac})$.

Clearly, L' is generable by a [propagating] deterministic partition-limited ET0L system. Now, we are going to show that each of the languages $\{ab\}L_{ab}$ is generable by a [propagating] deterministic partition-limited ET0L system. By the lemma stated above, it follows that L itself is generable by a [propagating] deterministic partition-limited ET0L system.

Let $G = (V_N, V_T, P, S)$ be a programmed grammar with label set $\mathrm{Lab}(P) = \{p_1, \ldots, p_n\}$ generating L_{ab}.

We construct a [propagating] deterministic partition-1-limited ET0L system

$$G' = (V, V_T, H, \hat{S}, \pi, 1) \quad \text{with}$$

$$V = V_N \cup V_T \cup \{\hat{S}, F, \#, \#'\} \cup \underbrace{\{\, A' \mid A \in V_N \,\}}_{=:V_N'} \cup \mathrm{Lab}(P)\,,$$

where the partition π is given by

$$\left\{ V_T, \{\hat{S}, F\}, V_N \cup \{\#\}, \mathrm{Lab}(P), V_N' \cup \{\#'\} \right\}.$$

The set H of tables is described in the following.

$$\begin{aligned} h_{init} &= \{\, \hat{S} \to p\#S \mid p \in \mathrm{Lab}(P)\,\} \cup \{\, X \to F \mid X \neq \hat{S}\,\}\,, \\ h_{fin} &= \{\, p \to a \mid p \in \mathrm{Lab}(P)\,\} \cup \{\, c \to c \mid c \in V_T\,\} \cup \{\# \to b\} \\ &\quad \cup \{\, X \to F \mid X \in \{\hat{S}, F, \#'\} \cup V_N \cup V_N'\,\}\,; \end{aligned}$$

for every $(p_i : A \to w, \sigma(p_i), \phi(p_i))$, $1 \leq i \leq n$ and for every $q \in \sigma(p_i)$, we introduce a table

$$\begin{aligned} h_{i,q} &= \{p_i \to q\} \cup \{\, c \to c \mid c \in V_T\,\} \cup \{A \to w\} \\ &\quad \cup \{\, X \to F \mid X \notin \{p_i, A\} \cup V_T\,\}\,, \end{aligned}$$

and, (if $\phi(p_i) \neq \emptyset$) tables

$$\begin{aligned} h_i' &= \{p_i \to p_i\} \cup \{B \to B', B' \to B' \mid A \neq B, B \in V_N\} \\ &\quad \cup \{\, c \to c \mid c \in V_T\,\} \cup \{\# \to \#', \#' \to \#'\} \\ &\quad \cup \{\, X \to F \mid X \in (\mathrm{Lab}(P) \setminus \{p_i\}) \cup \{A, A', \hat{S}, F\}\,\}\,, \end{aligned}$$

as well as, for every $q \in \phi(p_i)$,

$$\begin{aligned} h_{i,q}'' &= \{p_i \to q\} \cup \{\, c \to c \mid c \in V_T\,\} \cup \{\#' \to \#\} \\ &\quad \cup \{\, X \to F \mid X \in (\mathrm{Lab}(P) \setminus \{p_i\}) \cup \{\hat{S}, F, \#\} \cup V_N \cup V_N'\,\}\,, \\ h_i''' &= \{q \to q \mid q \in \phi(p_i)\} \cup \{\, c \to c \mid c \in V_T\,\} \\ &\quad \cup \{B' \to B, B \to B \mid A \neq B, B \in V_N\} \\ &\quad \cup \{\, X \to F \mid X \in (\mathrm{Lab}(P) \setminus \phi(p_i)) \cup \{A, A', \hat{S}, F, \#, \#'\}\,\}\,. \end{aligned}$$

The tables $h_{i,q}$ simulate an application of p_i in the success case. If the symbol A is not present in the current sentential form, an application of $h_{i,q}$ would introduce the failure symbol F, since A and $\#$ belong to the same part of the partition.

The sequence $(h_i')^+ h_{i,q}'' (h_i''')^*$ allows for negative appearance checks. h_i' has to be applied until every nonterminal B is converted to its primed version B', and $\#$ is converted to $\#'$. Then, the application of $h_{i,q}''$ checks the non-occurrence of A, since all other symbols in the part $V_N \cup \{\#\}$ of the partition π have been turned into their primed counterparts. At the same time, the label p_i is safely changed into some $q \in \phi(p_i)$. Finally, applications of h_i''' permit turning each primed nonterminal B' back into its unprimed version B. The colourings obtained by tables h_i' and h_i''' are necessary, since otherwise a non-occurrence check of the symbol A contained in a part of the partition where other symbols also belong to is difficult.

'$\Longleftarrow$': We only sketch the basic ideas. In principle, a sequentialization technique similar to [5] is applicable. First, nondeterministically a table h to be simulated is chosen, then, every part π_i of a partition π of the alphabet is scanned:

- non-occurrence assumption: no symbol in π_i occurs in the current string; this can be checked sequentially by $(t_1 : A_1 \to F, \emptyset, \{t_2\}), \ldots, (t_n : A_n \to F, \emptyset, ?)$, where $\pi_i = \{A_1, \ldots, A_n\}$ and F is a special failure symbol.
- occurrence assumption of A_j: some production $A_j \to w \in h$ is selected, leading to a production of the form $(? : A_j \to w, ?, \emptyset)$. $\square$

As in the case of 1-limitation, it is easy to show the following:

Lemma 3. *Let $k \geq 1$. For every given [propagating] deterministic partition-k-limited ET0L system, there exists an equivalent [non-erasing] programmed grammar with appearance checking.* □

From the well-known closure properties of programmed languages [6], we infer:

Corollary 4. *The family of languages generated by propagating (deterministic) partition-1-limited ET0L systems is an AFL. The family of languages generated by (deterministic) partition-1-limited ET0L systems is a FAFL and coincides with the recursively enumerable sets.* □

The following lemma corrects and generalizes the proof of [13, Satz 6.3.2].

Lemma 5. *Let $k \geq 1$. The family of languages generated by propagating partition-k-limited ET0L systems is a trio. The family of languages generated by partition-k-limited ET0L systems is a full trio.*

Proof. By our remark, we have to show the closure of the corresponding language families under 1-a-transducer mappings. So, let $G = (V, \Delta, \{P_1, \dots, P_r\}, \omega, \pi, k)$ be a partition-k-limited ET0L system with $\pi = \{\pi_1, \dots, \pi_s\}$ and let $M = (Q, \Delta, \Delta', H, q_0, q_f)$ be a 1-a-transducer.

We construct a partition-k-limited ET0L system $G' = (V', \Delta', P, S, \pi', k)$ such that $L(G') = M(L(G))$. The idea is a modification of the classical triple construction. Let

$$V' = (Q \times (V \cup \{L\}) \times Q) \cup \{S, F, L\} \cup \Delta \quad \text{and}$$
$$\pi' = \{\pi'_1, \dots, \pi'_s\} \cup \{\{S, F, L\} \cup \Delta' \cup (Q \times \{L\} \times Q)\},$$

where $\pi'_j = \{(q, a, q') \mid q, q' \in Q, a \in \pi_j\}$.
Let the mapping $\sigma : V^* \to 2^{V'^*}$ be given by $\lambda \mapsto Q \times \{L\} \times Q$, $a \mapsto Q \times \{a\} \times Q$ for $a \in V$, and in general, with $\forall 1 \leq i \leq n (w_i \in V)$, $w_1 \cdots w_n \mapsto \{(q_1, w_1, q_2) \cdots (q_n, w_n, q_{n+1}) \mid \forall 1 \leq i \leq n+1 (q_i \in Q)\}$.
P contains the following tables:

$$\begin{aligned} h_{init} = \{\, & S \to v_1 \cdots v_n \mid v_1 \cdots v_n \in \sigma(\omega) \wedge \forall 1 \leq i \leq n (v_i \in (Q \times V \times Q)) \\ & \wedge q_0 = \mathrm{pr}_1(v_1) \wedge q_f = \mathrm{pr}_3(v_n)\,\} \cup \{\, X \to F \mid X \neq S \,\}; \end{aligned}$$

for every $1 \leq \rho \leq r$, let

$$\begin{aligned} h_\rho = \{\, & \hat{a} \to \hat{w} \mid \hat{a} \in \sigma(a) \wedge \hat{w} \in \sigma(w) \wedge a \to w \in P_\rho \\ & \wedge \mathrm{pr}_1(\hat{w}_1) = \mathrm{pr}_1(\hat{a}) \wedge \mathrm{pr}_3(\hat{w}_{|\hat{w}|}) = \mathrm{pr}_3(\hat{a})\} \\ \cup \{\, & (q, L, q') \to (q, L, q') \mid q, q' \in Q \,\} \\ \cup \{\, & X \to F \mid X \in \{S, F, L\} \cup \Delta' \,\}; \end{aligned}$$

$$\begin{aligned} h_L = \{\, & (q, B, q'') \to (q, B, q')(q', L, q''), (q, B, q'') \to (q, L, q')(q', B, q''), \\ & (q, B, q'') \to (q, B, q'') \mid q, q', q'' \in Q, B \in V \cup \{L\}\,\} \end{aligned}$$

$$\cup \{ X \to F \mid X \in \{S, F, L\} \cup \Delta' \};$$

$$\begin{aligned} h_{fin,1} = & \{ (q, a, q') \to w \mid (q, a, w, q') \in H, a \in \Delta, w \in \Delta'^{+} \} \\ & \cup \{ (q, L, q') \to w \mid (q, \lambda, w, q') \in H, w \in \Delta'^{+} \} \\ & \cup \{ (q, a, q') \to L \mid (q, a, \lambda, q') \in H, a \in \Delta \} \\ & \cup \{ (q, L, q') \to L \mid (q, \lambda, \lambda, q') \in H \} \\ & \cup \{ (q, L, q) \to L \mid q \in Q \} \\ & \cup \{ a \to a \mid a \in \Delta' \cup \{L\} \} \\ & \cup \{ X \to F \mid X \in \{S, F\} \cup (Q \times (V \setminus \Delta) \times Q) \}; \end{aligned}$$

$$h_{fin,2} = \{L \to \lambda\} \cup \{ a \to a \mid a \in \Delta' \} \cup \{ X \to F \mid X \notin \Delta' \cup \{L\} \}.$$

A few comments on the construction may be in order:

- The initialization table h_{init} is the only one which can be successfully applied to the start symbol S. h_{init} guarantees that the 1-a-transducer M whose application is to be simulated starts in q_0 and ends up in q_f.
- Since the partitions are inherited from the original system G properly, the tables h_ρ simply simulate the original tables P_ρ in some shadow alphabet $Q \times (V \cup \{L\}) \times Q$ instead of V. L serves as some placeholder for the empty word λ.
- h_L helps to deal with λ-moves of the 1-a-transducer.
- In case only symbols from $Q \times (\Delta \cup \{L\}) \times Q$ occur in the present string, the table $h_{fin,1}$ can be applied, which immediately introduces some symbols from $\Delta' \cup \{L\}$ whose occurrence prevents further applications of the tables h_ρ.
- In order to get rid of the non-terminal L, the last table $h_{fin,2}$ has to be applied. At this stage, the string contains only terminal letters and the symbol L.

Observe that, in case of propagating systems and λ-free transducers, the last table (which is the only one containing erasing productions) can be omitted. □

Theorem 6. *Let $k \geq 1$. The family of languages generated by propagating partition-k-limited ET0L systems is an AFL. The family of languages generated by partition-k-limited ET0L systems is a full AFL.*

Proof. By our previous statements, it remains to be proved that the considered language families are closed under iteration-$^+$. This can be done as in [22, Theorem 4.14]. □

The following construction generalizes [13, Satz 6.3.4] to propagating systems.

Theorem 7. *Let $k \geq 2$. For every given [propagating] partition-1-limited ET0L system there exists a [propagating] partition-k-limited ET0L system and vice versa.*

Proof. '$\Longleftarrow$': This assertion is seen combining Theorem 2 with Lemma 3.

'$\Longrightarrow$': Let $G = (V, \Delta, \{P_1, \dots, P_r\}, \omega, \pi, 1)$ be a [propagating] partition-1-limited ET0L system with $\pi = \{\pi_1, \dots, \pi_s\}$. Let $k \geq 2$. We will construct a partition-k-limited ET0L system $G' = (V', \Delta', H, \omega', \pi', k)$ generating

$$\{\#_1^{k-1} \cdots \#_s^{k-1} \#^s \$[\$, 0]\} L(G) ,$$

where the symbols $\#_\kappa$, $[\$, 0]$, and \$ are new symbols, not contained in Δ. Since partition-k-limited ETOL languages form an AFL, they are closed under left derivatives. Hence, by our construction, $L(G)$ is a partition-k-limited ETOL language.

We start with $\omega' = \#_1^{k-1} \cdots \#_s^{k-1} \#^s \$[\$, 0]\omega$. Let $A = \{\#_i \mid 1 \leq i \leq s\}$. Moreover, there are barred versions of the symbols in V and A, whose sets are also denoted by $\bar{V}$ and $\bar{A}$, respectively. So, let $\Delta' = \Delta \cup A \cup \{\$, \#, [\$, 0]\}$. Let $V' = \Delta' \cup \bar{V} \cup V \cup \bar{A} \cup \{F, \#'\} \cup \{\, [\$, i] \mid 1 \leq i \leq s + 1 \,\}$.

As partition, we take

$$\begin{aligned} \pi' = \{\, &\{\#_i\} \cup \{\bar{B} \mid B \in \pi_i\} \mid 1 \leq i \leq s \,\} \cup \{V\} \\ &\cup \{\, \{[\$, i] \mid 0 \leq i \leq s + 1\} \,\} \cup \{\, \{\#, F\} \,\} \cup \{\, \bar{A} \cup \{\#', \$\} \,\} . \end{aligned}$$

The tables are defined as follows:

$$\begin{aligned} h_c = {} & \{\, B \to \bar{B}, \bar{B} \to \bar{B} \mid B \in V \,\} \cup \{\, B \to B \mid B \in A \,\} \\ & \cup \{\, B \to F \mid B \in \bar{A} \,\} \cup \{[\$, 0] \to [\$, 1], [\$, 1] \to [\$, 1]\} \\ & \cup \{\, [\$, i] \to F \mid 2 \leq i \leq s + 1 \,\} \cup \{\$ \to \$, \# \to \#, \#' \to F, F \to F\}; \end{aligned}$$

for $1 \leq \rho \leq r$:

$$\begin{aligned} h_\rho = {} & \{\, \bar{B} \to v \mid B \to v \in P_\rho \,\} \cup \{\, B \to \bar{B} \mid B \in A \,\} \\ & \cup \{\, \bar{B} \to F \mid B \in A \,\} \cup \{\, B \to F \mid B \in V \,\} \\ & \cup \{[\$, 0] \to F, [\$, 1] \to [\$, 2], [\$, 2] \to F\} \\ & \cup \{\, [\$, i] \to F \mid 3 \leq i \leq s + 1 \,\} \cup \{\$ \to \$, \# \to \#, \#' \to F, F \to F\}; \end{aligned}$$

$$\begin{aligned} h'_c = {} & \{\, \bar{B} \to B, B \to B \mid B \in V \,\} \\ & \cup \{\, B \to F \mid B \in A \,\} \cup \{\, B \to B \mid B \in \bar{A} \,\} \\ & \cup \{[\$, 0] \to F, [\$, 1] \to F, [\$, 2] \to [\$, 2]\} \\ & \cup \{\, [\$, i] \to F \mid 3 \leq i \leq s + 1 \,\} \\ & \cup \{\# \to \#', \$ \to \$, \#' \to \#', F \to F\}; \end{aligned}$$

for $1 \leq i \leq s$:

$$\begin{aligned} g_i = {} & \{\, \bar{B} \to F \mid B \in V \,\} \cup \{\, B \to B \mid B \in V \,\} \\ & \cup \{\, \bar{B} \to B \mid B \in A \,\} \cup \{\, B \to B \mid B \in A \,\} \\ & \cup \{[\$, i + 1] \to [\$, (i + 2) \bmod (s + 2)]\} \\ & \cup \{\#' \to \#, \# \to \#, \$ \to F, F \to F\}. \end{aligned}$$

Assume we want to simulate a derivation step $w \Rightarrow v$ of the original grammar G. In G', this corresponds to a number of steps, deriving

$$\#_1^{k-1} \cdots \#_s^{k-1} \#^s \$[\$,0] w \stackrel{*}{\Rightarrow} \#_1^{k-1} \cdots \#_s^{k-1} \#^s \$[\$,0] v \ .$$

When finishing with the colouring process via several applications of h_c, we have a string of the form $\#_1^{k-1} \cdots \#_s^{k-1} \#^s \$[\$,1]\bar{w}$, where $\bar{w}$ is obtained from w by barring all symbols in w. For simulating one application of P_ρ, we have to apply h_ρ. Now, there is a string of the form $\bar{\#}_1^{k-1} \cdots \bar{\#}_s^{k-1} \#^s \$[\$,2]v'$, where some symbols in v are barred, some are not. Several applications of h'_c lead to a string $\bar{\#}_1^{k-1} \cdots \bar{\#}_s^{k-1} \#'^s \$[\$,2]v$. So, we have to check whether we correctly selected one occurrence of a symbol from part π_i (if possible) and $(k-1)$ occurrences of $\#_i$ when applying h_ρ. A possible error would have occurred if we selected more than one occurrence of a symbol from part π_i and, accordingly, less than $(k-1)$ occurrences of $\#_i$ when applying h_ρ. (Furthermore, we check whether there are still some barred symbols from V.) In the errorfree case, there are $s \cdot (k-1) + s + 1 = s \cdot k + 1$ occurrences of symbols from the part $\bar{A} \cup \{\#', \$\}$. Using subsequently the tables $g_1, \ldots, g_s$ (this is enforced by the marker $[\$, i]$), we can check this, since the production $\$ \to F$ is always applicable. Now, we should have arrived at a string of the form $\#_1^{k-1} \cdots \#_s^{k-1} \#^s \$[\$,0]v$. □

Since the partition-k-limited ETOL system constructed in the last theorem is deterministic if the given partition-1-limited ETOL system is deterministic, combining Theorems 2 and 7, we obtain our main result:

Theorem 8. *Let $k, k' \geq 1$, $L \subseteq V^*$. The following assertions are equivalent:*

- *L is generated by a [propagating] partition-k-limited ETOL system.*
- *L is generated by a [propagating] deterministic partition-k'-limited ETOL system.*
- *L is generated by a [non-erasing] programmed grammar with appearance checking.* □

As a special case, this last theorem yields [13, Satz 6.3.5].

Corollary 9. *There are context-sensitive languages which cannot be generated by partition-limited propagating ET0L systems.*

Proof. This follows immediately from our theorem, keeping in mind the strict inclusion of $\mathcal{L}(\mathrm{P}, \mathrm{CF}-\lambda, \mathrm{ac})$ within the family of context-sensitive languages, as proved by Rosenkrantz [20]. □

Corollary 10. *The family of languages generated by partition-limited propagating ET0L systems is strictly included in the family of languages generated by partition-limited ET0L systems.* □

As mentioned in the introduction, partition-k-limited Lindenmayer systems are a generalization of both k-limited and uniformly k-limited Lindenmayer systems. Therefore, it is of interest to see whether or when these natural subclasses coincide with the class $\mathcal{L}(\mathrm{P}, \mathrm{CF}[-\lambda], \mathrm{ac})$ of languages generable by partition-k-limited [propagating] ET0L systems.

Corollary 11. – *The class of* 1*-limited [propagating] ET0L languages (which equals the class of languages generated by [non-erasing] programmed grammars with unconditional transfer, see [5, 10]) coincides with the class of partition-*1*-limited [propagating] ET0L languages if and only if the class of* 1*-limited [propagating] ET0L languages is closed under left derivatives.*
- *The class of k-limited ET0L languages coincides with the class of partition-k-limited ET0L languages if and only if the class of k-limited ET0L languages is closed under intersection with regular sets.*
- *The class of uniformly k-limited [propagating] ET0L languages is strictly included in the class of partition-k-limited [propagating] ET0L languages.*

Proof. The first two assertions basically follow from [7], for the propagating case refer to [10].

The third assertion follows from the fact that uniformly limited [propagating] systems can be simulated by [non-erasing] programmed grammars without appearance checking [11], and this family of languages is strictly included in $\mathcal{L}(\mathrm{P}, \mathrm{CF}[-\lambda], \mathrm{ac})$, see [8, 9, 14, 15]. □

Observe that the second assertion is only known in the case admitting erasing productions.

3 Discussion

In addition, let us mention that there exists still another proof of [13, Satz 6.3.3], namely, since partition-limited ET0L languages are closed under intersection of regular sets [13, Satz 6.3.1] and they are containing the 1lET0L languages, by [7, Remark 4.2.2] (which readily transfers to this case), they characterize the recursively enumerable sets. A similar argument applies to the propagating case, too, cf. [10].

It has been shown by Gärtner [13, Satz 6.4.1] that two tables suffice in order to generate every partition-limited ET0L language. The proof — which is basically an adaptation of the proof given in [25] in the case of k-limited ET0L systems — carries over to the propagating case as well.

The question whether systems with just one table (E0L systems) are already sufficient to generate every partition-limited ET0L language remains open, both in the propagating and in the nonpropagating case. For limited ET0L system, this problem was solved by Wätjen [24].

In this connection, it would be interesting to know whether partition-limited E0L systems are strictly more powerful than their natural subclasses, the limited E0L systems and uniformly limited E0L systems, respectively. Possible candidates would be $\{a^{2^n} \mid n \geq 0\}$ and $\{a^{n^2} \mid n \geq 1\}$. These languages cannot be generated neither by limited E0L systems [24] nor by uniformly limited E(T)0L systems, see [11, 14].

There is an interesting connection between partition-1-limited ET0L systems and cooperating distributed grammars with prescribed teams, see [12, 16, 19]. Since partitions on the alphabet implicitly impose a partition on the tables, which could be viewed as sets of rule sets in such a way, tables correspond to teams.

We only sketch some differences in the following:

- A team is applied by applying exactly one element of the team members (which are sets of productions) in parallel. On the contrary, in partition-1-limited ET0L systems, a "team" (= table) is still applicable when some members are not applicable, since none of the left-hand sides of their productions is present in the current string.
- Teams in cooperating distributed grammars with prescribed teams may be formed quite freely, while the teams in partition-1-limited ET0L systems are formed once and forever, due to the partition of the alphabet.
- There is no such thing as the completeness condition in the team formation. Especially, terminal symbols are not rewritten in cooperating distributed grammars with prescribed teams.
- This last seemingly minor difference is the main reason why there is nothing like the t-mode in partition-1-limited ET0L systems. More precisely, only the star-mode (a team may work on the sentential form as long as it likes) has been considered in partition-1-limited ET0L systems (here, we adopt the notions from the theory of cooperating distributed grammar systems, see, e.g., the monograph [3]).
- As regards the obtained language classes, we can state the maybe surprising fact that cooperating distributed grammars with prescribed teams working in the star-mode (characterizing the programmed languages without appearance checking) are strictly contained in the partition-1-limited ET0L languages (characterizing the programmed languages with appearance checking) due to [8, 9, 14, 15].

A similar comment applies to so-called stratified grammar systems, see [2, 4].

Finally, let us mention that intuitively there is a close connection between partition-limited systems and grammars controlled by a bicoloured digraph with unconditional transfer, see [10]. In both cases, we obtain a characterization of appearance checking via unconditional transfer due to the applicability definition of rule sets (instead of single production as in limited systems or, again equivalently, in programmed grammars with unconditional transfer). This might be a third idea to prove [13, Satz 6.3.3] along the lines of [5].

References

1. J. Berstel. *Transductions and Context-Free Languages*, volume 38 of *LAMM*. Stuttgart: Teubner, 1979.
2. E. Csuhaj-Varjú. Grammar systems: a multi-agent framework for natural language generation. In Gh. Păun, editor, *Mathematical Aspects of Natural and Formal Languages*, volume 43 of *World Scientific Series in Computer Science*, pages 63–78. Singapore: World Scientific, 1994.
3. E. Csuhaj-Varjú, J. Dassow, J. Kelemen and Gh. Păun. *Grammar Systems: A Grammatical Approach to Distribution and Cooperation.* London: Gordon and Breach, 1994.
4. E. Csuhaj-Varjú, J. Dassow, J. Kelemen and Gh. Păun. Stratified grammar systems. *Computers and Artificial Intelligence*, 13(5):409–422, 1994.
5. J. Dassow. A remark on limited 0L systems. *J. Inf. Process. Cybern. EIK*, 24(6):287–291, 1988.
6. J. Dassow and Gh. Păun. *Regulated Rewriting in Formal Language Theory*, volume 18 of *EATCS Monographs in Theoretical Computer Science.* Berlin: Springer, 1989.

7. H. Fernau. Membership for 1-limited ET0L languages is not decidable. *J. Inf. Process. Cybern. EIK*, 30(4):191–211, 1994.
8. H. Fernau. Observations on grammar and language families. Technical Report 22/94, Universität Karlsruhe, Fakultät für Informatik, August 1994. Most of this report will appear in Fundamenta Informaticae in 1996.
9. H. Fernau. A predicate for separating language classes. *EATCS Bulletin*, 56:96–97, June 1995.
10. H. Fernau. On unconditional transfer. In W. Penczek and A. Szalas, editors, *Proceedings of MFCS'96*, volume 1113 of *LNCS*, Berlin: Springer, 1996.
11. H. Fernau and H. Bordihn. Remarks on accepting parallel systems. *International Journal of Computer Mathematics*, 56:51–67, 1995.
12. R. Freund and Gh. Păun. A variant of team cooperation in grammar systems. *Journal of Universal Computer Science*, 1:105–130, 1995.
13. S. Gärtner. *Partitions-limitierte Lindenmayer-Systeme.* Berichte aus der Informatik. Aachen: Shaker-Verlag, 1995. (Dissertation Technische Universität Braunschweig.) The basic results of this thesis appeared under the title "On partition limited 0L systems". In J. Dassow, G. Rozenberg and A. Salomaa, editors, *Developments in Language Theory II*, pages 230–236. Singapore: World Scientific, 1996.
14. D. Hauschildt and M. Jantzen. Petri net algorithms in the theory of matrix grammars. *Acta Informatica*, 31:719–728, 1994.
15. F. Hinz and J. Dassow. An undecidability result for regular languages and its application to regulated rewriting. *EATCS Bulletin*, 38:168–173, 1989.
16. L. Kari, A. Mateescu, Gh. Păun and A. Salomaa. Teams in cooperating distributed grammar systems. *Journal of Experimental and Theoretical AI*, 7:347–359, 1995.
17. A. Lindenmayer. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology*, 18:280–299, 1968.
18. A. Lindenmayer. Developmental systems without cellular interactions, their languages and grammars. *Journal of Theoretical Biology*, 30:455–484, 1971.
19. Gh. Păun and G. Rozenberg. Prescribed teams of grammars. *Acta Informatica*, 31:525–537, 1994.
20. D. J. Rosenkrantz. Programmed grammars and classes of formal languages. *Journal of the Association for Computing Machinery*, 16(1):107–131, 1969.
21. P. M. B. Vitányi. Development, growth and time. In G. Rozenberg and A. Salomaa, editors, *The Book of L*, pages 431–444. Berlin: Springer, 1985.
22. D. Wätjen. k-limited 0L systems and languages. *J. Inf. Process. Cybern. EIK*, 24(6):267–285, 1988.
23. D. Wätjen. On k-uniformly-limited T0L systems and languages. *J. Inf. Process. Cybern. EIK*, 26(4):229–238, 1990.
24. D. Wätjen. A weak iteration theorem for k-limited E0L systems. *J. Inf. Process. Cybern. EIK*, 28(1):37–40, 1992.
25. D. Wätjen and E. Unruh. On the degree of synchronization of klET0L systems. *Information Processing Letters*, 29:87–89, 1988.
26. D. Wätjen and E. Unruh. On extended k-uniformly-limited T0L systems and languages. *J. Inf. Process. Cybern. EIK*, 26(5/6):283–299, 1990.

Acknowledgements

The work was supported by Deutsche Forschungsgemeinschaft DFG La 618/3-1.