INTERACTIVE L SYSTEMS WITH ALMOST INTERACTIONLESS BEHAVIOUR[†]

by

K. Culik II[*]

and

J. Karhumäki[⊕]

Research Report CS-78-47

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

December, 1978

[*] Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

[⊕] Department of Mathematics
University of Turku
Turku, Finland

[†] This research was supported by the Natural Sciences and Engineering Council of Canada, Grant No. A7403.

A B S T R A C T

A restricted version of interactive L systems is introduced. A P2L system is called an essentially growing 2L-system (e-G2L system) if every length-preserving production is interactionless (context-free). It is shown that the deterministic e-G2L systems can be simulated by codings of propagating interactionless systems, and that this is not possible for the nondeterministic version. Some interesting properties of e-GD2L systems are established, the main result being the decidability of the sequence equivalence problem for them.

1. Introduction

The area of L systems has had a rapid growth (see Rozenberg and Salomaa, 1976), however this is mainly due to their mathematical investigation rather than their biological application. Lindenmayer (1977) has stressed the importance of determinism in developmental models. For this reason the most important systems for a biologist are DOL and DIL systems , the deterministic versions of the basic interactionless and interactive systems.

Since the latter are much more powerful than the former it seems to us important to investigate systems of intermediate capability. One way to get such systems is if we allow interaction only when cells are dividing but not when they are merely changing states. Quite surprisingly in the case of propagating systems (no cells dying) the behaviour of such systems will be shown to be closer to the interactionless systems rather than to the interactive ones.

Our results seem to be well motivated mathematically, too. It is well known, see Baker's Theorem in (Harrison , 1978), that certain restrictions on the form of productions of a context-sensitive grammar make the grammar essentially lose its "context-sensitiveness", i.e. to generate a context-free language. We will introduce a different kind of restriction on the form of productions of a deterministic context-sensitive parallel rewriting system (D2L system) which has essentially the same effect, that is, it makes an interactive (context-sensitive) system behave in an almost interactionless (context-free) manner. Similarly, as in the case of sequential context-sensitive grammars, the restriction seems to be a mild one, and therefore the obtained results are rather surprising. We say that a D2L system is essentially growing (an e-GD2L system) if the system is propagating (nonerasing) and each of its productions which is actually context-sensitive is strictly growing. In other words an e-GD2L system is a PD2L system such that each of its nongrowing productions is actually context independent.

As our basic result we show that every e-GD2L system can be simulated by a coding of a propagating DOL system (CPDOL system). Then it is easy to show that both the languages and the sequences generated by e-GD2L systems are properly between those generated by PDOL and CPDOL systems. Hence each e-GD2L language can be generated by a nondeterministic context-free system with nonterminals (EOL system).

In section 4 we obtain several applications of the basic simulation result and the method of its proof. First we observe that the length sequence equivalence problem for e-GD2L systems is decidable. Then we demonstrate that despite the fact that e-GD2L growth functions are the same as the PDOL growth functions, it is possible to realize some growth functions with a considerably smaller number of symbols by e-GD2L systems than by PDOL systems. We show a sufficient condition for the so called "cell number minimization problem" (see, Salomaa and Soittola, 1978, pg.116) to be decidable. These conditions are satisfied by e-GD2L systems, so the cell number minimization problem for them is decidable.

We conclude section 4 with the main result of this paper, namely, the decidability of the sequence equivalence problem for e-GD2L systems. Hence e-GD2L systems are the most complicated type of L systems known for which this important problem is decidable. Our result is somewhat surprising in the view that this problem is undecidable for PD1L-systems (Vitanyi, 1974).

In the last section we show that Theorem 1 cannot be extended to nondeterministic e-G2L systems. This extension would mean that e-G2L languages are included in CPOL languages, therefore also in EOL languages (see Rozenberg and Salomaa, 1976.) However, this is impossible since we will show that each ETOL language can be expressed as $h(L) \cap R$ for some homomorphism h, e-G2L language L and regular set R.

2. Preliminaries and basic definitions

A 2L system is a triple $G = \langle\Sigma,P,w\rangle$ where Σ is a non-empty alphabet, w is an element of Σ^* and P is a finite relation from $V = \{\$\} \times \Sigma \times \{\$\} \cup \{\$\} \times \Sigma^2 \cup \Sigma^2 \times \{\$\} \cup \Sigma^3$ into Σ^* satisfying the following completeness condition: For each $u \in V$ there exists at least one v in Σ^* such that $(u,v) \in P$. An element (u,v) of P is called a production and usually written $u \to v$; the letter $\$$ not in V is called the environmental symbol. The relation $\Rightarrow_G$ (or $\Rightarrow$ in short) on Σ^* is defined as follows. For words x and y $x \Rightarrow y$ holds true if and only if one of the following conditions is satisfied: (i) $x \in \Sigma$ and $(\$,x,\$) \to y \in P$, (ii) $x = x_1x_2$, $y = y_1y_2$, with $x_1,x_2 \in \Sigma$, $y_1,y_2 \in \Sigma^*$, and $(\$,x_1,x_2) \to y_1$, $(x_1,x_2,\$) \to y_2 \in P$, (iii) $x = x_1 \ldots x_n$, $y = y_1 \ldots y_n$, with $n \geq 3$, $x_1,\ldots,x_n \in \Sigma$, $y_1,\ldots,y_n \in \Sigma^*$ and $(\$,x_1,x_2) \to y_1$, $(x_1,x_2,x_3) \to y_2,\ldots,(x_{n-2},x_{n-1},x_n) \to y_{n-1}$, $(x_{n-1},x_n,\$) \to y_n \in P$. Let $\Rightarrow^*$ be the transitive and reflexive closure of $\Rightarrow$. The language generated by G is $L(G) = \{x \mid w \Rightarrow^* x\}$.

In this paper propagating systems, i.e. systems where erasing productions are not allowed, are considered. Moreover, in most cases systems are assumed to be deterministic in the following sense. A 2L system $G = \langle\Sigma,P,w\rangle$ is deterministic (abbreviated a D2L system) if the relation P is a function from V into Σ^*. For a D2L system G the conditions $x \Rightarrow_G y$ and $x \Rightarrow_G y'$ imply $y = y'$, and hence G

defines the sequence

$$s(G) = w_0, w_1, \ldots$$

where $w_0 = w$ and $w_i \Rightarrow_G w_{i+1}$ for $i = 0,1,\ldots$. Such sequences are called D2L sequences. In the deterministic case we will also write $\delta(a,b,c) = d$ when $(a,b,c) \to d$.

Let $G = \langle\Sigma,P,w\rangle$ be a 2L system. A production $(x,a,y) \to \alpha$, with $x,y \in \Sigma \cup \{\$\}$, is called <u>context-free</u> if $\{(z,a,v) \to \alpha \mid z,v \in \Sigma \cup \{\$\}\} \subseteq P$. So the abbreviation $a \to \alpha$ for the context-free production $(x,a,y) \to \alpha$ can be used. The production of G , which are not context-free , are called <u>context-sensitive</u>. In the deterministic case we may also talk about context-free and context-sensitive letters.

For $w \in \Sigma^*$, $|w|$ denotes the length of w , for a set S $|S|$ denotes the cardinality of S. Now we introduce the basic notions of this paper.

<u>Definition</u> A 2L system $G = \langle\Sigma,P,w\rangle$ is <u>strictly growing</u> (an s-GL2 system) iff $|v| \geq 2$ for each $u \to v$ in P . System G is <u>essentially growing</u> 2L system (e-GL2 system) iff it is propagating and $|v| \geq 2$ for each context-sensitive production $u \to v$ in P . We will be mainly interested in deterministic e-GL2 systems (e-GD2L systems).

Every production of an s-G2L system must be length-increasing, while an e-G2L system may have length-preserving productions, if they are context-free. Therefore, any propagating context free productions are allowed in e-G2L systems and thus the e-GD2L systems include all PDOL systems. Also the s-GD2L systems are a special case of the e-GD2L systems and we introduce them mainly to facilitate the explanation of some proof techniques in a simple setting before a general proof. However, all the results concerning deterministic systems will be proved for the more general case of e-GD2L systems.

Throughout this paper we use the basic notions and results of formal language theory and L systems, we refer the reader, e.g., to Salomaa (1973) and Rozenberg and Salomaa (1976). In particular, by a coding we mean a letter-to-letter homomorphism. Moreover, the maximal prefix (resp. suffix) of a string x not longer than k is denoted by $pref_k(x)$ (resp. $suff_k(x)$).

3. The interactionless simulation of restricted interaction

In this section we consider essentially (strictly) growing D2L systems. We first observe that s-GD2L systems can be simulated by CPDOL systems in the sense that any s-GD2L sequence is obtained as a coding of a PDOL sequence. The result is seen as follows. Let $G = \langle\Sigma,\delta,w\rangle$ be a s-GD2L system and let $(b,c,d) \to \gamma$ be one of its productions. Now, we consider the letter c in the context ...abcde... for some letters a,b,d and e. We know how to rewrite c in that context (for this purpose the context ...bcd... is sufficient) but we also know what are the neighbours of the result (i.e. γ) of the length two. This follows since they are determined by words $\delta(a,b,c)$ and $\delta(c,d,e)$. So the use of quintuples (a,b,c,d,e) makes it possible to simulate the derivations of G by a PDOL system. We omit the details since the result is only a special case of the following stronger theorem.

Theorem 1 For any e-GD2L system G there exist a PDOL system G' and a coding c such that $s(G) = c(s(G'))$ and hence also $L(G) = c(L(G'))$.

Proof Let $G = \langle\Sigma,\delta,w\rangle$ and let $\delta(x)$ be the word derived from x in a fixed context. Moreover, let $\overline{\delta}(x)$ (resp. $\overline{\delta}_\ell(x)$ or $\overline{\delta}_r(x)$) denote the subword of $\delta(x)$ which can be obtained independently of the (resp. left or right) neighbours of x . So the meaning of $\overline{\delta}(x)$ is always clear while the notation $\delta(x)$ can be used only when the neighbours of x are fixed for our considerations. Further let

$$\Sigma_1 = \{a \in \Sigma \mid a \text{ is context-free}\}$$

and

$$\Sigma_2 = \{a \in \Sigma_1 \mid \text{there exists an infinite sequence } a = a_0, a_1, \ldots \text{ such that } a_i \in \Sigma_1 \text{ and } a_{i+1} \text{ occurs in } \delta(a_i) \text{ for all } i = 0,1,\ldots\}.$$

We also use the following notation. Let $x \in \Sigma^*$. Then $rc(x)$ is the shorter of the following two words: (i) $pref_{2|\Sigma|}(x)$, (ii) the shortest prefix of x ending with a letter from Σ_2. The notation of $\ell c(x)$ is defined similarly when using sufficies.

Now, we claim that for any $a \in \Sigma$ and $x,y \in \Sigma^*$, with $|x|,|y| \geq 2\,|\Sigma|$, the sequence

$$\overline{\delta}^n(\ell c(x)\; a\; rc(y)) , \qquad n \geq 0$$

is infinite, i.e., the derivation does not terminate (because of the lack of information about neighbours). In other words the above means that the immediate neighbours of all descendants of a are uniquely determined by the words not longer than $2\,|\Sigma|$ (i.e., $\ell c(x)$ and $rc(y)$).

Clearly, because of symmetry, it is sufficient to prove the claim for the sequence obtained from the word $a\; rc(y)$ with the further assumption that the left neighbour of a and its descendants are always known.

First, assume that $rc(y)$ ends with an element from Σ_2. Then the word $rc(\delta(rc(y)))$ is either of the length $2|\Sigma|$ or it ends with a letter in Σ_2. Hence, the right neighbouring word of $\delta(a)$ is obtained exactly in the form we want, i.e., either ending with a letter from Σ_2 or having length $2|\Sigma|$.

Secondly, assume that $rc(y)$ does not contain any letters from Σ_2. There are two subcases. Case A. A symbol from $\Sigma - \Sigma_1$ occurs in $rc(y)$ elsewhere than as the last symbol. This case causes no problems. Namely, writing $rc(y) = zc$, with $c \in \Sigma$, we see that $\delta(z)$ is at least of the length $2|\Sigma|$, and so the neighbouring word for $\delta(a)$ in the form we want is determined by $rc(y)$ independently of its right neighbours.

Case B. All letters (except possibly the last one) of $rc(y)$ are from Σ_1. Then the length of the word derived from $rc(y)$ without knowing its right neighbour may decrease by one (and hence we possibly do not get the neighbouring word for $\delta(a)$ in the form we want). This may happen again during the next derivation step. However, let us see what happens when we take $|\Sigma|-1$ steps. If a letter from Σ_2 appears during these steps, then the right neighbouring word for a descendant of a has been found. If no letters from Σ_2 appear, then each of the $|\Sigma|$ first symbols of $rc(y)$ generates during these steps at least one letter from $\Sigma - \Sigma_1$. Hence

$$\delta^{|\Sigma|}(\mathrm{pref}_{|\Sigma|}(rc(y)))$$

is of the length at least $2\,|\Sigma|$ and since it is a prefix of $\overline{\delta}_r^{\,|\Sigma|}(rc(y))$ we get a "periodic situation" guaranteeing that the right neighbouring word for descendants of a are really determined by $rc(y)$ independently of its right neighbours (although now the neighbouring words are not necessarily obtained in the same form as before; they may be shorter).

Hence the claim is established and we complete the proof of the theorem.

The PDOL system G' is defined as follows. Its alphabet is

$$V = \left(\bigcup_{i=1}^{2|\Sigma|} (\Sigma \cup \{\$\})^i\right) \times \Sigma \times \left(\bigcup_{i=1}^{2|\Sigma|} (\Sigma \cup \{\$\})^i\right) .$$

The axiom is obtained from the word $\$^{2|\Sigma|}w\,\$^{2|\Sigma|}$ by forming first its all subwords of length $4\,|\Sigma| + 1$, then writing them as elements of V and finally catenating letters thus obtained without changing their order (that is the order of subwords in $\$^{2|\Sigma|}w\,\$^{2|\Sigma|}$). The productions for G' are defined in the following way. If G contains the production

$$(a,b,c) \to b_1 \ldots b_n \qquad \text{with } b_1,\ldots,b_n \in \Sigma ,$$

then G' contains for all x and y in $\displaystyle\bigcup_{i=1}^{2|\Sigma|-1} \Sigma^i$ the production

$$(xa,b,cy) \to (\ell c(\overline{\delta}_\ell(xa)), \quad b_1, \; rc(b_2 \ldots b_n \overline{\delta}_r(cy)))$$
$$(\ell c(\overline{\delta}_\ell(xa)b_1), \; b_2, \; rc(b_3 \ldots b_n \overline{\delta}_r(cy))) \ldots$$
$$(\ell c(\overline{\delta}_\ell(xa)b_1 \ldots b_{n-1}), \; b_n, \; rc(\overline{\delta}_r(cy))).$$

This definition contains also the simulation of context-free productions of G, but does not contain productions for letters containing the environmental symbol \$. The definition of these productions is similar. (In fact, the situation is even easier, since δ can be used instead of above $\overline{\delta}_r$ and $\overline{\delta}_\ell$).

Finally, the coding $c : V^* \to \Sigma^*$ is defined by

$$c(x,a,y) = a$$

for all $a \in \Sigma$ and $x,y \in \bigcup_{i=1}^{2|\Sigma|} (\Sigma \cup \{\$\})^i$. Then clearly

$$s(G) = c(s(G'))$$

and the theorem is proved. □

The PDOL system G' above contains many useless letters, i.e. letters which are never encountered in the rewriting process. Of course the reduced system, i.e. the system without useless letters, can be found simply starting from the axiom of G' and defining the productions step by step according to the proof of Theorem 1.

Later on in (Theorem 7) it will be shown that the determinism is an essential assumption for Theorem 1 to hold. Also Theorem 1 cannot be generalized for D2L systems with strictly growing context-sensitive productions and arbitrary (possibly erasing) context-free productions. This is seen as follows. Let G be any PD2L system. We show that it can be simulated, in a sense, by a D2L system with strictly growing context-sensitive productions and arbitrary context-free productions. Such a system G' is defined in the following way. For any length-preserving production $(a,b,c) \to d$ in G

G' contains productions

$$(a,b,c) \to \bar{d}\, a_\lambda \ , \qquad \bar{d} \to d \quad \text{and} \quad a_\lambda \to \lambda$$

where $\bar{d}$ denotes the "bared copy" of d. Further for any production

$$(a,b,c) \to b_1 \dots b_n \quad \text{in } G \ , \quad \text{with } n \geq 2 \ ,$$

G' contains productions

$$(a,b,c) \to \bar{b}_1 \dots \bar{b}_n \quad \text{and} \quad \bar{b}_i \to b_i \quad \text{for } i = 1,\dots,n \ .$$

Then clearly

$$L(G) = L(G') \cap \Sigma^*$$

where Σ denotes the alphabet of G. Since we may choose L(G) not to be an EOL language we may also choose L(G') not to be an EOL language. Hence G' cannot be simulated by any DOL system in the sense of Theorem 1.

Let us denote by $\mathcal{L}_{e\text{-GD2L}}$ (resp. $\mathcal{S}_{e\text{-GD2L}}$) the family of languages (resp. sequences) generated by e-GD2L systems. Then using Theorem 1 we get

Theorem 2 $\mathcal{L}_{PDOL} \subsetneq \mathcal{L}_{e\text{-}GD2L} \subsetneq \mathcal{L}_{CPDOL}$

and

$$S_{PDOL} \subsetneq S_{e\text{-}GD2L} \subsetneq S_{CPDOL} .$$

Proof All inclusions follow from Theorem 1 and the definition of the e-GD2L system. That the first inclusions are proper follows from Example in the next section. The strictness of the second inclusions, in turn, are seen by considering some language over a^*, for example the language $\{a^2, a^4, a^8\} \cup \{a^{2^n} \mid n \geq 5\}$ and the corresponding sequence. □

4. Applications of the simulation result

After establishing Theorem 1 and the position of our language family $\mathcal{L}_{e\text{-GD2L}}$ within the hierarchy of L families we now derive some interesting properties of our systems. We first observe

Theorem 3 The length sequence equivalence problem for e-GD2L systems is decidable.

Proof The result follows from Theorem 1 and the fact that the problem is decidable for PDOL systems, see Salomaa and Soittola (1978). □

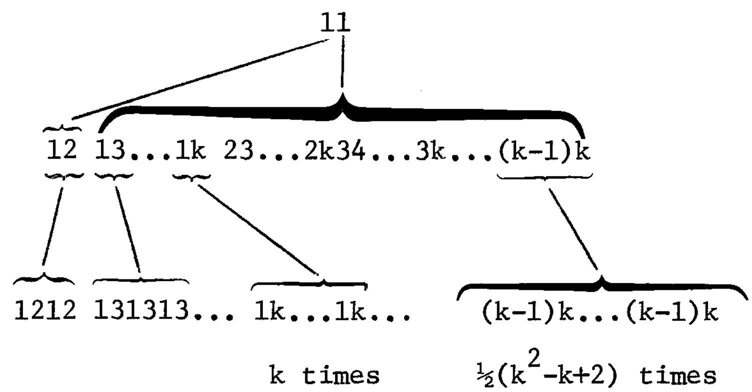
One of the consequences of Theorem 1 is that any e-GD2L growth function is a PDOL growth function, too. However, as is seen in the next example, the number of letters needed to realize a given function by an e-GD2L system may be much smaller than that needed to realize the same function by a PDOL system.

Example Let us define an e-GD2L system (or in fact an s-GD2L) G as follows. Its alphabet equals $\{1,\dots,k\}$ and its axiom is 11. To define the productions let γ be the function which gives the lexicographic order of the set $\{(i,j) \mid i,j = 1,\dots,k\ ,\ i < j\}$, i.e. $\gamma(1,2) = 1$, $\gamma(1,3) = 2,\dots,$ $\gamma(k-1,k) = \frac{1}{2}k(k-1)$. Now the productions of G are as follows.

$$(\$,1,1) \to 12 \ ,$$
$$(1,1,\$) \to 13\ldots 1k23\ldots 2k\ldots(k-1)k \ ,$$
$$(-,i,j) \to ij \ , \qquad \text{for} \quad i < j \ ,$$
$$(i,j,-) \to (ij)^{\gamma(i,j)} \ , \quad \text{for} \quad i > j \ .$$

where $-$ denotes that the element there is arbitary.

So the derivation starts as follows



If we denote by f the growth function just defined we get

$$f(n) = 2 \sum_{j=1}^{\frac{1}{2}(k^2-k)} (j+1)^n \ , \qquad \text{for} \quad n \geq 1 \ .$$

This formula implies that any PDOL (or DOL) system generating the growth function of G must contain at least $\frac{1}{2}(k^2-k)$ letters. However, f is realized by an e-GD2L system with k letters only.

The above example gives the motivation for the following definition. Let $\mathcal{C}$ be a class of deterministic L systems. <u>The cell number minimization problem</u> for $\mathcal{C}$ is the following :

Given an arbitrary function f realized by a system in $\mathcal{C}$. Is there an algorithm to find a system from $\mathcal{C}$ with an alphabet of the minimal cardinality such that its growth function equals f ?

Lemma 1 The cell number minimization problem is decidable for any class $\mathcal{C}$ of deterministic L systems satisfying the following two conditions:

(i) The growth equivalence problem is decidable in $\mathcal{C}$.

(ii) For any G in $\mathcal{C}$ one can effectively find a constant N_G, dependent on the size of the alphabet Σ of G only, such that for any $a \in \Sigma$ a occurs in $L(G)$ if and only if it occurs in the N_G first words generated by the system.

Proof Let H be an arbitrary system from $\mathcal{C}$ and let f be its (effectively given) growth function. We show that the number of "suitable candidates" to generate f with a smaller alphabet is finite and that the set of the candidates can be effectively found. First, the alphabet of a candidate is a subset of the alphabet of G (since we may always rename the letters). Secondly, the axiom of a candidate is of length $f(0)$, and hence the set of possible axioms is finite. Finally, the right-hand side of any production in a candidate system is at most of length $\max(f(0),\ldots,f(N_H))$. So, by (ii), the finite set of suitable candidates can be effectively found. Hence the lemma follows, since by the first condition we may test the equality of a candidate chosen and H. $\square$

Now, we are ready for

<u>Theorem 4</u> The cell number minimization problem for e-GD2L systems is decidable.

<u>Proof</u> The conditions of Lemma 1 are valid for e-GD2L systems. The first one follows from Theorem 3 and the second one from the proof of Theorem 1 (see also the discussion after Theorem 1). □

Now, we turn to consider the sequence equivalence problem for e-GD2L systems. In a subcase, i.e. in the case of s-GD2L systems, the decidability of this problem is a consequence of the simulation of these systems by PDOL systems and the decidability of the problem for PDOL systems. Indeed, two s-GD2L sequences s(G) and s(H) are equivalent if and only if the PDOL sequences s(G') and s(H'), where G' and H' are the "quintuple PDOL systems" simulating G and H, are equivalent.

For e-GD2L systems the situation is more complicated, mainly due to the fact that the letters of G' in the proof of Theorem 1 are not of uniform length as words of Σ^*. Hence, two e-GD2L systems may be equivalent although the corresponding simulating systems of Theorem 1 are not. Moreover, the proof of Theorem 1 indicates that it is probably impossible to find for an arbitrary e-GD2L system a simulating PDOL system with uniform length of letters, i.e. with n-tuples for a fixed n. However, through the following sequence of arguments, based on Theorem 1, we will show that the sequence equivalence problem for e-GD2L systems is decidable.

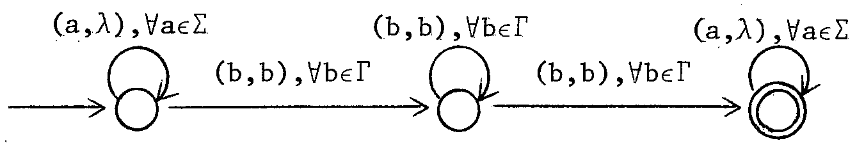
We start with few definitions. Let $G = \langle\Sigma,\delta,w\rangle$ be an e-GD2L system. Recalling that Σ_1 denotes the set of context-free letters of G we define

$$\Sigma_s = \{a \in \Sigma_1 \mid \text{there exists } t > 0 \text{ such that } \delta^t(a) = a\},$$
$$\Sigma_{ng} = \{a \in \Sigma_1 \mid \text{for all } t > 0 \ |\delta^t(a)| = 1\}$$
$$\Sigma_g = \Sigma - \Sigma_{ng}$$

Clearly $\Sigma_s \subseteq \Sigma_{ng}$. Letters in alphabets Σ_s, Σ_{ng} and Σ_g are called stable, nongrowing and growing, respectively. For an a in Σ_s let $[a]$ be the set of letters derived from a. Then Σ_s is a disjoint union of some such equivalence classes. A subalphabet $\Delta \subseteq \Sigma_s$ of G is called unbounded if for each natural number n and each $a \in \Delta$ there exists a subword $x \in \Delta^*$ of a word in $L(G)$ such that $\#_a(x) \geq n$, where $\#_a(x)$ denotes the number of a's in x. The maximal unbounded subalphabet is clearly unique and it is denoted by Δ_G. Clearly also the maximal unbounded subalphabet of G is a finite union of some equivalence classes of the form $[a]$.

Lemma 2 It is decidable whether an e-GD2L system contains a nonempty unbounded subalphabet. Moreover, the maximal unbounded subalphabet Δ_G can be effectively found.

<u>Proof</u> Let $G = <\Sigma,\delta,w>$ and let $\Gamma \subseteq \Sigma_s$. Define the gsm-mapping (Salomaa, 1973) g_Γ by the diagram



Then, clearly,

$$(*) \qquad g_\Gamma(L(G)) = \{x \in \Gamma^* \mid x \text{ is a subword of } L(G)\}$$

By Theorem 1, $L(G)$ is an EOL language, and so also $g_\Gamma(L(G))$ is an EOL language. Now the first sentence of the lemma follows since the finiteness of EOL languages is decidable and since G has an unbounded subalphabet if and only if some set of the form (*) is infinite.

The validity of the second sentence is seen as follows. For a given $\Gamma \subseteq \Sigma$ we first test whether (*) is infinite. If it is we search the maximal subset Γ' of Γ such that $g_h(g_\Gamma(L(G)))$ is finite, where g_h is the homomorphism from Γ^* into Γ'^* such that $h(a) = \lambda$ for $a \in \Gamma'$ and $h(a) = a$, otherwise. Clearly such a Γ' can be effectively found, and so we obtain an unbounded alphabet $\Gamma - \Gamma'$. Now, the maximal unbounded subalphabet can be found by checking all subsets Γ of Σ_s. □

Lemma 3 Let $G = \langle\Sigma,\delta,w\rangle$ be an e-GD2L system. One can effectively find a constant N such that any subword of a word in $L(G)$ longer than N contains an element from $\Sigma_g \cup \Delta_G$.

Proof As in the proof of Lemma 2 we may effectively find an EOL system generating the set

$$S = \{x \in (\Sigma-(\Sigma_g \cup \Delta_G))^* \mid x \text{ is a subword of } L(G)\}.$$

We claim that S is finite. From this the lemma follows since finite EOL language can be effectively found, and hence N is obtained as the maximal length of words in S.

To prove the claim we first observe that $\Sigma-(\Sigma_g \cup \Delta_G) = (\Sigma_{ng}-\Sigma_s) \cup (\Sigma_s-\Delta_G)$. Now the maximality of Δ_G (as unbounded subalphabet) implies the existence of k such that all subwords from $(\Sigma_s-\Delta_G)^*$ in $L(G)$ are of length at most k. So to establish the claim it suffices to show the existence of a constant m such that for any occurence $a \in \Sigma_{ng} - \Sigma_s$ in a word y in $L(G)$ there exists an occurence of $b \in \Sigma_g$ in y such that between a and b there are at most m letters. The existence of m is seen by looking at ancestors of a. Although we have context-sensitive rewriting we may do this by Theorem 1: We are actually looking for ancestors of a letter appearing in a certain context of length $2|\Sigma|$ from both sides. In this way we find a "recursive letter" c in the following sense. c appears twice in $s(G)$ in the same context of length $2|\Sigma|$ and $\delta^n(c)$ contains c

and $\delta^m(c)$ contains a for some $m,n > 0$ (here the meaning of δ must be understood as in the proof of Theorem 1). The letter $c \notin \Sigma_{ng}$ because $a \notin \Sigma_{ng}$. Hence c is a growing letter. Since it is also recursive in the above sense, it must produce (in the context we consider it) a growing letter in any string derived from it. So, in y there must be a growing letter not far from a. Making the above construction for all letters from $\Sigma_{ng} - \Sigma_s$ the required m can be found. Hence, the claim and also the lemma have been proved. □

<u>Lemma 4</u> Let Δ be an unbounded subalphabet of an e-GD2L system G. Then there exist a constant K and a finite set F such that any subword $x \in \Delta^*$ in $L(G)$ longer than K is of the form $x = s\, t^n\, r\, u^m\, v$ for some natural n and m and s, t, r, u and v from F.

<u>Proof</u> Again the result follows from the proof of Theorem 1. Namely, by this, Lemma 4 is true if and only if it is true for PDOL systems. That Lemma 4 is valid for PDOL systems is clear. Indeed, any stable substring x in a fixed PDOL language $L(G)$ admits the factorization $x = s\, t^n\, r\, u^m\, v$ for some s, t, u and v not longer than $n\, H$ and r not longer than $m\, H^n$ where n is the cardinality of the alphabet of G, H is the maximal length of the right-hand sides of the productions and m is the length of the axiom. □

For a word x we denote by $\psi(x)$ its Parikh vector. The order relation $\geq$ on $\mathbb{N}^k$ is defined componentwise in the standard way.

Lemma 5 For any PDOL system $G = \langle\Sigma,\delta,w\rangle$ there exist constants t and r such that $\psi(\delta^{t+n}(w)) \leq \psi(\delta^{t+r+n}(w))$ for all $n \geq 0$. Moreover, there exists an upper bound M for $t + p$ independently of the axiom w.

Proof The first sentence follows from the König Infinity Lemma (see, e.g., Harrison, 1978). To prove the second sentence let t_a and r_a be the constants of the lemma for the system $G_a = \langle\Sigma,\delta,a\rangle$ where $a \in \Sigma$. Then we may choose $M = \max\{t_a \mid a \in \Sigma\} + \prod_{a\in\Sigma} r_a$.

For two e-GD2L systems G and H we introduce the notion of a common subalphabet. An alphabet Δ is called a common subalphabet of the pair (G,H) if $\Delta \subseteq \Sigma_{s,G} \cap \Sigma_{s,H}$ where $\Sigma_{s,G}$ and $\Sigma_{s,H}$ denotes the stable subalphabets of G and H respectively. The next lemma is essential for the proof of our main result.

Lemma 6 Let G and H be e-GD2L systems with $s(G) = s(H)$. Then the maximal unbounded subalphabet Δ of G is a common subalphabet of (G,H).

Proof Let $G = \langle\Sigma,\delta,w_0\rangle$ and $H = \langle\Sigma,\nu,w_0\rangle$. Further let $s(G) = s(H) = w_0,w_1,\ldots$. The sets of stable, nongrowing and growing letters with respect to H (resp. G) are denoted by $\Sigma_{s,H}$, $\Sigma_{ng,H}$ and $\Sigma_{g,H}$ (resp. $\Sigma_{s,G}$, $\Sigma_{ng,G}$ and $\Sigma_{g,G}$). The notations $\overline{\delta}$ and $\overline{\nu}$ are used as in the proof of Theorem 1. Finally, let $H' = \langle\Sigma',\nu',w_0'\rangle$ be the system of Theorem 1 simulating H. What we should show is that $\Delta \subseteq \Sigma_{s,H}$, i.e. that all letters in Δ are stable (also) with respect to H. So let $c \in \Delta$.

We begin the proof with the following observations: (i) There exist constants $p_1 > 0$ and p_2 such that, for any subword x of $L(G)$ belonging to Δ^*, either $\#_{c'}(x) \geq p_1|x|$ for some $c' \in [c]$ or $\#_{c'}(x) \leq p_2$ for all $c' \in [c]$. Moreover, the first alternative holds for infinitely many subwords $x \in \Delta^*$. (ii) There exists a constant q such that, for any subword x in $L(G)$, $|sub_\Delta(\overline{\delta}(x))| \leq |sub_\Delta(x)| + q$ where $sub_\Delta(u)$ denotes (any of) the longest subword(s) of u belonging to Δ^*. The fact (i) is an immediate consequence of the definition of Δ and Lemma 4. The second condition, in turn, follows from the proof of Lemma 3. Indeed, similar arguments as used there show that for any occurence of a letter a in $L(G)$ which satisfies the conditions $a \notin \Delta$, $\delta(a) \in \Delta^*$ (or if a is context-sensitive $\delta(a_1,a,a_2) \in \Delta^*$ where a_1 and a_2 are the neighbours of the occurrence of a), there exists "near to a" an occurrence of a letter b such that $\delta(b) \notin \Delta^*$ (or if b is context-sensitive $\delta(b_1,b,b_2) \notin \Delta^*$ where b_1 and b_2 are the neighbours of b).

Now we fix $x \in \Delta^*$ satisfying the following three conditons: (a) x is a subword in $L(G)$, i.e. $w_n = yxz$ for some natural n and words $y,z \in \Sigma^*$; (b) $\#_{c'}(x) \geq p_1|x|$ for some $c' \in [c]$; (c) there exist for $j = 0,\ldots,M$, where M is the constant of Lemma 5 to the PDOL system H', words $x_0,\ldots,x_{-M}$ such that $\overline{\nu}^j(x_{-j}) = x_{-j+1}$ and x_0 is a subword of x (cf. the discussion on ancestors in the proof of Lemma 3).

Two further assumptions concerning the above choice are made. First x is chosen to be the longest possible in w_n and satisfying (b). Secondly, the choice of $x_0, \ldots, x_{-M}$ is made in such a way that $|x_{-M}| \geq 4|\Sigma| + 1$ and $|x| - |x_0| \leq s_1$ for some constant independent of x. Note that x may be assumed to be arbitrarily long.

According to the proof of Theorem 1 any subword v in $s(H)$ defines a set of subwords in $s(H')$. Let $\langle v \rangle$ denote the maximal such subword. Clearly, there exists a constant s_2 independent of v such that $|\nu'^n(\langle v \rangle)| \geq |\overline{\nu}^n(v)| - s_2$ for all $n \leq 2M$. Let now t and r, with $t + r \leq M$, be the constants of Lemma 5 for H'. Then $\psi(\nu'^{t+(M-t)}(\langle x_{-M} \rangle)) \leq \psi(\nu'^{t+r+(M-t)}(\langle x_{-M} \rangle))$. Moreover, we may assume, possibly by choosing a greater M, that exactly the same components are nonzero on the both sides of the above inequality. So we conclude that x_0 $(= \overline{\nu}^M(x_{-M}))$ has a subword $\overline{x}_0$ such that $|\overline{x}_0| \geq |x_0| - s_2$ and $\psi(\overline{x}_0) \leq \psi(\overline{\nu}^r(x_0))$. Further the word $\overline{\nu}^r(x_0)$ has a subword $\overline{x}_r$ such that

$$(*) \qquad \overline{x}_r \in \Delta^* \quad \text{and} \quad |\overline{x}_r| \geq |\overline{\nu}^r(x_0)| - s_2.$$

We may also suppose that $\overline{x}_r$ satisfies the first inequality in (i). This follows since for c' in (b)

$$\#_{c'}(\overline{x}_r) \geq \#_{c'}(x) - s_3 \geq p_1 |x| - s_3 \geq p_2$$

where $s_3 = s_1 + 2s_2$,

if only x is long enough. So

(**) $\qquad \#_{c''}(\overline{x}_r) \geq p_1|\overline{x}_r| \quad$ for some $c'' \in [c]$.

Now we estimate the length of $\overline{x}_r$ under the assumption $c \notin \Sigma_{s,H}$. We first observe that if c itself is not in $\Sigma_{g,H}$ (i.e. it is in $\Sigma_{ng,H}$) it is anyway very near to a letter from $\Sigma_{g,H}$. This has been shown in the proof of Lemma 3, where the limit m is also given. From this we conclude that x_0 contains at least $(3m)^{-1}(p_1|x| - s_1)$ growing letters. So, assuming $r \geq |\Sigma|$,

$$\begin{aligned}|\overline{\nu}^r(x_0)| &\geq |x_0| - 2r + (3m)^{-1}(p_1|x| - s_1)\\ &\geq |x| - s_1 - 2r + (3m)^{-1}(p_1|x| - s_1)\\ &= \alpha|x| - \beta'\end{aligned}$$

for constants α and β' with $\alpha > 1$. Hence also

(***) $\qquad |\overline{x}_r| \geq \alpha|x| - \beta$

for some constants α and β with $\alpha > 1$.

Let now w be any subword of w_n such that its all subwords from Δ^* satisfy the second condition in (i). Then any subword of $\overline{\delta}^r(w)$ from Δ^* contains at most $p_2 + 2qr$ occurrence of any fixed letter from $[c]$, which means, by (*) and (**) that $\overline{x}_r$ is not a subword in $\overline{\delta}^r(w)$ if only x is long enough. So $\overline{x}_r$ in w_{n+r} must be derived according to G from a word w' in w_n containing a subword from Δ^* which satisfies the first inequality in (i). But the longest subword of Δ^* in w_{n+r} obtained in this way, is not longer than $|x| + 2rq$. This follows from (ii) and from the fact that x is the longest subword in w_n satisfying the first inequality of (i). Hence

$$|\bar{x}_r| \leq |x| - \gamma$$

for some constant γ. This is contradictory to (***) when x is chosen long enough. Hence $c \in \Sigma_{s,H}$ and the lemma has been proved.

Now we are ready for the main result of this paper.

<u>Theorem 5</u> The sequence equivalence problem for e-GD2L systems is decidable.

<u>Proof</u> Let $G = <\Sigma,\delta,w_0>$ and $H = <\Sigma,\nu,w_0>$ be two e-GD2L systems. We first determine the constants N_G and N_H of Lemma 3 and put $N = \max\{N_G,N_H,2|\Sigma|\}$. Then we find, by Lemma 2, the maximal unbounded subalphabets Δ_G and Δ_H , respectively. If they do not coincide then the systems are not sequence equivalent, by Lemma 6. So assume that $\Delta_G = \Delta_H$ and let this common subalphabet of (G,H) be denoted by Δ.

Now, we refer to the proof of Theorem 1. We define, like there, a system G' (resp. H') simulating G (resp. H). The definition of G' here differs from that in Theorem 1 only in the respect that Σ_2 is replaced by Δ and that "the contexts of letters" are now longer. More specifically, the operation rc is defined as follows. For $x \in \Sigma^*$ rc(x) is the shorter of the words: (i) $\mathrm{pref}_N(x)$, (ii) the shortest prefix of x ending with a symbol in Δ. Similarly, ℓc is defined when using sufficies. Otherwise, the definition of G' here is identical to that in Theorem 1.

If all the letters of the reduced versions of G' and H' are from

$$\Theta = \left(\Delta\left(\bigcup_{i=0}^{N-1} \Sigma^i\right), \Sigma, \left(\bigcup_{i=0}^{N-1} \Sigma^i\right)\Delta\right) \cup ((\Sigma \cup \{\$\})^N, \Sigma, (\Sigma \cup \{\$\}^N))$$

then we have no difficulties. Namely, e-GD2L systems G and H are sequence equivalent if and only if PDOL systems G' and H' are sequence equivalent, and this latter condition is decidable by Culik and Fris (1977).

Unfortunately, we cannot be sure that the letters of G' and H' are from Θ. However, this difficulty can be overgone by speed-up. To do this we define, for $j= 0,\ldots,|\Sigma|-1$, a PDOL system G_i' as follows. Its axiom is the $(i-1)$st word in $s(G')$ and its alphabet is the minimal one needed to define the productions below. For a symbol (x,a,y) in Θ which has the property that $\$$ does not occur either in x or in y, the production in G_i' is

$$\begin{aligned}(x,a,y) \to\ & (\ell c(\bar{x}),\ a_1,\ rc(a_2\ldots a_n\ \bar{y}))\\ & (\ell c(\bar{x}\ a_1),\ a_2,\ rc(a_3\ldots a_n\ \bar{y}))\ldots\\ & (\ell c(\bar{x}\ a_1\ldots a_{n-1}),\ a_n,\ rc(\bar{y})),\end{aligned}$$

where $a_1\ldots a_n = \delta^{|\Sigma|-1}(a)$, $\bar{x} = \bar{\delta}_\ell^{|\Sigma|-1}(x)$ and $\bar{y} = \bar{\delta}_r^{|\Sigma|-1}(y)$. (Here the notations of the proof of Theorem 1 are employed). Observe that if x and y above are subwords in $s(G)$, as can be assumed, then the letters on the right-hand side are from Θ. This follows since, by the proof of Theorem 1, $\bar{x}$ (resp. $\bar{y}$) is either of the length N or starts with a symbol from Δ (resp. is either of the length N or ends with a symbol from Δ). The productions for the letters containing environmental symbols are defined similarly. The systems H_i', for $i = 0,\ldots,|\Sigma|-1$, are defined analogoulsy. Finally, by c we mean a coding defined in the proof of Theorem 1.

Now the proof of the theorem is easily completed. Indeed,

$$\begin{aligned} s(G) &= s(H), && \text{if and only if ,} \\ c(s(G')) &= c(s(H')), && \text{if and only if ,} \\ c(s(G_i')) &= c(s(H_i')) && \text{for } i = 0,\ldots,|\Sigma|-1, \text{ if and only if} \\ s(G_i') &= s(H_i') && \text{for } i = 0,\ldots,|\Sigma|-1, \end{aligned}$$

where the last equivalence follows from the fact that the alphabets of G_i' and H_i' are subsets of Θ. □

The definition of G' in the proofs of Theorems 1 and 5 differ only slightly. So, one may ask why do we introduce these both simulating systems. The answer is that to be able to define G' of Theorem 5 some properties of e-GD2L systems, especially Lemma 6, are needed and to prove these some kind of PDOL simulation is already necessary.

By Theorems 3, 4 and 5 e-GD2L systems have many favourable properties which general D2L systems do not have. Indeed, the decision problems in Theorems 3 and 5 are undecidable for PD1L systems, see Vitanyi (1974). We also want to point out that e-GD2L systems form the most complicated class of deterministic L systems known to have the decidable equivalence problem. For CPD0L systems the problem is still open.

5. Nondeterministic case

In this final section we show that Theorem 1 cannot be extended to nondeterministic systems. This is somewhat surprising since one might expect (after Theorem 1) that the strict growth in connection with context-sensitive rules essentially "blocks" the interaction here, too. However, because of the parallelism in the rewriting process, it is possible to use nondeterministic strictly growing context-sensitive rules to control the derivation, as it is done in the next proof.

Theorem 6 For any ETOL language L there exist an s-G2L system G, homomorphism h and a regular set R such that

$$L = h(L(G)) \cap R .$$

Proof Without loss of generality we may assume that L is generated by a propagating system, see Rozenberg and Salomaa (1976). (We consider languages modulo the empty string λ). Let $L = L(H)$ for a propagating ETOL system $H = \langle \Sigma, V, \{t_1, \ldots, t_r\}, w\rangle$. For each $i = 1,\ldots,r$ we define a homomorphism $c_i : \Sigma^* \to (\Sigma \cup \{i\})^*$ by

$$c_i(a) = ai , \quad \text{for all } a \in \Sigma.$$

A s-G2L system G is now defined. Its alphabet is

$$\Sigma_1 = \Sigma \cup \{1,\ldots,k\} \cup \{e, S, \#\} ,$$

where S denotes the axiom. The productions of G are as follows

$$
\begin{array}{lll}
S \to c_i(w) & \text{for} & i = 1,\dots,r \\
(\$,a,i) \to c_k(\alpha) & \text{for} & i,k = 1,\dots,r,\ a \in \Sigma,\ \text{if}\ \ a \to \alpha \in t_k, \\
(i,a,i) \to c_k(\alpha) & \text{for} & i,k = 1,\dots,r,\ a \in \Sigma,\ \text{if}\ \ a \to \alpha \in t_k, \\
(j,a,k) \to \#\# & \text{for} & j,k = 1,\dots,r,\ j \neq k,\quad a \in \Sigma, \\
(j,e,k) \to \#\# & \text{for} & j,k = 1,\dots,r,\ j \neq k, \\
(i,e,i) \to ek & \text{for} & i,k = 1,\dots,r, \\
i \to ek & \text{for} & i,k = 1,\dots,r, \\
\# \to \#\# & &
\end{array}
$$

Let $h : \Sigma_1^* \to (\Sigma \cup \{S,\#\})^*$ be a homomorphism defined by

$$h(e) = h(i) = \lambda \qquad \text{for} \qquad i = 1,\dots,r$$

and

$$h(a) = a\ , \qquad\qquad \text{otherwise.}$$

Then it is easy to see, by induction on the length of the derivation, that for any derivation

$$w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_n$$

according to H there exists a derivation

$$S \Rightarrow x_0 \Rightarrow x_1 \Rightarrow \dots \Rightarrow x_n$$

according to G such that $h(x_i) = w_i$ for $i = 0,\dots,n$. Moreover, each of the words $x_0,\dots,x_{n-1}$ contains only one type of letters from $\{1,\dots,r\}$. This follows since if a word containing two different numbers, i.e. j and k with $j \neq k$, appears to any derivation according to G, then the derivation is immediately "blocked" by the fourth or fifth rule, i.e. no terminal word is obtained later. Hence we conclude

$$L = h(L(G)) \cap V^*$$

and the theorem is proved.

As a consequence we can demonstrate the imposibility of a simulation similar like in Theorem 1 for the nondeterministic e-G2L systems.

Theorem 7 There are s-G2L language, and therefore also e-G2L languages, which are not in $\mathcal{L}_{EOL}$ ($= \mathcal{L}_{COL}$).

Proof Consider any language L in $\mathcal{L}_{ETOL} - \mathcal{L}_{EOL}$. By Theorem 6 we can write $L = h(L_0) \cap R$ where h is a homomorphism, L_0 an e-G2L language and R a regular set. Assume that $L_0 \in \mathcal{L}_{EOL}$, since $\mathcal{L}_{EOL}$ is closed under homomorphisms and intersection with a regular set, also $L \in \mathcal{L}_{EOL}$, a contradiction.

R E F E R E N C E S

Culik II, K. and Fris, I. (1977), The Decidability of the Equivalence Problem for DOL-Systems, Inf. and Control 35, 20-39.

Harrison, M. (1978), Introduction to Formal Language Theory, Addison-Wesley.

Lindenmayer, A. (1977), Private communication.

Rozenberg, G. and Salomaa A. (1976), The mathematical theory of L systems, In : J. Tou (ed.), Advances Information Systems Science, Vol. 6, Plenum Press, New York, 161-206.

Salomaa, A. (1973), Formal Languages, Academic Press.

Salomaa, A. and Soittola, M. (1978), Automata-Theoretic Aspects of Formal Power Series, Springer-Verlag.

Vitanyi, P.M.B. (1974), Growth of strings in context dependent Lindenmayer systems, In: L systems, edited by G. Rozenberg and A. Salomaa, Lecture Notes in Computer Science Vol. 15, Springer-Verlag.