A Mathematical Investigation of Parallel Graph OL Systems*

by

Karel Culik II†
and
Derick Wood††

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Research Report CS-78-36
August 1978

* Work supported by the National Research Council of Canada Grant Nos. A-7403 and A-7700.

† Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

†† Department of Applied Mathematics, McMaster University, Hamilton, Ontario, Canada, L8S 4K1.

Abstract

Parallel graph OL systems are investigated mathematically. It is shown that bounded degreeness is decidable for a "doubly interactionless" subclass of these systems. Various subgraph occurrence problems are shown to be decidable as well as various notions of growth or size equivalence.

1. Introduction

The notion of sequential graph rewriting systems has been available for some time, for example, see Montanari (1970), Rosenfeld and Milgram (1972), Abe, Mizumoto, Toyoda and Tanaka (1973), and more recently Rosen (1975), Della Vigna and Ghezzi (1978). However, the advent of parallel graph rewriting systems is more recent. Some first approaches are to be found in Mayoh (1973 and 1974), while formal models have been introduced by Culik II and Lindenmayer (1974 and 1976), Ehrig and Tischer (1975), Ehrig and Kreowski (1976) and Nagl (1976).

It has appeared to us that the area of graph rewriting systems has been plagued with an over-abundance of definitional suggestions, while at the same time only limited investigations of the proposed models have been carried out. It is our thesis that it is necessary to investigate one particular model in some depth, rather than introducing new models willy-nilly. (Not surprisingly della Vigna and Ghezzi (1978) make a similar observation for sequential graph rewriting systems.) Such an investigation should result in a greater understanding of the chosen model and also give insight into graph rewriting systems per se. Clearly, the chosen model should be both reasonable for its intended area of application and natural mathematically.

Our choice is the PGOL system, that is, the propagating graph OL system. Biologically it is a well motivated model for multi-cellular development, see Lindenmayer and Culik II (1978), and mathematically it is a pleasing and natural generalization of string OL systems, see Ehrig and Rozenberg (1976) for a discussion of these points. Our aim has been to carry out an in-depth mathematical investigation of these systems.

Some results are already available in Culik II and Lindenmayer (1976), which are mentioned at appropriate points in the current paper. Many questions which we consider to be fundamental mathematically and interesting biologically are, unfortunately, intractable at this time. Functionality, connectivity and equivalence are examples of such questions.

The paper consists of a further six sections. Section 2 briefly surveys the various string OL system notions and terminology, while Section 3 is devoted to an extended introduction to PGOL systems. We feel this is necessary because parallel graph rewriting is much more complex than sequential graph rewriting. We also give some examples and discuss the role of "stencils". We informally demonstrate that stencils can be replaced by "full stencils" as is proved in Lindenmayer and Culik II (1978). This serves to place the general notion of a stencil in perspective, that is, they are simply an abbreviatory mechanism.

Section 4 is devoted to determinism, functionality, growth and size of PGOL systems. For example, deterministic PGOL systems are introduced, various notions corresponding to growth and Parikh functions of deterministic OL systems are investigated, and size and Parikh sets corresponding to length sets are discussed. This leads to the definition of a "doubly interactionless" PGOL system, known as a PGOOL system.

The generative capacity of DPGOL, PGOL and PGOOL systems is demonstrated in Section 5. In particular a "universal" self-reproducing system is exhibited.

Section 6 deals with decidability results. The most important result is that bounded degreeness of PGOOL systems is decidable. Whether it is decidable for arbitrary PGOL systems remains open. Finally, Section 7 presents a discussion of the "context-free-ness" of PGOL systems. Since every DPGOL system is a PGOOL system, we have the surprising situation that while DPGOL systems are, in our opinion, context-free, their non-deterministic counterparts (the PGOL systems) are not. We discuss the evidence for this conclusion and also show how easily slight perturbations in the definition of PGOL systems make them even more non-context-free.

2. Tabled String OL Systems

In this section we briefly review the various notions from tabled string OL systems that are necessary to our investigation.

An extended tabled OL system (ETOL system) is an n+3-tuple $G = (V, \Sigma, P_1, \ldots, P_n, S)$, $n > 0$ where V is an alphabet, $\Sigma \subseteq V$ is the terminal alphabet, $V-\Sigma$ the nonterminal alphabet, $P_i \subseteq V \times V^*$ are the tables of productions, $1 \leq i \leq n$, and S in $V-\Sigma$, is the start symbol. Each table P_i is finite and for each X in V , there is a production (X, α) in P_i , for some α in V^* . We usually write (X, α) as $X \to \alpha$.

For α , β in V^* , we write $\alpha \Rightarrow \beta$ in G if $\alpha = X_1 \ldots X_m$, $\beta = \beta_1 \ldots \beta_m$ and for some i , $1 \leq i \leq n$, $X_j \to \beta_j$ is in P_i , $1 \leq j \leq m$. We denote by $\Rightarrow^+$ and $\Rightarrow^*$ the transitive and the reflexive transitive closure of $\Rightarrow$. The language generated by G , denoted $L(G)$, is defined by:

$$L(G) = \{x : S \Rightarrow^* x \text{ in } G \text{ and } x \text{ in } \Sigma^*\} .$$

We say $L \subseteq \Sigma^*$ is an ETOL language if there is an ETOL system G such that $L = L(G)$.

We now consider various restrictions of ETOL systems. If $V = \Sigma$, then we replace S by a word in Σ^* to give a TOL system, usually denoted $G = (\Sigma, P_1, \ldots, P_n, \sigma)$. If $n = 1$ we obtain EOL and OL systems. We say G is propagating if $P_i \subseteq V \times V^+$, $1 \leq i \leq n$ and deterministic if each P_i is a map of V to V^* , $1 \leq i \leq n$. Hence we obtain POL, DOL and PDOL systems, for example.

A homomorphism $\theta : \Sigma \to \Delta$ is said to be a coding, and a homomorphism $\theta : \Sigma \to \Delta \cup \{\lambda\}$ is said to be a <u>weak coding</u>, where λ denotes the empty word.

We now obtain homomorphisms, codings and weak codings of the various L systems. For example, a WDOL system is a pair (G, θ) where G is a DOL system and θ is a weak coding. Similarly we obtain HTOL systems, where θ is a homomorphism. It is well known that every EOL (ETOL) language is a CPOL (CPTOL) language and vice versa, where C denotes coding. The language of an XDOL system (G, θ) is defined to be $L(G, \theta) = \{x : y$ is in $L(G)$ and $\theta(y) = x\}$, where $X = C$, H or W.

Finally, we need the notion of the <u>length set of an L</u> system. Let G be an ETOL system (or one of its restrictions). Then $LS(G) = \{|x| : x$ is in $L(G)\}$, where $|x|$ denotes the length of x, that is, the number of symbols in x. Clearly we can define $LS(G, \theta)$ similarly.

All other undefined notions will be found in Herman and Rozenberg (1975) or Rozenberg and Salomaa (1978).

3. Graph OL Systems

We now generalize the notion of string OL systems introduced in Section 2 to give graph OL systems. The definition of such systems is quite complex for a number of reasons, which we shall discuss as we go along. It is worthwhile reading Della Vigna and Ghezzi (1978) whose recent paper investigates context-free graph grammars, the analogous but simpler generalization of context-free string grammars.

Let ∞ denote the universal environment node which is, in the following always labelled with e, the universal environment label.

A node-labelled edge-labelled directed e-graph α over Σ (the node alphabet) and Δ (the edge alphabet) is a triple (V, φ, E) where V is a finite nonempty set of nodes, ∞ is not in V, $\varphi : V^{\infty} \to \Sigma^{e}$, $V^{\infty} = V \cup \{\infty\}$, $\Sigma^{e} = \Sigma \cup \{e\}$ is the node labelling function and $E \subseteq V^{\infty} \times \Delta \times V^{\infty}$ is a set of labelled directed edges, such that for each (a, h, b) in E $a \neq b$. φ also satisfies the condition that for all u in V^{∞}, $\varphi(u) = e$ implies $u = \infty$, that is the environment node is the only node labelled with the environment label. In the following when no confusion results we refer to α as a graph over Σ,Δ.

Let $(\Sigma, \Delta)_{*}$ denote the family of all graphs over Σ,Δ.

Since graphs are defined in terms of sets of nodes and edges it is important to grasp the notions of concrete and abstract graphs. Clearly, we wish to specify rewriting systems that replace "mother" nodes with "daughter" graphs just as in string systems we wish to replace symbols by words. However, if we specify a production simply by "labelled node" is replaced by "graph" then this production is applicable to at most one node in any given graph. This is because the node is designated in a unique manner as an element of a set and therefore any given graph cannot "use" the same designation more than once. However, our intention is clear, simply replace all nodes labelled in this particular way by the given daughter graph, taking care that unique isomorphic copies of the given daughter graph are used in the replacemnt (that is, disjoint unions are used).

Let $\alpha = (V_\alpha, \varphi_\alpha, E_\alpha)$, $\beta = (V_\beta, \varphi_\beta, E_\beta)$ be two graphs over Σ,Δ then α and β are <u>isomorphic</u> if there is an isomorphism $\varepsilon : V_\alpha \to V_\beta$ such that

(i) for all v in V_α , $\varphi_\alpha(v) = \varphi_\beta(\varepsilon(v))$, and

(ii) for all (u, a, v) in $V_\alpha \times \Delta \times V_\alpha$, (u, a, v) is in E_α iff $(\varepsilon(u), a, \varepsilon(v))$ is in E_β .

Hence under isomorphism $(\Sigma, \Delta)_*$ is partitioned into equivalence classes of isomorphic graphs. We say a graph α in $(\Sigma, \Delta)_*$ is <u>concrete</u> while $[\alpha]$, the equivalence class defined by α , denotes an <u>abstract</u> graph and α is a (<u>concrete</u>) <u>representation</u> of $[\alpha]$. The family of all <u>abstract</u> graphs over Σ,Δ is denoted by $[\Sigma, \Delta]_*$. The <u>empty graph</u> is the graph with no edges and only the environment node, denoted by λ . We denote $[\lambda]$ by λ . Then $(\Sigma, \Delta)_+ = (\Sigma, \Delta)_* - \{\lambda\}$ and $[\Sigma, \Delta]_+ = [\Sigma, \Delta]_* - \{\lambda\}$.

<u>Notation</u>

We use early lower case Greek letters to denote concrete graphs and upper case Roman letters to denote abstract graphs.

We say α is a (concrete) subgraph of β , $\alpha \leq \beta$ if $V_\alpha \subseteq V_\beta$, $E_\alpha \subseteq E_\beta$ and $\varphi_\alpha(u) = \varphi_\beta(u)$, for all u in V_α . Moreover, $\alpha = (V_\alpha, \varphi_\alpha, E_\alpha)$ is a full subgraph of β, written $\alpha \leq_f \beta$, if $\alpha \leq \beta$ and α is the subgraph of β induced by the nodes V_α . Similarly A is an (abstract) subgraph of B , A, B over Σ,Δ if there exist α , β such that $\alpha \leq \beta$, $[\alpha] = A$ and $[\beta] = B$.

Note that we are forced to deal with concrete representants, whenever we wish to specify an abstract graph.

We assume in the following that our model of graph OL systems will fulfill the following conditions:

(1) only nodes are to be rewritten,

(2) the rewriting of a node is independent of its context, and

(3) all nodes are rewritten in parallel.

Clearly (1)-(3) are the graph analogues of the corresponding conditions for OL systems. In fact, since nodes represent cells in the biological context and edges communication and/or contact, condition (1) reflects the assumption that only cells develop.

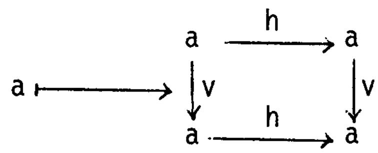
A graph OL production over Σ,Δ will be specified by a pair (a, A) , usually written $a \mapsto A$, where a is in Σ and A is in $[\Sigma, \Delta]_+$. We say a is the mother "node" and A is the daughter graph. Given a set of such productions P over Σ,Δ and an abstract graph B over Σ,Δ then $a \mapsto A$ can be applied to B if it has nodes labelled a . Note that $a \mapsto \lambda$ is not allowed, node erasure adds much complication and hence we only deal with propagating productions. We also ensure that P is complete, that is, for all a in Σ , there is a production $a \mapsto A$ in P for some A in $[\Sigma, \Delta]_+$.

Given a graph B and a complete set of productions P , both over Σ,Δ , it is straightforward to apply the productions to the nodes of B . This results, however, in a derived abstract graph which consists of disconnected daughter graphs. The major question is: How are they connected together?

Returning for a moment to the context-free graph grammars of Della Vigna and Ghezzi (1978) we find a much simpler situation. Only one node at a time is replaced. Hence the connecting rules are quite simple. Each daughter graph has a specified source and target node (possibly the same node). After replacement of a given mother node, the incoming edges are connected to the target node of the daughter graph and the outgoing edges to the source node of the daughter graph. Clearly,this can be carried across to graph OL systems. However,we follow the approach of Culik II and Lindenmayor (1976) where a much more general technique is used. Remark that the connecting rules of Della Vigna and Ghezzi (1978) have many drawbacks. For example: (i) Edges are always preserved, when there are situations for which edges should be removed or extra edges added , (ii) More than one source and one target node are often necessary.

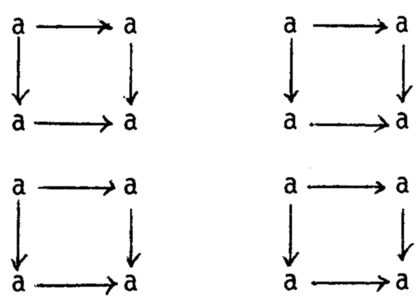
Consider the problem of generating the set of $2^n \times 2^n$-arrays, $n \geq 0$ over $\{a\}$, $\{h, v\}$. Edges labelled h are interpreted as horizontal, those labelled v as vertical.

We have one production p ,



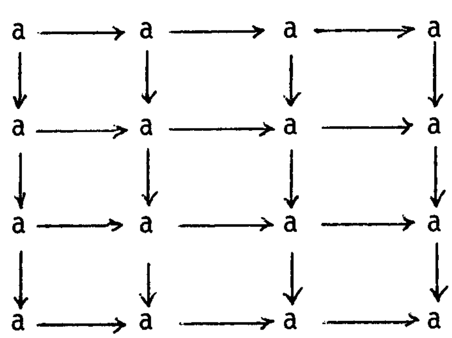
(1)

Beginning with a , the 1×1-array, we easily obtain the derived 2×2-array using p . Consider generating a 4×4-array from the 2×2-array. We obtain



(2)

where we have omitted edge labels for clarity. How are these 4 daughter graphs connected? Our intention is to obtain.



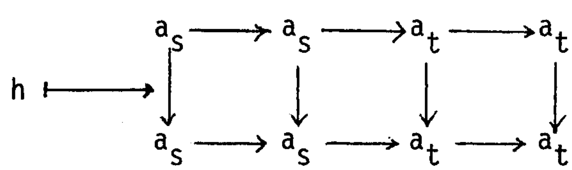
(3)

Observe that the connecting rules of Della Vigna and Ghezzi (1978) do not allow this kind of connection, since the top left daughter graph has 3 source nodes giving rise to 4 edges, while the mother node has only two edges.

We choose to specify connecting rules for each edge which depend upon the daughter graphs. For Σ an alphabet, letting s, t denote source and target, respectively, define $s(\Sigma) = \{a_s : a \text{ in } \Sigma\}$, $t(\Sigma) = \{a_t : a \text{ in } \Sigma\}$, which are denoted by Σ_s and Σ_t respectively. We also define $s^{-1}(a_s) = a$ and $t^{-1}(a_t) = a$, for all a in Σ . Let $\Sigma_{st} = \Sigma_s \cup \Sigma_t$ and $st^{-1}(a_p) = a$, for all a_p in Σ_{st} . A <u>stencil</u> (an <u>abstract stencil</u>) over Σ,Δ is a graph (an abstract graph) over Σ,Δ .

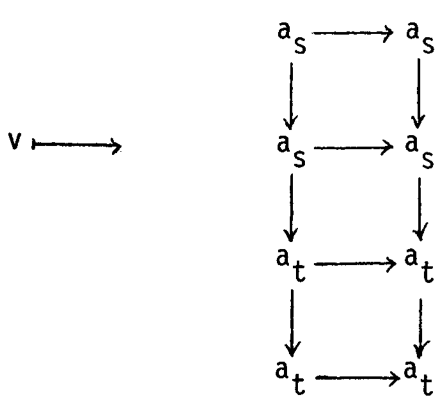
A <u>connection rule</u> over Σ,Δ is a pair (h, H) , usually written $h \to H$, where h is in Δ and H is an abstract stencil over Σ,Δ .

Continuing our example, we specify two connection rules, one for h and one for v .



(4)

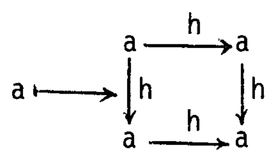
and



(5)

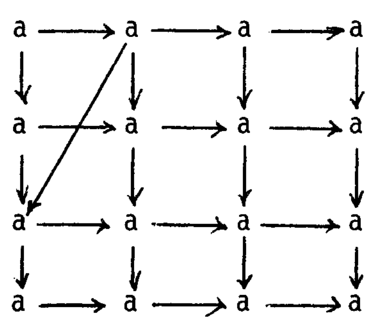
Informally, we apply the appropriate connection rule to each pair of daughter graphs in turn, matching the one daughter graph to the source part of the stencil and the other to the target, adding the specified edges between source and target nodes. For example, in (2) the top left hand daughter graph is the source for the h-connection rule, since its mother node has an outgoing h-edge, while the top right daughter graph is the target for the same connection rule. Hence we do in fact obtain (3) from a in two derivation steps using production (1) and connection rules (4) and (5).

Observe that the daughter graph in (1) has no nontrivial automorphisms, therefore the only way to connect is given by (3). However, if we had only one edge label h and



(6)

then the daughter graph in (6) does have nontrivial automorphisms, in which case we obtain:



(7)

and other variants.

Note that we have added three more conditions to be fulfilled by our model for graph OL systems in the above discussion, namely:

(4) graph OL systems are node propagating,

(5) the production set in a graph OL system must be complete,

(6) daughter graphs may only be connected if their mothers are connected.

We are now in a position to define a node propagating graph OL system, however we delay the definition in order to consider a more general notion of stencil.

Let $\alpha = (V_\alpha, \varphi_\alpha, E_\alpha)$ be a graph over Σ, Δ and $X \subseteq V_\alpha$ a subset of its nodes. Define merge(X, α) as the graph $(V_\beta, \varphi_\beta, E_\beta)$, where $V_\beta = V_\alpha - X$, φ_β is the restriction of φ_α to V_β, and

$$\begin{aligned} E_\beta = {} & (E_\alpha \cap (V_\beta \times \Delta \times V_\beta)) \\ & \cup \left\{(u, h, \infty) : (u, h, v) \text{ is in } E_\alpha,\ u \text{ in } V_\beta,\ v \text{ in } X\right\} \\ & \cup \left\{(\infty, h, v) : (u, h, v) \text{ is in } E_\alpha,\ u \text{ in } X,\ v \text{ in } V_\beta\right\} . \end{aligned}$$

Essentially, the nodes in X are merged into ∞, the environment node.

Using full daughter graphs in stencils as we did above, then such a full stencil γ fulfills $\text{merge}(\varphi_\gamma^{-1}(\Sigma_t), \gamma)$ is the source daughter graph and $\text{merge}(\varphi_\gamma^{-1}(\Sigma_s), \gamma)$ is the target daughter graph. To strip off the appropriate subscripts we need to apply st^{-1} to both these graphs. Our notion of applicability of a full stencil is simply that the source merged graph is isomorphic to the source daughter graph and similarly for the target graph.

We now relax this condition to "subgraph of". Let $\gamma_S = st^{-1}(\text{merge}(\varphi_\gamma^{-1}(\Sigma_t), \gamma))$, $\gamma_T = st^{-1}(\text{merge}(\varphi_\gamma^{-1}(\Sigma_s), \gamma))$ and

$$\gamma_C = st^{-1}((V_\gamma, \varphi_\gamma, \{(u, h, v) : (u, h, v) \text{ in } E_\gamma \text{ and either } \varphi_\gamma(u) \text{ in } \Sigma_s \text{ and } \varphi_\gamma(v) \text{ in } \Sigma_t \text{ or } \varphi_\gamma(u) \text{ in } \Sigma_t \text{ and } \varphi_\gamma(v) \text{ in } \Sigma_s\}) .$$

We say that a stencil γ over Σ, Δ is applicable to an ordered pair of graphs (α, β) over Σ, Δ if:

(i) $V_\alpha \cap V_\beta = \phi$, that is, the only common node is ∞ .

(ii) $\gamma_S \leq \alpha$ and $\gamma_T \leq \beta$.

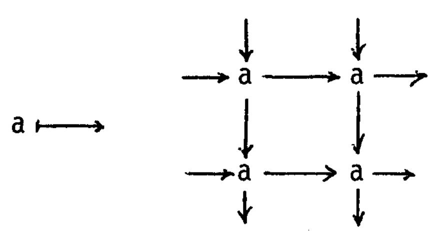
Let γ be applicable to (α, β) then *the joining of (α, β) by γ* is defined as the graph

$$(V_\alpha \cup V_\beta, \varphi_\alpha \cup \varphi_\beta , E_\alpha \cup E_\beta \cup E_{\gamma_C}) .$$

Let Q be a set of stencils. Then γ in Q is said to be *Q-maximal* with respect to (α, β) if γ is applicable to α,β and there is no δ in Q such that δ is applicable to α,β and $\gamma_S \cup \gamma_T \leq \delta_S \cup \delta_T$.

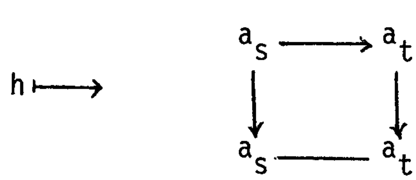
To illustrate these notions let's return to our running example.

An edge from any node to the environment node is called a *hand*. As we never represent the envirnoment node in our diagrams, hands are represented as broken directed edges. We first modify the production for a to:



(8)

where hands have been added. It is now sufficient to consider the following simpler stencils



(9)

and

$$v \longmapsto \begin{array}{ccc} a_s & \longrightarrow & a_s \\ \downarrow & & \downarrow \\ a_t & \longrightarrow & a_t \end{array} \qquad (10)$$

and the initial graph is

$$a \qquad (11)$$

Letting γ be the stencil of (9) then

$$\gamma_S = \begin{array}{cc} a & \longrightarrow \\ \downarrow & \\ a & \longrightarrow \end{array} \qquad \text{and} \qquad \gamma_T = \begin{array}{cc} \longrightarrow & a \\ & \downarrow \\ \longrightarrow & a \end{array}$$

while

$$\gamma_C = \begin{array}{ccc} a & \longrightarrow & a \\ & & \\ a & \longrightarrow & a \end{array} \quad .$$

Observe that γ is only applicable to a source and target daughter from (8) if they are in the appropriate orientation. The hands in γ_S and γ_T must match those in the source and target daughters, respectively.

We are now ready for the central notion of this paper.

Definition

A propagating graph OL system (PGOL system) is a quintuple $G = (\Sigma, \Delta, P, C, S)$ where

Σ is an alphabet of node labels,

Δ is an alphabet of edge labels,

P is a finite subset of $\Sigma \times [\Sigma, \Delta]_+$ of productions

C is a finite subset of $\Delta \times [\Sigma_{st}, \Delta]_+$ of connection rules, and

S in $[\Sigma, \Delta]_+$ is the start graph.

P must be complete, i.e. for each $a \in \Sigma$ there is at least one production $a \mapsto \alpha$ in P . For each h in Δ , $h \rightarrow \lambda$ is implicitly in C , therefore C is complete. Note that G is specified by abstract rather than concrete graphs. It is assumed that $e \mapsto e$ is implicitly available as a production for the environment node and that $h \mapsto a_s \overset{h}{\rightarrow} e_t$ and $h \mapsto e_s \overset{h}{\rightarrow} a_t$ are implicitly available for all a in Σ and h in Δ .

We define the yield relation over $[\Sigma, \Delta]_+$ informally, a rigorous definition can be found in Culik II and Lindenmayer (1976).

For two abstract graphs U , V in $[\Sigma, \Delta]_+$ we write $U \Rightarrow_q V$ (or simply $U \Rightarrow V$ if G is understood) if:

(i) there exist daughter graphs of each node of U , given by P and

(ii) there exist maximal stencils for each pair of daughter graphs whose mothers were connected, given by C ,

such that the simultaneous joining of all daughter graphs results in V . Note that in particular a hand of a certain kind which appears in a daughter graph will only appear in the derived graph if its mother has a hand of the same kind.

We obtain $\overset{+}{\Rightarrow}$ and $\Rightarrow^*$, the transitive and reflexive transitive closure of $\Rightarrow$ in the usual way. The graph language generated by G , denoted L(G) , is defined as:

$$L(G) = \{U : S \Rightarrow^* U\} .$$

$L \subseteq [\Sigma, \Delta]_+$ is a PGOL language if there exists a PGOL system G such that $L(G) = L$.

Let us formalize our earlier full stencil system. A PGOL system $G = (\Sigma, \Delta, P, C, S)$ is a full stencil PGOL system (fsPGOL system) if C fulfills the following conditions:

for each ordered pair $(a \mapsto A, b \mapsto B)$ of productions in P and h in Δ , there is a connection rule $h \mapsto D$ in C such that $D_S = A$ and $D_T = B$. D is a full stencil for A, B.

Clearly, if G is a fsPGOL system then for each triple (A, h, B) there is a full stencil D and D is maximal for (A, h, B) , where $a \mapsto A$, $b \mapsto B$ are in P and h is in Δ .

Let $\mathcal{L}(\text{PGOL})$ and $\mathcal{L}(\text{fsPGOL})$ denote the families of PGOL and fsPGOL languages, respectively. Clearly, every fsPGOL language is a PGOL language. Conversely, given a PGOL system G for each edge and each pair of daughter graphs in G extend their maximal applicable stencils to full stencils. In this way we obtain a fsPGOL system which simulates G step by step. The maximality condition is critical for this simulation to hold. Hence we have demonstrated:

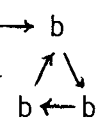
Lemma 3.1

$$\mathcal{L}(\text{PGOL}) = \mathcal{L}(\text{fsPGOL}) .$$

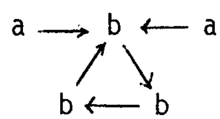
This serves to demonstrate that the nonfull stencil mechanism of Culik II and Lindenmayer (1976) adds no generative power but is rather a powerful abbreviatory tool. We can always replace a non-fsPGOL system by a fsPGOL system which generates the same graph language.

Let us consider the hands. Are they necessary or equivalently can the environment node be removed with restricting the generative power of PGOL systems? Consider the following example.

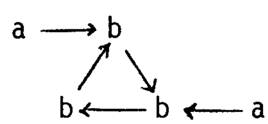
$$S = a \rightarrow b \leftarrow a \ ;$$

$a \mapsto a\rightarrow$, $b \mapsto$  and $h \mapsto a_s \rightarrow b_t$

then we obtain in one derivation step from S the graph



If we do not allow hands or equivalently assume all possible hands occur everywhere then we also obtain



which is not obtained with the above system. This demonstrates that "no hands" systems have weaker generative power than PGOL systems.

Let us denote by <u>allPGOL</u> the PGOL systems in which all possible hands occur at all possible nodes. Then we have shown:

<u>Lemma 3.2</u>

$$\mathcal{L}(\text{allPGOL}) \subsetneq \mathcal{L}(\text{PGOL}) \ .$$

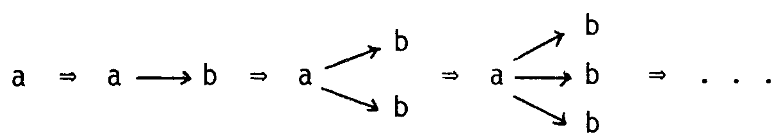
Let us close this section by giving a final example, the family of star graphs.

$$\text{Let} \quad S = a \ ;$$

$$a \mapsto \ \leftarrow a \rightarrow b \quad ; \qquad b \mapsto \ \rightarrow b \ ;$$

$$\text{and} \quad h \mapsto a_s \rightarrow b_t \quad ,$$

then we obtain



4. Determinism, Growth and Size

In Culik II and Lindenmayer (1976) the notions of deterministic and functional PG0L systems are introduced. They claim, wrongly, that these two notions are identical. We first introduce these notions and then compare them. Secondly, we show that various notions of growth functions, for DPG0L systems, and size sets, for PG0L systems are reducible to growth functions or length sets of various string 0L systems. One exception is the edge size sets of PG0L systems.

These reduction results depend heavily upon two constructions, which given a PG0L system derive an associated 0L or T0L system that is equivalent in a certain sense. The non-applicability of the second construction to PG0L systems leads to the notion of "doubly interactionless" PG0L systems, which are called PG00L systems. These reduction results are then used in Section 6 to derive numerous decidability results. We say a PG0L system $G = (\Sigma, \Delta, P, C, S)$ is reduced if it has no useless symbols, productions or connection rules. A symbol in $\Sigma \cup \Delta$ is said to be useless if it does not occur in any graph of $L(G)$. Similarly, a production in P or a connection rule in C is useless if it cannot be "used" in any derivation of G from S.

Recall from Culik II and Lindenmayer (1976) that a reduced PG0L system G is deterministic if:

(i) for each a in Σ there is only one production $a \mapsto A$ in P,

(ii) for each h in Δ and each edge (a_1, h, a_2) occurring in S, P or C, let $a_i \mapsto A_i$ be in P, $A_i = [\alpha_i]$, $i = 1,2$ and let $Q = \{\gamma : h \mapsto [\gamma]$ in $C\}$, then there is at most one γ in Q such that γ is Q-maximal for (α_1, α_2) .

We say that a reduced PGOL system G is functional if for all U for which $S \Rightarrow^* U$ there is exactly one V such that $U \Rightarrow V$.

In Culik II and Lindenmayer (1976) it is claimed that functionality and determinism are equivalent conditions. However this is not the case. Consider the following examples.

Example 4.1 G_1 defined by:

	the start graph	$a \rightarrow a$
	the production	$a \mapsto \rightarrow a \rightarrow$
and	the connection rules	$h \mapsto a_s \rightarrow a_t$
		$h \mapsto a_s \leftarrow a_t$.

Clearly, G_1 is functional and non-deterministic.

In a second example we show that a system can be functional even when the productions for nodes are non-deterministic.

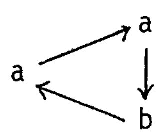
Example 4.2 G_2 defined by:

	the start graph	$c \rightleftarrows d = S$
	the productions	$c \mapsto \rightarrow a \rightarrow$
		$d \mapsto \rightarrow a \rightarrow b \rightarrow$
		$d \mapsto \rightarrow b \rightarrow a \rightarrow$
		$a \mapsto \rightarrow a \rightarrow$
		$b \mapsto \rightarrow b \rightarrow$
and	the connection rules	$h \mapsto a_s \rightarrow a_t$

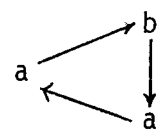
$$h \mapsto a_s \rightarrow b_t$$

$$h \mapsto b_s \rightarrow a_t$$

where h is the only edge label, which we are not showing explicitly in the diagrams. This convention will be used throughout the paper. From S one new graph is obtained, which then stays the same, namely



or



which are representants of the same abstract graph.

Although functionality does not imply determinism we do have the weaker result which we state without proof.

<u>Lemma 4.1</u>

Every DPGOL system is functional.

<u>Remarks</u>

(1) By definition each daughter graph in a DPGOL system has <u>no</u> non-trivial automorphisms.

(2) Example 4.2 demonstrates that there are functional PGOL systems which are <u>not</u> node deterministic, that is, a mother having only one daughter graph.

(3) It is an open problem whether functionality is decidable.

Lemma 4.1 implies that for each DPGOL system G there is an associated unique sequence of abstract graphs:

$$S = S_0,\ S_1,\ S_2,\ \ldots$$

Hence the notion of growth and Parikh functions can be introduced analogous to those for DOL systems. We can consider the node and edges separately or together. In all cases we ignore the environmental node and its connections.

Let A be an abstract e-graph. By $\#_V(A)$ we denote the number of nodes of A, by $\#_E(A)$ the number of edges of A and by $\#(A)$ the number of nodes and edges of A. Hence $\#(A) = \#_V(A) + \#_E(A)$. Let $G = (\Sigma, \Delta, P, C, S)$ be a DPGOL system. Then $f_{G,V}$, the node growth function of G, is defined by

$$f_{G,V}(i) = \#_V(S_i) \text{ , for all } i \geq 0 \text{ .}$$

Similarly we define $f_{G,E}$ and f_G, the edge growth function of G and the growth function of G.

We need the following construction:

Construction 4.1

Let $G = (\Sigma, \Delta, P, C, S)$ be a DPGOL system and Σ have some fixed arbitrary ordering. Let μ be the mapping from $[\Sigma, \Delta]_*$ to Σ^* which maps each abstract graph A over Σ, Δ to the alphabetically ordered string of all occurrences of Σ-symbols as node labels in A. Construct the associated PDOL system $F_G = (\Sigma, P_F, \mu(S))$ where $P_F = \{a \to \mu(A) : a \mapsto A \text{ is in } P\}$.

We now have our first theorem.

Theorem 4.2

Let $f : \mathbb{N} \to \mathbb{N}$ then

(i) f is a DPG0L node growth function

iff

(ii) f is a PD0L growth function.

Proof: (ii) clearly implies (i) since every string 0L system can be "simulated" by some PG0L system. Therefore consider the reverse implication. Let $G = (\Sigma, \Delta, P, C, S)$ be a DPG0L system with $f_{G,V} = f$. Using the above construction, obtain F_G a PD0L system, clearly $f_{F_G} = f_{G,V}$, hence the result.

We now have our second construction.

Construction 4.2

Let $G = (\Sigma, \Delta, P, C, S)$ be a DPG0L system. Let $\Omega = \Sigma \times \Delta \times \Sigma$ and let there be some fixed arbitrary ordering of Σ and Ω . Let μ be the mapping of Construction 4.1 and define a mapping η from $[\Sigma, \Delta]_*$ to Ω^* , which maps an abstract graph A over Σ,Δ to the alphabetically ordered string w of all occurrences of Ω-symbols as edges in A . That is, for each edge with source label a , target label b and edge label h in A , there is one occurrence of (a, h, b) in w . Let ν be the mapping from $[\Sigma, \Delta]_*$ to $\Sigma^*\Omega^*$ defined by $\nu(A) = \mu(A)\eta(A)$, for each A in $[\Sigma, \Delta]_+$. Now construct a PD0L system $H_G = (\Sigma\cup\Omega\cup\{d\}, P_H, \nu(S))$ where d is a new symbol and

$P_H = \{a \to \nu(A) : a \mapsto A$ is in $P\} \cup \{d \to d\} \cup \{(a, h, b) \to w :$ where $w = \eta([st^{-1}(\gamma_C)])$ if $\eta([st^{-1}(\gamma_C)]) \neq \lambda$ and $w = d$ otherwise, where (a, h, b) is in Ω and γ is the only stencil applicable to α , β , concrete daughter graphs of a, b respectively$\}$.

In the above construction the "edge" production in P_H replaces an "edge" by the "edges" it would be replaced by in the given DPGOL system. Since an edge may be replaced by no edges we take care of this possibility by using the dummy letter d to represent this situation.

We now obtain:

<u>Theorem 4.3</u>

Let $f : \mathbb{N} \to \mathbb{N}$ then:

(i) f is a DPGOL edge growth function

iff

(ii) f is a WPDOL growth function.

<u>Proof</u>: (i) $\Rightarrow$ (ii)

Let $G = (\Sigma, \Delta, P, C, S)$ be a DPGOL system with $f_{G,E} = f$. Using Construction 4.2 construct H_G . Now define a weak coding $\theta : \Sigma \cup \Omega \cup \{d\} \to \Delta \cup \{\lambda\}$ by:

$$\theta(a) = \lambda \text{ , for all } a \text{ in } \Sigma \cup \{d\} \text{ ,}$$

$$\theta((a, h, b)) = h \text{ , for all } (a, h, b) \text{ in } \Omega \text{ .}$$

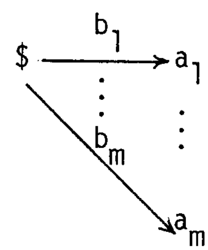
Clearly the growth function of (H_G, θ) is identical to $f_{G,E}$.

(ii) $\Rightarrow$ (i)

Let $G = (\Sigma, P, \sigma)$ be a PDOL system and $\theta : \Sigma \to \Delta \cup \{\lambda\}$ be a weak coding. Construct a DPGOL system $G' = (\Sigma \cup \{\$\}, \Delta, P', \{h \mapsto \lambda : h \text{ in } \Delta\}, S)$ where \$ is a new symbol not in $\Sigma \cup \Delta$. Without loss of generality we assume $\Sigma \cap \Delta = \phi$. Define a mapping ν from Σ^* to $[\Sigma \cup \{\$\}, \Delta]_*$ by:

$$\nu(\lambda) = \lambda$$

and for all $a_1 \ldots a_m$ in Σ^+ , $\nu(a_1 \ldots a_m)$ is the abstract graph



where $\theta(a_i) = b_i$ in $\Delta \cup \{\lambda\}$ and if $\theta(a_i) = \lambda$ then there is no edge incident to a_i . Clearly $\#_E(\nu(a_1 \ldots a_m)) = |\theta(a_1 \ldots a_m)|$.

Let $S = \nu(\sigma)$ and

$$P' = \{a \mapsto \nu(x) : a \to x \text{ is in } P\} \cup \{\$ \mapsto \$\} .$$

Now P' has only one production for each a in $\Sigma \cup \{\$\}$ since G does. Further since no edges are preserved at each derivation step, G' is a DPGOL system. By the observation above we therefore have $f_{G',E} = f_G$. This complete the theorem.

Finally we consider the total growth function (both nodes and edges).

<u>Theorem 4.4</u>

(i) Let $f : \mathbb{N} \to \mathbb{N}$ then f is a DPGOL growth function implies f is a WPDOL growth function, and

(ii) There exist WPDOL growth functions which are not DPGOL growth functions.

<u>Proof</u>: (i) This follows from the proof of (i) $\Rightarrow$ (ii) in Theorem 4.3, observing that deleting an "edge" (a, h, b) is carried

out by introducing a dummy symbol d , that is,

(a, h, b) $\to$ d ; d $\to$ d and removing d by a weak coding.

(ii) Consider the PDOL system defined by:

initial word a

productions $a \to b$; $b \to b$

and weak coding θ , $a \overset{\theta}{\to} a$; $b \overset{\theta}{\to} \lambda$.

Then the WPDOL growth sequence is 1,0,0,... . Clearly no DPGOL system can generate such a sequence.

Remark

It should be noted that we can replace WPDOL growth function by HPDOL or HDOL growth function in Theorems 4.3 and 4.4. This also holds true in the following corresponding theorems for Parikh functions.

We now turn to the notion of Parikh function for DPGOL systems.

Let Σ and Δ be given alphabets and A an abstract graph over Σ,Δ . Define some arbitrary but fixed ordering of the symbols of Σ,Δ and $\Sigma \cup \Delta$. Then we denote by $\pi_V(A)$ an element of $\mathbb{N}^n$, where $\#\Sigma = n$, such that the value in the i-th position denotes the number of occurrences of the i-th node label in A . Similarly we define $\pi_E(A)$ and $\pi(A)$. Clearly $\#_V(A) = \pi_V(A)(1,\ldots,1)^T$ and similarly for $\#_E(A)$ and $\#(A)$. Let $G = (\Sigma, \Delta, P, C, S)$ be a DPGOL system. Then $\pi_{G,V}$, the node Parikh function of G , is defined by:

$$\pi_{G,V}(i) = \pi_V(S_i) \text{ , for all } i \geq 0 .$$

We can define $\pi_{G,E}$ and π_G , the edge Parikh function of G and the Parikh function of G , in a similar manner.

Immediately by the proof techniques of Theorems 4.2, 4.3 and 4.4 we have the following results.

<u>Theorem 4.5</u>

Let $f : \mathbb{N} \to \mathbb{N}^n$, $n > 0$, then

(i) f is a DPGOL node Parikh function

iff

(ii) f is a PDOL Parikh function.

<u>Theorem 4.6</u>

Let $f : \mathbb{N} \to \mathbb{N}^n$, $n > 0$, then:

(i) f is a DPGOL edge Parikh function

iff

(ii) f is a WPDOL Parikh function.

<u>Theorem 4.7</u>

(i) Let $f : \mathbb{N} \to \mathbb{N}^n$, $n > 0$, then f is a DPGOL Parikh function implies f is a WPDOL Parikh function, and

(ii) There exist WPDOL Parikh functions which are not DPGOL Parikh functions.

We now investigate which of the above reducibility results hold in the non-deterministic case. First note that in this setting growth and Parikh functions are replaced by size and Parikh sets. For example, letting $G = (\Sigma, \Delta, P, C, S)$ be a PGOL system we define $ES(G)$, the <u>edge size set of G</u> , as:

$$ES(G) = \{\#_E(U) : U \text{ in } L(G)\} .$$

We define VS(G), S(G), $E\pi(G)$, $V\pi(G)$ and $\pi(G)$ similarly.

For VS and $V\pi$ we have:

<u>Theorem 4.8</u>

For $N \subseteq \mathbb{N}$ $(N_n \subseteq \mathbb{N}^n, \ n \geq 0)$

(i) N (N_n) is a POL length set (Parikh set)

iff

(ii) N (N_n) is a PGOL node size set (node Parikh set) .

<u>Proof</u>: Straightforward since we consider nodal growth independently of edges.

However when considering edge size sets Construction 4.2 just does not work.

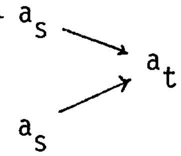
<u>Example 4.3</u>

Consider the PGOL system G defined by:

the start graph $a \rightarrow a$

the productions $a \mapsto\rightarrow a\rightarrow \; ; \; a \mapsto\rightarrow a \longrightarrow a\rightarrow$

and the connection rules $h \mapsto$



Consider the string representation of Construction 4.2. Initially we have aa(a, h, a) , but we can now obtain aa(a, h, a)(a, h, a) which does not correspond to any valid replacement in G since the connection rule can only be applied when a source node a is replaced by two a-labelled nodes. In fact we obtain all ETOL length sets when considering edge size sets of PGOL systems.

<u>Theorem 4.9</u>

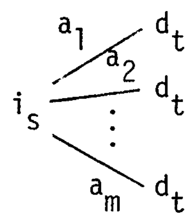
Let $G = (\Sigma, P_1, \ldots, P_n, \sigma)$, $n > 0$, be a TOL system. There exists a PGOL system $G' = (\Sigma', \Sigma, P^*, C, S)$ such that $LS(G) = ES(G')$ and $\Pi S(G) = E\Pi(G')$.

<u>Proof</u>: Let $\Sigma' = \{d\} \cup \{1,\ldots,n\}$. Define a mapping ν from $\{1,\ldots,n\} \times \Sigma^*$ to $[\Sigma', \Sigma]_*$ by:

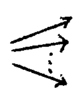
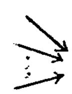
for all i , $1 \le i \le n$:

$$\nu(i, \lambda) = \lambda$$

and for all $a_1 \ldots a_m$ in Σ^+

$\nu(i, a_1 \ldots a_m) =$ 

Let $S = st^{-1}(\nu(1, \sigma))$,

$P = \{i \mapsto j$  $: 1 \le i,j \le n\} \cup \{d \to$  $d^m\}$, where

$m = \max(\{|x| : a \to x$ in P_i for a in Σ , for i , $1 \le i \le n\})$,

and j and d have all possible labelled hands, and

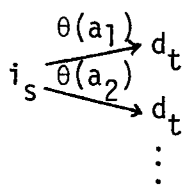
$C = \{a \mapsto \nu(i, x) : a \to x$ in P_i for some i , $1 \le i \le n\}$.

It should be clear that at each rewriting step the appropriate connections are made since the central node labelled i selects the appropriate connection rules corresponding to the table P_i .

Corollary 4.10

Let G be an ETOL system then there is a PGOL system G' such that $LS(G) = ES(G')$ and $\pi S(G) = E\pi(G')$.

Proof: Each ETOL language can be represented as a coding of a TOL language (Ehrenfeucht and Rozenberg, 1975). Therefore let G'' be a TOL system and θ a coding such that $\theta(L(G'')) = L(G)$. Proceed to construct G' , the required PGOL system by means of the proof technique of Theorem 4.9. However define $\nu(i, a_1 \ldots a_m)$ to be



Clearly $ES(G') = LS(G'') = LS(G')$ and $E\pi(G') = \pi S(G)$.

The situation for size and Parikh sets of PGOL systems is not so clear. We therefore close this section by considering a sufficient condition on PGOL systems which ensures Construction 4.2 (suitably modified for the non-deterministic situation) works.

Theorem 4.9 and the remarks preceeding it lead to the notion of an edge-context-free PGOL system. Let $G = (\Sigma, \Delta, P, C, S)$ be a PGOL system such that for all edges $a \overset{h}{\to} b$ occurring in S, P, or $st^{-1}(C)$ and for all concrete daughter graphs α_i , β_i , $i = 1,2$ of a and b , respectively, the set of maximal stencils applicable to (α_1, β_1) is isomorphic to the set of maximal stencils applicable to (α_2, β_2) . We say that G is a doubly interactionless PGOL system, denoted PGOOL system, that is, zero node-interactions and zero edge-interactions.

Immediately for PGOOL systems we have a result corresponding to Theorem 4.3 for edge size sets.

Theorem 4.11

Let $f : \mathbb{N} \to \mathbb{N}$ then:

(i) f is a PGOOL edge size or Parikh set

iff

(ii) f is a WPOL length or Parikh set, respectively.

Proof: (i) $\Rightarrow$ (ii)

Extend the proof of Theorem 4.3 to the PGOOL case. The problem discussed in Example 4.3 is avoided since for each triple (a, h, b) in Ω the possible new edges that can be produced by a derivation step are independent of the choice of productions for a and b.

(ii) $\Rightarrow$ (i)

The proof technique of Theorem 4.3 can be immediately extended to the non-deterministic case, giving a PGOL system with no connection rules, which is then, trivially, a PGOOL system.

Theorems 4.9 and 4.11 are of interest since they contribute to the discussion on the context-freeness of PGOL systems. Theorem 4.9 makes us aware that as far as edge size sets are concerned non-EOL length sets can be obtained. On the other hand the PGOOL systems under this measure can be said to be truly interactionless.

5. Generative Capacity of PGOL Systems

In Culik II and Lindenmayer (1976) various restricted classes of PGOL systems are investigated analogous to those for string OL systems, namely D = deterministic, T = tabled, F = finite start set of graphs, C = coding . It is shown that for any two combinations X, Y of "operators" D, T, F or C, L(XPGOL) $\subsetneq$ or is incomparable with L(YPGOL) if L(XPOL) $\subsetneq$ or is incomparable with L(XPOL) , respectively. Hence, for example, L(DPGOL) $\subsetneq$ L(PGOL) . This is proved by considering just those languages of line graphs with a single edge label which correspond to strings in a natural way. We immediately obtain L(DPGOL) $\subsetneq$ L(PGOOL) by the same technique. We also have L(PGOOL) $\subseteq$ L(PGOL) by definition, while proper containment is a result of the following non-PGOOL graph language.

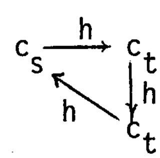
Consider G over {a, b, c, d} , {g, h} where

the initial graph is $a \xrightarrow{g} b$

the productions are $a \mapsto \xrightarrow{h} c \xrightarrow{h}$, $a \mapsto d$

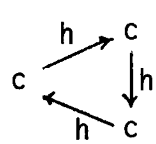
$b \mapsto \xrightarrow{h} c \xrightarrow{h} c \xrightarrow{h}$

$c \mapsto \xrightarrow{h} c \xrightarrow{h}$, $d \mapsto d$

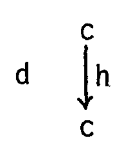
the connection rules are $g \mapsto$ 

$h \mapsto c_s \xrightarrow{h} c_t$

giving the initial graph,



and



Clearly G is not a PGOOL system. Consider any PGOL system G' such that L(G') = L(G) . Now the initial graph of G' must be $a \xrightarrow{g} b$, since this graph has fewest nodes. Since we are dealing with propagating systems each of a and b must give rise to at least one node, and in fact, one of them must give rise to two nodes only while the other only one. Clearly, the productions for a and b in G can be interchanged and the connection rules modified appropriately to give L(G) once more in a non-PGOOL fashion. Hence the only other possibility is that a (or b) gives $c \rightarrow c$ or c d and b (or a) gives $\rightarrow c \rightarrow$. However the connection rules must differentiate between the two possible daughter graphs of a to ensure that the appropriate connections are made. Hence this system is also non-PGOOL. Therefore L(G) is a non-PGOOL language.

Let us now restrict our attention to graphs with a single-edge-label, in other words to the webs of Rosenfeld and Milgram (1972). Cheung (1978) has proved that any bounded degree PGOL system can be "simulated" by a single-edge-labelled PGOL system, in the sense that the underlying graph structures obtained are the same. By underlying graph structure we mean the coding of the graph language which identifies all edge labels and all node labels, that is an unlabelled graph. However for arbitrary PGOL systems this "simulation" result does not hold.

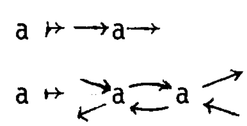
On the other hand if we restrict attention to single-node-labelled graphs and PGOL systems the generative power of such systems is drastically reduced, hence the star graph language of Section 3 cannot be generated when only a single node label is allowed. Therefore the trade-off of edge-labels and node-labels is in one direction only. For bounded degree systems the edge label set can be reduced at the expense of the node label set.

We now give two examples which generate all complete graphs over Σ , Δ and all graphs over Σ, Δ .

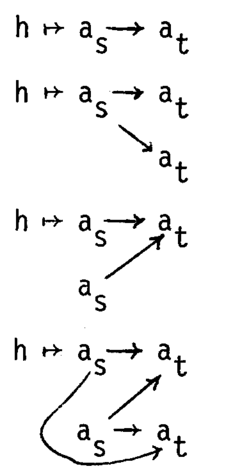
Let $G = (\Sigma, \Delta, P, C, S)$ where $\Delta = \{h\}$,

S is a

P contains



and C contains



Now C preserves all edges at each derivation step and moreover adds the appropriate new edges to maintain completeness. For each complete graph it is straightforward to construct a derivation in G from S which generates it. Assume G generates at least one incomplete graph. Assume U is an incomplete graph in L(G) such that there is no incomplete graph V in L(G) with either fewer nodes or more edges. Consider all T such that $T \Rightarrow U$ in G and $T \neq U$. Now T is complete, since either T has fewer nodes than U or T has the same number of nodes as U but more edges (fewer edges is not possible since edges cannot be created with C . But if T is complete then U cannot be incomplete by examination of C . Hence all U in L(G) are complete. This can be proved rigorously by induction on the number of nodes.

We now turn to a system generating all graphs. In this case there are a lot of connection rules, which are straightforward to specify. In the completeness example only one kind of connection was necessary, but in the following system all possible connections, including no connections, must be specified.

Let $G = (\Sigma, \Delta, P, C, S)$ where $\Delta = \{h\}$,

S is $\rightarrow a \rightarrow$

P contains $a \mapsto \rightleftarrows a \rightleftarrows$

$a \mapsto \rightleftarrows a \leftarrow a \rightleftarrows$

$a \mapsto \rightleftarrows a \leftarrow a \rightleftarrows$

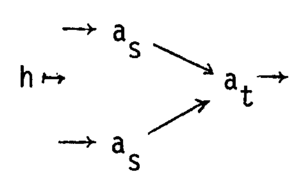
$a \mapsto \rightleftarrows a \rightleftarrows a \rightleftarrows$

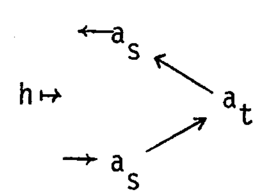
$a \mapsto \rightarrow a \rightarrow \quad \rightarrow a \rightarrow$

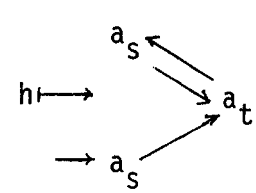
C contains $h \mapsto \rightarrow a_s \rightarrow a_t \rightarrow$

$h \mapsto \leftarrow a_s \leftarrow a_t \leftarrow$

$h \mapsto \rightarrow a_s \rightarrow \quad \rightarrow a \rightarrow$





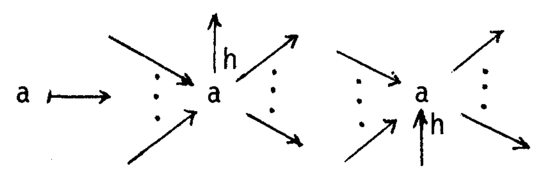


and all other possible edge combinations between 3 nodes, together with those in which s and t are interchanged. It is easy to see that every graph is generated by G .

We close this section by demonstrating a self-reproducing system, that is one which generates multiple copies of the start graph.

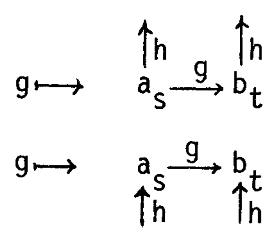
Let S in $[\Sigma, \Delta]_+$ be the start graph.

P contains



for all a in Σ , that is, each node duplicates itself with all possible hands over Δ (labels not shown in the diagram) and the hands shown for h , h not in Δ .

C contains



for all g in Δ and all a, b in Σ .

At each derivation step two new copies of S are made from each old copy of S and each new copy is reconnected correctly by use of the extra hands labelled h in the productions. Since S does not contain a hand labelled h neither do any of its offspring.

6. Decidability Results

Many of the basic questions about PGOL systems involve decidability issues. For example, does a given PGOL system G generate a finite or infinite graph language, is $L(G) = [\Sigma, \Delta]_+$, is membership decidable, are all graphs in $L(G)$ connected (that is, is the underlying undirected graph connected), and so on.

Our first result concerns growth and Parikh functions for DPGOL systems.

Theorem 6.1

Let G_1 and G_2 be two arbitrary DPGOL systems, then for $X = V$, E or λ we have:

X-growth and X-Parikh equivalence of G_1 and G_2

are decidable.

Proof: By the reduction results in Section 4 and since both X-growth and X-Parikh equivalence of two WPDOL systems are decidable properties.

Since it is decidable if an EOL language or its length set is finite we have:

Theorem 6.2

(1) Let G be a PGOL system. It is decidable whether VS(G) is finite or infinite.

(2) Let G be a PGOL system. It is decidable whether L(G) is finite or infinite (follows from (1)).

(3) Let G be a PGOOL system. It decidable whether ES(G) is finite or infinite.

(4) Let G be a PGOOL system. It is decidable whether L(G) has bounded surface area.

We say a PGOL system G has <u>bounded surface area</u> if there is an integer $k \geq 0$ such that for all U in L(G) the set of nodes $\{u$: either (u, h, ∞) or (∞, h, u) is in S, P or C for some h in $\Delta\}$ has cardinality $\leq k$. We suitably modify Construction 4.2 to take the environment node into account.

In Culik II and Lindenmayer (1976) it is proved to be decidable whether a given abstract graph U is generated as a (full) subgraph in a PGOL system G . In other words whether there is an abstract graph V in L(G) such that $U \leq V$ or $U \leq_f V$ is decidable. This also solves the membership problem for PGOL systems. We reprove this result somewhat differently, which enables us to solve other subgraph occurrence problems, which are of independent interest. We present results only for the full subgraph case, the corresponding subgraph results follow similarly.

<u>Definition</u>

Let $G = (\Sigma, \Delta, P, C, S)$ be a PGOL system. An abstract graph U in $[\Sigma, \Delta]_*$ <u>occurs in G</u> if there is a derivation $S \Rightarrow^* V$, for some V in $[\Sigma, \Delta]_+$ with $U \leq_f V$. Similarly we say U <u>occurs finitely</u> (<u>infinitely</u>) <u>in G</u> if the set $\{i : S \Rightarrow^i V$, for some V , $U \leq_f V\}$ is finite (infinite). We say U is <u>preserved</u> (<u>ultimately preserved</u>) <u>in G</u> if for all $i \geq 0$ (for $i \geq t$, some $t \geq 0$) , $S \Rightarrow^i V$ and $U \leq_f V$, for some V in L(G) .

Biologically these variants of the subgraph occurrence problem correspond to the survival of a subunit or sub-assemblage of cells during development. We now provide the notion of a subgraph derivation graph.

Construction 6.1

Given a PGOL system $G = (\Sigma, \Delta, P, C, S)$ and k an integer, $k > 0$, construct the associated k-subgraph-derivation graph $H_{G,k}$ over K, where $K = 2^J$ and J is the set of all graphs with at most k nodes over Σ, Δ, as follows:

Let $H_{G,k} = (V, \varphi_H, E)$ where V is a set of nodes, $\#V = 2^{\#J}$, φ_H: $V \to K$ is a node labelling isomorphism and $E \subseteq V \times V$ is a set of directed edges. $E = \{(u, v) : u, v$ in V, $\varphi_H(v) = \{D : A$ is in $\varphi_H(u)$, $A \Rightarrow B$, $D \leq_f B$ and D is in $J\}\}$.

For all A in $[\Sigma, \Delta]_+$, $k > 0$ an integer, let $k : A$ denote the set of all graphs B with at most k nodes such that $B \leq_f A$.

Clearly, $H_{G,k}$ is a finite graph. Moreover a graph U occurs in G iff there is path from u, the node labelled by $k : S$, $k = \#_N(U)$, in $H_{G,k}$ to a node v labelled with a set of graphs containing U. This follows since a PGOL system does not erase nodes and therefore a k-node full subgraph is obtained in one derivation step from a full subgraph with at most k nodes. This gives the first result, which was proved somewhat differently in Culik II and Lindenmayer (1976).

Theorem 6.3

Given a graph U over Σ, Δ and PGOL system G over Σ, Δ it is decidable whether U occurs in G.

We have a number of interesting corollaries:

Corollary 6.4

Given a graph U over Σ,Δ and a PGOL system G over Σ,Δ it is decidable whether U is in L(G) . That is, the membership problem is decidable.

Corollary 6.5

Given a PGOL system G it is decidable whether G is reduced.

Proof: Straightforward. □

Corollary 6.6

Given a PGOL system G it is decidable whether for all U in L(G) U is complete.

Proof: It is decidable whether a two node unconnected full subgraph over Σ occurs in L(G). Clearly for all U in L(G) , U is complete iff no such subgraph exists. □

Theorem 6.7

Given a graph U over Σ,Δ and a PGOL system G over Σ,Δ it is decidable:

(i) whether U occurs finitely or infinitely.

(ii) whether U is preserved or ultimately preserved.

Proof: Consider the k-subgraph-derivation graph, $H_{G,k}$, where $k = \#_N(U)$. Let u be the node labelled with k : S . First observe that each node in $H_{G,k}$ has only one successor. Hence there is exactly one path from u to some other node, which may be u itself. Thus, the sequence of nodes $u = u_0, u_1, \ldots$ representing a path in $H_{G,k}$ from u has a cycle, that is, there is an i such that $u_i = u_j$ for some $j < i$. Clearly, there can only be one cycle.

(i) U occurs finitely in G if U is not associated with any node in the cycle, and infinitely if it is.

(ii) U is preserved if U is associated with every node in the cycle-path from u and ultimately preserved if it is associated with all nodes in the cycle. □

We may define the various notions of occurrence with respect to L(G) rather than to the derivations of G. Let us call them the language variants of subgraph occurrence and preservation. For DPG0L systems there is only one derivation sequence and hence the language subgraph problems can be reduced to consideration of the derivation subgraph problems. For PG00L or PG0L systems these problems appear to be difficult.

We now consider perhaps the most interesting PG0L systems from a biological viewpoint, namely, those of bounded degree. We say a PG0L system G is of <u>bounded degree</u> if there is an integer $k \geq 0$ such that every graph generated by G has at most degree k (that is, each node has a total number of edges, leading into and out of it, which is less than or equal to k). Similarly we say G has <u>bounded in-degree</u> (<u>bounded out-degree</u>) analogously. We exclude the environment node from consideration.

We now prove that bounded degreeness is a decidable property. By suitable modifications of the proof technique we also obtain the decidability of bounded in-and out-degreeness.

Theorem 6.8

For DPGOL systems bounded degreeness is a decidable property.

Proof: Our proof technique is an indirect one. We reduce bounded degreeness of DPGOL systems to finiteness of ETOL systems. Since this latter property is decidable we have the result.

Let $G = (\Sigma, \Delta, P, C, S)$ be an arbitrary DPGOL system. We first produce a PGOL system G' which simulates a derivation step of G and nondeterministically marks one node in the derived graph. Secondly, using the string representation technique of Construction 4.2, G' is simulated by a DTOL system G''. Finally, we define a homomorphism θ which deletes node and edge labels and identifies "marked" edge labels. Then $\theta(L(G''))$ is finite iff G has bounded degree.

Without loss of generality assume S is a single node graph. Let $\overline{\Sigma} = \{\overline{a} : a \text{ in } \Sigma\}$ be the marked node alphabet. Construct a PGOL system $G' = (\Sigma \cup \overline{\Sigma}, \Delta, P', C', Z)$. Let

$$P' = P \cup \{\overline{a} \mapsto A' : a \mapsto A \text{ is in } P \text{ and } A' \text{ is in } \text{marked}(A)\} .$$

For each abstract graph A in $[\Sigma, \Delta]_+$, marked(A) is the set of all abstract graphs obtained from A by marking just one appearance of one node label. Let

$$C' = C \cup \{h \mapsto A' : h \mapsto A \text{ is in } C \text{ and } A' \text{ is in } \text{marked}(A)\} .$$

We have ensured that one and only one node label will be marked at each derivation step of G' by the **simple** expedient of marking one node label in the daughter graph of a marked node. Observe that for each triple $(\bar{a}, h, b)$ or $(b, h, \bar{a})$ representing an edge of a derived graph in G' and each A' a daughter graph of $\bar{a}$, there is exactly one connection rule applicable to it, as is the case in the original system G. This is the crucial observation that enables the simulation of G' to be carried out by a DTOL system.

Now, construct a DTOL system $G'' = (\Sigma'', P_1, \ldots, P_m, \sigma)$, where $m = \max(\{\#\ \text{marked}(A) : a \mapsto A \text{ in } P, a \text{ in } \Sigma\})$. $\Sigma'' = \Sigma \cup \bar{\Sigma} \cup \Omega \cup \{N\}$ where $\Omega = ((\Sigma \cup \bar{\Sigma}) \times \Delta \times \Sigma) \cup (\Sigma \times \Delta \times (\Sigma \cup \bar{\Sigma}))$. Consider the representation of Construction 4.2. Let μ map abstract graphs from $[\Sigma \cup \bar{\Sigma}, \Delta]_+$ to ordered strings of node labels, and η map abstract graphs to ordered strings of Ω-symbols. Note that η is only a partial map. Again ν is a partial map from $[\Sigma \cup \bar{\Sigma}, \Delta]_+$ to $(\Sigma \cup \bar{\Sigma})^+ \Omega^*$ defined by $\nu(A) = \mu(A)\eta(A)$, for each A having at most one marked node.

Now each table P_i contains all "productions" from P and C, that is,

$$\{a \to \nu(A) : a \mapsto A \text{ in } P\} \cup \{(a, h, b) \to \eta([st^{-1}(\gamma_C)]) : \text{if } a \overset{h}{\to} b$$
is an edge over Σ, Δ and γ is the maximal stencil applicable to some α and β, concrete daughter graphs of a and $b\}$.

Secondly, we now ensure that each possible marked variant of each production $a \mapsto A$ in P is found in at least one table. For each $a \mapsto A$ in P arbitrarily number the elements of marked(A) as $A_1, \ldots, A_r$ where $1 \le r \le m$. Take $\bar{a} \to \nu(A_i)$ into P_i, $1 \le i \le r$ and complete the remaining tables by taking $\bar{a} \to \nu(A_1)$ into P_i, $r+1 \le i \le m$. Now take the appropriate edge productions into each P_i, $1 \le i \le m$:

for each $\bar{a} \to \nu(A)$ and $b \to \nu(B)$ in P_i and for each h in Δ take:

$$(\bar{a}, h, b) \to w \text{ into } P_i$$

where $w = \eta([st^{-1}(\gamma_C)])$ and γ is the maximal stencil applicable to α , β , for $[\alpha] = A$ and $[\beta] = B$, and similarly for $(b,h,\bar{a}) \to x$.

Finally, for any table P_i which is incomplete for some X in Σ'' add $X \to N$ to P_i , $1 \le i \le m$.

For each triple (a, h, b) there is clearly only one edge production and this also holds true for each triple $(\bar{a}, h, b)$ or $(a, h, \bar{b})$, with respect to each table, since only one production for $\bar{a}$ is taken into each table. Hence we have a DTOL system.

The symbol N never occurs in any derivation from $\sigma = \nu(S)$ in G'' , it merely plays the role of enabling each table to be completed.

Define a homomorphism $\theta : \Sigma'' \to \{d\}^*$ by

$\theta(X) = \lambda$ for X in $\{N\} \cup \Sigma \cup \bar{\Sigma} \cup (\Sigma \times \Delta \times \Sigma)$, and

$\theta(X) = d$ otherwise.

Then $\theta(L(G'')) \subseteq \{d\}^*$ and clearly $\theta(L(G''))$ is finite iff G has bounded degree.

□

Now consider the situation when G is a PGOOL system. In this case the simulation by a PGOL system G' still holds. Moreover G' has the property that for each $(\bar{a}, h, b)$ (or $(a, h, \bar{b})$) and each $\bar{a} \mapsto A$ the stencils applicable to α and β , $[\alpha] = A$ and $\beta = [B]$ where $b \mapsto B$ is in P' , are independent of the choice of production for b . Hence in the simulation of G' by a TOL system G'' there is a separate table for each production $\overline{a} \mapsto A$ for $\overline{a}$ in G' and each table contains exactly those edge productions for $(\overline{a}, h, b)$ and $(b, h, \overline{a})$ which are applicable given $\overline{a} \to A$. The remainder of the proof follows analogously. Hence we have shown:

Theorem 6.9

For PGOOL systems bounded degreeness, bounded in-degreeness and bounded out-degreeness are decidable properties. Moreover, if the degree (in-degree, out-degree) is bounded then it is computable.

The second sentence follows from the proof technique of Theorem 6.8.

The decidability of bounded degreeness for PGOL systems remains open.

Since it has recently been shown that the DOL sequence equivalence problem is decidable (Culik II and Fris 1977), it is natural to ask whether the DPGOL sequence equivalence problem is decidable. That is, given two DPGOL systems G_1 and G_2 whether it is decidable if they generate the same sequence of graphs. We have shown that various necessary conditions for sequence equivalence to hold are decidable, namely, node Parikh function equivalence, edge Parikh function equivalence and whether both are of the same degree. However, the DPGOL sequence equivalence problem appears to be complex and although it is still open, we conjecture that it is decidable.

We close this section by mentioning a number of interesting open problems. Firstly, functionality is such a basic notion that whether it is decidable or not is an important problem. Secondly, connectedness is both biologically and mathematically interesting. Whether or not there are generated graphs containing 2 or more disconnected components in a PGOL (or even DPGOL) system is an open decidability question. Thirdly, a question of mathematical interest is whether or not all graphs generated by a PGOL (or DPGOL) system are planar. This is also open.

7. The Context-Freeness of PGOL Systems

When speaking of context-freeness for string rewriting systems we usually mean that each symbol is rewritten independently of its context. For example, in context-free grammars and EOL systems. However, in PGOL systems and other graph rewriting systems the situation is not as clear. We not only have node rewriting but also connection rules. In the case of PGOL systems node replacement is context-free, hence the choice of acronym. However whether PGOL systems are "context-free" in total, that is also as far as the connection rules are concerned, is subject to question. Some measure of their context-freeness is given by the positive results for the decidability of the membership problem, the effectiveness of the construction of reduced PGOL systems, and the decidability of the finiteness problem. Similarly the reduction to a full stencil normal form is also evidence in favor of the context-freeness of PGOL systems.

On the other hand we have the results of Section 4 in which it is shown that PGOL systems generate at least ETOL length sets when considering edge size sets. Whether ETOL length sets exhaust the PGOL edge size sets is an open problem. However this leads to the introduction of the "doubly interactionless" systems, namely the PGOOL systems. These have the pleasant property that the deterministic restrictions of PGOL and PGOOL systems coincide. We claim that the PGOOL systems are the truly context-free parallel graph rewriting systems. This is reflected in the result that PGOOL edge size sets are EOL length sets in contradistinction to PGOL edge size sets. Moreover bounded degreeness is decidable for PGOOL systems whereas this is still open for PGOL systems.

Observe that minor contextual change in the definition of PGOL systems gives them the ability to simulate P1L systems (L systems in which rewriting of a symbol depends upon the left neighboring symbol). For example, (1) let the productions for nodes depend upon a neighboring node (one that has an edge to the given node). Clearly P1L systems can be simulated. (2) let the productions for nodes depend upon an edge leading into the node. Again, P1L systems can be simulated. (3) Finally, assume the connection rule for each edge is chosen first and then secondly daughter graphs are generated to which the given stencils apply. This is essentially predictive context as in Culik II and Opatrny (1974). It again enables P1L systems to be simulated.

Thus, it seems to us that while PGOL systems may be considered to be context-free, PGOOL systems should be considered to be truly context-free.

References

Abe, N., Mizumoto, M., Toyoda, J., and Tanaka, K. (1973), Web grammars and several graphs, J. of Computer and System Sciences 7, 37-65.

Cheung, (1978) Private communication.

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

Culik II, K., and Lindenmayer, A. (1975), Graph OL-systems and recurrence systems on graphs, Proceedings of the 8th Hawaii Conference on Systems Science, January 1975.

Culik II, K., and Lindenmayer, A. (1976), Parallel graph generating and graph recurrence systems for multicellular development, Int. J. General Systems 3, 53-66.

Culik II, K., and Opatrny, J. (1974), Context in parallel rewriting, in L Systems (eds., Rozenberg and Salomaa), Springer-Verlag Lecture Notes in Computer Science 15, Heidelberg, 230-243.

Ehrenfeucht, A., and Rozenberg, G. (1974), Nonterminals versus homomorphisms in defining languages for some classes of rewriting systems, Acta Informatica 3, 265-283.

Ehrig, H., and Kreowski, H.J. (1976), Parallel graph grammars, in Automata, Languages, Development (eds. Lindenmayer and Rozenberg), North-Holland Publishing Co., Amsterdam, 425-442.

Ehrig, H., Kreowski, H.J., Maggiolo-Schettini, A., Rosen, B.K., and Winkowski, J. (1978), Deriving structures from structures, IBM Research Report RC 7046.

Ehrig, H., and Rozenberg, G. (1976), Some definitional suggestions for parallel graph grammars, in Automata, Languages, Development (eds. Lindenmayer and Rozenberg), North-Holland Publishing Co., Amsterdam, 443-468.

Ehrig, H., and Tischer, K.W. (1975), Graph grammars and applications to specialization and evolution in biology, J. of Computer and Systems Sciences 11, 212-236.

Herman, G.T., Lee, K.P., Leeuwen, J. van, and Rozenberg, G. (1973), Characterization of unary developmental languages, Discrete Mathematics 6, 235-247.

Herman, G.T. and Rozenberg, G. (1975), Developmental Systems and Languages, North-Holland Publishing Co., Amsterdam.

Lindenmayer, A., and Culik II, K. (1978), Growing cellular systems: generation of graphs by parallel rewriting, Int. J. General Systems, to appear.

Mayoh, B.H. (1973), Mathematical models for cellular organisms, Aarhus University Computer Science Report No. DAIMI PB-12.

Mayoh, B.H. (1974), Multidimensional Lindenmayer organisms, in L Systems (eds. Rozenberg and Salomaa), Springer-Verlag Lecture Notes in Computer Science 15, Heidelberg, 302-326.

Montanari, U.G. (1970), Separable graphs, planar graphs, and web grammars, Information and Control 16, 243-267.

Nagl, M. (1976), On a generalization of Lindenmayer-systems to labelled graphs, in Automate, Languages, Development (eds. Lindenmayer and Rozenberg), North-Holland Publishing Co., Amsterdam, 487-508.

Rosen, B.K. (1975), Deriving graphs from graphs by applying a production, Acta Informatica 4, 337-357.

Rosenfeld, A., and Milgram, D. (1972), Web automata and web grammars, in Machine Intelligence 7 (eds. Meltzer and Michie), Edinburgh University Press, 307-324.

Rozenberg, G., and Salomaa, A. (1978), The Mathematical Theory of L Systems, to appear.

Schneider, H.J., and Ehrig, H. (1976), Grammars on partial graphs, Acta Informatica 6, 297-316.

Della Vigna, P., and Ghezzi, C. (1978), Context-free graph grammars, Information and Control 37, 207-233.