Accepted Manuscript

Lobo D., Vico F.J., Dassow J.
"Graph grammars with string-regulated rewriting"
Theoretical Computer Science 412(43), pp. 6101-6111, 2011.
DOI: 10.1016/j.tcs.2011.07.004

Graph grammars with string-regulated rewriting

Daniel Lobo[a,b,*], Francisco J. Vico[a], Jürgen Dassow[b]

[a] *Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Severo Ochoa 4, E-29590 Málaga, Spain*
[b] *Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, PSF 4120, D-39016 Magdeburg, Germany*

Abstract

Multicellular organisms undergo a complex developmental process, orchestrated by the genetic information in their cells, in order to form a newborn individual from a fertilized egg. This complex process, not completely understood yet, is believed to have a key role in generating the impressive biotic diversity of organisms found on earth. Inspired by mechanisms of Eukaryotic genetic expression, we propose and analyse graph grammars with string-regulated rewriting. In these grammatical systems a genome sequence is represented by a regulatory string, a graph corresponds to an organism, and a set of graph grammar rules represents different forms of implementing cell division. Accordingly, a graph derivation by the graph grammar resembles the developmental process of an organism. We give examples of the concept and compare its generative power to the power of the traditional context-free graph grammars. We demonstrate that the power of expression increases when genetic regulation is included in the model, as compared to non-regulated grammars. Additionally, we propose a hierarchy of string-regulated graph grammars, arranged by expressive-power. These results highlight the key role that the transmission of regulatory information during development has in the emergence of biological diversity.

1 Introduction and Motivation

The great variety in the size, form, organization, and patterning of living beings raises deep questions about their evolutionary origins. Nowadays, our planet is populated by some 1 to 20 million animal species. This astonishing diversity of forms has emerged from the evolution of novel features, a process not fully understood yet [5, 25].

Conventionally, the modern evolutionary synthesis of evolution [16], which integrated Mendelian genetics [24] with Darwinian natural selection [8], is still, to a large extent, the current paradigm in evolutionary biology [23]. However, the standard modern synthesis framework cannot explain the origins of this multitude of animal forms in mechanistic terms. This problem might originate in the neglect of the intrinsic complexity of the developmental process [26], through which a single-celled egg gives rise to a complex, multi-celled organism.

The process of development is regulated by a genome, the entire DNA sequence of an organism inherited from its ancestors and present almost identical in every cell of an organism. However, the familiar idea of a genome as a blueprint is hopelessly misleading [2]. The straightforward mapping between plan and product does not apply for genome and organism body [22].

In contrast, development occurs through a series of simultaneously regulated cell divisions, a process where a parent cell is replaced by two or more daughter cells. An important aspect of the biological developmental process is its regulation by the genome; indeed, development has been increasingly viewed as a process orchestrated by the selective expression of genes [5].

Understanding the role of development, and its regulation, in generating the impressive biotic diversity of organisms found on earth is an active subject of research [3]. In order to shed light on the role of the

*Corresponding author; current address: Biology Department and Center for Regenerative and Developmental Biology, Tufts University, 200 Boston Ave., Suite 4600, Medford MA 02155, USA. Tel: +1 6176274171

E-mail addresses: dlobo@geb.uma.es (D. Lobo), fjv@geb.uma.es (F.J. Vico), dassow@iws.cs.uni-magdeburg.de (J. Dassow)

genetic regulation in the biological developmental process, we propose and study in this work a formal model of development based on the concept of graph grammars with string-regulated rewriting.

The proposed formal model abstracts a series of biological developmental concepts. First, an organism is represented by a graph, whose edges represent the organism's cells. The nodes in the graph embody the connectivity among cells, i.e., two cells are connected if the edges representing them share a node. Second, the developmental process of the organism is implemented by an edge-replacement graph grammar. In this way, the rewriting rules in the grammar define the types of divisions that cells can perform during development. Hence, a developmental process consists in a derivation of the grammar: starting with a single edge that connects two nodes (the zygote or axiom in the grammar) the graph is rewritten successively according to the set of rules defined in the grammar. Finally, this developmental process is regulated by a genetic regulatory mechanism, where a symbol represents a structural gene and a string of symbols represents a genome. In the same way that every cell in an organism contains a genome, every edge in a graph is labelled with a string which regulates the rules of the grammar to be applied. The grammar associates symbols with rewriting rules and the regulation during a derivation step is performed by applying to the edge a rule corresponding to the first symbol of its string label. In this way, the sequence of structural genes that are expressed in every cell during biological development is abstracted in the proposed model by a sequence of symbols that activates rewriting rules.

An important aspect during biological development is that the genome propagates through the organism's cells as part of the myriad of cell divisions that take place while the organism develops. Indeed, the original genetic material stored in a parent cell is copied equally to its daughter cells during a division. Similarly to the biological cell division, the regulatory string in the proposed model propagates through the graph's edges in each rewriting step. The string label of the replaced edge (except the first symbol, which defines the rule to apply) is copied equally to the new edges after the application of a grammar rule.

However, there exist mechanisms that occasionally cause daughter cells to gain additional regulatory information as development progresses. One such mechanism is DNA methylation [18], which consists in the addition of methyl groups to specific genes, blocking their current or future expression. The new expression state, which excludes the silenced genes by methyl groups, can be inherited by daughter cells following cell division. In the same way, a rule in the proposed model may define a disjoint or incomplete repartition of the string among the new edges, equivalent to silencing the genes contained in the substrings not copied to the new edges. In this case, similarly to methylated cells, the edges might continue the derivation applying different rules, and hence they can have different fates.

In order to study the role of a genetic regulation in development, we have analysed the expression power of the proposed model of string-regulated graph grammars, comparing it with the traditional non-regulated context-free graph grammars. The results demonstrate that the genetic regulation incorporated in the model permits an increased power of expression, i.e., a bigger collection of graph topologies represented by these regulated grammars. Additionally, it is demonstrated that the expression power depends both on the types of genetic repartition during cell division and the types of genome strings set allowed in the zygote, which yields to an expressive-power hierarchy of variants of string-regulated graph grammars. Furthermore, the biological knowledge incorporated in the model concerns also the evolutionary level, since this form of implicit encoding allows the evolution of topologies which resemble interesting properties of biological organisms [20, 21].

The rest of the paper is organized as follows. In Section 2, the related work of the proposed model is reviewed. In Section 3, we define formally the proposed model. In Section 4, we illustrate the model through a series of examples. In Section 5, we provide results on the power of string-regulated graph grammars. Finally, the conclusions derived from the results are discussed in Section 6.

2 Related work

As shown before, development occurs through a series of simultaneously regulated cell divisions, a process where a parent cell is replaced by two or more daughter cells. Due to this replacing mechanism, multicellular development has been investigated and found to inspire some parallel rewriting systems in the field of formal language theory [14].

In a pioneering work, Lindenmayer proposed L-systems [19], a parallel rewriting string grammar

extensively applied for the modelling of growing forms, especially plants [28]. In order to transform the resulting derived string into a graphical form, L-systems are usually combined with turtle geometry [1] by associating every final symbol in the grammar to a movement of a drawing cursor (the turtle) in a plane. A resulting string of the grammar is then interpreted by drawings of the turtle, resulting in a graphical form. In consequence, a restriction of L-systems combined with turtle graphics is that, by themselves, they can only model filamentous organisms. This limitation is not present in the case of graphs grammars, which can serve as a simple, but powerful, description of the developmental process of an organism abstracted as a graph [27].

Graph grammars are interesting from the theoretical point of view because they are a natural generalization of formal language theory based on strings and the theory of term rewriting based on trees. As opposed to the case of strings, there exist many types, depending on the properties of the graphs generated or the embeddings defined in the rewriting. The two most basic choices for rewriting a graph are node replacement and edge replacement [29]. In a node-replacement type graph grammar a node is replaced by a new subgraph which is connected to the remainder of the graph by new edges. In contrast, in an edge replacement graph grammar an edge in the graph is replaced by a new subgraph which is fused to the remainder of the graph through the nodes of the replaced edge. Additionally, other generalizations of edge replacement graph grammars have been proposed. Hyperedge replacement graph grammars are a natural extension of edge replacement graph grammars, where hypergraph are composed by hyperedges that can connect any number of nodes [12]. Moreover, handle-rewriting hypergraph grammars extend hyperedge replacement graph grammars; they are based on the replacement of handles, which means that an edge together with its incident vertices is replaced [7]. In general, handle-rewriting grammars are more powerful than node replacement grammars, which in turn have more power than edge rewriting grammars [7].

Star and parallel grammars have been also proposed as generalizations of edge replacement grammars. Adaptive star grammars are characterized by rule schemas that replace a nonterminal node together with its outgoing edges (a star) with another graph which is glued via a cloning operation to the nodes pointed to by the outgoing edges of the replaced node [11]. On the other hand, parallel graph transformations [13], generalized to parallel high-level replacement systems [32], are used in parallel graph grammars. Parallel graph grammars operates by defining amalgamation schemes that are applied the actual graph; an amalgamation scheme is composed by a (possibly infinite) set of rules that are applied in parallel and that share certain regularity.

Additionally, as we have seen above, an important aspect of the biological developmental process is its regulation by a genome, a string that can be represented as a word over an alphabet of nucleotides. Formal methods have been also applied in the investigation of gene regulation, especially grammar formalisms. Brendel and Busse [4] reported a description of very simple genes by means of regular grammars. Collado-Vides [6] used transformational grammars when modeling the regulation structure of genomes. Grate et al. [15] and Sakakibara et al. [30] considered stochastic context-free grammars for modeling RNA. Searls [31] used definite clause grammars for modeling biological sequences, including genomes. Modeling genomes as formal languages and mutation operartions as operations on strings and languages, Dassow et al. [10] investigated the evolution of genomes.

Finally, it is worth mentioning here that the proposed formalism resembles string grammars with regulated rewriting, reported in [9], where the generative power of string grammar increases by imposing restrictions to the derivation process.

3 The Model and Some Definitions

Definition 1. *(1) Let C be an arbitrary, but fixed alphabet, called set of label symbols.*
(2) A (directed) unlabelled graph is a system (E, V, s, t), where

- *E is a set of edges;*
- *V is a set of nodes; and*
- *$s, t : E \to V$ are maps from E into V, assigning a source and a target to each edge;*

(3) A (directed) string-labelled graph is a system $G = (E, V, s, t, l)$, where

- (E, V, s, t) *is an unlabelled graph, called underlying graph and denoted by* $U(G)$*; and*
- $l : E \to C^*$ *is a mapping, called labelling.*

(4) The set of all string-labelled graphs over a set of string labels C^* *is denoted by* $\mathcal{G}_{C^*}$*. The set of string-labelled graphs whose edges are all labelled with a string* $\omega \in C^*$ *is denoted by* $\mathcal{G}_\omega$*.*

Remark. The components of a graph G are denoted by E_G, V_G, s_G, t_G, and l_G, respectively.

Definition 2. *A string handle is a string-labelled graph of the form* $H = (\{e_1\}, \{v_0, v_1\}, s, t, l)$ *with* $s(e_1) = v_0$ *and* $t(e_1) = v_1$*. If* $l(e_1) = \omega \in C^*$*, then* H *is called the string handle induced by* ω *and is denoted by* $\omega^\S$*.*

Definition 3. *(1) Let* G *be a string-labelled graph, and let* $e \in E_G$ *be an edge in* G*. The result of removing* e *from* G *is a string-labelled graph* $H = (E_H, V_G, s_H, t_H, l_H)$*, where* $E_H = E_G - \{e\}$ *and* s_H*,* t_H*,* l_H *are the restrictions of* s_G*,* t_G*,* l_G *to* E_H*, respectively. The graph* H *is denoted by* $\text{REMOVE}(G, e)$*.*
(2) Let G *and* B *be string-labelled graphs and* (g_1, g_2)*,* (b_1, b_2) *be ordered pairs of nodes of* G *and* B*, respectively. The result of inserting* B *at* (b_1, b_2) *in the graph* G *at* (g_1, g_2) *is the graph* $H = (E_H, V_H, s_H, t_H, l_H)$*, where*

$$\begin{aligned} E_H &= E_G + E_B, \\ V_H &= V_G + (V_B - \{b_1, b_2\}), \end{aligned}$$

and s_H*,* t_H*,* l_H *are defined as follows*

$$s_H(e) = \begin{cases} s_G(e) & \text{if } e \in E_G, \\ g_1 & \text{if } e \in E_B \wedge s_B(e) = b_1, \\ g_2 & \text{if } e \in E_B \wedge s_D(e) = b_2, \\ s_B(e) & \text{otherwise,} \end{cases}$$

$$t_H(e) = \begin{cases} t_G(e) & \text{if } e \in E_G, \\ g_2 & \text{if } e \in E_B \wedge t_B(e) = b_2, \\ g_1 & \text{if } e \in E_B \wedge t_B(e) = b_1, \\ s_B(e) & \text{otherwise,} \end{cases}$$

$$l_H(e) = \begin{cases} l_G(e) & \text{if } e \in E_G, \\ l_B(e) & \text{if } e \in E_B. \end{cases}$$

The graph H *is denoted by* $\text{INSERT}(G, (g_1, g_2), B, (b_1, b_2))$
(3) Let D *be an unlabelled graph,* $m : E_D \to \{1, \ldots, n\}$*, where* $n \in \mathbb{N}$*, a mapping of edges of* D*, and* ω *a string of label symbols. The result of labelling* D *with* ω *according to the mapping* m *is the string-labelled graph* $H = (E_D, V_D, s_D, t_D, l_H)$*, where*

$$\begin{aligned} \omega &= \gamma_1 \gamma_2 \ldots \gamma_n,\ \gamma_i \in C^*, \\ l_H(e_i) &= \begin{cases} \epsilon & \text{if } e_i \notin Dom(m), \\ \gamma_{m(e_i)} & \text{otherwise;} \end{cases} \end{aligned}$$

and

$$|\gamma_i| = \begin{cases} \left\lceil \frac{|\omega|}{n} \right\rceil & \text{if } i \leq mod(|\omega|, n), \\ \\ \left\lfloor \frac{|\omega|}{n} \right\rfloor & \text{otherwise.} \end{cases}$$

The graph H *is denoted by* $\text{LABEL}(D, m, \omega)$*.*

Definition 4. *(1) A (string-regulated graph grammar) production over* C *is an ordered tuple* $p = (X, D, (d_1, d_2), m)$*, where*

- $X \in C$ *is the edge label symbol,*
- $D \in \mathcal{G}$ *is an unlabelled graph,*
- (d_1, d_2) *is an ordered pair where* d_1 *and* d_2 *are nodes of* D*, and*
- $m : E_D \to \{1, \ldots, n\}$*, where* $n \in \mathbb{N}$*, is an mapping of edges of* D*.*

(2) Given string-labelled graphs G *and* H*, a production* $p = (X, D, (d_1, d_2), m)$*, and an edge* $e \in E_G$ *with* $l_G(e) = X\omega$*,* $\omega \in C^*$*,* G *directly derives* H *(through* p *applied to* e*) if* H *is isomorphic to the graph*

$$\text{INSERT}(\text{REMOVE}(G, e), (s_G(e), t_G(e)), \text{LABEL}(D, m, \omega), (d_1, d_2)).$$

A direct derivation from G *to* H *(through* p *applied to* e*) is denoted by* $G \underset{p,e}{\Longrightarrow} H$*. A sequence of direct derivations* $G_0 \underset{p_1,e_1}{\Longrightarrow} G_1 \underset{p_2,e_2}{\Longrightarrow} \ldots \underset{p_m,e_m}{\Longrightarrow} G_m$ *is called a derivation from* G_0 *to* G_m *. If* P *is a set of productions and* $p_1, \ldots, p_m \in P$*, this is abbreviated by* $G_0 \overset{*}{\underset{P}{\Longrightarrow}} G_m$*.*

Definition 5. *(1) A string-regulated graph grammar is a system* $S = (C, P, W)$*, where*

- C *is a set of label symbols;*
- P *is a finite set of productions over* C*;*
- $W \subseteq C^*$ *is a set of possible string labels for the initial string-labelled graph handle, the axiom.*

(2) Let $S = (C, P, W)$ *be a string-regulated graph grammar. The graph language generated by* S *consists of all string-labelled graphs with empty string edge labels derivable from the set* W *of initial labels, i.e.,*

$$L(S) = \left\{ G \in \mathcal{G}_\varepsilon | \omega^\S \overset{*}{\underset{P}{\Longrightarrow}} G, \omega \in W \right\}.$$

We define several variations of string-regulated graph grammars attending to the type of mapping function and set of string labels.

Definition 6. *(1) A string regulated graph grammar is simple if all the mappings in its productions are injective. Otherwise, it is general.*
(2) A string regulated graph grammar is regular if its set of initial string labels is a regular set. Otherwise, it is arbitrary.

Below, some string-regulated graph grammars are defined, being Example 1 simple and regular, Example 2 general and regular, and Example 3 simple and arbitrary.

We denote the families of all languages generated by the different types of string regulated graph grammars as follows:

- the languages generated by simple regular string-regulated graph grammars by $srSR$,
- the languages generated by general regular string-regulated graph grammars by $grSR$,
- the languages generated by simple arbitrary string-regulated graph grammars by $saSR$,
- the languages generated by general arbitrary string-regulated graph grammars by $gaSR$, and
- the set of all recursively enumerable graph languages by AL.

We now formalize context-free graph grammars and their languages used in the rest of the paper.

Definition 7. *(1) Let* C *be an arbitrary, but fixed alphabet, called set of label symbols.*
(2) A (directed) symbol-labelled graph is a system $G = (E, V, s, t, l)$*, where*

- (E, V, s, t) *is an unlabelled graph, called underlying graph and denoted by* $U(G)$*; and*
- $l : E \to C$ *is a mapping, called labelling.*

(3) The set of all symbol-labelled graphs over a set of symbol labels C is denoted by $\mathcal{G}_C$. The set of symbol-labelled graphs whose edges are all labelled with a symbol $\alpha \in C$ is denoted by $\mathcal{G}_\alpha$.

We define a symbol handle and a symbol handle induced by a symbol $\alpha \in C$, and denoted by $\alpha^{\S}$, analogously as in the case for string-labelled graphs (see Definition 2).

Additionally, we define the graphs denoted by REMOVE(G, e) and INSERT$(G, (g_1, g_2), B, (b_1, b_2))$, being G and B symbol-labelled graphs, analogously as in the case for string-labelled graphs (see Definition 3).

Definition 8. *(1) A (context-free graph grammar) production over N is an ordered tuple $p = (X, R, (r_1, r_2))$, where*

- *$X \in N$ is the edge label symbol,*
- *$R \in \mathcal{G}_C$ is a symbol-labelled graph, and*
- *(r_1, r_2) is an ordered pair where r_1 and r_2 are nodes of R,*

(2) Given symbol-labelled graphs G and H, a production $p = (X, R, (r_1, r_2))$, and an edge $e \in E_G$ with $l_G(e) = X$, G directly derives H (through p applied to e) if H is isomorphic to the graph

$$\text{INSERT}(\text{REMOVE}(G, e), (s_G(e), t_G(e)), R, (r_1, r_2)).$$

A direct derivation from G to H (through p applied to e) is denoted by $G \underset{p,e}{\Longrightarrow} H$. A sequence of direct derivations $G_0 \underset{p_1,e_1}{\Longrightarrow} G_1 \underset{p_2,e_2}{\Longrightarrow} \ldots \underset{p_m,e_m}{\Longrightarrow} G_m$ is called a derivation from G_0 to G_m . If P is a set of productions and $p_1, \ldots, p_m \in P$, this is abbreviated by $G_0 \overset{*}{\underset{P}{\Longrightarrow}} G_m$.

Definition 9. *(1) A context-free graph grammar is a system $S = (N, T, P, Z)$, where*

- *N is a set of nonterminal symbols;*
- *T is a set of terminal symbols;*
- *P is a finite set of productions over N;*
- *$Z \in N$ is the axiom.*

(2) Let $S = (N, T, P, Z)$ be a context-free graph grammar. The graph language generated by S consists of all terminal-labelled graphs derivable from the the handle induced by Z, i.e.,

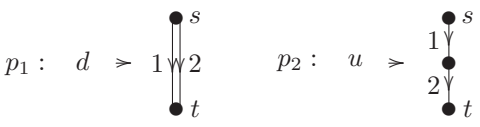
$$L(S) = \left\{ G \in \mathcal{G}_T | Z^{\S} \overset{*}{\underset{P}{\Longrightarrow}} G \right\}.$$

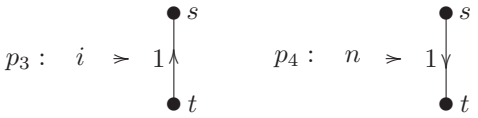
We finish this section with some notations used in the rest of the paper. We denote a directed path graph with n edges by $\leftarrow^n$ or $\rightarrow^n$. Similarly, a concatenation of n cycle graphs of two vertices is denoted by $\rightleftarrows^n$. By $G \oplus H$ we denote the concatenation of G and H, being G and H directed path graphs or a concatenation of cycle graphs of two vertices.

4 Some Examples

In this section we present some examples which illustrate the concepts given in the preceding section and show that some languages can and some languages cannot be generated.

Example 1. We consider a simple regular string-regulated graph grammar $S = (\{d, u, i, n\}, P, \{uduunn\})$ with P consisting of the rules





where the pair (d_1, d_2) is indicated by labelling d_1 with s and d_2 with t, and the mapping m is indicated by the numeric labels in the edges. Figure 1 shows the only derivation of the grammar that leads to the unique graph included in $L(S)$. The symbol that heads the label of the edge that is replaced in the next derivation step is underlined.

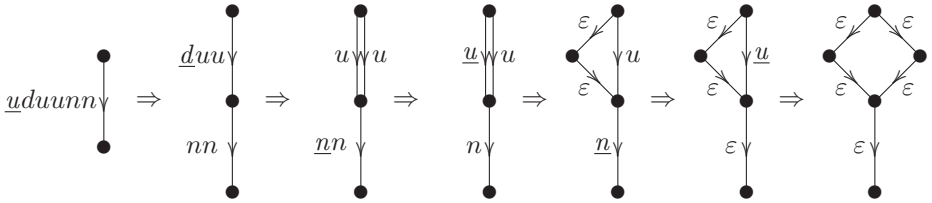


Figure 1: A derivation of the string-regulated graph grammar defined in Example 1.

We now show the following fact:

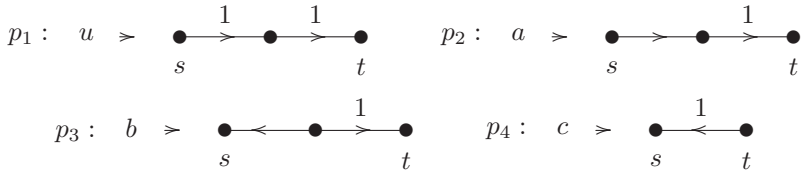
There does not exist a string label ω for the string-regulated graph grammar S defined above that can generate the graph K_4.

Proof. We assume that there exists such string label ω for the string-regulated graph grammar S that generates the graph K_4, and we obtain a contradiction. Let

$$\omega^{\S} \underset{p_1,e_1}{\Longrightarrow} G_1 \underset{p_2,e_2}{\Longrightarrow} \ldots \underset{p_m,e_m}{\Longrightarrow} K_4$$

be a sequence of direct derivations. Now, one of such direct derivations $\underset{p_i,e_i}{\Longrightarrow}$ must generate the fourth node of the graph. Moreover, there is no production in the grammar that generates an edge between two non-connected nodes, so, the production p_i that generates the fourth node of the graph must generate edges between the new node and the other three nodes. Though, there is no production in the grammar that can create a node connected to other three nodes, and a contradiction occurs. □

Example 2. Let $S = (\{u, a, b, c\}, P, u\{a, b, c\}^*)$ be a general regular string-regulated graph grammar where P is given by



where the pair (d_1, d_2) is indicated by labelling d_1 with s and d_2 with t, and the mapping m is indicated by the numeric labels in the edges.

Any derivation has to start with p_1, which yields a linear graph of two edges with the same labelling. Hence, the two edges of such graph derive the same linear subgraph, both sharing the last node and the first node, respectively. If H is a linear graph it can be noted as a word over $\{\leftarrow, \rightarrow\}$. Thus the generated language is the set of all linear graphs where the first half subgraph equals to the second half subgraph, i.e., $L(S) = \{H \oplus H | H \in \{\leftarrow, \rightarrow\}^*\}$.

As a result, we now show that our concept differs from that of context-free graph grammars with respect to generative power.

Proposition 1. *Let S be the string-regulated graph grammar defined in Example 2. There does not exist a context-free graph grammar S' such that $L(S') = L(S)$, i.e., the graph language $L(S)$ is a non-context-free graph language.*

Proof. We show that the language $L(S)$ is a non-context-free language using the pumping lemma of context-free graph languages [17]. We assume that $L(S)$ is a context-free graph language and shall obtain a contradiction.

Let p and q be the constants that satisfy the conditions set by the pumping lemma and are guaranteed to exist. Select the graph $G = \rightarrow^{p+q} \oplus \leftarrow^{p+q} \oplus \rightarrow^{p+q} \oplus \leftarrow^{p+q}$. Clearly, G is a member of $L(S)$ and its number of edges is greater than p. We show that G cannot be pumped. The pumping lemma states that G can be pumped by gluing G of a sequence of three subgraphs $FIRST$, $LINK$, and $LAST$, where $|E_{LINK}| + |E_{LAST}| \leq q$.

Since the graph G is linear, it can be divided into five linear subgraphs $u \oplus v \oplus w \oplus x \oplus y$, where $FIRST$ corresponds with u and y, $LINK$ with v and x, and $LAST$ with w.

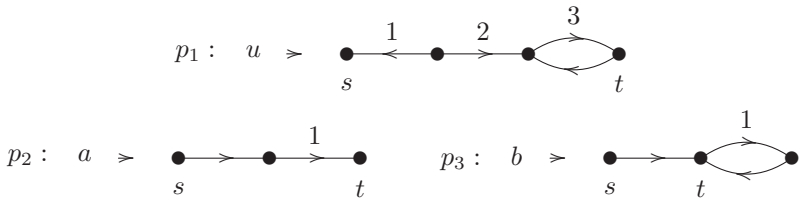
Then, we show that the linear subgraph $v \oplus w \oplus x$, which corresponds with $LINK$ and $LAST$, must straddle the midpoint of G. Otherwise, if the subgraph occurs only in the first half of G, pumping up G produces the graph $u \oplus v \oplus v \oplus w \oplus x \oplus x \oplus y$, which moves at least a $\leftarrow$ into the first position of the second half, and so it cannot be of the form $H \oplus H$. Similarly, if the linear subgraph $v \oplus w \oplus x$ occurs in the second half of G, pumping up G produces the graph $u \oplus v \oplus v \oplus w \oplus x \oplus x \oplus y$, which moves at least a $\rightarrow$ into the last position of the first half, and so it cannot be of the form $H \oplus H$.

But if the subgraph $v \oplus w \oplus x$ straddles the midpoint of G, when we try to pump down G it is produced the graph $u \oplus w \oplus y$ which has the form $\rightarrow^{p+q} \oplus \leftarrow^{i} \oplus \rightarrow^{j} \oplus \leftarrow^{p+q}$, where i and j cannot both be $p+q$ since $|E_{v \oplus w \oplus x}| \leq q$. This graph is not of the form $H \oplus H$.

Finally, when G is divided into five linear subgraphs $u \oplus v \oplus w \oplus x \oplus y$, we should also consider the possibility that $FIRST$ corresponds with w, $LINK$ with v and x, and $LAST$ with u and y. But, when we try to pump down G in this case it is produced the graph $u \oplus w \oplus y$ which has the form $\rightarrow^{i} \oplus \leftarrow^{p+q} \oplus \rightarrow^{p+q} \oplus \leftarrow^{j}$, where i and j cannot both be $p+q$ since $|E_{u \oplus v}| + |E_{x \oplus y}| \leq q$. This graph is not of the form $H \oplus H$.

Thus G cannot be pumped, and $L(S)$ is a non-context-free language. □

Example 3. Let $S = (\{u, a, b\}, P, \{ua^{2n}b^{n} \mid n \geq 1)$ be a simple arbitrary string-regulated graph grammar with P given by



where the pair (d_1, d_2) is indicated by labelling d_1 with s and d_2 with t, and the mapping m is indicated by the numeric labels in the edges.

Any derivation has to start with p_1 which yields a linear graph of two opposite direction edges followed by a loop. In this first direct derivation, the label is distributed among three edges, namely a^n, a^n, and b^n, respectively. Hence, the grammar derives a linear graph of n left edges connected to a linear graph of n right edges connected to n linear loops, i.e.,

$$L(S) = \{\leftarrow^{n} \oplus \rightarrow^{n} \oplus \rightleftarrows^{n} \mid n \geq 1\}.$$

We give a statement analogous to Proposition 1.

Proposition 2. *Let S be the string-regulated graph grammar defined in Example 3. There does not exist a context-free graph grammar S' such that $L(S') = L(S)$, i.e., the graph language $L(S)$ is a non-context-free graph language.*

Proof. We show that the language $L(S)$ is a non-context-free language using the pumping lemma of context-free graph languages. We assume that $L(S)$ is a context-free graph language and we obtain a contradiction. Let p and q be the constants that satisfy the conditions set by the pumping lemma and are guaranteed to exist. Select the graph $G = \leftarrow^{p+q} \oplus \rightarrow^{p+q} \oplus \rightleftarrows^{p+q}$. Clearly G is a member of $L(S)$ and its number of edges is at least p. The pumping lemma states that G can be pumped by gluing G of a sequence of three subgraphs $FIRST$, $LINK$, and $LAST$, where $|E_{LINK}| + |E_{LAST}| \leq q$, but we consider four cases to show that this result is impossible.

1. The graph LINK does not have left edges. In this case, when we try to pump down the graph G, $FIRST$ and $LAST$ contain more left edges than right edges or loops. Therefore it is not a member of $L(S)$, and a contradiction occurs.
2. The graph LINK does not have right edges. In this case, $FIRST$ and $LAST$ contain more right edges than left edges or loops. Therefore it is not a member of $L(S)$, and a contradiction occurs.
3. The graph LINK does not have loops. In this case, $FIRST$ and $LAST$ contain more loops than right or left edges. Therefore it is not a member of $L(S)$, and a contradiction occurs.
4. When LINK contains the tree types (right edges, left edges, and loops), when we try to pump up the graph G, gluing $FIRST$, $LINK$, $LINK$, and $LAST$, the resulting graph cannot contain the edges in the correct order. Hence it cannot be a member of $L(S)$, and a contradiction occurs.

□

5 The Power of String-Regulated Graph Grammars

In this section we study the expression power of string-regulated graph grammars, including a comparison with the power of traditional context-free graph grammars. Additionally, we demonstrate a hierarchy of families of string-regulated graph grammars according to their expression power.

5.1 On the Size of Graphs

We first start showing some facts about the number of edges of graphs generated by string-regulated graph grammars.

For $k \geq 2$, k sets $M_1, M_2, \ldots, M_k$ of natural numbers, $j \geq 0$, and a set M of natural numbers, we set

$$M_1 + M_2 + \cdots + M_k = \{a_1 + a_2 + \cdots + a_k \mid a_i \in N_i\}$$

and

$$j \cdot M = \underbrace{M + M + \cdots + M}_{j \text{ times}}.$$

Let U be a set of graphs. We set

$$a(U) = \{\#(E) \mid (E, V, s, t) \in U\}.$$

Lemma 1. Let $SRGG = (C, P, W)$ be a simple string-regulated graph grammar. For any $X \in C$, let

$$a(X) = \{\#(E') - 1 \mid (X, (E', V', s', t'), (d_1, d_2), m) \in P\}.$$

Then

$$a(L(SRGG)) = \bigcup_{w \in W} \{1 + \sum_{X \in C} \#_X(w) \cdot a(X)\}.$$

Proof. Obviously, because the grammar is simple, any occurrence of $X \in C$ in some word $w \in W$ leads to one application of a rule $(X, D, (d_1, d_2), m) \in P$. Moreover, any application of the rule $(X, D, (d_1, d_2), m)$ with $D = (E', V', s', t')$ adds $\#(E') - 1$ new edges to the graph (one edge replaces the edge which is removed). Hence, if (E_2, V_2, s_2, t_2) is obtained from (E_1, V_1, s_1, t_1) by application of $(X, D, (d_1, d_2), m)$ with $D = (E', V', s', t')$, then $\#(E_2) = \#(E_1) + \#(E') - 1$. Hence $\#(E_2) = \#(E_1) + s$ for some $s \in a(X)$. Taking into consideration all occurrences of labels in a word of W, and the fact that we start with one edge, we get the statement. □

Theorem 2. For any simple regular string-regulated graph grammar S, there are natural numbers $n \geq 0$, $m \geq 0$, $r_1 \leq r_2 \leq \cdots \leq r_n \leq s_1 \leq s_2 \leq \cdots \leq s_m$ and $k > s_m - s_1$ such that

$$a(L(S)) = \{r_1, r_2, \ldots, r_n\} \cup \bigcup_{i=1}^{m} \{s_i + kj \mid j \geq 0\}.$$

Proof. Let $S = (C, P, W)$ be a simple regular string-regulated graph grammar. We define the finite substitution $h : C^* \to \{b\}^*$ where $b \notin C$ by $h(X) = \{b^a \mid a \in a(X)\}$ for $X \in C$. Obviously, for $w \in W$, we have $b^n \in h(w)$ if and only if $n \in \sum_{X \in C} \#_X(w) \cdot a(X)$. Thus, by Lemma 1, $b^n \in \{b\}\{h(w) \mid w \in W\}$ if and only if $n \in a(L(S))$. Because W is regular and the family of regular languages is closed under finite substitutions and concatenation, $\{b\}\{h(w) \mid w \in W\}$ is a regular language over the unary alphabet $\{b\}$. It is a well-known fact that, for a regular language L over a unary alphabet $\{b\}$, there exist natural numbers $n \geq 0$, $m \geq 0$, $r_1 \leq r_2 \leq \cdots \leq r_n \leq s_1 \leq s_2 \leq \cdots \leq s_m$ and $k > s_m - s_1$ such that $b^n \in L$ if and only if $n = r_i$ for some i, $1 \leq i \leq n$, or $n = s_j + kl$ for some j, $1 \leq j \leq m$, and some $l \geq 0$. Now the statement follows immediately. □

From Theorem 2, we get the following corollary.

Corollary 1. For any graph language $L \in srSR$, there are constants k_1 and k_2 (depending on L) such that, for any graph $Z = (E, V, s, t) \in L$ with $\#(E) \geq k_1$, there is a graph $Z' = (E', V', s', t') \in L$ such that $|\#(E) - \#(E')| \leq k_2$. □

5.2 The Hierarchy of String-Regulated Graph Grammars

We start with a comparison of the set of languages generated by context-free graph grammars, noted as GG, and those generated by string-regulated graph grammars, where we have already partial results in the Propositions 1 and 2.

Theorem 3. $GG \subset saSR$.

Proof. For a context-free graph grammar $S = (N, T, P, Z)$, define the string-regulated graph grammar $S' = (C, P', W)$ as follows. $C = N \cup \{\eta\}$, being η a symbol not included in N. For each production $p \in P$, define an equivalent production $p' \in P'$ with the same edge label symbol, an unlabelled graph obtained by unlabelling the graph defined in p, and same ordered node pair than p; the mapping of production p' is defined as an arbitrary bijection between the edges of its unlabelled graph and the set $\{1, \ldots, n\}$, being n the number of edges of the graph. P' also includes an additional production p_η, the neutral production, with edge label symbol η, a directed handle as the graph, and a trivial complete mapping. Clearly, the application of the neutral production to a graph results in the same graph, except for its labelling. Finally, W is defined such that it includes a word for each possible derivation of the S grammar in the following way. Let define a sequence of direct derivations

$$G_1 \underset{p_1,e_1}{\Longrightarrow} G_2 \underset{p_2,e_2}{\Longrightarrow} \ldots \underset{p_{k-2},e_{k-2}}{\Longrightarrow} G_{k-1} \underset{p_{k-1},e_{k-1}}{\Longrightarrow} G_k$$

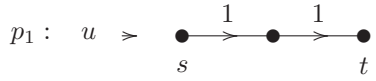
of the graph grammar S. In order to define the corresponding word included in W for such derivation we label with strings the graphs of the sequence. The graph G_k has all its edges labelled with ε. For each graph G_i, $i < k$, the labels of its edges will be the same than in G_{i+1}, except for the replaced edge, e_{i-1}, that will have the string $\omega = cu_1\eta^{s-|u_1|}u_2\eta^{s-|u_2|}\ldots u_m\eta^{s-|u_m|}$, being c the replaced edge label symbol, $u_1, \ldots, u_m$ the labels of the corresponding edges in the graph G_{i+1} according to the mapping of the production p' in S', and $s = \{max|u_j|, 1 \leq j \leq m\}$. The graph G_1 is a handle graph, and the label of its only edge is the word included in W for that derivation.

The strictness of the inclusion follows by Example 2 and Proposition 1. □

We now compare the different types of string-regulated graph grammars with respect to their power.

Lemma 4. $srSR \subset grSR$

Proof. The inclusion holds by definition. The strictness of the inclusion can be proved showing that the graph language $L = \{\rightarrow^{2^n} \mid n \geq 0\}$ can be generated by a general regular string-regulated graph grammar, but that it cannot be generated by a simple regular string-regulated graph grammar. Let $S = (\{u\}, P, u^*)$ be a general regular string-regulated graph grammar, being P:



Clearly, the grammar S produces the language L.

However, $L \notin srSR$ by Corollary 1 since there the difference $2^n - 2^{n-1}$, $n \geq 1$, can be arbitrarily large. □

Lemma 5. *$saSR = gaSR$*

Proof. *$saSR \subseteq gaSR$.* The inclusion holds by definition.

$gaSR \subseteq saSR$. Let $S = (C, P, W)$ be a general arbitrary string-regulated graph grammar. Then we construct the simple arbitrary string-regulated graph grammar $S' = (C, P', W')$ as follows. For each production $p \in P$, define an equivalent production $p' \in P'$ with same edge label symbol, unlabelled graph, and ordered node pair than p; the mapping of production p' is defined as an arbitrary bijection between the edges of its unlabelled graph, being n the number of edges, and the set $\{1, \ldots, n\}$. P' also includes an additional production p_η, the neutral production, with edge label symbol η, a directed handle as graph, and a trivial complete mapping. Finally, W is defined such that it includes a word for each possible derivation of S in the following way: let define a sequence of direct derivations

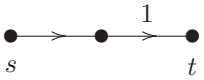
$$G_1 \underset{p_1,e_1}{\Longrightarrow} G_2 \underset{p_2,e_2}{\Longrightarrow} \ldots \underset{p_{k-2},e_{k-2}}{\Longrightarrow} G_{k-1} \underset{p_{k-1},e_{k-1}}{\Longrightarrow} G_k$$

of the graph grammar S. In order to define the corresponding word included in W for such derivation we relabel the graphs of the sequence, starting with G_k and ending with G_1. The graph G_k remains with all its edges labelled with ε. For each graph G_i, $i < k$, the labels of its edges will be the same than in G_{i+1}, except for the replaced edge, e_{i-1}, that will have the string $\omega = cu_1\eta^{s-|u_1|}u_2\eta^{s-|u_2|}\ldots u_m\eta^{s-|u_m|}$, being c the replaced edge label symbol, $u_1, \ldots, u_m$ the labels of the corresponding edges in the graph G_{i+1} according to the mapping of the production p' in S', and $s = \{max|u_j|, 1 \leq j \leq m\}$. The graph G_1 is a handle graph, and the label of its only edge is the word included in W for that derivation. □

Lemma 6. *$grSR \subset saSR$*

Proof. Obviously, $grSR \subseteq gaSR$ follows by definition. Now the inclusion $grSR \subseteq saSR$ follows by Lemma 5.

The strictness of the inclusion can be proved showing that the graph language $L = \{\rightarrow^{n^n}\}$ can be generated by a simple arbitrary string-regulated graph grammar, but it cannot be generated by a general regular string-regulated graph grammar. Let $S = (\{u\}, P, u^{n^n})$ be a simple arbitrary string-regulated graph grammar, being P:

$p_1:$ $u \succ$ 

Clearly, the grammar S produces the language L. Now we show that the language L cannot be generated by a general regular string-regulated graph grammar. We assume that $S' = (C, P', W)$ is a general regular string-regulated graph grammar that generates the language L and we obtain a contradiction. Let define a string $\omega \in W$ at least of length k. Since W is a regular language, according to the pumping lemma of regular languages, there exists a constant k such that the string ω, whose lenght is at least k, can be divided into three substrings, $\omega = xyz$, satisfying that $xy^iz \in W$. Since S' produces the language L, we assume that $xyz^{\S} \underset{P}{\overset{*}{\Longrightarrow}} \rightarrow^{m^m}$. Because $xy^2z \in W$, $|xy| \leq k$, and a sequence of productions starting with an edge labelled with a word of length at most k can generate at most c^k edges, being c a constant, we get that at most c^k additional edges can be generated for each edge. Therefore $xy^2z^{\S} \underset{P}{\overset{*}{\Longrightarrow}} \rightarrow^{m^m \cdot c^k}$, but for an m large enough, $m^m \cdot c^k < (m+1)^{m+1}$, hence $\rightarrow^{m^m \cdot c^k} \notin L$, and a contradiction occurs. □

Lemma 7. *$gaSR \subset AL$.*

Proof. The inclusion follows by definition.

We show that the recursively enumerable graph language $L = \{K_i | i > 1\}$ cannot be generated by any string-regulated graph grammar. We assume that there exists a grammar S that generates the graph language L, and we obtain a contradiction. Let define a sequence of direct derivations

$$\omega \underset{p_1,e_1}{\Longrightarrow} G_1 \underset{p_2,e_2}{\Longrightarrow} \ldots \underset{p_m,e_m}{\Longrightarrow} K_i$$

for some $i > 2$. Now, one of such direct derivations $\underset{p_j,e_j}{\Longrightarrow}$ must generate the i-th node of the graph. Moreover, there cannot exist a production in the grammar that generates an edge between two non-connected nodes because the grammar always replaces a single edge. So, the production p_j that generates the i-th node of the graph must replace an edge with the K_i graph. Clearly, in order to obtain a derivation of the K_{i+1} graph, we need a production that replaces the edge with the K_{i+1} graph. Though, we need an infinite set of productions to generate the language L, but a string-regulated graph grammar must have a finite set of productions, and a contradiction occurs. □

Summarizing our results we get the following statement.

Theorem 8. $srSR \subset grSR \subset saSR = gaSR \subset AL$. □

6 Conclusions and discussion

We proposed here a novel graph rewriting formalism inspired by the development of biological organisms. More precisely, the formalism is based on an edge-replacement graph grammar, where the edges of a graph represent the cells in an organism. The model abstracts three key aspects of biological development: cell division, genetic regulation, and expression state inheritance by new cells. First, cell divisions are represented by the rewriting of edges and a developmental process is abstracted as a derivation in the grammar. The types of possible cell divisions are formalized by the set of rules in the grammar. Second, every edge is labelled with a string, abstracting the genes to be expressed in the cell. The head symbol of a labelling string determines the rule to apply to the edge. In addition, the grammar defines a set of strings to label the initial handle, the axiom or zygote. Third, the string is inherited by the new edges, either by copying it literally (hence, new edges will express the same genes, i.e., they share the same fate) or by distributing it unevenly, equivalent to the biological silencing of genes by DNA methylation (hence new edges will express different genes, i.e., their fate will diverge). In this way, the transmission of regulatory information through the derivation process is the key difference of the proposed model with respect to the traditional context-free graph grammars, where the edges are labelled with just a nonterminal symbol that only determines the next rule to be applied. Consequently, the proposed formalism of string-regulated graph grammars models biological development in a more realistic way than the traditional context-free graph grammars.

Furthermore, string regulation has an impact in the grammar's generative power; in particular, it determines the next rule to be applied to the edge, as well as the rules to be applied to the consequent new edges of the lineage. We have demonstrated in this work that the formalism of string-regulated graph grammars can generate more powerful languages that the classical context-free graph grammars. This result suggests that the transmission of regulatory information, over the lineages of cells, during cell division in the development of an organism has indeed a key role in the emergence of the impressive diversity of organisms found on earth. In the same way that the integration of a regulatory string in a graph grammar allows the generation of a more diverse set of graph topologies, the mechanism by which cells transmit regulatory information to their cell descendants may allow the generation of a more diverse spectrum of morphologies.

Additionally, different types of string-regulated graph grammars have being studied, according to the type of mapping function, injective or general, and set of initial string labels, regular or arbitrary. It has been demonstrated that the generative power of string-regulated graph grammars whose set of initial string labels is restricted to a regular set depends on the type of mapping function allowed in their rules. More precisely, regular string-regulated graph grammars with only rules with injective mappings, which restrain the copy of the same parts of the regulatory string over several cell descendants, have less generative power than grammars with no restrictions in their rules' mappings. Consequently, we conclude that the copy and expression of the same genes among different cells during development has a significant role into the generation of biological diversity. On the other hand, arbitrary string-regulated graph grammars have a larger generative power than regular string-regulated graph grammars, independently on the type of rule mappings. However, the relationship in power of both the simple regular and general regular string-regulated graph grammars with the traditional context-free graph grammars is still open for research. An investigation towards this goal may characterize better the role of the different types of regulation.

Finally, it has been demonstrated that, even if the regulation by a string increases the generative power beyond the power of context-free graph grammars, string-regulated graph grammars cannot generate all recursively enumerable graph languages, such as the language containing all complete graphs. As a consequence, it is worth studying other formalisms to regulate graph grammars that could generate more powerful languages or that represent closer models of biological genetic regulation. For example, the model of graph-grammars regulated by Boolean networks presented in [21] is a good candidate for a formal study.

Acknowledgments

The authors are greatful to the anonymous reviewers for valuable comments and suggestions. D.L. was supported in part by a research stay fellowship at Otto-von-Guericke-Universität Magdeburg from the Spanish Ministerio de Educación.

References

[1] H. Abelson and A. diSessa. *Turtle Geometry: The Computer as a Medium for Exploring Mathematics (Artificial Intelligence)*. The MIT Press, 1986.

[2] P. Bateson. Where does our behaviour come from? *Journal of Biosciences*, 26(5):561–570–570, 2001.

[3] E. Borenstein and D. C. Krakauer. An end to endless forms: Epistasis, phenotype distribution bias, and nonuniform evolution. *PLoS Computational Biology*, 4(10):e1000202+, 2008.

[4] V. Brendel and H. G. Busse. Genome structure described by formal languages. *Nucl. Acids Res.*, 12(5):2561–2568, 1984.

[5] S. Carroll, J. Grenier, and S. Weatherbee. *From DNA to Diversity: Molecular Genetics and the Evolution of Animal Design*. Wiley-Blackwell, 2nd edition, 2004.

[6] J. Collado-Vides. A transformational-grammar approach to the study of the regulation of gene expression. *Journal of Theoretical Biology*, 136(4):403–425, 1989.

[7] B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, 46(2):218–270, 1993.

[8] C. Darwin. *The Origin of Species by Means of Natural Selection or The Preservation of Favoured Races in the Struggle for Life*. John Murray, Albemarle Street, London, 1859.

[9] J. Dassow. Grammars with regulated rewriting. In Carlos Martín-Vide, Victor Mitrana, and Gheorghe Paun, editors, *Formal Languages and Applications. Studies in Fuzziness and Softcomputing*, volume 148, pages 249–273. Springer, 2004.

[10] J. Dassow, V. Mitrana, and A. Salomaa. Operations and language generating devices suggested by the genome evolution. *Theoretical Computer Science*, 270(1-2):701–738, 2002.

[11] F. Drewes, B. Hoffmann, D. Janssens, and M. Minas. Adaptive star grammars and their languages. *Theoretical Computer Science*, 411(34-36):3090–3109, 2010.

[12] F. Drewes, H. J. Kreowski, and A. Habel. Hyperedge replacement graph grammars. In *Handbook of Grammars and Computing by Graph Transformation*, pages 95–162. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1997.

[13] H. Ehrig and H. J Kreowski. Parallel graph grammars. In A. Lindenmayer and G. Rozenberg, editors, *Automata, Languages, Development*, pages 425–447. Amsterdam: North Holland, 1976.

[14] D. Gernert. Graph grammars as an analytical tool in physics and biology. *Biosystems*, 43(3):179–187, 1997.

[15] L. Grate, M. Herbster, R. Hughey, D. Haussler, S. I. Mian, and H. Noller. Rna modeling using gibbs sampling and stochastic context free grammars. In *2nd Internat. Conf. on Intelligent Systems for Molecular Biology, ISMB-94*, volume 2, pages 138–146, 1994.

[16] J. Huxley. *Evolution: The Modern Synthesis*. Allen & Unwin, London, 1942.

[17] H. Kreowski. A pumping lemma for context-free graph languages. In *Lecture Notes in Computer Science*, volume 73, pages 270–283. 1979.

[18] Julie A. Law and Steven E. Jacobsen. Establishing, maintaining and modifying DNA methylation patterns in plants and animals. *Nature Reviews Genetics*, 11(3):204–220, February 2010.

[19] A. Lindenmayer. Mathematical models for cellular interaction in development: Parts i and ii. *Journal of Theoretical Biology*, 18:280–315, 1968.

[20] D. Lobo and F. J. Vico. Evolutionary development of tensegrity structures. *Biosystems*, 101(3):167–176, 2010.

[21] D. Lobo and F. J. Vico. Evolution of form and function in a model of differentiated multicellular organisms with gene regulatory networks. *Biosystems*, 102(2-3):112–123, 2010.

[22] Gary Marcus. *The Birth Of The Mind: How A Tiny Number Of Genes Creates The Complexities Of Human Thought*. Basic Books, 2003.

[23] E. Mayr. *What Evolution Is*. Basic Books, 2001.

[24] G. Mendel. Versuche über pflanzen-hybriden. *Verhandlungen des naturforschenden Vereines in Brünn*, 42:3–47, 1866.

[25] A. P. Moczek. On the origins of novelty in development and evolution. *BioEssays*, 30(5):432–447, 2008.

[26] G. B. Müller. Evo-devo: extending the evolutionary synthesis. *Nature Reviews Genetics*, 8(12):943–949, 2007.

[27] M. Nagl. Graph-grammars and their application to computer science and biology. In *Graph-Grammars and Their Application to Computer Science and Biology*, chapter A tutorial and bibliographical survey on graph grammars, pages 70–126. 1979.

[28] P. Prusinkiewicz, Y. Erasmus, B. Lane, L. D. Harder, and E. Coen. Evolution and development of inflorescence architectures. *Science*, 316(5830):1452–1456, 2007.

[29] G. Rozenberg. *Handbook of Grammars and Computing by Graph Transformation*. World Scientific Publishing Company, 1997.

[30] Y. Sakakibara, M. Brown, R. Hughey, I. S. Mian, K. Sjölander, R. C. Underwood, and D. Haussler. Stochastic context-free grammars for trna modeling. *Nucl. Acids Res.*, 22(23):5112–5120, 1994.

[31] D. Searls. String variable grammar: A logic grammar formalism for the biological language of dna. *The Journal of Logic Programming*, 24(1-2):73–102, 1995.

[32] G. Taentzer. *Parallel and Distributed Graph Transformation: Formal Description and Application to Communication-Based Systems*. PhD thesis, Technische Universität Berlin, 1996.