On the number of different dynamics in Boolean networks with deterministic update schedules

J. Aracena [a] J. Demongeot [c] E. Fanchon [d] M. Montalva [b,*]

[a] *CI^2MA and Departamento de Ingeniería Matemática, Universidad de Concepción, Av. Esteban Iturra s/n, Casilla 160-C, Concepción, Chile.*

[b] *Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Peñalolén, Santiago, Chile.*

[c] *AGIM FRE3405 - CNRS/UJF/EPHE, Université Joseph Fourier, Domaine de la Merci, 38710 La Tronche, France.*

[d] *TIMC-IMAG, Université Joseph Fourier CNRS-UMR 5525, Domaine de la Merci, 38710 La Tronche, France.*

Abstract

Deterministic Boolean networks are a type of discrete dynamical systems widely used in the modeling of genetic networks. The dynamics of such systems is characterized by the local activation functions and the update schedule, i.e., the order in which the nodes are updated. In this paper, we address the problem of knowing the different dynamics of a Boolean network when the update schedule is changed. We begin proving that problem about the existence of different dynamics is NP-complete. However, we show that certain structural properties of the interaction digraph are sufficient for guarantee distinct dynamics of a network. In (Aracena et al., 2009) the authors define equivalence classes which have the property that all the update schedules of a given class yield the same dynamics. In order to determine the dynamics associated to a network, we develop an algorithm to efficiently enumerate the above equivalence classes by selecting a representative update schedule for each class with a minimum number of blocks. Finally, we run it over the well known Arabidopsis Thaliana network to know the full spectrum of its different dynamics.

Key words: Boolean network, update schedule, update digraph, dynamics.

* Corresponding author.
Email addresses: jaracena@ing-mat.udec.cl (J. Aracena), Jacques.Demongeot@imag.fr (J. Demongeot), Eric.Fanchon@imag.fr (E. Fanchon), marco.montalva@uai.cl (M. Montalva).

1 Introduction

Deterministic Boolean networks have been introduced in Systems Biology by S. Kauffman (1969, 1993) to model the dynamics of genetic networks. In the original scheme all the nodes are updated at each time step, in parallel (this scheme is also called synchronous updating). This kind of updating has given rise to an enormous mathematical literature.

A more general scheme is to consider that the set of network nodes is partitioned into blocks and that the nodes in a block are updated simultaneously, the blocks being considered in a given sequence. This generalizes the previous case because the parallel case corresponds to a single block. It also generalizes the so-called sequential Boolean systems where every node is updated in a defined sequence at every time step.

On different grounds it was realized that the purely synchronous (parallel) updating was not satisfactory for the modeling of gene networks and several extensions were proposed in the literature. Gershenson (2002) defined the so-called Deterministic Generalized Asynchronous RBNs (DGARBNs), for which a node i is updated if it satisfies an updating condition (depending on two parameters P_i and Q_i associated to the node). When several nodes satisfy their condition simultaneously they are updated synchronously. Gershenson calls this kind of Boolean network semi-synchronous. This scheme bears some ressemblance with our block-sequential updating because the nodes which share the same values of P_i and Q_i can be said to belong to the same block (they are updated at the same time steps). The difference is that nodes i with short period (the parameter P_i) are updated more often than nodes with longer periods whereas in the block-sequential scheme all the blocks are updated exactly once, following a predefined order, at each time step. Another important difference is a difference of perspective: S. Kauffman initiated an approach in which ensembles of Boolean networks are considered, rather than a network with a given architecture and given activation functions representing a specific biological system. In the ensemble approach promoted by S. Kauffman the focus is on representative properties of the statistical ensemble of networks. By contrast our focus here will be on the modeling of specific biological phenomena.

In a different spirit, Thomas (1973); Thomas et al. (1995); Thomas and Kaufman (2001) developped a different view. The Boolean model is viewed as an abstraction of a system of piecewise-linear differential equations with diagonal matrix, and is consequently non-deterministic (in the sense that a given state may have several successors). Thomas also introduced time delays and even considered the possibility that these delays may be stochastic, but the occurence of non-determinism is intrinsically linked to the fact that the Boolean

model is a discrete abstraction of a dynamical system : the continuous state space is partitioned into rectangular domains. The loss of information induced by the abstraction entails an uncertainty in the successor of a state and the formalism of Thomas is designed to include all the admissible transitions from a state. A transition graph computed with these rules includes all the possible dynamics compatible with a given network architecture (but conversely an arbitrary path from the transition graph does not necessarily represent a valid behavior).

The formalism of Thomas is at first sight quite different from the Boolean networks with deterministic updating rules. It was nevertheless recognized that deterministic synchronous updating can often be recovered as a simplification of the Thomas dynamics (Fauré et al., 2006).

In the present paper we will call Boolean network an entity made of (i) a directed graph (called interaction graph, the nodes of which represent the genes); (ii) an activation function for each node, which specifies the next state of the node given the state of the predecessor nodes; (iii) an update schedule s which specifies the order in which the nodes are updated. In other word the function s defines the partition into blocks of the set of nodes.

In this framework, Aracena et al. (2009) proved that two Boolean networks differing only by their update schedules may have exactly the same dynamics. They introduced a new kind of signed digraph, called update digraph, which defines for each arc whether the tail is updated before or after the head of the arc. The information carried by an update digraph is weaker than the one contained in the complete definition of a Boolean network, but the theorem proved in (Aracena et al., 2009) shows that it is sufficient to define completely the dynamics. Equivalence classes of update schedules have been defined on these grounds and the robustness of the dynamics with respect to perturbation of the schedules has been studied. In Aracena et al. (2011) the combinatorial and algorithmical aspects of update digraphs were studied. In particular bounds on the number of equivalence classes were obtained.

Our perspective in this paper is the modeling of specific biological phenomena. The problem is thus one of inference: how to infer a Boolean network whose behavior matches the observed behaviors? We will focus here more precisely on the update schedules and their equivalence classes. We showed in (Aracena et al., 2012) that for complete digraphs there is exactly 1 update schedule per class. The architecture of networks encountered in biology is generally rather sparse and in that case a given class may contain many update schedules (thus associated to the same dynamics). This means that the information contained in the dynamics of a system pertains only to the equivalence classes. In other words such observations do not allow to distinguish two update schedules belonging to the same class. Consequently in the context of building models

from data it is very important to characterize the classes in order to optimize the inference process. In Aracena et al. (2012) we gave exact formulae for the number of equivalence classes for a large class of digraphs. In the present paper we focus on the enumeration of the equivalence classes of a given digraph.

Section 2 provides the necessary definitions for the sequel. All the schedules belonging to a given class generate the same dynamics, but conversely two different classes are not necessarily associated to different dynamics. For some network architectures it is indeed possible that two different equivalence classes generate the same dynamics.

Section 3 gives somes results related to this issue; first, we point out the difficulty of to know the different dynamics for a given network, problem that, under our knowledge, has not yet been study in depth. Especifically, we prove that the problem of knowing whether there exist two different udpdate schedules for a given network such that its associated dynamics are different is, in fact, NP-Complete. We prove a propertie of how the structural properties of the digraph can assure different dynamics as well as we ilustrate an extreme case example of a particular family of digraphs where the definition of the activation function of each node can imply that all its dynamics are identical or can imply that all them are different. We explain how the analysis of update schedules yielding the same dynamics give us bounds for the number of different dynamics in a given network. At the end of section, the reader arrive to better understand the computational savings that he can achieve if is to develop an efficient enumeration algorithm for the equivalence classes compatible with a given network architecture (digraph).

In section 4 we propose such efficient algorithm whose principal subroutine we call it `DigraphUD`. This one, rougly speaking, start with the set of vertices $V(G)$ associated to a given network and through a process of sucessive partitions, to check if the current update schedule compatible with such partition has been saved in the output through a polynomial test. A priori, this process seems to be a brute force enumerating algorithm but a set of lemmas and propositions that we proved, guarantee that such algorithm to do only the necessary partitions and the update schedule saved in the output has the property of to have a minimal number of blocks, i.e. it is the most parallel possible. Finally, we conclude this section with some special cases where the algorithm is more efficient: in digraphs with a big number of equivalence classes and in digraphs with representative schedules having a small number of blocks.

Finally in section 5, our theoretical and computational tools of the previous sections are applied to the study of the flower morphogenesis of the plant Arabidopsis Thaliana (Mendoza and Alvarez-Buylla, 1998). Especifically, we work with the reduced model defined by Demongeot et al. (2010) which has two non-trivial connected components of 3 and 4 genes. Our results allow us

to consider only one update schedule for each equivalence class, and we show that this entails a reduction of computational work that allow us to have the full spectrum of all the different dynamics associated with each component.

2 Definitions

We begin this section giving some basic definitions and introducing the necessary notations. Besides, we remember some known results that allow us to develop the following sections.

In the sequel, for any integers a and b with $a \leq b$, we will denote $[\![a, b]\!] = \{i \in \mathbb{Z} : \ a \leq i \leq b\}$.

A **digraph** is an ordered pair of sets $G = (V, A)$ where $V = \{1, \ldots, n\}$ is a set of elements called **vertices** (or **nodes**) and A is a set of ordered pairs (called **arcs**) of vertices of V. The vertex set of G is referred to as $V(G)$, its arc set as $A(G)$. For a vertice $i \in V$ we denote $V^-(i) = \{j \in V : \ (j, i) \in A\}$.

A **subdigraph** of G is a digraph $G' = (V', A')$ where $V' \subseteq V$ and $A' \subseteq (V' \times V') \cap A$. We write $G' \subseteq G$.

A **path** from a vertex v_1 to a vertex v_m in a digraph G is a sequence of vertices $v_1, v_2, \ldots, v_m$ of $V(G)$ such that $(v_k, v_{k+1}) \in A(G)$ for all $k = 1, \ldots, m-1$. A **cycle** is a path $v_1, \ldots, v_m$ such that $v_i \neq v_j$ for all $i \neq j$ with $i, j \in \{2, \ldots, m-1\}$ and $v_1 = v_m$.

More terminology about digraphs can be found in (West, 1996).

A **(deterministic) update schedule** over the vertices of G with $|V(G)| = n$, is a function $s : [\![1, n]\!] \to [\![1, n]\!]$ such that $s(V(G)) = [\![1, m]\!]$ for some $m \leq n$. A **partial update schedule** is an update schedule over the vertices of some $G' \subseteq G$. A **block** of s is the set $B_i = \{v \in V(G') : \ s(v) = i\}$, $1 \leq i \leq m$. The number of blocks of s is denoted by $nb(s) \equiv m$. Frequently, s will be denoted by $s = (j \in B_1)(j \in B_2) \cdots (j \in B_{nb(s)})$ or more compactly $s = (B_i)_{i=1}^{nb(s)}$.

Let $G' = G$. If $nb(s) = 1$, then s is said to be a **parallel** update schedule. In this case, we will write $s = s_p$. If s is a permutation over the set $[\![1, n]\!]$, i.e. $nb(s) = n$, s is said to be a **sequential** update schedule. In all other cases, i.e. when $2 \leq nb(s) \leq n-1$, s is said to be a **block sequential** update schedule. As was mentionned in Demongeot et al. (2008), the number T_n of deterministic update schedules associated to a digraph of n vertices is equal to the number of ordered partitions of a set of size n, that is:

$$T_n = \sum_{k=0}^{n-1} \binom{n}{k} T_k \tag{1}$$

where $T_0 = T_1 \equiv 1$.

2.1 Update digraph

Let $G = (V, A)$ be a digraph and s an update schedule, we define the label function $lab_s : A \to \{\ominus, \oplus\}$ in the following way :

$$\forall (j,i) \in A, lab_s(j,i) = \begin{cases} \oplus & \text{if } s(j) \geq s(i) \\ \ominus & \text{if } s(j) < s(i). \end{cases}$$

An arc $a \in A$ such that $lab_s(a) = \oplus$ is called a **positive arc** and an arc $a \in A$ such that $lab_s(a) = \ominus$ is called a **negative arc**. Labeling every arc a of A by $lab_s(a)$, we obtain a labeled digraph (G, lab_s) named **update digraph**. We denote

$$U(G) = \{lab : A(G) \to \{\ominus, \oplus\} |\ (G, lab) \text{ is an update digraph}\} \tag{2}$$

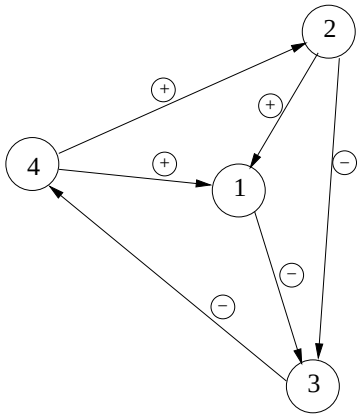

Fig. 1. A digraph $G = (V, A)$ labeled by the function lab_s where $\forall i \in V = \{1, \ldots, 4\}$, $s(i) = i$.

2.2 Boolean network

A **Boolean network** $N = (G, F, s)$ is defined by:

- A digraph $G = (V, A)$ with n vertices, named **interaction graph**.

- A **global activation function** $F = (f_1, \ldots, f_n) : \{0,1\}^n \to \{0,1\}^n$, where the component functions $f_i : \{0,1\}^n \to \{0,1\}$ are called **local activation functions** and satisfy the following property:
  $(j,i) \in A \iff \exists x \in \{0,1\}^n,\ f_i(x) \neq f_i(\overline{x}^j)$, where for all $x \in \{0,1\}^n$, $\overline{x}^j \in \{0,1\}^n$ is defined by $\overline{x}^j_j = \overline{x}_j = 1 - x_j$ and $\overline{x}^j_k = x_k$ for all $k \neq j$.
- An update schedule $s : V \to [\![1, n]\!]$of the vertices of G.

The iteration of the discrete network with an update schedule s is given by:

$$x_i^{r+1} = f_i(x_1^{l_1}, \ldots, x_j^{l_j}, \ldots, x_n^{l_n}),$$

where $l_j = r$ if $s(i) \leq s(j)$ and $l_j = r + 1$ if $s(i) > s(j)$. The exponent r represents the time step.

This is equivalent to apply a function $F^s : \{0,1\}^n \to \{0,1\}^n$ in a parallel way, with $F^s(x) = (f_1^s(x), \ldots, f_n^s(x))$ defined by:

$$f_i^s(x) = f_i(g_{i,1}^s(x), \ldots, g_{i,n}^s(x)),$$

where the function $g_{i,j}^s$ is defined by $g_{i,j}^s(x) = x_j$ if $s(i) \leq s(j)$ and $g_{i,j}^s(x) = f_j^s(x)$ if $s(i) > s(j)$. Thus, the function F^s corresponds to the **dynamical behavior** of the network N. We will say that two networks $N_1 = (G, F, s_1)$ and $N_2 = (G, F, s_2)$ have the same dynamics if $F^{s_1} = F^{s_2}$.

Since $\{0,1\}^n$ is a finite set, we have two limit behaviors for the iteration of a network:

- **Fixed Point**. We define a fixed point as $x \in \{0,1\}^n$ such that $F^s(x) = x$.

- **Limit Cycle**. We define a limit cycle of length $p > 1$ as the sequence $x^0, \ldots, x^{p-1}$ such that $x^j \in \{0,1\}^n$, x^j are pairwise distinct and $F^s(x^j) = x^{j+1}$, for all $j = 0, \ldots, p-2$ and $F^s(x^{p-1}) = x^0$.

Fixed points and limit cycles are called **attractors** of the network.

3 Different dynamics in Boolean networks

For a given Boolean network $N = (G, F, s)$, determining the existence of an update schedule $s' \neq s$ such that the network $N' = (G, F, s')$ has a different dynamics to that of N is contrarily to intuition a difficult problem as stated in the following theorem.

Theorem 1 *Let G be an interaction graph and F a global activation function. The problem of knowing whether there exist udpdate schedules $s' \neq s$ such that*

$F^s \neq F^{s'}$ is NP-complete.

PROOF.

It is easy to see that to check that $F^s \neq F^{s'}$ for a given udpdate schedule $s' \neq s$. For this, it is sufficient to verify that $f_i^s(x) \neq f_i^{s'}(x)$ for some $x \in \{0,1\}^n, i \in \{1,\ldots,n\}$ where $|V(G)| = n$.

We show a polynomial reduction from SAT problem. Let ϕ a conjunctive normal form (cnf) formula with variables $x_1,\ldots,x_n$ in $\{0,1\}$. We construct a Boolean network with interaction graph G, as shown in Fig. 2, with $n+2$ nodes as follows. For each variable x_i there is a node i with local activation function $f_i(x) = \bar{x}_i$. In addition, there are two nodes $n+1$ and $n+2$ with local functions $f_{n+1}(x) = x_{n+1} \wedge \overline{\phi(x_1,\ldots,x_n)}$ and $f_{n+2}(x) = x_{n+1}$ for every $x \in \{0,1\}^{n+2}$. To prove the correctness of the reduction consider first a satisfiable formula ϕ, with $\phi(\alpha) = 1,\ \alpha = (\alpha_1,\ldots,\alpha_n) \in \{0,1\}^n$. Then, for $s = (1,2,\ldots,n)(n+1)(n+2)$ and $s' = (1,2,\ldots,n)(n+2)(n+1)$, $F^s(\alpha,1,0) = (\bar{\alpha},0,0)$ and $F^{s'} = (\alpha,1,0) = (\bar{\alpha},0,1)$, hence $F^s \neq F^{s'}$. On the other hand, if ϕ is not satisfiable, that is for every $x \in \{0,1\}^n,\ \phi(x) = 0$, then $f_{n+1}(x) = x_{n+1}$. Thus, for every update schedule s, $F^s(x,0,*) = (\bar{x},0,0)$ and $F^s(x,1,*) = (\bar{x},1,0)$ for every $x \in \{0,1\}^n$ and where $* \in \{0,1\}$. Therefore, for every update schedules $s \neq s'$, $F^s = F^{s'}$. □

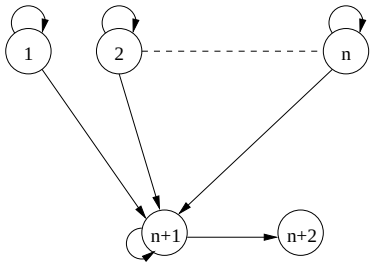

Fig. 2. Interaction graph G for the proof of Theorem 1

However, for some Boolean networks, with interaction graphs having certain structural properties, it is possible to ensure the existence of update schedules yields different dynamical behaviors as shown in the following proposition.

Proposition 2 *Let $N = (G,F,s)$ a Boolean network such that $\exists i \in V(G)$, $V^-(i) = \{j\}$ and $\exists k \in V^-(j)$, $k \neq j$. Then, there exist update schedules s_1, s_2 such that $F^{s_1} \neq F^{s_2}$.*

PROOF. Let $i \in V(G)$, $V^-(i) = \{j\}$. Hence, $f_i(x) = x_j \vee f_i(x) = \overline{x}_j$. In both cases, $f_i(x) \neq f_i(\overline{x}^j),\ \forall x \in \{0,1\}^n$ where $n = |V(G)|$.

Besides, if $\exists\, k \in V^-(j)$ with $k \neq j$, $\exists\, y \in \{0,1\}^n$, $f_j(y) = \overline{y}_j$, since otherwise $\forall x \in \{0,1\}^n$, $f_j(x) = x_j$, which means $V^-(j) = \{j\}$, a contradiction.

Let s_1, s_2 be update schedules such that $s_1(j) \geq s_1(i)$, $s_1(k) \geq s_1(k)$, $\forall k \in V^-(j)$ and $s_2(j) < s_2(i)$. Thus, $f_i^{s_1}(y) = f_i(y)$ and $f_i^{s_2}(y) = f_i(y_1, \ldots, f_j(y), \ldots, y_n) = f_i(\overline{y}^j) \neq f_i(y)$. Therefore, $F^{s_1} \neq F^{s_2}$. □

On the other hand, in (Aracena et al., 2009), the authors in their study of the robustness of the dynamical behavior of a Boolean network with respect to different update schedules, they established the following result:

Theorem 3 *Let $N_1 = (G, F, s_1)$ and $N_2 = (G, F, s_2)$ be two Boolean networks that differ only in their update schedules s_1 and s_2 respectively. If $(G, lab_{s_1}) = (G, lab_{s_2})$, then N_1 and N_2 have the same dynamics.*

Hence, in the same paper and due to this Theorem, the authors naturally grouped the update schedules yielding the same update digraph into equivalence classes defined as follows:

$$[s]_G = \{s' : \ (G, lab_s) = (G, lab_{s'})\} \tag{3}$$

Thus, there is a one to one relation between the elements of $U(G)$ in (2) and each different equivalence class as in (3). Therefore,

$$|U(G)| = |\{[s]_G : \ s \text{ is an update schedule over } V(G)\}|.$$

In other words, Theorem 3 says that $|U(G)|$ gives us the maximum number of different dynamics that can be obtained by iterating a Boolean network with the T_n deterministic update schedules, where $|V(G)| = n$. In this context, some theoretical bounds and exact formulas have been established for $|U(G)|$ in particular families of digraphs such as connected digraphs, complete digraphs (i.e., digraphs $G = (V, A)$ where $A = \{(u, v) : \ u, v \in V \wedge u \neq v\}$), digraphs containing a tournament as a subdigraph, etc. as well as the NP-completness of the update digraph decision problem associated (Aracena et al., 2011, 2012).

For example, in the particular case of complete digraphs, the autors in (Aracena et al., 2011) proved that there are many equivalence classes as update schedules, i.e., eventually a complete digraph could have T_n different dynamics (the maximum value for $|U(G)|$). However, often in the applications, the networks are not complete digraphs, then it is in these cases where the number of equivalence classes is strictly smaller than T_n.

It is also possible that two non equivalent update schedules, i.e. each one of them belonging to a different equivalence class, yield the same dynamical behavior. The following example exhibits families of networks where all pairs of non equivalent update schedules yield either the same or different dynamical behaviors.

Example 1 *Let G be the digraph, with $|V(G)| = n$ as shown in Fig 3. F and $\tilde{F}$ defined by $f_i(x) = \tilde{f}_i(x) = x_n$ $\forall i = 1, \ldots, n-1$ and $f_n(x) = x_n$ and $\tilde{f}_n(x) = \overline{x}_n$ for every $x \in \{0,1\}^n$. Hence, $|U(G)| = 2^{n-1}$ and since $f_n^s(x) = x_n$ and $\tilde{f}_n^s = \bar{x}_n \neq x_n$ for every update schedule s and $x \in \{0,1\}^n$, then $F^s = F^{s'}$ and $\tilde{F}^s \neq \tilde{F}^{s'}$ for every s, s' non equivalent update schedule. Therefore, the dynamics of $N_1 = (G, F, s_1)$ and $N_2 = (G, F, s_2)$ are the same and of $\tilde{N}_1 = (G, \tilde{F}, s_1)$ and $\tilde{N}_2 = (G, \tilde{F}, s_2)$ are different for every update schedules $s_1 \neq s_2$.*

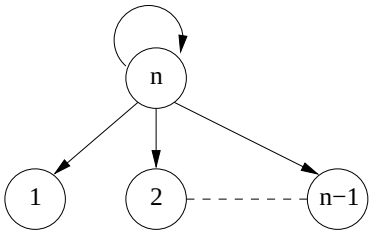

Fig. 3. Interaction graph G for Example 1. Note that $N = (G, F, s)$ gives the same dynamic for each update schedule s while $\tilde{N} = (G, \tilde{F}, s)$ has $|U(G)| = 2^{n-1}$ different dynamics, one for each equivalence class.

Note that from Proposition 2, if G is a cycle, then for any global activation function associated F, all update schedule classes yield different dynamics.

At this point, we can to compare the maximum number of different dynamics with the total number of update schedules for the networks of Example 1, i.e., to compare $|U(G)| = 2^{n-1}$ with T_n respectively. The values are summarized in Table 1.

n	2^{n-1}	T_n	$2^{n-1}/T_n$
1	1	1	1
2	2	3	0.667
3	4	13	0.308
4	8	75	0.107
5	16	541	0.030
6	32	4683	0.007
⋮	⋮	⋮	⋮

Table 1
Number of equivalence classes vs. total number of update schedules for Example 1.

Cases like this, help us to understand more intuitively the computational savings that we can achieve if we want to know, for example, the full spectrum of all the dynamics associated with a given network.

From previous results, there is not an easy way, in the general case, for determing the different dynamics of a given Boolean network when we change only the deterministic update schedule. An approach for this, is first knowing the update schedule equivalence classes and after determining the distinct dynamical behaviors.

4 Enumerating update digraphs

In this section, we exhibit an algorithm for enumerating a representative update schedule, with the smallest number of blocks, for each one of the equivalence classes in order to determine the different dynamics of a Boolean network when we use distinct deterministic update schedules.

Definition 4 *Let $G = (V, A)$ be a digraph and $C, D \subseteq V$. We define the subdigraph of G associated to C and D by $G_{(C,D)} = (C \cup D, A(G) \cap ((C \cup D) \times (C \cup D))$. Also we define $lab_{(C,D)} : A(G_{(C,D)}) \longrightarrow \{\oplus, \ominus\}$ by:*

$$lab_{(C,D)}(u,v) = \begin{cases} \ominus, & u \in C \ \wedge \ v \in D, \\ \oplus, & otherwise \end{cases}$$

Definition 5 *Let G be a digraph, $s = (j \in B_1)(j \in B_2)\cdots(j \in B_{nb(s)})$ a partial update schedule over G and $X \subseteq V(G) \setminus \bigcup_{i=1}^{nb(s)} B_i$. We define the operation $*$ as follows:*

$$s * X = (j \in B_1)(j \in B_2)\cdots(j \in B_{nb(s)})(j \in X).$$

In addition, we define

$$s_e * X = (j \in X)$$

where s_e is an element named **empty update schedule** *with $nb(s_e) = 0$ and $B_0 = \emptyset$.*

The following is an algorithm to determine a representative update schedule for each one of the equivalence classes of a given digraph.

Lemma 6 *Let G be a digraph with $|V(G)| = n$, $s = (B_i)_{i=1}^{k}$ an update schedule*

Algorithm 1. `EqClass`(G)
Input: $G = (V, A)$ a digraph
Output: `UD`, a set of representative update schedules of each equivalence classes associated to G
begin
  `UD` $\leftarrow$ `DigraphUD`$(s_e, \emptyset, V)$;
end

Algorithm 2. `DigraphUD`(s, A, B)
Input: A, B subsets of vertices of a digraph G and s a partial update schedule of a subdigraph of G
Output: `UD`, a set of partial update schedules for G
begin
  `UD`$\leftarrow \emptyset$;
  $U \leftarrow B_{nb(s)}$;
  if $U = A = \emptyset$ **then**
    `UD` $=$ `UD` $\cup \{s_B = (j \in B)\}$;
    forall $A_0 \subset B$ *such that* $A_0 \neq \emptyset$ *with decreasing size* **do**
      $B_0 = B - A_0$;
      `UD` $=$ `UD` $\cup$ `DigraphUD`(s_e, A_0, B_0);
    end
  else
    if `MoveTest`$(U, A) = 0$ **then**
      if `MoveTest`$(A, B) = 0$ **then**
        `UD` $=$ `UD` $\cup \{(s * A) * B\}$;
      end
      if $|B| > 1$ **then**
        forall $A_1 \subset B$ *such that* $A_1 \neq \emptyset$ *with decreasing size* **do**
          $B_1 = B - A_1$;
          `UD` $=$ `UD` $\cup$ `DigraphUD`$(s * A, A_1, B_1)$;
        end
      end
    end
  end
  return(`UD`);
end

for G, $k \in [\![1, n]\!]$, *and* `UD` *the output of the algorithm* `EqClass`(G). *Then,* $s \in$ `UD` $\Leftrightarrow s = s_p$ *or* `MoveTest`$(B_i, B_{i-1}) = 0$, $\forall i \in [\![2, k]\!]$.

PROOF. Let G be a digraph with $|V(G)| = n$, $s = (B_i)_{i=1}^k$ an update schedule for G, $k \in [\![1, n]\!]$, and `UD` the output of the algorithm `EqClass`(G).

Algorithm 3. `MoveTest`(C, D)
Input: C, D subsets of vertices of a digraph G
Output: An index 1 if it is possible to move nodes from D to C without changing the update digraph induced by $C \cup D$, an index 0 in other case

```
begin
    if C = ∅ then
        return(0);
    else
        if ∃H ⊆ D such that (G_(C,D), lab_(C,D)) = (G_(C∪H,D−H), lab_(C∪H,D−H)) then
            return(1);
        else
            return(0);
        end
    end
end
```

$\Rightarrow$) Let $s \in \texttt{UD}$. If $k = nb(s) = 1$, then clearly $s = s_p$ and the statement is true. If $k > 1$, then, by definition of `DigraphUD`, $\texttt{MoveTest}(B_{k-2}, B_{k-1}) = \texttt{MoveTest}(B_{k-1}, B_k) = 0$ (eventually $B_{k-2} = \emptyset$). This also implies that $\texttt{MoveTest}(B_{k-3}, B_{k-2}) = 0$, because in contrary case, s would not be invocated in the recursive process of algorithm. Thus, applying recursively this argument, we have that $\texttt{MoveTest}(B_i, B_{i-1}) = 0$, $\forall i \in [\![2, k]\!]$.

$\Leftarrow$) If $s = s_p$, then the result is direct. If $k \geq 2$ and $\texttt{MoveTest}(B_i, B_{i-1}) = 0$, $\forall i \in [\![2, k]\!]$, then s is invocated in the recursion and it satisfies the conditions $\texttt{MoveTest}(U = B_{k-2}, A = B_{k-1}) = \texttt{MoveTest}(B_{k-1}, B = B_k) = 0$. Therefore, s is added to `UD`. □

Lemma 7 *Let G be a digraph and* `UD` *the output of algorithm* $\texttt{EqClass}(G)$. *Then, $\forall s, s' \in \texttt{UD}$, $s \neq s'$: $[s]_G \neq [s']_G$.* □

PROOF. Let G be a digraph with $|V(G)| = n$ and let `UD` the output of algorithm $\texttt{EqClass}(G)$. Let $s, s' \in \texttt{UD}$, $s \neq s'$ with $s = (B_i)_{i=1}^k$ and $s' = (B'_i)_{i=1}^t$ for some $k, t \in [\![1, n[\![$. By Lemma 6, we have that $s = s_p$ or $\texttt{MoveTest}(B_i, B_{i-1}) = 0$, $\forall i \in [\![2, k]\!]$ and $s' = s_p$ or $\texttt{MoveTest}(B'_i, B'_{i-1}) = 0$, $\forall i \in [\![2, t]\!]$.

Let $i = \min\{j : B_j \neq B'_j\}$. Let us suppose w.l.o.g. that $\exists w \in B_i \setminus B'_i$. Then, $B'_j = B_j$, $\forall j < i$. Hence, $w \in B_p$, for some $p > i$. Since $\texttt{MoveTest}(B_{p-1}, B_p) = 0$, then $\exists y \in B_{p-1}$, $v \in B_p$ such that the arc (y, v) is negative. In addition, there exists a path from w to v in G (eventually $v = w$) because in contrary case, there exists $H = \{w\} \cup \{v \in B_p : \text{ there is a path from } w \text{ to } v\} \subseteq B_p$ such that $(G_{(B_{p-1}, B_p)}, lab_{(B_{p-1}, B_p)}) = (G_{(B_{p-1} \cup H, B_p - H)}, lab_{(B_{p-1} \cup H, B_p - H)})$, there-

fore $\texttt{MoveTest}(B_{p-1}, B_p) = 1$ which is a contradiction. Since the arc (y, v) is negative with respect to s, this necessarily implies that $y \in B'_r$ for some $r < i$, i.e. $B'_r \neq B_r$, a contradiction because $B'_j = B_j$, $\forall j < i$, in particular for $j = r$. □

Lemma 8 *Let G be a digraph with $|V(G)| = n$, $s = (B_i)_{i=1}^k$ an update schedule for G, $k \in [\![1, n]\!]$, and* UD *the output of the algorithm* $\texttt{EqClass}(G)$*. Then, $s \notin \texttt{UD} \Leftrightarrow \exists s' \in \texttt{UD}$, $s' \neq s$, such that $s \in [s']_G$.*

PROOF. Let G be a digraph with $|V(G)| = n$, $s = (B_i)_{i=1}^k$ an update schedule for G, $k \in [\![1, n]\!]$, and UD the output of the algorithm $\texttt{EqClass}(G)$.

$\Rightarrow$) If $s \notin \texttt{UD}$, then by Lemma 6, $\exists r = \min\{i : \texttt{MoveTest}(B_{i-1}, B_i) = 1\}$. Hence, we consider H_r the biggest subset of B_r such that $(G_{(B_{r-1},B_r)}, lab_{(B_{r-1},B_r)}) = (G_{(B_{r-1}\cup H_r, B_r - H_r)}, lab_{(B_{r-1}\cup H_r, B_r - H_r)})$. Thus, there exists update schedule $s_1 = (B_i^1)_{i=1}^k \neq s$ where $B_j^1 = B_j$, $\forall j \in [\![1, k]\!] \backslash \{r-1, r\}$, $B_{r-1}^1 = B_{r-1} \cup H_r$ and $B_r^1 = B_r - H_r$ (eventually $B_r^1 = \emptyset$ and therefore $nb(s_1) < k$) such that $(G, lab_{s_1}) = (G, lab_s)$. Now, considering s_1, if there exists $r = \min\{i : \texttt{MoveTest}(B_{i-1}^1, B_i^1) = 1\}$, then we construct s_2 depending of s_1, i.e., we consider H_r the biggest subset of B_r^1 such that $(G_{(B_{r-1}^1, B_r^1)}, lab_{(B_{r-1}^1, B_r^1)}) = (G_{(B_{r-1}^1 \cup H_r, B_r^1 - H_r)}, lab_{(B_{r-1}^1 \cup H_r, B_r^1 - H_r)})$. Thus, there exists an update schedule $s_2 = (B_i^2)_{i=1}^{nb(s_1)} \notin \{s, s_1\}$ where $B_j^2 = B_j^1$, $\forall j \in [\![1, nb(s_1)]\!] \backslash \{r-1, r\}$, $B_{r-1}^2 = B_{r-1}^1 \cup H_r$ and $B_r^2 = B_r^1 - H_r$ (eventually $B_r^2 = \emptyset$ and therefore $nb(s_2) < nb(s_1)$) such that $(G, lab_{s_2}) = (G, lab_{s_1})$. If such r does not exist, then Lemma 6 implies that $s_1 \in \texttt{UD}$ and since $(G, lab_{s_1}) = (G, lab_s)$, then $s \in [s_1]_G$. Applying recursively these arguments to s_2, s_3 and so on and since each block has a finite number of nodes, we deduce that there exists $l \geq 1$ such that $s_l \in \texttt{UD}$, $s_l \neq s$ and $s \in [s_l]_G$.

$\Leftarrow$) Let $s' \in \texttt{UD}$, $s' \neq s$, such that $s \in [s']_G$, i.e. $[s]_G = [s']_G$. Suppose on the contrary that $s \in \texttt{UD}$, then due to Lemma 7, we conclude that $[s]_G \neq [s']_G$, a contradiction. □

Theorem 9 *Let G be a digraph. Then, the output of algorithm* $\texttt{EqClass}(G)$ *is* $\texttt{UD} = \{s_1, \ldots, s_k\}$ *where $\{[s_i]_G | 1 \leq i \leq k\}$ is a partition of the set of update schedules of G and such that $\forall 1 \leq i \leq k$, $\forall s \in [s_i]_G$, $nb(s) \geq nb(s_i)$.*

PROOF.

The fact that $\{[s_i]_G | 1 \leq i \leq k\}$ is a partition of the set of update schedules of G is directly obtained from Lemmas 6, 7 and 8. On the other hand, Let us suppose that there exist $s_i = (B_j^i)_{j=1}^l \in \texttt{UD}$ and $s = (B_j)_{j=1}^k \in [s_i]_G$ such that $nb(s) = k < nb(s_i) = l$. From proof of Lemma 6, there exists $\exists w \in B_p \setminus B_p^i$

with $p = \min\{j : B_j \neq B_j^i\}$ and applying the same arguments we obtain a contradiction. □

Note that the algorithm `MoveTest` is polynomial in the size of $C \cup D$. Indeed, if for any $u \in D$ we denote $D(u) = \{v \in D :$ there exists a path from u to $v\} \cup \{u\}$ the set of vertices reached from u in D, then the condition $\texttt{MoveTest}(C, D) = 1$ is equivalent to $\exists u \in D, \forall y \in C, \forall v \in D(u)$, $(y, v) \notin A(G)$ and this last condition can be easily tested in polynomial time. On the other hand, the number of recursive invocations made by `EqClass` depends on $|U(G)|$, because $|\texttt{UD}| = |U(G)|$, and the minimum number of blocks of update schedules in each equivalence class. Thus, `EqClass` is efficient in digraphs with a big number of equivalence classes and in digraphs with representative schedules having a small number of blocks (as in Example 1). We also observe that `DigraphUD` is a base algorithm that obviously can be optimized (as a future work), for example, with properties related with not to do partitions in the latest blocks of a given update schedule or in function of the structural properties of the digraph.

An example of the steps of the algorithm `EqClass` applied over the left side digraph in Fig. 4 is given in Table 2:

s	A	B	UD
s_e	$\emptyset$	$\{5,6,7\}$	$\{(5,6,7)\} \cup \texttt{DigraphUD}(s_e, \{5,6\}, \{7\})$
s_e	$\{5,6\}$	$\{7\}$	$\{(5,6,7),(5,6)(7)\} \cup \texttt{DigraphUD}(s_e, \{5,7\}, \{6\})$
s_e	$\{5,7\}$	$\{6\}$	$\{(5,6,7),(5,6)(7),(5,7)(6)\} \cup \texttt{DigraphUD}(s_e, \{6,7\}, \{5\})$
s_e	$\{6,7\}$	$\{5\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5)\} \cup \texttt{DigraphUD}(s_e, \{5\}, \{6,7\})$
s_e	$\{5\}$	$\{6,7\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7)\} \cup \texttt{DigraphUD}(s_e * \{5\}, \{6\}, \{7\})$
(5)	$\{6\}$	$\{7\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7)\} \cup \texttt{DigraphUD}(s_e * \{5\}, \{7\}, \{6\})$
(5)	$\{7\}$	$\{6\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7),(5)(7)(6)\} \cup \texttt{DigraphUD}(s_e, \{6\}, \{5,7\})$
s_e	$\{6\}$	$\{5,7\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7),(5)(7)(6),(6)(5,7)\} \cup \texttt{DigraphUD}(s_e * \{6\}, \{5\}, \{7\})$
(6)	$\{5\}$	$\{7\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7),(5)(7)(6),(6)(5,7)\} \cup \texttt{DigraphUD}(s_e * \{6\}, \{7\}, \{5\})$
(6)	$\{7\}$	$\{5\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7),(5)(7)(6),(6)(5,7),(6)(7)(5)\} \cup \texttt{DigraphUD}(s_e, \{7\}, \{5,6\})$
s_e	$\{7\}$	$\{5,6\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7),(5)(7)(6),(6)(5,7),(6)(7)(5),(7)(5,6)\} \cup \texttt{DigraphUD}(s_e * \{7\}, \{5\}, \{6\})$
(7)	$\{5\}$	$\{6\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7),(5)(7)(6),(6)(5,7),(6)(7)(5),(7)(5,6)\} \cup \texttt{DigraphUD}(s_e * \{7\}, \{6\}, \{5\})$
(7)	$\{6\}$	$\{5\}$	$\{(5,6,7),(5,6)(7),(5,7)(6),(6,7)(5),(5)(6,7),(5)(7)(6),(6)(5,7),(6)(7)(5),(7)(5,6)\}$

Table 2
`EqClass` applied over the left side digraph in Fig. 4. The respective equivalence classes and update digraphs associated to each representative update schedule of `UD` are showed in Table 4 and Fig. 5 respectively.

5 Running `EqClass` in Arabidopsis Thaliana

In this section, we will use `EqClass` algorithm in a real genetic regulation network of the floral morphogenesis in the plant *Arabidopsis thaliana* with the aim to discuss the ideas of the previous sections and thus show which is the importance added by our algorithm. We will consider the *reduced Mendoza*

and Alvarez-Buylla network which has two non-trivial strongly connected symmetric components and whose asymptotic dynamics has the same attractors as the original network (see (Elena, 2009; Demongeot et al., 2010) for more details). Thus, we will focus on work with the subdigraphs G and F depicted in Fig. 4, where the states of the network at time t, $x_i(t) \in \{0,1\}$, $i = 1, ..., 7$ are defined as follows:

$$x_1(t) = H(-2x_3(t-1) - 2x_2(t-1) - 1), \qquad x_5(t) = x_7(t-1),$$
$$x_2(t) = H(-2x_4(t-1) - 2x_1(t-1) - 2), \qquad x_6(t) = x_7(t-1),$$
$$x_3(t) = x_4(t-1), \qquad x_7(t) = H(x_5(t-1) + x_6(t-1) - 1),$$
$$x_4(t) = x_4(t-1), \qquad H(x(t)) = 1 \text{ if } x(t) > 0 \text{ and}$$
$$H(x(t)) = 0 \text{ if } x(t) \leq 0.$$

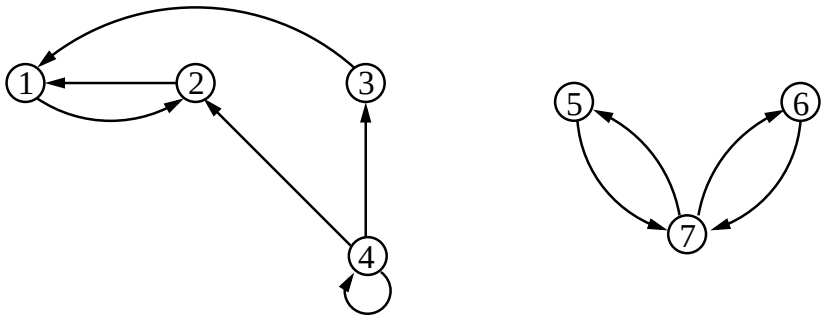

Fig. 4. The subdigraph of the reduced Mendoza and Alvarez-Buylla network composed by two connected components: G (left side) and F (rigth side). The vertices 1,...,7 represent the following genes of the plant Arabidopsis thaliana involved in its floral morphogenesis: *AGAMOUS (AG)*, *APETALATA 1 (AP1)*, *TERMINAL FLOWER 1 (TF1)*, *EMBRYONIC FLOWER 1 (EMF1)*, *APETALATA 3 (AP3)*, *PISTILLATA (PI)* and *BURST FORMING UNIT (BFU)*, respectively.

Components G and F are digraphs whose attractors have been enterily determined in (Demongeot et al., 2010) but not all its different dynamics, that apriori can be numerous, therefore through our algorithm, we need evaluate in a efficient way, only some (not necessarily all) update schedules in order to know the full spectrum of different dynamics associated to each component.

Let $N = (G, F, s)$ be a Boolean network. We define $D(G)$ as the set of all the different dynamics of N, then when we execute `EqClass` over the components G and F, the following values are summarized in Table 3.

We can see how these two components G and F of the reduced Mendoza and Alvarez-Buylla network shows a little number of different dynamics for G while for F the number of different dynamics does not decrease with respect to the number of update digraphs. The respective equivalence classes, update digraphs and dynamical behavior are detailed, for a more didactical purpose, only for F in Table 4, Fig. 5 and Tables 5 and 6 respectively.

X	$n = \|V(X)\|$	T_n	$\|\texttt{UD}\|$	$\|\texttt{UD}\|/T_n$	$\|D(X)\|$
G	4	75	20	0.27	6
F	3	13	9	0.69	9

Table 3
For the component G (F) of 4 (3) nodes there are 75 (13) ways to iterate the network (update schedule) that can be grouped into 20 (9) equivalence classes (update digraphs), each one of them with a representative update schedule obtained in the output `UD` of `EqClass` algorithm. Then to have the full spectrum of the dynamical behavior of the network, we need to evaluate a 27 (69) percent of all update schedules for G (F) which give us 6 (9) different dynamics.

$[s_1]_F$	$[s_2]_F$	$[s_3]_F$	$[s_4]_F$	$[s_5]_F$	$[s_6]_F$	$[s_7]_F$	$[s_8]_F$	$[s_9]_F$
(5,6,7)	(6)(5,7)	(5,7)(6)	(6,7)(5)	(6)(7)(5)	(7)(5,6)	(5)(6,7)	(5,6)(7)	(5)(7)(6)
					(7)(5)(6)		(5)(6)(7)	
					(7)(6)(5)		(6)(5)(7)	

Table 4
The different equivalence classes associated to F.

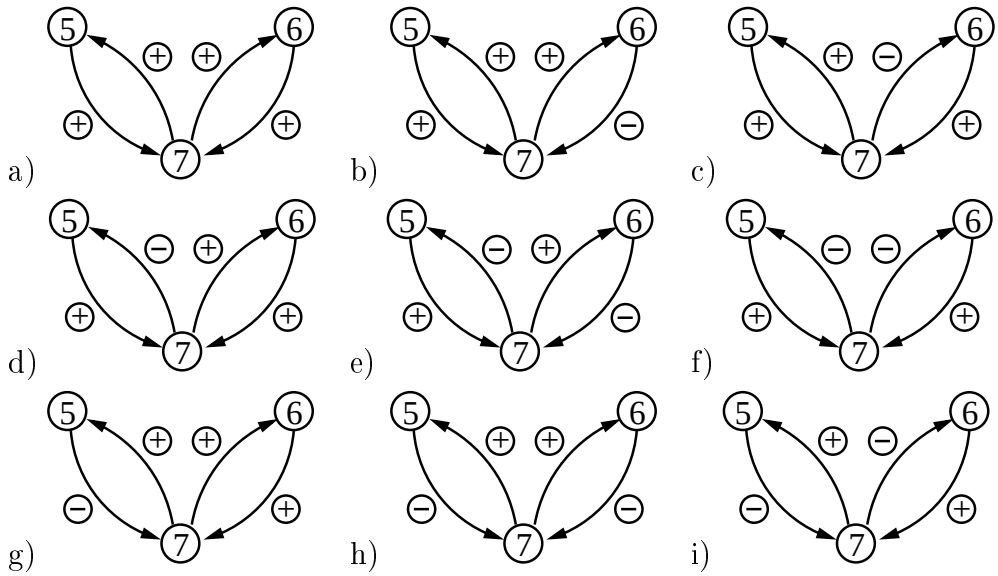

Fig. 5. The update digraphs F_1,...,F_9 associated to the equivalence classes of s_1,...,s_9 are showed in the sub-figures a),...,i) respectively.

6 Acknowledgments

This work was partially supported by FONDECYT project 1090549 (J.A.), by FONDAP and BASAL projects CMM (J.A. M.M.), Universidad de Chile, by Centro de Investigación en Ingeniería Matemática (CI^2MA), Universidad de Concepción and ANILLO ACT-88 (M.M.).

State	Sched. 1 $s_1(5)=1$ $s_1(6)=1$ $s_1(7)=1$	Sched. 2 $s_2(5)=2$ $s_2(6)=1$ $s_2(7)=2$	Sched. 3 $s_3(5)=1$ $s_3(6)=2$ $s_3(7)=1$	Sched. 4 $s_4(5)=2$ $s_4(6)=1$ $s_4(7)=1$	Sched. 5 $s_5(5)=3$ $s_5(6)=1$ $s_5(7)=2$
000	000	000	000	000	000
001	110	110	100	010	010
010	000	000	000	000	000
011	110	110	100	010	010
100	000	000	000	000	000
101	110	111	100	010	111
110	001	000	011	101	000
111	111	111	111	111	111

Table 5
Dynamics associated to F_1,...,F_5.

State	Sched. 6 $s_6(5)=2$ $s_6(6)=2$ $s_6(7)=1$	Sched. 7 $s_7(5)=1$ $s_7(6)=2$ $s_7(7)=2$	Sched. 8 $s_8(5)=1$ $s_8(6)=1$ $s_8(7)=2$	Sched. 9 $s_9(5)=1$ $s_9(6)=3$ $s_9(7)=2$
000	000	000	000	000
001	000	110	111	100
010	000	000	000	000
011	000	111	111	111
100	000	000	000	000
101	000	110	111	100
110	111	000	000	000
111	111	111	111	111

Table 6
Dynamics associated to F_6,...,F_9.

References

J. Aracena, E. Goles, A. Moreira, L. Salinas, On the robustness of update schedules in Boolean networks, Biosystems 97 (2009) 1–8.

S. Kauffman, Metabolic stability and epigenesis in randomly connected nets, Journal of Theoretical Biology 22 (1969) 437–467.

S. Kauffman, The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, New York, 1993.

C. Gershenson, Classification of random Boolean networks, in: Proc. of the 8th Int. Conf. on Artificial Life, MIT Press, 1–8, 2002.

R. Thomas, Boolean formalization of genetic control circuits, Journal of Theoretical Biology 42 (1973) 563–585.

R. Thomas, D. Thieffry, M. Kaufman, Dynamical behaviour of biological regulatory networksâĂŤI. Biological role of feedback loops and practical use of the concept of the loop-characteristic state., Bull. Math. Biol. 57 (1995) 247–276.

R. Thomas, M. Kaufman, Multistationarity, the basis of cell differentiation and memory. I. Structural conditions of multistationarity and other nontrivial behavior., Chaos 11 (1) (2001) 170–179.

A. Fauré, A. Naldi, C. Chaouiya, D. Thieffry, Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle, Bioinformatics 22 (14) (2006) 124–131.

J. Aracena, E. Fanchon, M. Montalva, M. Noual, Combinatorics on update digraphs in Boolean networks, Discrete Applied Mathematics 159 (6) (2011) 401–409.

J. Aracena, J. Demongeot, E. Fanchon, M. Montalva, On the number of update digraphs and its relation with the feedback arc sets and tournaments (2012) Submitted.

L. Mendoza, E. Alvarez-Buylla, Dynamics of the genetic regulatory network for Arabidopsis thaliana flower morphogenesis, Journal of Theoretical Biology 193 (1998) 307–319.

J. Demongeot, E. Goles, M. Morvan, M. Noual, S. Sené, Attraction Basins as Gauges of Robustness against Boundary Conditions in Biological Complex Systems, PLoS ONE 5 (8) (2010) doi:10.1371/journal.pone.0011793.

D. B. West, Introduction to Graph Theory, Prentice Hall, 1996.

J. Demongeot, A. Elena, S. Sené, Robustness in Regulatory Networks: a Multi-Disciplinary Approach, Acta Biotheoretica 56 (1-2) (2008) 27–49.

A. Elena, Robustesse des réseaux d'automates booleéns a seuil aux modes d'itération. Application a la modélisation des réseaux de régulation génétique, PhD thesis, Université Joseph Fourier (Grenoble I), Grenoble, France, 2009.