Mathematical Biosciences 242 (2013) 188–194

Contents lists available at SciVerse ScienceDirect

Mathematical Biosciences

journal homepage: www.elsevier.com/locate/mbs

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^{2}MA 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] UJF-Grenoble 1/CNRS/TIMC-IMAG UMR 5525, Grenoble, F-38041, France

ARTICLE INFO

Article history:
Received 13 April 2012
Received in revised form 11 January 2013
Accepted 18 January 2013
Available online 4 February 2013

Keywords:
Boolean network
Update schedule
Update digraph
Dynamics

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 by proving that the problem of the existence of a pair of update schedules with different dynamics is *NP*-complete. However, we show that certain structural properties of the interaction digraph are sufficient for guaranteeing distinct dynamics of a network. In [1] 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 this algorithm on the well known *Arabidopsis thaliana* network to determine the full spectrum of its different dynamics.

1. Introduction

Deterministic Boolean networks have been introduced in Systems Biology by Kauffman [2,3] 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 (block-sequential schedule). 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 modeling genetic networks and several extensions were proposed in the literature. Gershenson [4] 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.

Thomas, Thomas et al. and Thomas and Kauffman [5–7] developed a different approach in which 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, and so is the parameter space. 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

* Corresponding author. Tel.: +56 223311642; fax: +56 223311906.
E-mail 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).

0025-5564/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.mbs.2013.01.007

can often be recovered as a simplification of the Thomas dynamics [8].

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 words the function s defines the ordered partition into blocks of the set of nodes.

In this framework, Aracena et al. [1] 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. 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 [9] the combinatorial and algorithmical aspects of update digraphs were studied. In particular bounds on the number of equivalence classes were obtained.

One of the main analytical studies of equivalent update schedules in discrete networks has been made in sequential dynamical systems (SDS) [10,11]. These systems correspond to networks with sequential schedules and where the connection digraph is symmetric. Reidys [12] characterized the set of equivalent sequential schedules yielding a same dynamical behavior of a given SDS and gave a sharp upper bound for the number of different SDS. In [1] it was proved that these equivalence classes coincide with those defined in this paper in the particular case of SDS. For a more general case of SDS when the update order is a word (the alphabet is the set of nodes) and the interaction graph is a digraph, an upper bound was given in [13]. However, this case does not include the block-sequential update schedule considered in this work.

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 [9] 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 [14] we give an exact formula 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. Section 3 gives some results related to this issue; first, we point out the difficulty of knowing the different dynamics for a given network, problem that, to our knowledge, has not been studied in depth, yet. More specifically, we prove that the problem of determining whether there exist two different update schedules for a given network such that the associated dynamics are different is, in fact, *NP*-Complete. We prove a proposition which ensures the existence of different dynamics providing that the corresponding digraph has some structural property. We illustrate this result in the case of a particular family of digraphs where the choice of the activation function of each node can lead to two extreme situations: either all the dynamics are identical or they are all different. We explain how the analysis of update schedules yielding the same dynamics gives us bounds for the number of different dynamics in a given network. At the end of the section it should be clear that large computational savings would be achieved by an efficient algorithm enumerating the equivalence classes compatible with a given network architecture (digraph). In Section 4 we propose such an efficient algorithm.

Finally in Section 5, our theoretical and computational tools are applied to the study of the flower morphogenesis of the plant *A.* thaliana [15]. More specifically, we work with the reduced model defined by Demongeot et al. [16] which has two non-trivial connected components of 3 and 4 genes. Our results allow us to compute just one update schedule for each equivalence class, instead of enumerating all update schedules, and we show that this entails a significant reduction of the computational work. We are then able to compute the full spectrum of all the different dynamics associated with each component of the *A.* thaliana network.

2. Definitions

This section provides basic definitions and introduces the necessary notations.

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

A **digraph** is an ordered pair of sets $G = (V,A)$ where $V = \{1,\dots,n\}$ 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 $v_1, v_2, \dots, v_m$ of $V(G)$ such that $(v_k, v_{k+1}) \in A(G)$ for all $k = 1,\dots,m-1$. A **cycle** is a path $v_1,\dots,v_m$ such that $v_i \neq v_j$ for all $i \neq j$ with $i,j \in \{2,\dots,m-1\}$ and $v_1 = v_m$.

More terminology about digraphs can be found in [17].

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 $m \leqslant n$. A **partial update schedule** is an update schedule over the vertices of some $G' \subseteq G$. A **block** of s is the set $B_i = \{v \in V(G') : s(v) = i\}, 1 \leqslant i \leqslant m$. The number of blocks of s is denoted by $nb(s) \equiv m$. Frequently, s will be denoted by $s = (j \in B_1)(j \in B_2)\dots(j \in B_{nb(s)})$ 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 $2 \leqslant nb(s) \leqslant n-1, s$ is said to be a **block sequential** update schedule. As was mentionned in [18], 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 $T_0 \equiv 1$.

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:

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

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**. By labeling

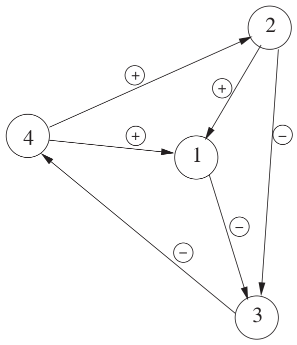

Fig. 1. A digraph $G = (V, A)$ labeled by the function lab_s where $\forall i \in V = \{1,\ldots,4\}, s(i) = i$.

every arc a of A by $lab_s(a)$, we obtain a labeled digraph (G, lab_s) named **update digraph** (see Fig. 1). We denote

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

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:

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

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

This is equivalent to applying 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 $s(i) \leqslant s(j)$ 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 from 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 update schedules $s' \neq s$such that $F^s \neq F^{s'}$ is NP-complete.*

Proof. It is easy to see that the problem of knowing whether there exist update schedules $s' \neq s$ such that $F^s \neq F^{s'}$ is *NP*. Indeed, for a given update schedule $s' \neq s, x \in \{0,1\}^n$ and $i \in \{1,\ldots,n\}$, where $|V(G)| = n$, we can check in polynomial time that $f_i^s(x) \neq f_i^{s'}(x)$, i.e. $F^s \neq F^{s'}$.

We present a polynomial reduction from the *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, let us consider first a satisfiable formula ϕ, with $\phi(\alpha) = 1$, $\alpha = (\alpha_1,\ldots,\alpha_n) \in \{0,1\}^n$. Then, for $s = (n+1)(n+2)(1,2,\ldots,n)$ and $s' = (n+2)(n+1)(1,2,\ldots,n), F^s(\alpha,1,0) = (\bar{\alpha},0,0)$ 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 schedule $s \neq s', F^s = F^{s'}$. □

However, for some Boolean networks, with interaction graphs having certain structural properties, it is possible to ensure the existence of update schedules which yield 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, $f_i(x) = x_j$ or $f_i(x) = \bar{x}_j$. 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 $k \neq j$, then $\exists y \in \{0,1\}^n, f_j(y) = \bar{y}_j$, 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 $s_1(j) \geqslant s_1(i), s_2(j) < s_2(i)$ and $\forall k \in V^-(j), s_1(k) \geqslant s_1(j) \wedge s_2(k) \geqslant s_2(j)$. 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, $F^{s_1} \neq F^{s_2}$. □

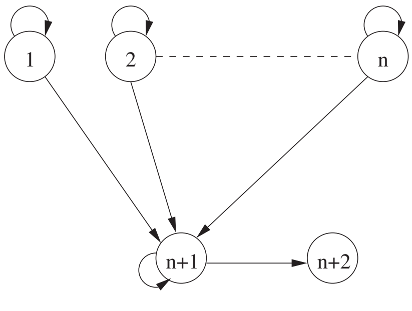

Fig. 2. Interaction graph G for the proof of Theorem 1.

On the other hand, in [1], the authors studied the robustness of the dynamical behavior of a Boolean network with respect to different update schedules and 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.*

Using 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 correspondence between the elements of $U(G)$ in (2) and the equivalence classes defined 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 ([9,14]).

For example in [9] the authors proved that in the particular case of complete digraphs the number of equivalence classes is T_n (the maximum value for $|U(G)|$), with exactly one update schedule in each class, i.e. a complete digraph could possibly have T_n different dynamics. However in applications the networks are rarely complete digraphs, except for the n-switches often met in plant morphogenesis ([19,20]). For digraph that are not complete the number of equivalence classes is strictly smaller than T_n.

It is also possible that two nonequivalent update schedules, i.e. belonging to distinct equivalence classes, yield the same dynamical behavior. The following example exhibits families of networks where all pairs of nonequivalent 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 $\tilde{F}$ are defined by $f_i(x) = \tilde{f}_i(x) = x_n \ \forall i = 1, \ldots, n-1$ and $f_n(x) = x_n$ and $\tilde{f}_n(x) = \bar{x}_n$ for every $x \in \{0,1\}^n$. Hence, $|U(G)| = 2^{n-1}$ and since $f_n^s(x) = x_n$ and $\tilde{f}_n^s(x) = \bar{x}_n \neq x_n$ 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 pair (s, s') of nonequivalent update schedules. Therefore, the dynamics of $N_1 = (G, F, s_1)$ and $N_2 = (G, F, s_2)$ are the same, and that of $\tilde{N}_1 = (G, \tilde{F}, s_1)$ and $\tilde{N}_2 = (G, \tilde{F}, s_2)$ are different for every pair (s_1, s_2) of nonequivalent update schedules.

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

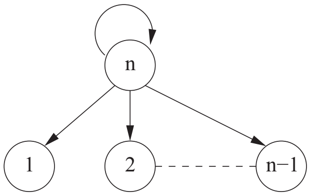

Fig. 3. Interaction graph G for Example 1. Note that $N = (G, F, s)$ gives the same dynamic whatever the update schedule s, while $\tilde{N} = (G, \tilde{F}, s)$ has $|U(G)| = 2^{n-1}$ different dynamics, one for each equivalence class.

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

At this point, we can 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.

Such cases help to understand more intuitively the computational savings that can be achieved if one wants to know 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 determining the different dynamics of a given Boolean network when only the deterministic update schedule is changed. A straightforward approach for doing this is, first, to determine the set of update schedule equivalence classes, and then to compute the dynamical behavior of each class.

4. Enumerating update digraphs

In this section, we exhibit an algorithm for enumerating a representative update schedule, with the smallest number of blocks, for all the equivalence classes. This will then allow to determine the different dynamics of a Boolean network when the deterministic update schedule is varied.

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 $lab_{(C,D)} : A(G_{(C,D)}) \rightarrow \{\oplus, \ominus\}$ by:

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

Definition 5. Let G be a digraph, $s = (j \in B_1)(j \in B_2) \ldots (j \in B_{nb(s)})$ 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:

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

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.

Algorithm 1. `EqClass`(G)

```
Input: $G = (V, A)$ a digraph
Output: UD, a set of representative update schedules, one for
  each equivalence class associated to $G$
begin
  UD ← DigraphUD($s_e, \emptyset, V$);
end
```

Lemma 6. *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,* $s \in \texttt{UD} \iff (s = s_p \vee \texttt{MoveTest}(B_i, B_{i-1}) = 0, \forall i \in [[2,k]])$.

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 $\texttt{UD}$ the output of the algorithm $\texttt{EqClass}(G)$.

$\Rightarrow$) 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 $\texttt{DigraphUD}$, $\texttt{MoveTest}(B_{k-2}, B_{k-1}) = \texttt{MoveTest}(B_{k-1}, B_k) = 0$ (possibly $B_{k-2} = \emptyset$). This also implies that $\texttt{MoveTest}(B_{k-3}, B_{k-2}) = 0$, because otherwise s would not be invoked 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 $k \geqslant 2$ and $\texttt{MoveTest}(B_i, B_{i-1}) = 0, \forall i \in [[2,k]]$, then s is invoked 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 $\texttt{UD}$. □

Algorithm 2 $\texttt{DigraphUD}(s, A, B)$

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

Lemma 7. *Let G be a digraph and* $\texttt{UD}$ *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 $\texttt{UD}$ 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 exist a path from w to v in G (possibly $v = w$) because otherwise there exist $H = \{w\} \cup \{v \in B_p :$ there is a path from w to $v\} \subseteq B_p$ such that $(G_{(B_{p-1},B_p)}, lab_{(B_{p-1},B_p)}) = (G_{(B_{p-1}\cup H, B_p - H)}, lab_{(B_{p-1}\cup H, B_p - H)})$, therefore $\texttt{MoveTest}(B_{p-1}, B_p) = 1$ 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 $j = r$. □

Algorithm 3: $\texttt{MoveTest}(C, D)$

```
Input: $C, D$ subsets of vertices of a digraph $G$
bf 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 otherwise
begin
  if $C = \emptyset$then
    return (0);
  else
      if $\exists H \subseteq D$ such that
   $(G_{(C,D)}, lab_{(C,D)}) = (G_{(C\cup H, D-H)}, lab_{(C\cup H, D-H)})$
    then
      return (1);
    else
      return (0);
    end
  end
end
```

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* $\texttt{UD}$ *the output of the algorithm* $\texttt{EqClass}(G)$. *Then,* $s \notin \texttt{UD} \iff \exists s' \in \texttt{UD}, s' \neq s$, *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 $\texttt{UD}$ 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 exist an update schedule $s_1 = (B_i^1)_{i=1}^{k} \neq s$ where $B_j^1 = B_j, \forall j \in [[1,k]] \setminus \{r-1, r\}$, $B_{r-1}^1 = B_{r-1} \cup H_r$ and $B_r^1 = B_r - H_r$ (it is possible that $B_r^1 = \emptyset$ and therefore $nb(s_1) < k$) such that $(G, lab_{s_1}) = (G, lab_s)$. Now, considering s_1, if there exist $r = \min\{i : \texttt{MoveTest}(B_{i-1}^1, B_i^1) = 1\}$, 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 exist an update schedule $s_2 = (B_i^2)_{i=1}^{nb(s_1)} \notin \{s, s_1\}$ where $B_j^2 = B_j^1, \forall j \in [[1, nb(s_1)]] \setminus \{r-1, r\}$, $B_{r-1}^2 = B_{r-1}^1 \cup H_r$ and $B_r^2 = B_r^1 -$

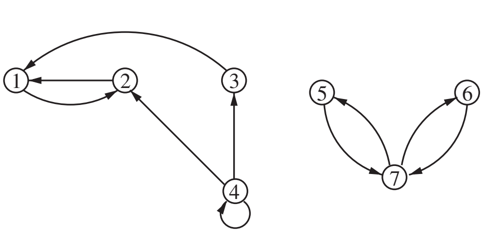

Fig. 4. The subdigraph of the reduced Mendoza and Alvarez-Buylla network composed of two connected components: G (left side) and F (right side). The vertices 1,…,7 represent the following genes of the plant *A. thaliana* involved in its floral morphogenesis: *AGAMOUS (AG)*, *APETALATA 1 (AP1)*, *TERMINAL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

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

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

H_r (it is possible that $B_r^2 = \emptyset$ 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 $s_1 \in \mathtt{UD}$ 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 exist $l \geqslant 1$ such that $s_l \in \mathtt{UD}, s_l \neq s$ and $s \in [s_l]_G$.

$\Leftarrow$) Let $s' \in \mathtt{UD}, s' \neq s$, such that $s \in [s']_G$, i.e. $[s]_G = [s']_G$. Suppose on the contrary that $s \in \mathtt{UD}$, 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 $\mathtt{EqClass}(G)$ is $\mathtt{UD} = \{s_1, \ldots, s_k\}$ where $\{[s_i]_G \mid 1 \leqslant i \leqslant k\}$ is a partition of the set of update schedules of G and such that $\forall 1 \leqslant i \leqslant k, \forall s \in [s_i]_G, nb(s) \geqslant nb(s_i)$.*

Proof. The fact that $\{[s_i]_G \mid 1 \leqslant i \leqslant k\}$ is a partition of the set of update schedules of G is directly obtained from Lemmas 6–8. On the other hand, Let us suppose that there exist $s_i = (B_j^i)_{j=1}^l \in \mathtt{UD}$ 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 exist $w \in B_p \setminus B_p^i$ 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 exist a path from $u$to $v\} \cup \{u\}$ the set of vertices reached from u in D, then the condition $\mathtt{MoveTest}(C, D) = 1$ 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 $|\mathtt{UD}| = |U(G)|$) and on the minimum number of blocks of update schedules in each equivalence class. Thus, EqClass is efficient in digraphs with a large 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 allowing to avoid the partitioning in the latest blocks of a given update schedule.

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 gives us 6 (9) different dynamics.

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

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

5. Running **EqClass** in *A. Thaliana*

In this section, we use the EqClass algorithm in a real genetic regulation network of the floral morphogenesis in the plant *A. thaliana* with the aim to discuss the ideas of the previous sections and thus show the gain provided 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 [21,16] 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 $t, x_i(t) \in \{0,1\},\ i = 1, \ldots, 7$ are defined as follows:

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

Components G and F are digraphs whose attractors have been entirely determined in [16] but not all their different dynamics, which a priori can be numerous. Our algorithm evaluates in an efficient way only part of the update schedules (unless the digraph is complete) in order to determine the full spectrum of the different dynamics associated to components G and F.

Let $N = (G, F, s)$ be a Boolean network. We define $D(G)$ as the set of all the different dynamics of N. The result of the execution of EqClass over the components G and F are summarized in Table 3.

In this Table we observe that the reduction factor is 0.27 for component G and 0.69 for component F. This means that in order to determine the truly different dynamics one needs to compute the dynamics of only 27% of all the possible update schedules for G (69% for F). When this computation is performed, we note that the number of truly different dynamics is reduced from 20 to 6 for G, but is not further reduced for F, meaning that for F all the equivalence classes have different dynamics. On the other hand, from the above computational analysis for G and F components, it was also possible to observe that in their asymptotic behaviors, only 2 of the 6 different dynamics in G have a limit cycle while this kind of attractor exists in only 1 of the 9

Table 4
The different equivalence classes associated to F.

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

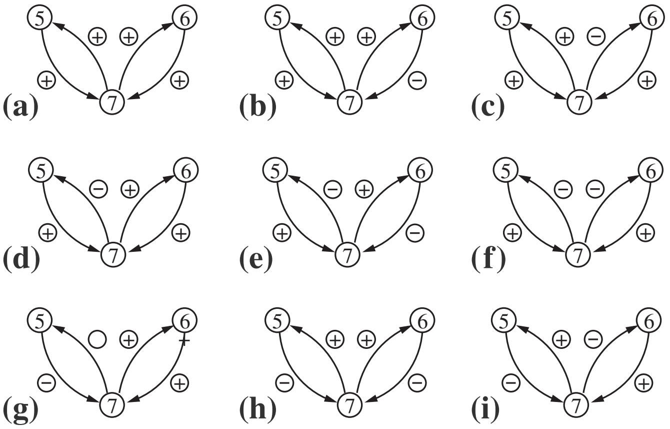

Fig. 5. The update digraphs $F_1,\ldots,F_9$ associated to the equivalence classes of $s_1,\ldots,s_9$ are showed in the sub-figures $a),\ldots,i)$ respectively.

Table 5
Dynamics associated to $F_1,\ldots,F_5$.

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 6
Dynamics associated to $F_6,\ldots,F_9$.

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

different dynamics in F. In all these cases, the limit cycles are of length 2 (which can be deduced from results in [22,16]). This is another way to validate the fact that most of the dynamics of the original network have only fixed points (which represents the phenotypes of the flower) as was showed in [16]. The equivalence classes, update digraphs and dynamical behavior are detailed, for a didactical purpose, only for F in Table 4, Fig. 5 and Tables 5 and 6, respectively.

Acknowledgement

This work was partially supported by FONDECYT 1090549 and ANILLO ACT-87 (J.A.), by FONDECYT 3130466 and ANILLO ACT-88 (M.M.), and by FONDAP and BASAL projects CMM, Universidad de Chile, by Centro de Investigación en Ingeniería Matemática (CI^2MA), Universidad de Concepción.

References

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

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

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

[4] C. Gershenson, Classification of Random Boolean Networks, Proc. of the 8th Int. Conf. on Artificial Life, MIT Press, 2002, pp. 1–8.

[5] R. Thomas, Boolean formalization of genetic control circuits, Journal of Theoretical Biology 42 (1973) 563.

[6] R. Thomas, D. Thieffry, M. Kaufman, Dynamical behaviour of biological regulatory networks-I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state, Bulletin of Mathematical Biology 57 (1995) 247.

[7] 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.

[8] 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.

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

[10] H. Mortveit, C. Reidys, Discrete sequential dynamical systems, Discrete Mathematics 226 (2001) 281.

[11] H. Mortveit, C. Reidys, An Introduction to Sequential Dynamical Systems, Springer, 2007.

[12] C. Reidys, On acyclic orientations and sequential dynamical systems, Advances in Applied Mathematics 27 (2001) 790.

[13] L. Garcia, A. Jarrah, R. Laubenbacher, Sequential dynamical systems over words, Applied Mathematics and Computation 174 (2006).

[14] J. Aracena, J. Demongeot, E. Fanchon, M. Montalva, On the number of update digraphs and its relation with the feedback arc sets and tournaments, Discrete Applied Mathematics (2013), http://dx.doi.org/10.1016/j.dam.2012.12.018.

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

[16] 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), http://dx.doi.org/10.1371/journal.pone.0011793.

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

[18] J. Demongeot, A. Elena, S. Sené, Robustness in regulatory networks: a multi-disciplinary approach, Acta Biotheoretica 56 (1–2) (2008) 27.

[19] O. Cinquin, J. Demongeot, High-dimensional switches and the modeling of cellular differentiation, Journal of Theoretical Biology 233 (2005) 391.

[20] J. Demongeot, M. Thellier, R. Thomas, Storage and recall of environmental signals in a plant: modelling by use of a differential (continuous) formulation, Comptes Rendus Biologies 329 (2006) 971.

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

[22] E. Goles, J. Olivos, Comportement périodique des functions à seuil binaires et applications, Discrete Applied Mathematics 3 (1981) 95.