Automata 2010 — 16th Intl. Workshop on CA and DCS DMTCS proc. **AL**, 2010, 41–50

Block-sequential update schedules and Boolean automata circuits

Eric Goles[1,2] and Mathilde Noual[3,4†]

[1]*University Adolfo Ibañez, Peñalolen, Santiago, Chile*
[2]*Complex Systems Institute of Valparaiso, ISCV, Valparaiso, Chile*
[3]*University of Lyon, ÉNS-Lyon, LIP, CNRS UMR 5668, F-69007, Lyon, France*
[4]*Rhône-Alpes Complex Systems Institute, IXXI, F-69007, Lyon, France*

Our work is set in the framework of complex dynamical systems and, more precisely, that of Boolean automata networks modeling regulation networks. We study how the choice of an update schedule impacts on the dynamics of such a network. To do this, we explain how studying the dynamics of any network updated with an arbitrary block-sequential update schedule can be reduced to the study of the dynamics of a different network updated in parallel. We give special attention to networks whose underlying structure is a circuit, that is, Boolean automata circuits. These particular and simple networks are known to serve as the "engines" of the dynamics of arbitrary regulation networks containing them as sub-networks in that they are responsible for their variety of dynamical behaviours. We give both the number of attractors of period p, $\forall p \in \mathbb{N}$ and the total number of attractors in the dynamics of Boolean automata circuits updated with any block-sequential update schedule. We also detail the variety of dynamical behaviours that such networks may exhibit according to the update schedule.

Keywords: Boolean automata network, cycles/circuits, attractors, discrete dynamical system, update/iteration schedule

1 Introduction

From the point of view of theoretical biology as well as that of theoretical computer science, it seems to be of great interest to address the question of the number of different asymptotic dynamical behaviours of a regulation network. Close to the 16th Hilbert problem concerning the number of limit cycles of dynamical systems [10], this question has already been considered in a certain number of works [3, 2, 13]. In the same lines and with a similar will to understand the dynamical properties of (regulation) networks, we decided to focus on the dynamics of Boolean automata networks.

Two aspects of these networks caught our attention. The first one is that, as Thomas [15] already noticed, the "driving force" of their dynamics lies in their underlying circuits. Indeed, a network whose underlying interaction graph is an acyclic digraph can only eventually end up in a configuration that will never change over time (*aka.* fixed point). A network with retroactive loops, on the contrary, exhibits

† corresponding author

1365–8050 © 2010 Discrete Mathematics and Theoretical Computer Science (DMTCS), Nancy, France

more diverse dynamical behaviour patterns. This is why, before attempting to explain theoretically the dynamics of Boolean automata networks whose interaction graphs are arbitrary, we decided to pay special attention to the simple instance of Boolean automata networks that are Boolean automata circuits[(i)].

The other essential aspect of Boolean networks, or more generaly, of regulation networks, that we concentrated on is their update schedule, that is, the order according to which the different interactions that define the system occur. Robert [14] highlighted the importance of update schedules on the dynamics of a system. In [7], the focus was put on the parallel update schedule that updates all automata of a network synchronously at each time step of a discretised time scale. Now, although biological knowledge about the precise schedules of gene regulations lack, one may argue reasonably that genes involved in a same cellular physiological function are highly unlikely to perform there regulations in perfect synchrony although biologists seem to agree that a certain amount of synchrony is not, on the whole, implausible. In this paper, we consider a looser version of the parallel update schedule, namely, the general *block-sequential* schedule that updates every automaton of a network exactly once at every step according to a predefined order but which does not impose that all automata be updated at once. In other words, block-sequential schedules define *blocks* of automata to be updated sequentially while within the blocks, the automata are updated synchronously.

Section 2 introduces some definitions relative to general Boolean automata networks as well as some preliminary results. Section 3 focuses on Boolean automata circuits and on their dynamics under arbitrary block-sequential update schedules[(ii)].

2 Networks and their dynamics

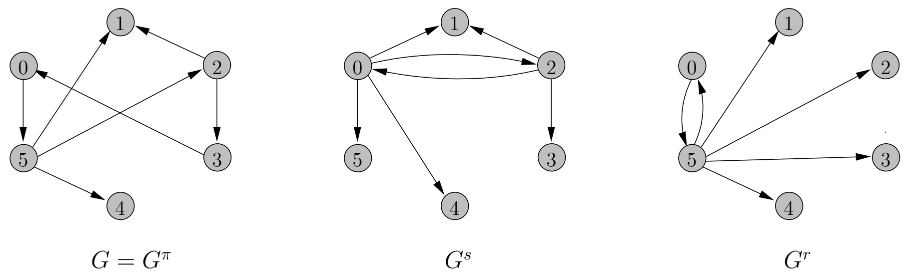
We define a *Boolean automata network* of size n as a couple $N = (G, \mathcal{F})$ where $G = (V, A)$ is a digraph of order $|V| = n$ called the *interaction graph* of the network. The nodes of G are assimilated to the automata of N. Vectors of $\{0,1\}^n$ are seen as *configurations* of N. Their i^{th} components are the states of nodes $i \in V$. $\mathcal{F} = \{f_i : \{0,1\}^n \to \{0,1\} \mid i \in V\}$ is the set of *local transition functions* of the network. For each node $i \in V$, and each configuration $x \in \{0,1\}^n$, $f_i(x)$ depends only on the components x_j such that $(j,i) \in A$. For the sake of simplicity, we consider abusingly, in some cases, that f_i is a function of arity $deg^-(i) = |\{j \in V \mid (j,i) \in A\}|$ instead of n.

To define the dynamics of N, an update schedule s of the states of nodes needs to be specified. In this paper, we consider only *block-sequential update schedules*, that is, functions $s : V \to \{0, \ldots, n-1\}$ such that for any node $i \in V,\ s(i)$ gives the date of update of node i ($t + \frac{s(i)}{s_{max}},\ s_{max} = max\{s(i) \mid i \in V\}$) between any two time steps t and $t+1$. Thus, *within* a time step t, the states of all nodes are updated exactly once. Without loss of generality, we suppose that update schedules s impose no "waiting period" within a time step: $min\{s(i) \mid i \in V\} = 0$ and $\forall 0 \leq d < n-1,\ \exists i \in V,\ s(i) = d+1 \Rightarrow \exists j \in V,\ s(j) = d$. The *parallel update schedule* denoted here by π is the update schedule such that $\forall i \in V,\ \pi(i) = 0$. It updates all nodes at once. A *sequential update schedule* is a block-sequential update schedule s that updates only one node at a time: $\forall i \neq j,\ s(i) \neq s(j)$. The number of different update schedules of a set of n elements is known to be exponential in n [6].

Example 2.1 *Let* $V = \{0, \ldots, 5\}$. *The function* $r : V \to \{0, \ldots, 5\}$ *such that* $r(2) = 0$, $r(3) = r(4) = 1$ *and* $r(0) = r(1) = r(5) = 2$ *is a block sequential update schedule. The function* $s : V \to \{0, \ldots, 5\}$

(i) and which also happen to be a simple instance of *threshold* Boolean automata networks [11].

(ii) Results presented in this paper and their proofs are detailed in [12].



$i \in V$	$x_i(t+1)$		
	π	$s = (5)(3)(1)(0)(2)(4)$	$r = (2)(3,4)(0,1,5)$
0	$f_0(x_3(t))$	$f_0(x_3(t+1)) = f_0 \circ f_3(x_2(t))$	$f_0(x_3(t+1)) = f_0 \circ f_3 \circ f_2(x_5(t))$
1	$f_1(x_2(t), x_5(t))$	$f_1(x_2(t), x_5(t+1)) = f_1(x_2(t), f_5(x_0(t)))$	$f_1(x_2(t+1), x_5(t)) = f_1(f_2(x_5(t)), x_5(t))$
2	$f_2(x_5(t))$	$f_2(x_5(t+1)) = f_2 \circ f_5(x_0(t))$	$f_2(x_5(t))$
3	$f_3(x_2(t))$	$f_3(x_2(t))$	$f_3(x_2(t+1)) = f_3 \circ f_2(x_5(t))$
4	$f_4(x_5(t))$	$f_4(x_5(t+1)) = f_4 \circ f_5(x_0(t))$	$f_4(x_5(t))$
5	$f_5(x_0(t))$	$f_5(x_0(t))$	$f_5(x_0(t))$

Fig. 1: Above: interaction graphs associated to the three different update schedules considered in example 2.1. Below: a table giving the dependencies between states of nodes according to the update schedule of the network.

such that $s(5) = 0,\ s(3) = 1,\ s(1) = 2,\ s(0) = 3,\ s(2) = 4$ *and* $s(4) = 5$ *is a sequential update schedule. A more practical way of denoting* r*,* s *and the parallel update schedule is the following:*

$$r \equiv (2)(3,4)(0,1,5) \qquad s \equiv (5)(3)(1)(0)(2)(4) \qquad \pi \equiv (0,1,2,3,4,5).$$

A network $N = (G, \mathcal{F})$, updated according to a block-sequential update schedule s is denoted by $N(s)$. Its dynamics is defined by the following *global transition function*:

$$F_s : \left\{ \begin{array}{ccc} \{0,1\}^n & \to & \{0,1\}^n \\ x & \mapsto & (f_0^s(x), \dots, f_{n-1}^s(x)) \end{array} \right. \tag{1}$$

where $\forall i \in V,\ f_i^s$ is the *local transition function of node* i *relative to* s and is defined by:

$$f_i^s(x) = f_i(x^{(s,i)}), \qquad \forall j \in V,\ x_j^{(s,i)} = \begin{cases} x_j & \text{if } s(j) \geq s(i) \\ f_j^s(x) & \text{if } s(j) < s(i). \end{cases} \tag{2}$$

In particular, if $s = \pi$ then $\forall i \in V,\ f_i^s = f_i$ and the global transition function simplifies to: $F^{(\pi)}(x) = (f_0(x), \ldots, f_{n-1}(x))$. When there is no ambiguity as to what network is being considered, for any initial configuration $x \in \{0,1\}^n$, we write $x = x^s(0)$ and $x^s(t) = F_s^t(x)$ (where F_s is composed t times) and when there is no ambiguity either on s, we write $x = x(0)$ and $x(t) = F_s^t(x)$. With this notation, (1) and (2) mean that $\forall i, j \in V$ such that $(j,i) \in A$, $x_i(t+1)$ depends on $x_j(t)$ if $s(j) \geq s(i)$, and on $x_j(t+1)$ if $s(j) < s(i)$.

For a network N updated with a particular update schedule s, we define a new interaction graph $G^s = (V, A^s)$, the interaction graph relative to s (see figure 1) such that $A^s = \{(j,i) \mid x_i^s(t{+}1) \text{ depends on } x_j^s(t)\}$. By an easy induction, this set of arcs can be shown to be equal to:

$$A^s = \{(j,i) \mid \text{there exists in } G \text{ a directed path } \{v_0 = j, v_1 \ldots, v_l = i\} \\ \text{from } j \text{ to } i \text{ such that } s(j) \geq s(v_1) \text{ and } \forall\ 1 \leq k < l,\ s(v_k) < s(v_{k+1})\}. \quad (3)$$

An important point is that when $s = \pi$, $G^\pi = G$. Further, define $N^s = (G^s, \mathcal{F}^s)$ to be the network whose interaction graph is G^s and whose set of local transition functions is $\mathcal{F}^s = \{f_i^s \mid i \in V\}$. Then, as one may check, the dynamics of $N^s(\pi)$ is identical to that of $N(s)$: the global transition functions of both networks are equal to F_s. As a result, provided a characterisation of the graphs G^s, we may bring our study of networks updated with arbitrary block-sequential update schedules back to the study of networks updated in parallel.

The dynamics of a network N updated with an update schedule s is described by its *iteration graph* $\mathcal{I}(N(s))$ (and also, from the previous paragraph by the iteration graph $\mathcal{I}(N^s(\pi))$) whose nodes are the configurations of N and whose arcs are the transitions $(x(t), x(t+1))$ from one configuration to another. Since the set of configurations of any finite sized network is finite, all trajectories necessarily end up looping, *i.e.,* $\forall\ x(0) \in \{0,1\}^n,\ \exists t, p,\ x(t+p) = x(t)$. *Attractors* of $N(s)$ are orbits of such configurations $x(t)$ for which there exists a $p \in \mathbb{N}$ such that $x(t) = x(t+p)$. The smallest such p is called the period of the attractor. Attractors of period one are called fixed points.

3 Boolean automata circuits

As mentioned in the introduction, we pay special attention here to a particular instance of Boolean automata networks called *Boolean automata circuits* [7]. A *circuit* of size n is a digraph denoted by $\mathbb{C}_n = (V, A)$. Its set of nodes $V = \{0, \ldots, n-1\}$ is identified with $\mathbb{Z}/n\mathbb{Z}$ so that, considering two nodes i and j, $i + j$ designates the node $i + j \mod n$. The set of arcs of $\mathbb{C}_n$ is $A = \{(i, i+1) \mid i \in \mathbb{Z}/n\mathbb{Z}\}$. A *Boolean automata circuit* is a Boolean automata network whose interaction graph is a circuit. Since any node i in this graph has a unique incoming neighbour, $i-1$, its local transition function f_i is either equal to the identity function $id : a \in \{0,1\} \mapsto a$ or to the negation function $neg : a \in \{0,1\} \mapsto \neg a = 1 - a$. In the first case, the arc $(i-1, i)$ is said to be *positive* and in the second case it is said to be *negative*. When there is an even number of negative arcs in the circuit, then the the *sign* of the (Boolean automata) circuit is said to be positive. Otherwise it is said to be negative.

Let $C = (\mathbb{C}_n, \mathcal{F})$ be a Boolean automata circuit of size n whose set of local transition functions is $\mathcal{F} = \{f_i \mid i \in \mathbb{Z}/n\mathbb{Z}\})$. Let s be an arbitrary block-sequential update schedule of C. For any node $i \in V$, let us note:

$$i^* \ = \ max\{k < i \mid s(k) \geq s(k+1)\}.$$

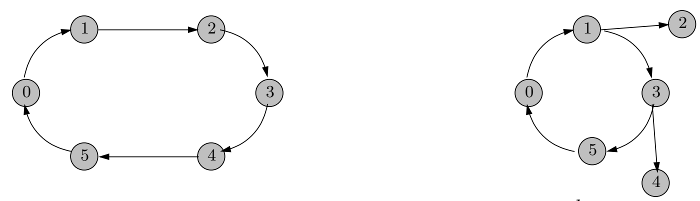
where the maximum is taken cyclically so that the number of arcs on a path from i^* to i is is minimal. From (3), it holds that $A^s = \{(i^*, i) \mid i \in V\}$ and it can be shown that $\forall i \in \mathbb{Z}/n\mathbb{Z},\ f_i^s = F[i, i^* + 1]$ where:

$$\forall i, j \in V,\ F[j,i] = \begin{cases} f_j \circ f_{j-1} \circ \ldots \circ f_i & \text{if } i \leq j \\ f_j \circ f_{j-1} \circ \ldots \circ f_0 \circ f_{n-1} \circ \ldots \circ f_i & \text{if } j < i \end{cases}$$

Following the remarks made in the previous section, the dynamics of $C(s)$ is identical to that of $C^s(\pi) = (\mathbb{C}_n^s, \mathcal{F}^s)$ where $\mathcal{F}^s = \{F[i, i^* + 1] \mid i \in \mathbb{Z}/n\mathbb{Z}\}$. Let us describe the digraph $\mathbb{C}_n^s$. To do this, we first define the *inversions* of C relative to s:

$$inv(s) = A \setminus A^s = \{(i, i+1) \mid s(i) < s(i+1)\}.$$

For nodes of an inversion $(i, i+1)$, $x_{i+1}^s(t+1)$ depends on $x_i^s(t+1)$ instead of $x_i^s(t)$ as is the case when $s(i+1) \leq s(i)$ and when, in particular, $s = \pi$. Obviously, the number of inversions is strictly smaller than n. The only block-sequential update schedule that has no inversions is the parallel update schedule π. From the characterisation of A^s given in equation 3, we derive that the nodes i^* (*i.e.*, the nodes $i \in \mathbb{Z}/n\mathbb{Z},\ \exists j \in \mathbb{Z}/n\mathbb{Z},\ i = j^*$) form a circuit in $\mathbb{C}_n^s$ of size $n - |inv(s)|$. The $|inv(s)|$ other nodes that do not belong to this circuit depend on one and only one node in it (as in Figure 2). And since the composition of all functions f_i is necessarily equal to the composition of all functions $F[i, i^* + 1]$, the sign of this circuit is equal to the sign of the original circuit $\mathbb{C}_n$. From this description of the network C^s, we may now derive the following result:



a. b.

Fig. 2: a. The underlying interaction graph $\mathbb{C}_6$ of a network $C = (\mathbb{C}_6, \mathcal{F})$. b. The interaction graph of C^s where $s \equiv (2)(3,4)(0,1,5)$ and $inv(s) = \{(2,3),(4,5)\}$. The underlying circuit of size 4 in this second interaction graph has as set of nodes $\{0,1,3,5\} = \{i \in \mathbb{Z}/6\mathbb{Z} \mid \exists j \in \mathbb{Z}/6\mathbb{Z},\ i = j^*\}$.

Proposition 3.1 *Let $C = (\mathbb{C}_n, \mathcal{F})$ be a Boolean automata circuit of size n and let s and r be two block-sequential update schedules of C. Then:*

(i) The dynamics induced by s, that of $C(s)$, and the dynamics induced by r, that of $C(r)$, are identical if and only if $inv(s) = inv(r)$.

(ii) If $inv(s) \neq inv(r)$, then the dynamics induced by s and by r have no attractor of period $p > 1$ in common.

(iii) If $|inv(s)| = k$, then for any $p \in \mathbb{N}$, $C(s)$ has as many attractors of period p than any Boolean automata circuit of size $n - k$, of same sign as C and updated with the parallel update schedule.

Proof: *(i)* follows directly from theorem 1 of [5] and *(iii)* is derived from the description of the structure of $\mathbb{C}_n^s$ made in the previous paragraph. To prove *(ii)*, suppose that $(i, i+1) \in inv(r) \setminus inv(s)$ and that there exists $x = x^s(t) = x^r(t) \in \{0,1\}^n$ such that $x^s(t+1) = x^r(t+1)$. Then:

$$x^s_{i+1}(t+2) = f_{i+1}(x^s_i(t+2)) = F[i+1, i^*+1](x^s_{i^*}(t+1))$$

and

$$x^r_{i+1}(t+2) = f_{i+1}(x^r_i(t+1)) = f_{i+1}(x^s_i(t+1)) = F[i+1, i^*+1](x^s_{i^*}(t))$$

where $i^* = max\{k < i \mid s(k) \geq s(k+1)\}$ (as above). By the injectivity of $F[i+1, i^*+1]$, this implies that if $x^s(t+2) = x^r(t+2)$ then $x_{i^*}(t+1) = x_{i^*}(t)$. Now, if x belongs to an attractor that is induced identically by both s and r, then $\forall t \in \mathbb{N},\ x^s(t) = x^r(t)$. As result, in this case, $\forall t \in \mathbb{N},\ x^s_{i^*}(t+1) = x^r_{i^*}(t) = x^s_{i^*}(t)$ (*i.e.,* the state of node i^* is fixed in the attractor). As one can check this leads to states of all nodes being fixed in the attractor which therfore is a fixed point. □

In relation with point *(ii)* of Proposition 3.1 above, recall that if the dynamics of a network has fixed points for a certain update schedule, then it has the same fixed points for every other update schedule. The important consequence of point *(iii)* of Proposition 3.1 is that from the results in [7] concerning the number of attractors of Boolean automata circuits updated in parallel, we may derive the number of attractors of each period and in total of any Boolean automata circuit updated with any block-sequential update schedule:

Corollary 3.1 *Let $C = (\mathbb{C}_n, \mathcal{F})$ be a Boolean automata circuit of size n and s a block-sequential update schedule of C such that $|inv(s)| = k$:*

- *If C is positive, then the total number of attractors in the dynamics of $C(s)$ is given by $\mathtt{T}^+_p$ below. For any integer p, the number of attractors of period p is either 0 if p does not divide $n-k$ or it is $\mathtt{A}^+_p$:*

$$\mathtt{T}^+_p \ = \ \frac{1}{n-k} \cdot \sum_{p \mid n-k} \psi(\frac{n-k}{p}) \cdot 2^p, \qquad \mathtt{A}^+_p \ = \ \frac{1}{p} \cdot \sum_{d \mid p} \mu(\frac{p}{d}) \cdot 2^d.$$

- *If C is negative, then the total number of attractors in the dynamics of $C(s)$ is given by $\mathtt{T}^-_p$ below. For any integer p, the number of attractors of period p is either 0 if $n-k$ cannot be written $n-k = q \times \frac{p}{2}$ where $q \in \mathbb{N}$ is odd, or it is $\mathtt{A}^-_p$:*

$$\mathtt{T}^-_p \ = \ \frac{1}{2n} \cdot \sum_{odd\ p \mid n} \psi(\frac{n}{p}) \cdot 2^p, \qquad \mathtt{A}^-_p \ = \ \frac{1}{p} \cdot \sum_{odd\ d \mid \frac{p}{2}} \mu(d) \cdot 2^{\frac{p}{2d}}.$$

Above, μ is the Möbius (see [9, 1]) function and ψ the Euler totient function.

Following Proposition 3.1, we define the equivalence relation between update schedules that relates r and s if and only if $inv(s) = inv(r)$. $[s]$ denotes the equivalence class of s for this relation. Proposition 3.2 below sums up some results concerning this relation:

Proposition 3.2 *Let $C = (\mathbb{C}_n, \mathcal{F})$ be a Boolean automata circuit of size n.*

(i) The total number of distinct dynamics induced by the different update schedules of C is $\sum_{k=0}^{n-1} \binom{n}{k} = 2^n - 1$.

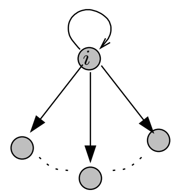


Fig. 3: Interaction graph relative to one of the n equivalence classes of update schedules that have $n-1$ inversions. Each one of these classes is characterised by the unique node $i \in \mathbb{Z}/n\mathbb{Z}$ that is such that $(i, i+1)$ is *not* an inversion and contains exactly one update schedule which is sequential, namely, the update schedule $s_i \equiv (i+1)(i+2)\dots(i-1)(i)$ such that $inv(s_i) = \{(j, j+1),\ j \neq i\}$. Because there is a loop over node i in this graph, the dynamics of $C(s_i)$ contains only fixed points if $\mathbb{C}_n$ is positive and only attractors of period 2 if $\mathbb{C}_n$ is negative.

(ii) *In every class $[s]$, $s \neq \pi$, there exists a sequential update schedule. Given the set of inversions of the class, a sequential update schedule can be constructed effectively in $\mathcal{O}(n)$ steps.*

(iii) *Given a set of $p > 1$ configurations of C, $\mathcal{A} = \{x(0), \dots, x(p-1)\}$, we can determine in $\mathcal{O}(p \cdot n)$ steps whether there exists a block-sequential update schedule s such that $C(s)$ has $\mathcal{A}$ as an attractor of period p. If such an update schedule exists, with Algorithm 1 below, in $\mathcal{O}(p \cdot n)$ steps, we can compute its set of inversions as well as a sequential update schedule inducing the same dynamics.*

Algorithm 1: Finding a sequential update schedule that induces a particular attractor of a given Boolean automata circuit if it exists

Input: $C = (\mathbb{C}_n, \mathcal{F})$ and $\mathcal{A} = \{x(0), \dots, x(p-1)\}$.

begin

In $\mathcal{O}(p \cdot n)$ steps, compute the set $\mathcal{A}^\pi = \{y(t) = F_\pi(x(t-1)) \mid 0 \leq t < p\}$;

In $\mathcal{O}(p \cdot n)$, compute the set $inv = \{(i-1, i) \mid \exists t \leq p,\ \ x_i(t) \neq y_i(t)\}$;

In $\mathcal{O}(n)$ steps, compute a sequential update schedule s using the set inv;

In $\mathcal{O}(p \cdot n)$ steps, compute the set $\mathcal{A}_s = \{F_s^t(x(0)) = x^s(t) \mid 0 \leq t < p\}$ and check that $\mathcal{A}_s = \mathcal{A}$. If not, then no update schedule induces $\mathcal{A}$ as an attractor;

Otherwise, output s.

end

Proof: Point *(i)* of Proposition 3.2 above is a direct consequence of points *(i)* and *(ii)* of Proposition 3.1 and of the fact that the number of distinct equivalence classes of update schedules with k inversions is $\binom{n}{k}$ (*i.e.,* the number of different sets of k inversions).

To prove Point *(ii)*, let us show that for every set of $k < n$ inversions, there exists a sequential update schedule s that satisfies exactly these k inversions. Thus, let inv be a set of $|inv| = k$ inversions and let $G = (V, A)$ be the acyclic digraph whose set of nodes is that of $\mathbb{C}_n$ (*i.e.,* $V = \{0, \ldots, n-1\}$) and whose set of arcs is $A = \{(i, i+1) \notin inv\} \cup \{(i+1, i) \mid (i, i+1) \in inv\}$ (in other words, G is obtained by inverting all arcs of $\mathbb{C}_n$ that belong to inv). Then, any sequential update schedule s whose set of inversions is inv satisfies the following:

$$\forall (i, j) \in A,\ s(i) > s(i)$$

so that such a sequential update schedule s can be obtained in linear time using a topological ordering algorithm on digraph G.

Finally, to prove Point *(iii)* and Algorithm 1, suppose that s is an existing block-sequential update schedule that induces $\mathcal{A}$ as an attractor, *i.e.,* $\forall t < p,\ x^s(t+1) = f_i^s(x(t)) = x(t+1)$. Let us show that its set of inversions $inv(s)$ is necessarily equal to inv. Suppose that $(i-1, i) \notin inv(s)$. Then, $\forall t < p,\ x_i(t+1) = x_i^s(t+1) = f_i^s(x(t)) = f_i(x_{i-1}(t)) = y_i(t+1)$ and consequently, $(i-1, i) \notin inv$. Now, $\forall i \in \mathbb{Z}/n\mathbb{Z}$, again, let $i^* = max\{j < i,\ (i^*, i^*+1) \notin inv(s)\}$. It is easy to prove that the state of any node j such that $\exists i,\ j = i^*$ necessarily changes in all attractors induced by s and in particular in $\mathcal{A}$. Suppose that $(i-1, i) \in inv(s)$. Let $T < p$ be such that $x_{i^*}(T) \neq x_{i^*}(T+1)$. Then, the following holds:

$$x_i(T+2) = x_i^s(T+2) = F[i, i^*+1](x_{i^*}(T+1)) \text{ and}$$
$$y_i(T+2) = f_i(x_{i-1}(T+1)) = f_i(x_{i-1}^s(T+1)) = f_i \circ F[i-1, i^*+1](x_{i^*}(T))$$

so that $x_i(T+2) \neq y_i(T+2)$ and consequently $(i-1, i) \in inv$. □

4 Conclusion

Following the work presented in this paper, we believe that most combinatoric problems concerning the dynamics Boolean automata circuits updated with block-sequential update schedules have now been dealt with. We know the exact value of both the total number of attractors and the number of attractors of period $p,\ \forall p \in \mathbb{N}$, in the dynamics of positive and negative Boolean automata circuits of any size updated with the synchronous, sequential and the block-sequential update schedules. We also know how many different dynamics can be induced by the set of block-sequential update schedules of a Boolean automata circuit.

One important question, however, remains unanswered: "What are the sizes of the equivalence classes of block-sequential update schedules that yield the same dynamics?". For the very particular cases of $[\pi]$ and of the classes of update schedules with $n-1$ inversions (where n is the size of the circuit) we know that the size of the classes is 1. We also obtained a very intricate formula for the size of classes of update schedules having *consecutive* inversions only. It implies that the sizes of such classes is exponential as may certainly be that of many other classes. One motive (amongst others) for studying this question follows from A. Elena's work. In his PhD thesis [8], Elena computed statistics of the number of attractors of threshold Boolean automata networks as well as of their periods averaging over all networks (of sizes between 3 and 6) and all update schedules. For both, he found particularly small values. Now, as we have already mentioned, it is known that underlying circuits play an important role in the dynamics of a network with an arbitrary structure. Knowing the answer to this question would help us to understand better the averages found by Elena.

Therefore, beyond this question, we believe that there are two obvious extensions needed of our combinatoric analysis of the dynamics of circuits: one towards more general networks, that is, networks with arbitrary underlying interaction graphs. In line, with [4], this would need to relate the dynamics of arbitrary networks with that of there embedded circuits. The second extension needed is in the direction of other update schedules. Although understanding the dynamics of networks under block-sequential update schedules is a first notable step, these update schedules remain rather unadapted to the modelisation of biological networks. One may indeed argue that it is rather unrealistic that a network updates infallibly every one of its nodes exactly once and according to the exact same order at every time step. It seems more likely, that, on the contrary, some nodes may be updated more often than others and that the updating of nodes may depend on some parameters in a way that cannot be translated by giving an *order* of update as do block-sequential update schedules.

Acknowledgements

We thank the Basal project-CMM and the Fondecyt 1100003.

References

[1] T. M. Apostol. *Introduction to analytic number theory*. Springer-Verlag, 1976.

[2] J. Aracena, J. Demongeot, and E. Goles. Fixed points and maximal independent sets in and-or networks. *Discrete Applied Mathematics*, 138:277–288, 2004.

[3] J. Aracena, J. Demongeot, and E. Goles. On limit cycles of monotone functions with symmetric connection graph. *Theoretical Computer Science*, 322:237–244, 2004.

[4] J. Aracena, J. Demongeot, and E. Goles. Positive and negative circuits in discrete neural networks. *IEEE Transactions on Neural Networks*, 15:77–83, 2004.

[5] J. Aracena, E. Goles, A. Moreira, and L. Salinas. On the robustness of update schedules in boolean networks. *Biosystems*, 97, 2009.

[6] J. Demongeot, A. Elena, and S. Sené. Robustness in regulatory networks: a multi-disciplinary approach. *Acta Biotheoretica*, 56(1-2):27–49, 2008.

[7] J. Demongeot, M. Noual, and S. Sené. On the number of attractors of boolean automata circuits. WAINA, Perth, Australia, 2010. IEEE Press. To appear.

[8] A. Elena. *Robustesse des réseaux d'automates booléens à seuil aux modes d'itération. Application à la modélisation des réseaux de régulation génétique*. PhD thesis, Université Joseph Fourier - Grenoble, 2009.

[9] C. F. Gauss and A. A. (tr.) Clarke. *Disquisitiones Arithemeticae*. Yale University Press, 1965.

[10] D. Hilbert. Mathematical problems. *Bulletin of the American Mathematical Society*, 8:437–479, 1902.

[11] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Journal of Mathematical Biology*, 5:115–133, 1943.

[12] M. Noual. On the dynamics of two particular classes of boolean automata networks: Boolean automata circuits and or networks. Technical report, TIMC-IMAG (Grenoble) and CMM (Santiago de Chile), 2009.

[13] A. Richard and J.P. Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Applied Mathematics*, 155(18):2403–2413, 2007.

[14] F. Robert. *Discrete Iterations*. Springer-Verlag, 1986.

[15] R. Thomas. On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. *Springer Series in Synergetics*, 9:180–193, 1981.