Nonlinear Dynamics and Systems Theory, **11** (3) (2011) 275–284

Exponentially Long Orbits in Boolean Networks with Exclusively Positive Interactions

W. Just [1]* and G.A. Enciso [2]

[1] *Department of Mathematics, Ohio University, Athens, OH 45701, USA*
[2] *Mathematics Department, University of California, Irvine, CA 92617 USA*

Received: May 5, 2011; Revised: July 28, 2011

Abstract: The absence of negative feedback in Boolean networks tends to result in systems with relatively short orbits. We present a construction of N-dimensional Boolean networks that use only AND, OR, COPY gates and nevertheless have an exponentially large orbit (of size c^N for arbitrary $c < 2$). The construction is based on pseudorandom number generation algorithms. A previously obtained nontrivial upper bound on the orbit length under certain limitations on the number of outputs per node is shown to be optimal.

Keywords: *Boolean networks; monotone systems; gene networks; systems biology.*

Mathematics Subject Classification (2000): 06E99, 34C12, 39A33, 92B99, 94C10.

1 Introduction

The concept of a *Boolean network* was originally proposed in the late 1960's by Stuart Kauffman to model gene regulatory behavior at the cell level [13]. This type of modeling can sometimes capture the general dynamics of continuous systems in a simplified framework without the choice of specific nonlinearities or parameter values; see for instance [1]. Boolean networks are used in several other disciplines such as electrical engineering, computer science, and control theory, and analogous definitions are known under various names such as sequential dynamical systems [16] or Boolean difference equations [6].

An N-dimensional *Boolean dynamical system* or *Boolean network* (Π, g) consists of N variables $s_1, \ldots, s_N$, each of which can have value 0 or 1 at any given time step t. The variables are updated according to $s_i(t+1) = g_i(s_1(t), \ldots, s_N(t))$.

* Corresponding author: mailto:mathjust@gmail.com

Usually, individual update functions g_i depend only on very few of the variables. Let us say that *s_j is an input of s_i* and *s_j sends output to g_i* if there are two Boolean vectors s, s^* that differ only in variable s_j for which $g_i(s) \neq g_i(s^*)$. The input-output relation defines a digraph on the set of Boolean variables that is called the *connectivity* of the Boolean network. We call a Boolean network (Π, g) *a K, M network* if each of its update functions takes at most K inputs and each variable has at most M outputs.

A key problem in the study of Boolean networks is how the dynamics depends on the updating functions and the connectivity. This problem has been largely studied for so-called *random Boolean networks (RBNs)* where both the updating functions and the network connectivity are randomly drawn from a specified network distribution. This allows to make estimates on quantities such as the number of orbits and their length [2, 7, 14, 22]. Such estimates can be either obtained from simulation studies or analytically.

1.1 Cooperative Boolean networks

An important topic in the study of dynamical systems is the role of negative feedback. This notion is usually defined in terms of *negative feedback loops* that contain an odd number of negative interactions. *Monotone systems* are systems that contain only positive feedback loops. Here we study *cooperative systems,* that are systems in which there are no negative interactions between any two variables. In the context of networks with at most $K = 2$ inputs per variable, cooperativity is equivalent to the use of only the following update functions: constants, COPY, AND, OR. No negations are allowed [11].

Comparative simulation studies show that Boolean networks with no or only few negative feedback loops tend to have relatively shorter orbits [26]. The question naturally arises whether the assumption of cooperativity imposes nontrivial provable upper bounds on the length of orbits in Boolean networks. This question seems especially interesting for cooperative $K = 2$ Boolean networks, since $K = 2$ random Boolean networks tend to have much shorter orbits than RBNs drawn from distributions where $K > 2$ [2, 7, 14].

Since the state space of an N-dimensional Boolean network has size 2^N, a bound on the attractor length should be considered "nontrivial" if it scales like $o(2^N)$. In [15, 20], a simple example of a $K = 2$, $M = 2$ Boolean network is constructed with N variables and an orbit of length $2^{N-1} - 1$, which comprises exactly half of the state space. In contrast, the orbits of cooperative Boolean systems cannot comprise a fixed fraction of the state space [4, 11, 25], which already gives an upper bound that scales like $o(2^N)$. Upper bounds on orbit length that scale like $O(c^N)$ for some $c < 2$ were derived in [12] for $K = 2$, $M = 2$ cooperative networks under the assumption that at least a fixed positive fraction of update functions take exactly two inputs (see Theorem 3.1 below).

Theorem 1 of the preprint [12] shows that for any constant $1 < c < 2$ and all sufficiently large N there exists an N-dimensional $K = 2$, $M = 2$ cooperative Boolean network with at least one orbit of length $\geq c^N$. This shows that some such additional assumptions are needed for nontrivial upper bounds of type $O(c^N)$ on orbit length. Our main theorem here, Theorem 2.1, consists of a simplified proof of a similar result with a mild additional assumption as described below. Theorem 3.2 of Section 3 provides variations on Theorem 2.1 and shows, among other things, that one of the upper bounds on orbit length that was derived in [12] is sharp.

1.2 Additive Lagged-Fubini Generators

An additive lagged-Fubini generator (ALFG) is an algorithm to produce ‘pseudorandom’ numbers. It is the basis for some of the most widely used random number generators such as the Mersenne twister [19]. In its Boolean version, the ALFG consists simply of a sequence of Boolean values x_i satisfying the formula

$$x_i = x_{i-p} + x_{i-q} \bmod 2, \tag{1}$$

for all i, where $0 < q < p$ are fixed numbers. For particular choices of p and q, it has been shown that x_i can be periodic with maximal period $2^p - 1$ [17]. The sufficient and necessary condition for producing this orbit length is that the polynomial $x^p + x^q + 1$ is primitive modulo 2 [8]. Many such pairs (p, q) are known, including ones with values as large as $p = 6972593$ and $q = 3037958$. It is an open but widely believed conjecture that there are infinitely many such pairs, see for instance [8]. For a list of all admissible pairs (p, q) with $p \leq 1000$, see [27].

In [15, 20], the authors build a Boolean network which is easily seen to be equivalent to an ALFG, using a single loop of length p, and a single internal connection between two nodes q variables apart, as in Figure 1a. All update functions are equal to the simple copy function $f(a) = a$, except for one with two inputs, $f(a, b) = a + b \bmod 2 = a$ XOR b. The authors of these papers also point out that this reversible update function (Table 1) can be replaced by three canalyzing ones (see Section 3 for the definition of canalyzing functions). Yet even such an implementation would contain negations and its main feedback loop is negative. In particular, this network is not cooperative.

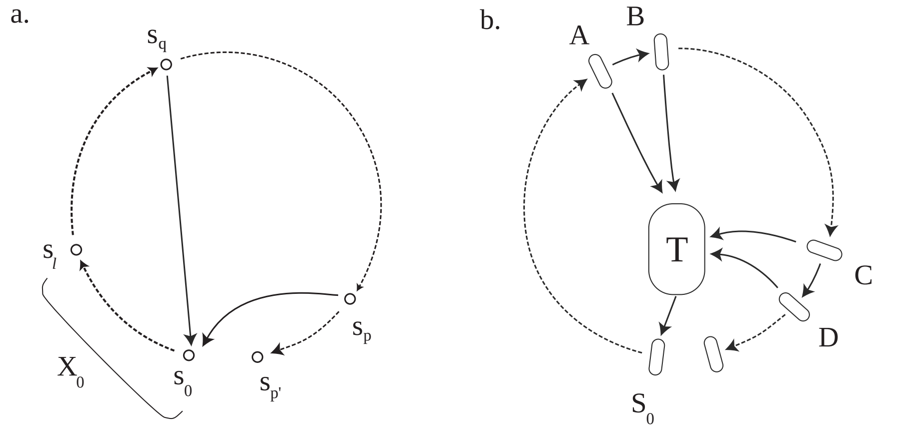

Figure 1: a. The original additive lagged-Fibonacci generator. A solid arrow represents a direct connection, and a dotted arrow is a chain of connected variables. For the purposes of our results, the variables are grouped into blocks of size ℓ such as shown for X_0, and additional nodes are added outside the main loop so that the total number of nodes is divisible by ℓ (see the dotted line joining x_s and $x_{s'}$). b. The cooperative network associated to a., where every variable corresponds to L different Boolean nodes. The inputs of the Boolean circuit T are $A = S_{\lfloor q/\ell \rfloor - m - 1}$, $B = S_{\lfloor q/\ell \rfloor - m}$, $C = S_{\lfloor p/\ell \rfloor - m - 1}$, $D = S_{\lfloor p/\ell \rfloor - m}$.

2 Cooperative Boolean Networks with an Exponentially Long Orbit

Let us state the version of Theorem 1 of [12] that we are going to prove here.

Theorem 2.1 *Assume that there exist infinitely many positive integers $p > q$ such that (1) defines an AFLG with maximal period $2^p - 1$. Let $1 < c < 2$ be an arbitrary constant. Then for infinitely many N there exists an N-dimensional $K = 2$, $M = 2$ cooperative Boolean network with at least one orbit of length $\geq c^N$. In this network the only update functions are $s_i \vee s_j$, $s_i \wedge s_j$, and the copy function $f(s_i) = s_i$.*

Proof We start with an ALFG with maximal period $2^p - 1$ as in Figure 1a, with sufficiently large delays (p, q). In the Boolean setting we have the following equation:

$$s_0(t+1) = s_q(t) \text{ XOR } s_p(t) = s_0(t-q) \text{ XOR } s_0(t-p). \tag{2}$$

This network will be referred to as *the AFLG network.* It has an orbit with length $2^p - 1$ and two negative feedback loops as shown in Figure 1a.

For a given $c < 2$ we construct a Boolean network with N variables that uses only update functions AND, OR, COPY and has an orbit of length $\geq c^N$ states. Its dynamics closely mimics the one of the AFLG network. The idea is to group the variables $s_0, \dots s_p$ into blocks of ℓ adjacent variables each, as in Figure 1b. The new network has as variables Boolean *vectors* $S_0, S_1, \dots$, each with an even number L of bits. The values of L and ℓ will be chosen as in Lemma 2.2 below. Importantly, the values of each S_i are not arbitrary but chosen from the image of an injective function $\Gamma : \{0,1\}^\ell \to \{0,1\}^L$, i.e. they are thought of as coded sequences of ℓ bits. Additionally, the values of $\Gamma(s)$ are required to have exactly $L/2$ nonzero entries. Such a function Γ exists as long as $2^\ell \leq \binom{L}{L/2}$, which will be ensured by Lemma 2.2. In order to guarantee that the nodes in the ALFG can be divided in groups of ℓ, we introduce some additional nodes as in Figure 1a, which are however not part of the loop (see the legend of Figure 1 for details).

Notice that after ℓ time steps, the ALFG network has rotated the values of each group of ℓ variables into the next – except for the group $s_0, \dots s_{\ell-1}$, whose values have been determined by a more complicated algorithm. The idea is that one time step in the new network will represent ℓ time steps in the ALFG. More precisely, each variable $S_i(t)$ is coding for the variables $X_i(\ell t) = (s_{i\ell}(\ell t), \dots, s_{i\ell+\ell-1}(\ell t))$ at time ℓt, or $S_i(t) = \Gamma(X_i(\ell t))$.

In order to achieve this, we set $S_i(t+1) = S_{i-1}(t)$ for $i > 0$. As for the updating function of S_0, it is defined as the encoding Γ of $s_0, \dots, s_\ell$ after iterating the ALFG for ℓ time steps. $S_0(t+1)$ is therefore a function G of the variables S_i encoding $s_{q-\ell+1}, \dots, s_q$ and $s_{p-\ell+1}, \dots, s_p$. In other words, given the arguments A, B, C, and D, one can compute S_0 at the next time step by first decoding them into their corresponding sequences of ℓ-vectors using the function Γ^{-1}, and assigning these values to the variables in ALFG, then iterating ALFG ℓ times, and finally encoding the resulting sequence X_0 using the function Γ.

The following technical lemma shows that this encoding function G can be implemented as a Boolean circuit (labeled T in Figure 1b) with only binary AND- and OR- and unary COPY gates and no negations. Such a Boolean circuit can be incorporated into our network without violating cooperativity or the commitment to build a $K = 2$, $M = 2$ network. The indegree and outdegree for a node of a Boolean circuits are defined analogously as for Boolean networks.

Our construction uses the fact that G is only used with arguments that have $L/2$ zeros and $L/2$ ones each. Given two Boolean $P-$vectors s, r, we say that $s \leq r$ if $s_i \leq r_i$

for all i. If either $s \leq r$ or $r \leq s$, the two vectors are called *comparable.* The following lemma will be applied to the proof of the main result for $P = 4L$.

Lemma 2.1 *Let $g : D \subseteq \{0,1\}^P \to \{0,1\}^L$ be an arbitrary function, defined on a domain D where no two elements are comparable. Then there exists a Boolean circuit B with input vector a of dimension P, and an output vector $b = (b_1, \dots, b_L)$, such that $b(t+m) = g(a)$, for some fixed delay m and any $a(t) \in D$. Furthermore, the circuit B uses only binary AND- and OR- and unary COPY gates and the indegree (outdegree) of every designated input (output) variable is 0.*

Proof The function g can be extended to a cooperative function h, i.e. one for which $s \leq r$ implies $h(s) \leq h(r)$, defined on all of $\{0,1\}^P$; see [11]. The result will follow from building a suitable Boolean circuit that computes the function h.

Consider a fixed component $h_i : \{0,1\}^P \to \{0,1\}$ of h. By the cooperativity of this function, one can write it in the normal form $h_i(s_1, \dots, s_P) = \Psi_1^i(s_1, \dots, s_P) \vee \dots \vee \Psi_{k_i}^i(s_1, \dots, s_P)$, where each Ψ_j^i is the conjunction of a number of variables, i.e., $\Psi_j^i(s_1, \dots, s_P) = s_{\alpha_{1i}} \wedge \dots \wedge s_{\alpha_{ji}}$. This suggests a way of computing h_i: define Boolean variables $\psi_j^i(t) := \Psi_j^i(s(t-1))$, and then let $h_i(t) := \psi_1^i(t-1) \vee \dots \vee \psi_{k_i}^i(t-1)$. Repeating this procedure for all components of h yields a Boolean circuit which computes h in $m = 2$ steps, and which is cooperative and has indegree (outdegree) 0 for every input (output).

In order to satisfy the condition that every node have in- and outdegree of at most 2, we need to modify this construction by introducing additional variables. First, note that the outdegree of every input s_i can be very large. One can define two additional variables which simply copy the value of $s_i(t)$, then four variables that copy the value of the previous two, etc. This procedure is repeated for each s_i so that at least as many copies of each variable are present as appear in the expressions of all ψ_j^i. A similar cascade can be used to define each ψ_j^i and h_i so that each indegree is at most two. If $\psi_i^j = s_{\alpha_1} \wedge s_{\alpha_2} \wedge s_{\alpha_3}$, say, then one can define $r_1(t) := s_{\alpha_1}(t-1)$, $r_2(t) := s_{\alpha_2}(t-1) \wedge s_{\alpha_3}(t-1)$, $\psi_i^j(t) := r_1(t-1) \wedge r_2(t-1)$. Similarly for longer disjunctions and each ψ_j^i and also similarly for h_i, in which case $\wedge$ is replaced by $\vee$ at each step. This produces a computation of h_i in m_i steps for each i. Finally, after introducing further additional variables at each component i if necessary to compensate for unequal lengths of the expressions for ψ_j^i, the Boolean vector $h(s_1, \dots, s_P)$ can be computed in exactly $m = \max(m_1, \dots, m_L)$ steps.

Notice that the function G is not computed by the Boolean circuit instantaneously, but after m steps. Since $S_i(t) = S_{i-m}(t-m)$, we correct for this by feeding the circuit an input which has been shifted back by m.

The new cooperative Boolean network has an orbit of length at least $(2^p - 1)/\ell$. Its dimension is

$$N = (p + \gamma + 1)L/\ell + T, \tag{3}$$

where $\gamma = p' - p < \ell$ reflects the need for dummy variables (see the legend of Figure 1) and T is the number of nodes involved in the Boolean circuit that computes G. Since T only depends on ℓ and L, not on p, the following lemma implies Theorem 2.1.

Lemma 2.2 *For arbitrary $1 < c < 2$ and sufficiently large p, there exist ℓ, N, and L as above such that*

$$\binom{L}{L/2} > 2^\ell, \quad \frac{2^p - 1}{\ell} > c^N.$$

Proof We prove first that there exist $L > 0$ even, an integer $\ell > 0$, and a real constant $\delta > 1$ such that

$$\binom{L}{L/2} > 2^{\ell} > c^{\delta L}. \tag{4}$$

Start by fixing an arbitrary $\delta > 1$ such that $c < c^{\delta} < 2$. The second inequality in (4) is equivalent to $L/\ell < \ln 2/(\delta \ln c)$. Let L be an even integer with $L = w\ell$, for some fixed $1 < w < \ln 2/(\delta \ln c)$. Since $\binom{L}{L/2} > \frac{2^L}{L+1} = \frac{2^{w\ell}}{w\ell+1}$ and $w > 1$, the first inequality in (4) is satisfied for sufficiently large ℓ.

It remains to show that $(2^p - 1)/\ell \geq c^N$ for some sufficiently large N as in (3). But since $c^{L/\ell} < 2^{1/\delta}$, expression (3) implies

$$\frac{c^{(p+\gamma+1)L/\ell+T}}{(2^p-1)/\ell} < \ell c^T \frac{2^{\frac{p+\gamma+1}{\delta}}}{2^p - 1} < \ell c^T 2^{\frac{1}{\delta}(p+\gamma+1)-(p-1)}$$

which is < 1 for sufficiently large p.

It follows that for sufficiently large p, we can choose ℓ, L so that the system we constructed will contain an orbit of length at least c^N, as stated in Theorem 2.1.

3 Biased Update Functions and Long Orbits

The *bias* Λ of a Boolean function is the fraction of input vectors for which the function outputs 1. A Boolean function would be considered *biased* if $\Lambda \neq 0.5$. More specifically, let us say that an update function is *ε-biased* if $|\Lambda - 0.5| \geq \varepsilon$. Simulation studies of random Boolean networks indicate that networks with only strongly biased update functions tend to have shorter orbits than generic networks with a given number of inputs (see [23] and references therein). The following result from [12] gives a provable upper bound on the length of orbits in some ε-biased networks.

Theorem 3.1 *Let $\varepsilon, \alpha > 0$ and let K, M be positive integers. Then there exists a positive constant $c(\varepsilon, \alpha, K, M) < 2$ such that for all $c > c(\varepsilon, \alpha, K, M)$ and all sufficiently large N, the length of any orbit in any N-dimensional K-M Boolean network in which a proportion of least α of the update functions are ε-biased does not exceed c^N.*

In particular, $c(0.25, 1, 2, 2) \leq 10^{1/4}$.

To put the last sentence of Theorem 3.1 into perspective, consider Table 1 of Boolean functions with two inputs.

Thus for $K = 2$ a Boolean function is biased iff it is 0.25-biased iff it is in $\mathcal{C}_2 \cup \mathcal{F}$. The classes $\mathcal{C}_1$ and $\mathcal{C}_2$ constitute the *canalyzing* functions, in which a certain value of one of the inputs determines the function output [7]. Note the $\mathcal{C}_2$ contains the AND and the OR functions. We call a $K = 2$ Boolean system a *$\mathcal{C}_2$-network* if all its update functions are in the class $\mathcal{C}_2$. Notice that the last sentence of Theorem 3.1 gives an upper bound of $O(10^{N/4})$ for orbit lengths in N-dimensional $K = 2, M = 2$ $\mathcal{C}_2$-networks.

We can prove the following variants of Theorem 2.1 for $\mathcal{C}_2$-networks. Part (b) of Theorem 3.2 implies that the upper bound in the last sentence of Theorem 3.1 is sharp. The theorem does not require assumptions about the existence of AFLGs [12], but we state and prove it here in this form to emphasize the connection with the earlier construction.

In	$\mathcal{F}$	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{R}$
0 0	1 0	0 1 0 1	1 0 0 0 0 1 1 1	1 0
0 1	1 0	0 1 1 0	0 1 0 0 1 0 1 1	0 1
1 0	1 0	1 0 0 1	0 0 1 0 1 1 0 1	0 1
1 1	1 0	1 0 1 0	0 0 0 1 1 1 1 0	1 0
cooperative	* *	* *	* *	
bias	1 0	$\frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}\ \frac{1}{2}$	$\frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}\ \frac{1}{4}\ \frac{3}{4}\ \frac{3}{4}\ \frac{3}{4}\ \frac{3}{4}$	$\frac{1}{2}\ \frac{1}{2}$

Table 1: The different Boolean update functions with $K = 2$ inputs (adapted from [7]).

s	1111	1110	1101	1100	1011	1010	0111	0101	0011	0000
$f(s)$	1111	1100	1010	1000	0101	0100	0011	0010	0001	0000
$h \circ f(s)$	1111	1110	1101	1100	1011	1010	0111	0101	0011	0000

Table 2: The values of f and $h \circ f$ on F.

Theorem 3.2 *Assume that there exist infinitely many positive integers $p > q$ such that (1) defines an AFLG with maximal period $2^p - 1$. Let c, c_1 be constants with $1 < c < 2$ and $1 < c_1 < 10^{1/4}$. Then for arbitrarily large N there exist cooperative Boolean networks with the following properties:*
(a) (Π, g) is a $\mathcal{C}_2$ network with at least one orbit of length $\geq c^N$,
(b) (Π, g) is a $\mathcal{C}_2$, $M = 2$ network with at least one orbit of length $\geq c_1^N$.

Proof For part (a), let (Σ, f) be a cooperative $K = 2$ Boolean network of dimension $N - 2$ that contains an orbit of length c^N. We show how to turn (Σ, f) into a cooperative $\mathcal{C}_2$ Boolean network (Π, g) of dimension N. The update functions g_k for $k < N - 1$ of the new system are defined as follows:

If f_k is already in $\mathcal{C}_2$, then $g_k = f_k$.

If $f_k = s_{i_k}$, then $g_k = s_{i_k} \wedge s_{N-1}$.

If f_k is constant with value 1, then $g_k = s_{N-1} \vee s_N$; if f_k is constant with value 0, then $g_k = s_{N-1} \wedge s_N$.

Finally, we let $g_{N-1} = s_{N-1} \vee s_N$ and $g_N = s_{N-1} \wedge s_N$. Then (Π, g) is a cooperative $\mathcal{C}_2$-system. Now let $s \in \Sigma$ be a state in an orbit of length at least c^N of (Σ, f), and define a state $s^* \in \Pi$ by $s^* = [s_1, \dots, s_{N-2}, 1, 0]$. Then the orbit of s^* in (Π, g) has the same length as the orbit of s in (Σ, f). This proves part (a).

For the proof of part (b), we need to implement the $\mathcal{C}_1$ functions that copy the value of one input variable by cooperative $\mathcal{C}_2$ functions in such a way that the overall dimension is not increased by more than a factor of $4 \log_{10} 2$.

Let us define Boolean vector functions f and h on four-dimensional Boolean vectors $s = (s_1, s_2, s_3, s_4)$ as follows:

$$f(s) = (s_1 \wedge s_2, s_1 \wedge s_3, s_2 \wedge s_4, s_3 \wedge s_4), \quad h(s) = (s_1 \vee s_2, s_1 \vee s_3, s_2 \vee s_4, s_3 \vee s_4).$$

Let $F = \{1111, 1110, 1101, 1100, 1011, 1010, 0111, 0101, 0011, 0000\}$ and $H = f(F)$. Table 2 gives the values of $f, h \circ f$ on F.

As Table 2 shows, $h \circ f$ is the identity on F. It follows that h maps H onto F and $f \circ h$ is the identity on H.

Let L be a positive integer divisible by eight, and let $p := L/4$. Write $[L]$ as a disjoint union of blocks of four consecutive integers $i(1,r), i(2,r), i(3,r), i(4,r)$ for $r \in [p]$. Let $\vec{s}_r := (s_{i(1,r)}, s_{i(2,r)}, s_{i(3,r)}, s_{i(4,r)})$. Call a Boolean vector $s \in \{0,1\}^{[L]}$ *L-compliant* if

(a) $\vec{s}_r \in F$ for $1 \leq r \leq p/2$, $\vec{s}_r \in H$ for $p/2 < r \leq p$, and

(b) s takes the value 1 exactly $L/2$ times.

Lemma 3.1 *Let $c_1 < 10^{1/4}$. Then there exist a positive integer ℓ and a positive integer L that is a multiple of eight such that $2^\ell > c_1^L$ and the number of L-compliant Boolean vectors is larger than 2^ℓ.*

Proof Let L be a positive integer that is an integer multiple of 8, and let V be the set of Boolean vectors $s \in \{0,1\}^L$ that satisfies condition (a) above. Since $|F| = |H| = 10$, it is clear that $|V| = 10^{L/4}$.

Let $|\vec{s}_r|$ denote the number of 1's in $\vec{s}_r$. For each $s \in V$ define the *signature of s* as $\sigma(s) = (\sigma_1(s), \ldots, \sigma_6(s))$, where
$\sigma_1(s) = |\{r : \ r \leq p/2 \ \& \ |\vec{s}_r| = 4\}|$, and $\sigma_4(s) = |\{r : \ p/2 < r \ \& \ |\vec{s}_r| = 4\}|$,
$\sigma_2(s) = |\{r : \ r \leq p/2 \ \& \ |\vec{s}_r| = 0\}|$, and $\sigma_5(s) = |\{r : \ p/2 < r \ \& \ |\vec{s}_r| = 0\}|$,
$\sigma_3(s) = |\{r : \ r \leq p/2 \ \& \ |\vec{s}_r| = 3\}|$, and $\sigma_6(s) = |\{r : \ p/2 < r \ \& \ |\vec{s}_r| = 1\}|$.

Let $\sigma^{max} = (\frac{1}{16}, \frac{1}{16}, \frac{1}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{4})$. Then the inequality

$$|\{s \in V : \sigma(s) = \sigma\}| \leq |\{s \in V : \sigma(s) = \sigma^{max}\}| \tag{5}$$

for any possible signature σ. Moreover, observe that if $s \in V$ and $\sigma(s) = \sigma^{max}$, then s takes the value 1 exactly $L/2$ times, and hence s is L-compliant. Since the total number of possible signatures is bounded from above by $(L/4+1)^6$, it follows from (5) that the total number Q of L-compliant Boolean vectors satisfies the inequality

$$Q \geq \frac{10^{L/4}}{(L/4+1)^6}.$$

Notice that $\lim_{L\to\infty} \ L \ln 10^{1/4} - 6\ln(L/4+1) - L \ln c_1 = \infty$.
Thus for sufficiently large L we can find a positive integer ℓ with

$$L \ln 10^{1/4} - 6 \ln(L/4+1) > \ell \ln 2 > L \ln c_1,$$

and the lemma follows.

Now fix $c_1 < 10^{1/4}$ and let L, ℓ be as in Lemma 3.1. Build an N-dimensional Boolean system (Π, g^-) as in the proof of Theorem 2.1, but with the following modifications for indices i where the value of S_i will just be copied to S_{i+2}:

We require that the values of S_i on the blocks S_i of length L are L-compliant vectors.

Instead of requiring $S_i(t+1) = S_{i+1}(t)$ and implementing this dynamics by $\mathcal{C}_1$ functions, we only require $S_i(t+2) = S_{i+2}(t)$ and implement this dynamics as follows: Let S_i be partitioned into blocks $b_{i,1}, \ldots, b_{i,L/4}$ of four Boolean values each, with $b_{i,r}(t) \in F$ for $r \leq L/8$ and $b_{i,r} \in H$ for $L/8 < r \leq L/4$. Define $b_{i,r}(t+1) = h(b_{i+1,r+L/8}(t))$ for $r \leq L/8$ and $b_{i,r}(t+1) = f(b_{i+1,r-L/8}(t))$ for $L/8 < r \leq L/4$.

This construction is possible by Lemma 3.1 and the observations on the functions f, h we made above, and the exact same argument as in the proof of Theorem 2.1 shows that one can choose initial states of (Π, g^-) that belong to an orbit of length $\geq c_2^N$, where

c_2 is a constant that depends only on L and ℓ and satisfies $c_1 < c_2 < 10^{1/4}$. It is also straightforward to verify that the resulting system is cooperative.

However, the system may not yet be a $\mathcal{C}_2$ system with $M = 2$, since the loops where copying occurs do not need to be of even length. So to ensure that we end up with an $M = 2$ system we may have at most two leftover sets S_i where the copying of some s_j needs to be implemented by $s_j \cap s_j^*$ using a few dummy variables s_j^*. Since the number of these 'leftover variables' is at most $2L$, this can be done without increasing N too much so that the resulting orbit will still have length $> c_1^N$ (see Appendix A of [12].)

4 Conclusion

Monotone and cooperative systems have been used as a modeling tool for gene regulatory systems, e.g. in [3, 5]. In the absence of negative feedback *continuous systems* converge generically towards an equilibrium under mild regularity hypotheses; see the work by Hirsch, Smith, Enciso, Mazco, and others [9, 10, 18, 24]. These generic convergence results have been generalized to the case of continuous monotone *maps*, in which case the generic iteration converges towards a periodic solution, with upper bounds for the maximum period [21]. In contrast, our results show that even very stringent conditions on *Boolean systems* with no negative interactions do not preclude very long orbits.

Our reasons for presenting a new proof of Theorem 1 in [12] are two-fold. First of all, the original construction given in [12] is somewhat difficult to read. We hope that the much simpler proof presented here will make the result more accessible to the mathematical community and will make the basic ideas that are common to both constructions more clearly visible. Second, as in [15, 20], this construction is based on additive lagged-Fubini generators (ALFG), which are the basis for the most commonly used pseudo-random number generators that are ubiquitous in applications from computer science and engineering. We hope that the proof presented here will highlight some important connections between number theory, computer science, and the study of Boolean networks and their applications, including the study of gene regulatory networks.

Acknowledgment

We thank Eduardo Sontag for suggesting this research topic to us, Xiaoping A. Shen and Maciej Malicki for valuable comments, and the reviewer of another manuscript for bringing ALFG-based constructions to our attention.

References

[1] Albert, R. and Othmer, H.G. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *J. Theor. Biol.* **223** (2003) 1–18.

[2] Aldana, M., Coppersmith, S. and Kadanoff, L. P. In: *Perspectives and Problems in Nonlinear Science.* (E. Kaplan, J. E. Marsden and K. R. Sreenivasan eds.). Springer Verlag, New York, 2003, 23–90.

[3] Angeli, D., Ferrell, J.E. and Sontag, E.D. Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. *Proc. Natl. Acad. Sci.* **101** (2004) 1822–1827.

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

[5] de Leenheer, P., Levin, S.A., Sontag, E.D. and Klausmeier, C.A. Global stability in a chemostat with multiple nutrients. *J. Math. Biol.* **52** (2006) 419–438.

[6] Dee, D. and Ghil, M. Boolean difference equations, I: Formulation and dynamic behavior. *SIAM J. Appl. Math.* **44** (1984) 111–126.

[7] Drossel, B. Random Boolean Networks. In: *Reviews of nonlinear dynamics and complexity, Volume 1.* (H.G. Schuster, ed.) Wiley-VCH, Weinheim, 2008, 69–110.

[8] Golomb, S.W. *Shift Register Sequences - A Retrospective Account.* Lecture Notes in Computer Science 4086, Sequences and Their Applications, Springer, 2006, 1–4.

[9] Enciso, G., Hirsch, M. and Smith, H. Prevalent behavior of strongly order preserving semiflows. *J. Dyn. Diff. Eq.* **20** (1) (2008) 115–132

[10] Hirsch, M. Stability and convergence in strongly monotone dynamical systems. *Reine und Angew. Math* **383** (1988) 1–53.

[11] Just, W. and Enciso, G.A. Analogues of the Smale and Hirsch theorems for cooperative Boolean and other discrete systems. To appear in the *J. Diff. Eqs. Appl.*

[12] Just, W. and Enciso, G.A. Extremely Chaotic Boolean Networks. Preprint. arXiv:0811.0115v1 (2008).

[13] Kauffman, S.A. Homeostasis and differentiation in random genetic control networks, *Nature* **224** (1969) 177–178.

[14] Kauffman, S. A. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, Oxford, 1994.

[15] Kaufman, V. and Drossel, B. On the properties of cycles of simple Boolean networks. *Europ. Phys. J. B* **43** (1) (2005) 115–124.

[16] Garcia, L, Jarrah, A.S. and Laubenbacher, R. Classification of finite dynamical systems. arXiv:math/0112216 (2001).

[17] Marsaglia, G. and Tsay, L.-H. Matrices and the structure of random number sequences. *Linear Algebra and its Applications* **67** (1985) 147–156.

[18] Mazko, A.G. Positive and monotone systems in partially ordered space. *Ukr. Mat. Zh.* **55**(N2) (2003) 164–173 [Russian].

[19] Matsumoto, M. et al. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Mod. and Comp. Sim.* **8** (1998) 3–30.

[20] Paul, U., Kaufman, V., and Drossel, B. Properties of attractors of canalyzing random Boolean networks. *Phys. Rev. Lett.* **73** (2006) 026118:1–9.

[21] Polacik, P. Parabolic equations: Asymptotic behavior and dynamics on invariant manifolds. In: *Handbook of Dynamical Systems, Vol. 2.* (B. Fiedler, ed.). Elsevier, Amsterdam, 2002, 835–884.

[22] Samuelsson, B., and Troein, C. Superpolynomial growth in the number of attractors in Kauffman networks, *Phys. Rev. Lett.* **90** (9) (2003) 098701:1–4.

[23] Shmulevic, I., Lahdesmaki, H., Dougherty, E., Astola, J. and Zhang, W. The role of certain Post classes in Boolean network models of genetic networks. *Proc. Nat. Acad. USA* **100** (19) (2003) 10734–10739.

[24] Smith, H.L. *Monotone dynamical systems,* Math Surv. and Monogr., AMS, Providence, RI, 1995.

[25] Sontag, D.E. Monotone and near-monotone biochemical networks. *J. Sys. Synth. Biol.* **1** (2007) 59–87.

[26] Sontag, E. D., Veliz-Cuba, A., Laubenbacher, R. and Jarrah, A.S. The effect of negative feedback loops on the dynamics of Boolean networks. *Biophys. J.* **95** (2) (2008) 518–526.

[27] Zierler, N. and Brillhart, J. On primitive trinomials (mod 2). *Information and Control* **13** (1968) 541–554.