JOURNAL OF DIFFERENCE EQUATIONS AND APPLICATIONS
2023, VOL. 29, NO. 5, 542–560
https://doi.org/10.1080/10236198.2023.2220811

NOTE

OPEN ACCESS

From Boolean networks to linear dynamical systems: a simplified route

Yunjiao Wang[a], Maria C. A. Leite[b,c] and Alona Ben-Tal[d]

[a]Department of Mathematics, Texas Southern University, Houston, TX, USA; [b]Department of Mathematics and Statistics, University of South Florida St. Petersburg, St. Petersburg, FL, USA; [c]Basque Center for Applied Mathematics (BCAM), Bilbao, Spain; [d]Insightful Modelling Limited, Auckland, New Zealand

ABSTRACT

Boolean Networks can be converted to discrete linear dynamical systems on finite spaces by a semi-tensor-product approach. This approach has been used by many to study the dynamics and control of Boolean systems. However, the process of getting the linear representation using the semi-tensor-product method is complicated even for a simple three-node network and requires the help of a computer program. In this work, we show that we can skip the semi-tensor process and obtain the same linear representation with a straightforward mapping. Moreover, our approach produces a large number of isomorphic representations which provides a flexible framework. Importantly, it could simplify the analytical study of networks with unspecified number of nodes that have some structure.

ARTICLE HISTORY
Received 27 June 2022
Accepted 25 May 2023

KEYWORDS
Truth matrix; semi-tensor product; discrete dynamical system

AMS CLASSIFICATIONS
37A99; 92C42; 06E30

1. Introduction

An n-node autonomous Boolean network is a discrete dynamical system with the form

$$x_i(t+1) = g_i(x_1(t), x_2(t), \dots, x_n(t)), \tag{1}$$

where x_i is the state variable of the ith node, t is the current time step, $t+1$ is the next time step and g_i is a Boolean function with a value of either 0 or 1. To simplify the notation, we rewrite Equation (1) as:

$$\mathbf{x}^+ = g(\mathbf{x}), \tag{2}$$

where $\mathbf{x}$ is a vector of size n, containing the states of all the Boolean nodes at the current step, $\mathbf{x}^+$ is the state of the n Boolean nodes in the next step and $g = [g_1, g_2, \dots, g_n]^T$.

Since first introduced by Kauffman [22], Boolean networks have been widely used to model biological regulatory networks [1,2,6,14,18,36,37,40,41,46]. They can be set up in situations where the detailed kinetic characterization of an interaction is not available or not essential and provide many valuable insights [4,12,13,18,35,38,42,43].

Identifying fixed points and cycles is crucial for understanding the overall dynamical behaviour of a Boolean network. When a network is small, cycles can be obtained

CONTACT Alona Ben-Tal alona@insightfulmodelling.com Insightful Modelling Limited, 6 Watervista Place, Mangere Bridge, Auckland, 2022, New Zealand

by exhaustive enumeration. However, as the number of nodes increases, the number of states grows exponentially. For large networks, numerical methods are the main tools for investigating the dynamics [7,10,25,30,31,34,44]. To facilitate better understanding of Boolean networks, several representations other than truth tables were proposed in recent years such as linear representation [9], polynomial representation [20,28,44] and directed graphs [27,32]. Among these, the linear representation proposed by Cheng and Qi [9] has been used by many to analyse the dynamics and control of Boolean networks [21,23,26,45,47,48]. The advantage of a linear representation is that it enables us to use existing theory from discrete dynamical systems to study Boolean networks.

While the resulting linear representation is very useful, without computer-aided calculation, the process of obtaining it using semi-tensor products is not easy even for a small three-node network. This makes it impossible to study more general network systems, where the number of nodes is not specified. To address this issue, we propose here a straightforward approach to obtain the same linear representation. Moreover, since there are an infinite number of isomorphic representations for a given Boolean network, our method makes it easier to choose the most convenient one to work with. In addition, we can construct a hybrid system that is in part Boolean and in part non-Boolean if it is needed. Our result thus provides a flexible way of representing Boolean networks that facilitates theoretical studies about their general properties.

This note is constructed as follows. In Section 2, we briefly review the semi-tensor-product approach for obtaining a linear representation, and then show that the representation can be obtained by a straightforward map. In Section 3, we introduce a *truth matrix*, which is a compact way to work with a linear representation. We then provide an algorithm for identifying cycles and fixed points. We show in the Appendix how our method can be applied for analysing Boolean representation of neural networks.

2. Linear representation of Boolean networks

The aim of this section is to write the Boolean network (2) as a discrete linear system:

$$\mathbf{y}^{+} = L\,\mathbf{y}. \tag{3}$$

We show that $\mathbf{y}$ is a $2^n \times 1$ vector and L is a $2^n \times 2^n$ matrix (recall that n is the number of nodes in the original Boolean network and that $\mathbf{y}^{+}$ is $\mathbf{y}$ in the next step). We introduce two methods for obtaining $\mathbf{y}$ and L. Both methods lead to the same $\mathbf{y}$ and L but the processes to derive the result are distinct. The first method was proposed by Cheng and Qi [9] and the second method is a simplification we propose here for the first time. We review the first method in Section 2.1 to make the comparison with our method easier. Then, in Section 2.2 we discuss our alternative approach.

2.1. Linear representation of Boolean networks based on semi-tensor product [9]

Semi-tensor product: Let $A_{m\times n}$ and $B_{p\times q}$ be two matrices. When $n = kp$ or $p = kn$ for some positive integer $k > 0$, the semi-tensor product of A and B is defined as follows

$$A \ltimes B = \begin{cases} A(B \otimes I_k), & \text{if } n = kp \\ (A \otimes I_k)B, & \text{if } p = kn \end{cases}$$

where $\otimes$ is the Kronecker product and I_k is a $k \times k$ identity matrix. For example, if A and B are two column vectors: $A = (a_1, \dots, a_m)^T$ and $B = (b_1, \dots, b_p)^T$, then

$$A \ltimes B = (a_1 b_1, \dots, a_1 b_p, a_2 b_1, \dots, a_2 b_p, \dots, a_m b_1, \dots, a_m b_p)^T$$

is an $mp \times 1$ vector.

By its definition, the semi-tensor product is a generalization of conventional matrix product. We thus will assume that the matrix product is a semi-tensor product and the notation $\ltimes$ will be omitted later on.

Matrix expression of logic: Let δ_k^i be the i^{th} column of the identity matrix I_k and

$$\Delta_k = \{\delta_k^i | i = 1, 2, \dots, k\}.$$

The logical states $T = 1, F = 0$ are mapped into:

$$T \equiv \delta_2^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } F \equiv \delta_2^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4}$$

Next, matrices for each logic operation are defined. For example, the matrices corresponding to negation ($\neg$), disjunction/or ($\vee$) and conjunction/and, $\wedge$ are M_n, M_d and M_c respectively as follows:

$$\begin{aligned} M_n &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = [\delta_2^2, \delta_2^1] \equiv \delta_2[2, 1], \\ M_d &= \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \equiv \delta_2[1, 1, 1, 2], \\ M_c &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \equiv \delta_2[1, 2, 2, 2] \end{aligned} \tag{5}$$

Logic functions can then be converted to matrix semi-tensor products. For example, the operation $F \wedge T = F$ is converted to the semi-tensor product:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{6}$$

Deriving linear representations of Boolean systems : Next we show an example ([9, Example IV.7]) of how the linear representation is derived through semi-tensor product.

Consider the system

$$\begin{aligned} x_1^+ &= x_2 \wedge x_3, \\ x_2^+ &= \neg x_1, \\ x_3^+ &= x_2 \vee x_3. \end{aligned} \tag{7}$$

where $x_1, x_2, x_3 \in \{0, 1\}$.

Based on the matrix expression of logic operators in (5) and the identification given in (4), system (7) can be rewritten as follows

$$\begin{cases} \mathbf{A}^+ = M_c\mathbf{BC} \\ \mathbf{B}^+ = M_n\mathbf{A} \\ \mathbf{C}^+ = M_d\mathbf{BC} \end{cases} \tag{8}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \Delta_2 = \{\delta_2^1, \delta_2^2\}$.

Next, define $\mathbf{y} = \mathbf{ABC}$ which gives $\mathbf{y} \in \Delta_8$. To derive L, we need two additional matrices: the *swap matrix* $W_{[m,n]}$, and the *power-reducing matrix* M_r (see details in [9]). Then using the semi-tensor product and its properties we get [9]:

$$\begin{aligned} \mathbf{y}^+ &= M_c\mathbf{BC}M_n\mathbf{A}M_d\mathbf{BC} \\ &= M_c(I_4 \otimes M_n)\mathbf{BCA}M_d\mathbf{BC} \\ &= M_c(I_4 \otimes M_n)(I_8 \otimes M_d)\mathbf{BCABC} \\ &= M_c(I_4 \otimes M_n)(I_8 \otimes M_d)W_{[2,4]}\mathbf{ABCBC} \\ &= M_c(I_4 \otimes M_n)(I_8 \otimes M_d) \otimes W_{[2,4]}\mathbf{AB}W_{[2]}\mathbf{BCC} \\ &= M_c(I_4 \otimes M_n)(I_8 \otimes M_d)W_{[2,4]}(I_4 \otimes W_{[2]})\mathbf{A}M_r\mathbf{B}M_r\mathbf{C} \\ &= M_c(I_4 \otimes M_n)(I_8 \otimes M_d)W_{[2,4]})(I_4 \otimes W_{[2]})(I_2 \otimes M_r)(I_4 \otimes M_r)\mathbf{ABC}, \\ &= \delta_8[3,7,7,8,1,5,5,6]\mathbf{y} = L\mathbf{y}, \end{aligned} \tag{9}$$

As can be seen, although the resulting linear expression is simple, the intermediate steps and expressions are rather complicated. The above calculation can be simplified by a recent development of the semi-tensor process in [24]. However, even with this simplification it is still not an easy calculation for such a small system. We show in the next subsection that we do not have to go through all these steps to find the linear representation.

2.2. Proposed alternative conversion to discrete dynamical system

In this section, we show in Theorem 2.1 that *any* bijection from $\{0,1\}^n \to \Delta_{2^n}$ converts a Boolean network to a discrete linear dynamical system $\mathbf{y}^+ = L\,\mathbf{y}$. Then we prove in Theorem 2.2 that the semi-tensor product used for the construction of $\mathbf{y}$ is the same as a *specific* bijection from $\{0,1\}^n \to \Delta_{2^n}$.

We begin by defining a bijection that takes a state of a Boolean network and convert it into a distinct number. Let $Q = \{q_1, \ldots, q_{2^n}\}$ be a set of 2^n distinct numbers and $h : \{0,1\}^n \to Q$ be a bijection function that associates each Boolean state with a single number. Depending on the set Q, the function h could take different forms and is not unique. For example, if $Q = \{0, 1, 2, \ldots, 2^n - 1\}$ the function h could be

$$h(x_1, x_2, \ldots, x_n) = x_1(2^{n-1}) + x_2(2^{n-2}) + \cdots + x_n, \tag{10}$$

and many more.

If $Q = \{1, 2, 3, \ldots, 2^n\}$ the function h could be:

$$h(x_1, x_2, \ldots, x_n) = x_1(2^{n-1}) + x_2(2^{n-2}) + \cdots + x_n + 1, \tag{11}$$

or

$$h(x_1, x_2, \ldots, x_n) = 2^n - \left(x_1(2^{n-1}) + x_2(2^{n-2}) + \cdots + x_n\right). \tag{12}$$

and many more.

Theorem 2.1: *Let $g : \{0,1\}^n \to \{0,1\}^n$ be a Boolean map and $U_n = \{\mathbf{e}_i\}_{i=1}^{2^n}$ be the standard basis of $\mathbb{R}^{2^n}$. Let $Q = \{1, 2, 3, \ldots, 2^n\}$ and $h : \{0,1\}^n \to Q$ be a bijection map. Define $H : \{0,1\}^n \to U_n$ by*

$$H(\mathbf{x}) = \mathbf{e}_{h(\mathbf{x})}, \quad \textit{for } \mathbf{x} \in \{0,1\}^n.$$

Then $\tilde{g} : U_n \to U_n$ defined by

$$\tilde{g} = H \circ g \circ H^{-1}$$

is isomorphic to g and can be represented as

$$\tilde{g}(\mathbf{y}) = L\mathbf{y}, \quad \textit{for } \mathbf{y} \in U_n$$

with $L = [\tilde{g}(\mathbf{e}_1), \tilde{g}(\mathbf{e}_2), \ldots, \tilde{g}(\mathbf{e}_{2^n})]$.

Proof: Since h is a bijection, it is invertible, which implies that H is also invertible. This makes $\tilde{g}$ isomorphic to g by definition. Next we show that $\tilde{g}(\mathbf{y}) = L\mathbf{y}$. Let L_i be the i^{th} column of L. Then $L_i = L\mathbf{e}_i = \tilde{g}(\mathbf{e}_i)$. So $\tilde{g}(\mathbf{y}) = L\mathbf{y}$, for $\mathbf{y} \in U_n$. ■

Remark 2.1: (1) Theorem 2.1 says that instead of solving the Boolean network $\mathbf{x}^+ = g(\mathbf{x})$, we can solve the linear system $\mathbf{y}^+ = L(\mathbf{y})$.

(2) The map H in Theorem 2.1 is an arbitrary bijection from $\{0,1\}^n$ to Δ_{2^n} since h is arbitrary. So the theorem implies that any bijection from $\{0,1\}^n$ to Δ_{2^n} takes a Boolean network system to a discrete linear dynamical system.

When h is as in Equation (12), we prove in Theorem 2.2 that we obtain the same linear representation as the one constructed by the semi-tensor product method developed by Cheng and Qi [9]. As an example, take System (7). Table 1 shows the isomorphic map of the system from $U_n \to U_n$.

Table 1. Truth table of the isomorphic linear dynamical system of the Boolean system (7). H is defined in Theorem 2.1 with h defined by Equation (12).

$H(\mathbf{x})$	$H(\mathbf{x}^+)$
$H(1,1,1) = \mathbf{e}_1$	$H(1,0,1) = \mathbf{e}_3$
$H(1,1,0) = \mathbf{e}_2$	$H(0,0,1) = \mathbf{e}_7$
$H(1,0,1) = \mathbf{e}_3$	$H(0,0,1) = \mathbf{e}_7$
$H(1,0,0) = \mathbf{e}_4$	$H(0,0,0) = \mathbf{e}_8$
$H(0,1,1) = \mathbf{e}_5$	$H(1,1,1) = \mathbf{e}_1$
$H(0,1,0) = \mathbf{e}_6$	$H(0,1,1) = \mathbf{e}_5$
$H(0,0,1) = \mathbf{e}_7$	$H(0,1,1) = \mathbf{e}_5$
$H(0,0,0) = \mathbf{e}_8$	$H(0,1,0) = \mathbf{e}_6$

By Theorem 2.1, the isomorphic linear representation of the Boolean system $\mathbf{x}^+ = g(\mathbf{x})$ is

$$\mathbf{y}^+ = [\mathbf{e}_3, \mathbf{e}_7, \mathbf{e}_7, \mathbf{e}_8, \mathbf{e}_1, \mathbf{e}_5, \mathbf{e}_5, \mathbf{e}_6]\mathbf{y},$$

which is the same as was obtained by the semi-tensor product in (9).

Theorem 2.2: *Let $x_i \in \{0, 1\}$ and $\mathbf{z}_i \in \Delta_2$ be the vector associated to x_i by (4) for $1 \le i \le n$. Then*

$$\mathbf{z}_1 \ltimes \mathbf{z}_2 \ltimes \cdots \ltimes \mathbf{z}_n = e_{h(\mathbf{x})} \tag{13}$$

where $h(\mathbf{x}) = 2^n - (x_1(2^{n-1}) + x_2(2^{n-2}) + \cdots + x_n) = 2^n - Dec(\mathbf{x})$.

Proof: We use induction to prove the theorem. When $n = 1$, by (4), $x_1 = 1$ is associated with $[1, 0]^T = e_1 = e_{2^1-1}$ and $x_1 = 0$ is associated with $[0, 1]^T = e_2 = e_{2^1-0}$. So (13) holds for $n = 1$.

Suppose (13) holds when $n = m$, that is, $\mathbf{z}_1 \ltimes \mathbf{z}_2 \ltimes \cdots \ltimes \mathbf{z}_m = \ltimes_{i=1}^m \mathbf{z}_i = e_{2^m - Dec(\mathbf{x})}$. When $n = m + 1$, we need to prove that $\ltimes_{i=1}^{m+1} \mathbf{z}_i = e_{2^{m+1} - Dec(\mathbf{x})}$. Note that $\ltimes_{i=1}^{m+1} \mathbf{z}_i = \mathbf{z}_1 \ltimes (\ltimes_{i=1}^m \mathbf{z}_{i+1})$. We will prove the result by considering two possible cases: (Case 1) when $x_1 = 0$ and (Case 2) when $x_1 = 1$. Let $\tilde{\mathbf{x}} = (x_2, \dots, x_{m+1})$.

(Case 1) When $x_1 = 0$, $\mathbf{z}_1 = [0, 1]^T$ and $\mathbf{x} = (0, \tilde{\mathbf{x}})$. Then

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \ltimes (\ltimes_{i=1}^m \mathbf{z}_{i+1}) = \begin{bmatrix} \mathbf{0} \\ \ltimes_{i=1}^m \mathbf{z}_{i+1} \end{bmatrix} = e_{2^m + (2^m - Dec(\tilde{\mathbf{x}}))} = e_{2^{m+1} - Dec(\mathbf{x})}$$

where the bold-faced $\mathbf{0}$ is a zero vector with dimension $2^m \times 1$. So (13) holds in this case when $n = m + 1$.

(Case 2) When $x_1 = 1$, $\mathbf{z}_1 = [1, 0]^T$ and $\mathbf{x} = (1, \tilde{\mathbf{x}})$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \ltimes (\ltimes_{i=1}^m \mathbf{z}_{i+1}) = \begin{bmatrix} \ltimes_{i=1}^m \mathbf{z}_{i+1} \\ \mathbf{0} \end{bmatrix} = e_{2^m - Dec(\tilde{\mathbf{x}})}$$

Note that $Dec(\mathbf{x}) = Dec(1, \tilde{\mathbf{x}}) = 2^m + Dec(\tilde{\mathbf{x}})$. So $2^m - Dec(\tilde{\mathbf{x}}) = 2^m + 2^m - (2^m + Dec(\tilde{\mathbf{x}})) = 2^{m+1} - Dec(\mathbf{x})$. Hence, $\ltimes_{i=1}^{m+1} \mathbf{z}_i = e_{2^{m+1} - Dec(\mathbf{x})}$ and (13) holds for this case as well.

Therefore, (13) holds for all n. ■

3. Truth matrix: an abreviated linear representation of Boolean networks

In this section, we show another way of analysing Boolean networks, similar to the way permutation matrices are reperesented [3].

Let $g : \{0, 1\}^n \to \{0, 1\}^n$ be a Boolean map and $h : \{0, 1\}^n \to Q$ be a bijection function as defined in Section 2. We can now define a new map $f : Q \to Q$ that is isomorphic to g as follows

$$f = h \circ g \circ h^{-1}. \tag{14}$$

We can represent the map f by the following matrix

$$T(f) = \begin{bmatrix} q_1 & q_2 & \cdots & q_{2^n} \\ f(q_1) & f(q_2) & \cdots & f(q_{2^n}) \end{bmatrix}, \tag{15}$$

Table 2. A Boolean map.

x	$g(x)$
(0, 0, 0)	(0, 1, 0)
(0, 0, 1)	(0, 1, 1)
(0, 1, 0)	(0, 1, 1)
(0, 1, 1)	(1, 1, 1)
(1, 0, 0)	(0, 0, 0)
(1, 0, 1)	(0, 0, 1)
(1, 1, 0)	(0, 0, 1)
(1, 1, 1)	(1, 0, 1)

Table 3. A map isomorphic to the one in Table 2.

$h(x)$	$f(h(x))$
0	2
1	3
2	3
3	7
4	0
5	1
6	1
7	5

where $q_i \in Q$. As an example, consider the Boolean map g in Table 2. If h is defined by Equation (10), then its isomorphic map f is shown in Table 3.

Note that $T(f)$ in Equation (15) is just another way to represent the truth table (such as the one shown in Table 3). However, it has a simpler appearance, which provides an easier way to identify the Boolean states and their corresponding next states and potential patterns. Such representation facilitates analytical studies of Boolean networks, even of networks with unspecified number of nodes. We illustrate this aspect in the Appendix where we show a few examples. For convenience, we call $T(f)$ in Equation (15) a *truth matrix* of f. Note that the exact components of truth matrices depend on the choice of h.

We now provide some general results.

Definition 3.1 (Limit set and transient state): Consider an n-node Boolean network, $\mathbf{x}^+ = g(\mathbf{x})$, where $\mathbf{x} \in \{0,1\}^n$. $V \subset \{0,1\}^n$ is called a *limit set* of g if $g^m(V) \subset V$ when m is sufficiently large. A *transient state* is any state $\mathbf{x} \notin V$.

Lemma 3.2: *Let T be a truth matrix of the isomorphic map f. If q_i is not in the second row of T, then q_i is a transient state.*

Proof: Since the second row of T consists of all the images of $Q = h(\mathbf{x})$, then the fact that q_i is not in the second row of T means that q_i is not an image of any element on the first row. So q_i is a transient state. ■

Theorem 3.3: *Let T be a truth matrix and $q_0 \in T$ be a transient state. Let $\tilde{T}$ be the matrix resulting from removing the columns containing q_0. Then the limit set of T is the same as the limit set of $\tilde{T}$.*

Proof: We only need to prove that if q_i is in the limit set of T, then it must be in the limit set of $\tilde{T}$. Suppose that q_i is in the limit set of T. By definition of a limit set, there must exist q_j in the limit set of T such that $f(q_j) = q_i$. Therefore, $q_i \neq q_0$ and is not a transient state by Lemma 3.2. It follows that q_i is in $\tilde{T}$. Hence, removing columns related to the transient state q_0 does not reduce the limit set. ■

Based on these results, we propose the following algorithm for finding all possible cycles of a given Boolean network.

Algorithm 1 Algorithm for finding all cycles in a Boolean network

1. Create a Truth Matrix, T.
2. Identify all transient states. That is, identify all elements on the first row that do not show up in the second row and remove the corresponding column from T.
3. Repeat the previous step for the updated T until all elements on the first row show up in the second row. Then the remaining elements in the first row forms the limit set V.
4. Pick any element in V, trace its trajectory until it repeats itself. Then mark the trajectory as a cycle and remove all columns corresponding to the elements of this cycle from the matrix.
5. Repeat the previous step until T is empty.
6. Convert the states back to the original Boolean representation.

Example 3.4: We use the example from Table 3 to show the algorithm. First remove those columns whose first element does not show up in the second row, which we have indicated in red. Repeat the process until all the elements in the first row are in the second row.

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 3 & 7 & 0 & 1 & 1 & 5 \end{bmatrix} \to \begin{bmatrix} 0 & 1 & 2 & 3 & 5 & 7 \\ 2 & 3 & 3 & 7 & 1 & 5 \end{bmatrix} \to$$
$$\to \begin{bmatrix} 1 & 2 & 3 & 5 & 7 \\ 3 & 3 & 7 & 1 & 5 \end{bmatrix} \to \begin{bmatrix} 1 & 3 & 5 & 7 \\ 3 & 7 & 1 & 5 \end{bmatrix}.$$

We then pick any element, say 1 and trace its trajectory. In this case, we find that there is only one cycle: $1 \to 3 \to 7 \to 5 \to 1$. Then in the original system, the cycle is

$$001 \to 011 \to 111 \to 101 \to 001.$$

Remark 3.1: (1) There is a considerable body of literature on how to find cycles in Boolean networks numerically [5,11,15–17,19,29,33,39,49]. Our algorithm is for facilitating analytical studies of networks. It has the advantage of dealing with networks with unspecified number of nodes. We show its flexibility and advantage by presenting some examples in Appendix A.

(2) By Theorem 3.3, in order to find cycles, we can throw away known transient states from the beginning, which cannot be done with the linear representation (3).

(3) For large networks, we could divide nodes into groups. Importantly, the variables of each group can be associated with a *different* isomorphic representation. We illustrate this idea in Appendix A.

4. Discussion and conclusions

In this work we suggest an alternative approach for converting a Boolean Network to a discrete linear dynamical system on a finite space. Our approach is straightforword and much simpler than the semi-tensor product approach developed by [9] and produces a large number of isomorphic representations. We proved that our approach, when used with a *specific* isomorphic representation, gives the same end result as the linear representation obtained through the semi-tensor product. In addition, we introduced a truth matrix and an algorithm to analyse Boolean networks. The truth matrix can be viewed as a different version of a truth table, however, we find it more tractable. Our methodology facilitates analytical studies of networks. Importantly, it allows us to obtain general results for Boolean networks with unspecified number of nodes if they have some structure, which cannot be done under the semi-tensor product framework. On the other hand, we did not consider how our methodology could deal with delayed and/or controlled systems, for which the semi-tensor product method has provided a platform [8]. Our ability to obtain general results using our methodology and the flexibility of our approach is illustrated with a few case studies in Appendix A.

Acknowledgments

The three authors carried-on the underlying research and wrote the manuscript.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

YW is supported by National Science Foundation (NSF) [grant number HRD-1800406] and National Science Foundation (NSF) [grant number CNS-1831980]. This work was also supported by the 2019 University of South Florida Nexus Initiative (UNI) Award. MCAL acknowledges the support from Basque Center for Applied Mathematics (BCAM) Research Visit Fellowship.

References

[1] W. Abuo-Jaoude, D. Ouattara, and M. Kaufman, From structure to dynamics: Frequency tuning in the p53-mdm2 network: I. Logical approach, J. Theor. Biol. 258(4) (2009), pp. 561–577.
[2] R. Albert and H. Othmer, The topology of the regulatory interactions perdicts the expression pattern of the segment polarity genes in drosophila melanogaster, J. Theor. Biol. 233 (2003), pp. 1–13.
[3] M. Artin, Algebra, Prentice-Hall, 1991.
[4] A. Ben-Tal, Y. Wang, and M. C. A. Leite, The logic behind neural control of breathing pattern, Sci. Rep. 9, 9078 (2019). https://doi.org/10.1038/s41598-019-45011-7.
[5] N. Benes, L. Brim, S. Pastva, and D. Safranek, *Symbolic coloured SCC decomposition*, in Tools and Algorithms for the Construction and Analysis of Systems 27th International Conference,

TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27, April 1, 2021, Proceedings, Part II. Vol. 12652, 2021, pp. 64–83.

[6] C. Campbell, J. Thakar, and R. Albert, Network analysis reveals cross-links of the immune pathways activated by bacteria and allergen, Phys. Rev. E, Stat. Phys. Plasmas, Fluids Related Interdisciplinary Topics. 84(3) (2011), pp. 031929.

[7] C. Chaouiya, A. Naldi, and D. Thieffry, *Logical modelling of gene regulatory networks with ginsim*, in *Bacterial Molecular Networks. Methods in Molecular Biology (Methods and Protocols)*, Vol. 804, J. van Helden, A. Toussaint, and D. Thieffry, eds., Springer, New York, 2012.

[8] D. Cheng and H. Qi, Controllability and observability of Boolean control networks, Automatica. 45(7) (2009), pp. 1659–1667.

[9] D. Cheng and H. Qi, A linear representation of dynamics of Boolean networks, IEEE. Trans. Automat. Contr. 55(10) (2010), pp. 2251–2258.

[10] J. Dorier, I. Crespo, A. Niknejad, R. Liechti, M. Ebeling, and I. Xenarios, Boolean regulatory network reconstruction using literature based knowledge with a genetic algorithm optimization method, BMC. Bioinform. 17(1) (2016), pp. 410.

[11] E. Dubrova and M. Teslenko, A sat-based algorithm for finding attractors in synchronous Boolean networks, Comput. Biol. Bioinform. 8(5) (2011), pp. 1393–1399.

[12] R. Edwards and L. Glass, Combinatorial explosion in model gene networks, Chaos: An Inerdisciplinary J. Nonlinear Sci. 10(3) (2000), pp. 691–704.

[13] R. Edwards, H. Siegelmann, K. Aziza, and L. Glass, Symbolic dynamics and computation in model gene models, Chaos: An Inerdisciplinary J. Nonlinear Sci. 11(1) (2001), pp. 160–169.

[14] C. Espionza-Soto, P. Padilla-Longoria, and E. Alvarez-Buylla, A gene regulatory network model for cell-fate determination during arabidopsis thaliana flower development that is robus and recovers experiental gene expression profiles, Plant. Cell. 16 (2004), pp. 2923–2939.

[15] A. Garg, I. Xenarios, L. Mendoza, and G. DeMicheli, *An efficient method for dynamic analysis of gene regulatory networks and in silico gene perturbation experiments*, in *Research in Computational Molecular Biology*, T. Speed and H. Huang, eds., Springer, Berlin, Heidelberg, 2007, pp. 2–76.

[16] A. Garg, A. Di Cara, I. Xenarios, L. Mendoza, and G. De Micheli, Synchronous versus asynchronous modeling of gene regulatory networks, Bioinformatics. 24(17) (2008), pp. 1917–1925.

[17] T.V. Giang and K. Hiraishi, *An efficient method for approximating attractors in large-scale asynchronous Boolean models*, in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2020, pp. 1820–1826.

[18] L. Glass and S.A. Kauffman, The logical analysis of continuous, nonlinear biochemical control network, J. Theor. Biol. 39 (1973), pp. 103–129.

[19] W. Guo, G. Yang, W. Wu, L. He, and M. Sun, A parallel attractor finding algorithm based on Boolean satisfiability for genetic regulatory networks, PLoS. ONE. 9(4) (2014), pp. 1–10.

[20] F. Hinkelmann, D. Murrugarra, A. Jarrah, and R. Laubenbacher, A mathematical framework for agent based models of complex biological networks, Bull. Math. Biol. 73 (2011), pp. 1583–1602.

[21] C. Huang, J. Lu, G. Zhai, J. Cao, G. Lu, and M. Perc, Stability and stabilization in probability of probabilistic Boolean networks, IEEE. Trans. Neural. Netw. Learn. Syst. 32(1) (2021), pp. 241–251.

[22] S.A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, J. Theor. Biol. 22(3) (1969), pp. 437–467.

[23] H. Li, G. Zhao, M. Meng, and J. Feng, A survey on applications of semi-tensor product method in engineering, Sci. China Inf. Sci. 61(1) (2017), pp. 010202.

[24] Z. Liu and D. Cheng, *Canonical form of Boolean networks*, 2019 Chinese Control Conference, 2019.

[25] H. Mandon, C. Su, S. Haar, J. Pang, and L. Pauleve, *Sequential reprogramming of Boolean networks made practical*, in *Computational Methods in Systems Biology. CMSB 2019. Lecture Notes in Computer Science*, Volume 11773, L. Bortolussi and G. Sanguinetti, eds., Springer, Cham, 2019.

[26] M. Meng, L. Liu, and G. Feng, Stability and l_1 gain analysis of Boolean networks with markovian jump parameters, IEEE. Trans. Automat. Contr. 62(8) (2017), pp. 4222–4228.
[27] L. Menini, C. Possieri, and A. Tornamb, Boolean network representation of a continuous-time system and finite-horizon optimal control: Application to the single-gene regulatory system for the lac operon, Int. J. Control. 90(3) (2017), pp. 519–552.
[28] L. Menini, C. Possieri, and A. Tornambè, Boolean network analysis through the joint use of linear algebra and algebraic geometry, J. Theor. Biol. 472 (2019), pp. 46–53.
[29] A. Mizera, J. Pang, H. Qu, and Q. Yuan, Taming asynchrony for attractor detection in large Boolean networks, IEEE/ACM Trans. Comput. Biol. Bioinform. 16(1) (2019), pp. 31–42.
[30] C. Mussel, M. Hopfensitz, and H.A. Kestler, Boolnet–an R package for generation, reconstruction and analysis of Boolean networks, Bioinformatics. 26(10) (2010), pp. 1378–1380.
[31] M. Ostrowski, L. Pauleve, T. Schaub, A. Siegel, and C. Guziolowski, Identification from perturbation time series data combining dynamics abstraction and logic programming, Biosystems. 149 (2016), pp. 139–153.
[32] C. Possieri and A.R. Teel, Asymptotic stability in probability for stochastic Boolean networks, Automatica. 83 (2017), pp. 1–9.
[33] H. Qu, Q. Yuan, J. Pang, and A. Mizera, *Improving BDD-based attractor detection for synchronous Boolean networks*, in *Proceedings of the 7th Asia-Pacific Symposium on Internetware*, Internetware '15, New York, NY, USA, Association for Computing Machinery, 2015, pp. 212–220.
[34] C. Rohr, W. Marwan, and M. Heiner, Snoopy–a unifying petri net framework to investigate biomolecular networks, Bioinformatics. 26(7) (2010), pp. 974–975.
[35] A. Saadatpour, R. Albert, and T.C. Reluga, A reduction method for Boolean network models proven to conserve attractors, SIAM J. Appl. Dyn. Sys. 12(4) (2013), pp. 1997–2011.
[36] J. Saez-Rodriguez, L. Simeoni, J.A. Lindquist, R. Hemenway, U. Bommhardt, R. Weismantel, E.D. Gilles, S. Klamt, and B. Schraven, A logical model provides insights into t cell receptor signaling, PLoS. Comput. Biol. 3 (2007), pp. e163.
[37] L. Sanchez and D. Thieffry, A logical analysis of the drosophila gap-gene system, J. Theor. Biol. 211 (2001), pp. 115–141.
[38] E. Snoussi, Qualitative dynamics of piecewise differential equations: A discrete mapping approach, Dyn. Stab. Sys. 4(3) (1989), pp. 189–207.
[39] R. Somogyi and L.D. Greller, The dynamics of molecular networks: Applications to therapeutic discovery, Drug. Discov. Today. 6(24) (2001), pp. 1267–1277.
[40] L. Song, S.L. AssMann, and R. Albert, Predicting essential components of signal transduction networks: A dynamic model of guard cell abscisic acid signaling, PLoS. Biol. 4 (2006), pp. e100–e312.
[41] J. Thakar, A.K. Pathak, L. Murphy, R. Albert, I. Cattadori, and R.J.D. Boer, Network model of immune responses reveals key efectors to single and co-infection dynamics by a respiratory bacterium and a gatrointestinal helminth, PLoS. Comput. Biol. 8(1) (2012), pp. e1002345.
[42] R. Thomas, Biological Feedback, CRC, 1990.
[43] A. Veliz-Cuba, A. Kumar, and K. Josic, *Piecewise linear and Boolean models of chemical reaction networks*. Accepted to J. Math. Bio, 2014a.
[44] A. Veliz-Cuba, B. Aguilar, F. Hinkelmann, and R. Laubenbacher, Steady state analysis of Boolean molecular network models via model reduction and computational algebra, BMC. Bioinform. 15, 221 (2014b). https://doi.org/10.1186/1471-2105-15-221.
[45] B. Wang, H. Li, and Y. Yu, On detectability of Boolean control networks, Nonlinear Anal: Hybri. Sys.36 (2020), pp. 100859.
[46] R.S. Wang and R. Albert, Discrete dynamical modeling of cellular signaling networks, Meth. Enzymol.467 (2009), pp. 281–306.
[47] J. Yang, J. Lu, J. Lou, and Y. Liu, Synchronization of drive-response Boolean control networks with impulsive disturbances, Appl. Math. Comput. 364 (2020), pp. 124679.
[48] Y. Yu, J. Feng, J. Pan, and D. Cheng, Block decoupling of Boolean control networks, IEEE. Trans. Automat. Contr. 64(8) (2019), pp. 3129–3140.
[49] Q. Yuan, A. Mizera, J. Pang, and H. Qu, A new decomposition-based method for detecting attractors in synchronous Boolean networks, Sci. Comput. Program. 180 (2019), pp. 18–35.

Appendix A. Case studies

In this appendix we present three case studies that correspond to three Boolean networks introduced in [4]. The dynamical properties of these networks were derived theoretically in [4] deploying a different methodology than the one presented in this appendix. To illustrate the advantages and flexibility of the approach we present in this paper, we prove the results found in [4] using the *truth matrix*, isomorphic decimal representation and the algorithm we proposed (Algorithm 1). Specifically, in Appendix A.1, we give the detailed construction of the *truth matrix* associated with the example, including the states of all nodes, and use Algorithm 1 to derive the network limit sets. In Appendix A.2, we exemplify how prior knowledge on the dynamics of certain sub-networks can be used to simplify significantly the construction of the *truth matrix* from the beginning. In Appendix A.3, we illustrate how to analyse a large network by decomposing it into smaller sub-networks and use *different* isomorphic representations simultaneously.

A.1. Periodic signal generator

In [4], we considered the following Boolean network, whose architecture is presented in Figure A1.

$$x_1^+ = \neg(\vee_{i=1}^n x_i),$$
$$x_i^+ = x_{i-1} \text{for} \quad 2 \le i \le n. \tag{A1}$$

This network system generates a periodic output with period n of the form $(\overline{1\underbrace{0\cdots0}_{n-1}})$, where the pattern under the line repeats.

Here we use the truth matrix and Algorithm 1 to prove the following result that is equivalent to the one in [4, Lemma 1].

Theorem A.1: *The limit set of System* (A1) *consists of a unique cycle*

$$(0\cdots0) \to (10\cdots0) \to (010\cdots0) \to \cdots \to (0\cdots01) \to (0\cdots0),$$

which corresponds to the cycle under the map h in Equation (10)

$$0 \to 2^{n-1} \to 2^{n-2} \to \cdots \to 2 \to 1 \to 0.$$

Proof: We first derive the truth matrix for the Boolean system from Equation (A1).

By Equation (A1), we obtain

(1) $(0,\ldots,0)^+ = (1,0,\ldots,0)$,
(2) $(x_1,x_2,\ldots,x_n)^+ = (0,x_1,\ldots,x_{n-1})$ when $x_j = 1$ for some $j \in \{1,2,\ldots,n\}$.

Then under the map h defined by Equation (10), (1) and (2) are equivalent to

$$0^+ = 2^{n-1}$$

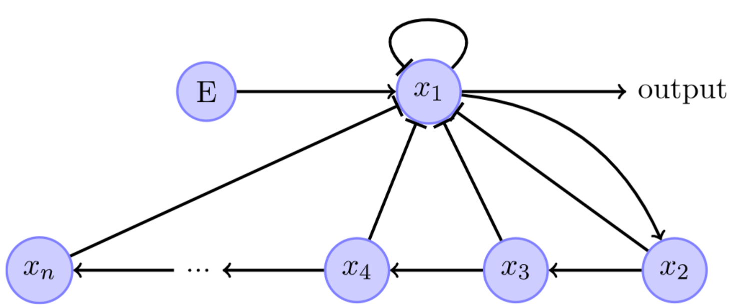


Figure A1. Boolean network with n nodes, x_1, $x_2, \ldots, x_n$ and an external signal E, which is equal to 1 at all steps. The network generates a periodic signal with period n.

and

$$(x_1 2^{n-1} + x_2 2^{n-2} \cdots + x_{n-1} 2 + x_n)^+ = x_1 2^{n-2} + x_2 2^{n-3} + \cdots + x_{n-1}. \tag{A2}$$

Note that a binary number is odd if and only if the last digit is 1. So Equation (A2) means that when the number $q_i \in \{0, 1, 2, \ldots, 2^n - 1\}$ is such that $q_i \neq 0$ then

$$q_i^+ = \begin{cases} \frac{q_i}{2}, & \text{when } q_i \text{ is even} \\ \frac{q_i - 1}{2}, & \text{when } q_i \text{ is odd.} \end{cases}$$

Hence, the truth matrix of the system is

$$T_0 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & \cdots 2q_i & 2q_i + 1 & \cdots & 2^n - 2 & 2^n - 1 \\ 2^{n-1} & 0 & 1 & 1 & 2 & \cdots q_i & q_i & \cdots & 2^{n-1} - 1 & 2^{n-1} - 1 \end{bmatrix}.$$

Note that all numbers larger than 2^{n-1} are not in the range (the second row). So we can remove all those columns whose first element $> 2^{n-1}$. The resulting truth matrix is

$$T_1 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & \cdots 2q_i & 2q_i + 1 & \cdots & 2^{n-1} - 1 & 2^{n-1} \\ 2^{n-1} & 0 & 1 & 1 & 2 & \cdots q_i & q_i & \cdots & 2^{n-2} - 1 & 2^{n-2} \end{bmatrix}. \tag{A3}$$

Next note that all numbers between 2^{n-2} and 2^{n-1} are not in the range of the truth matrix T_1 in Equation (A3). We can reduce the matrix by removing all those columns whose first elements are in $(2^{n-2}, 2^{n-1})$. Repeat this process, that is, remove those columns whose first elements are between 2^{n-3} and 2^{n-2} then those between $(2^{n-4}$ and 2^{n-3} etc., and we obtain the following matrix

$$\tilde{T} = \begin{bmatrix} 0 & 1 & 2 & 4 & \cdots 2^k & \cdots & 2^{n-2} & 2^{n-1} \\ 2^{n-1} & 0 & 1 & 2 & \cdots 2^{k-1} & \cdots & 2^{n-3} & 2^{n-2} \end{bmatrix}. \tag{A4}$$

It is easy to see from the matrix $\tilde{T}$ in Equation (A4) that the limit set of the map consists of a unique cycle

$$0 \to 2^{n-1} \to 2^{n-2} \to \cdots \to 2 \to 1 \to 0.$$

∎

A.2. Excitatory network with periodic input

In this section, we consider the Boolean system associated with the network in Figure A2, previously introduced in [4]. The system can be described by the following equations.

$$\begin{aligned} S_1^+ &= C_1 \\ S_i^+ &= S_{i-1} \qquad \text{for} \quad 2 \leq i \leq k \\ X_1^+ &= \textstyle\bigvee_{i_j \in K} (\bigwedge_{j=1}^{N} S_{i_j}), \end{aligned} \tag{A5}$$

where k is the number of S nodes, $K = \{1, \ldots, k\}$ and $C_1 = (\overline{1\underbrace{0 \cdots 0}_{p-1}})$ is a periodic signal with period p.

The System (A5) is of the form

$$\begin{cases} \mathbf{S}^+ = g_1(\mathbf{S}, C_1) \\ X_1^+ = g_2(\mathbf{S}), \end{cases} \tag{A6}$$

where $\mathbf{S} = (S_1, \ldots, S_k)$. Since it is a feed-forward network, we first look at the limit sets of $\mathbf{S}^+ = g_1(\mathbf{S}, C_1)$, then determine the dynamics of X_1. Our end result is the same as presented in [4] but the proof is different, and, due to the new representation we introduced in this paper, easier to validate.

Lemma A.2: *Consider the system*

$$\begin{aligned} S_1^+ &= C_1 \\ S_i^+ &= S_{i-1} \quad \textit{for} \quad 1 < i \leq k, \end{aligned} \tag{A7}$$

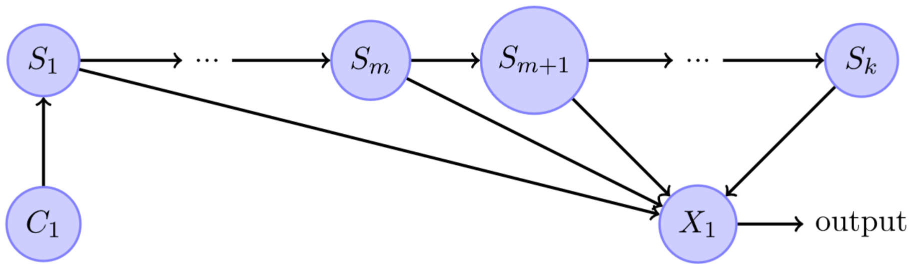


Figure A2. Excitatory network with periodic input [4]. C_1 is a periodic signal of period p, that is, it has a repeated pattern of one followed by $p-1$ zeros. $X_1 = 1$ in the next step whenever there are at least N ones in $(S_1, \ldots, S_k)$ at the current step.

where $C_1 = (\overline{1\underbrace{0\cdots0}_{p-1}})$ *and* $p \geq 1$. *Let* $l_0 = \lfloor \frac{k}{p} \rfloor$ *and* $r_0 = k - l_0 p$. *Denote*

$$\mathcal{Y}_j = 2^j(1 + 2^p + 2^{2p} + \cdots + 2^{l_0 p}), \quad \text{and} \quad \mathcal{U}_j = 2^j(1 + 2^p + 2^{2p} + \cdots + 2^{(l_0-1)p}),$$

where j *is an integer such that* $0 \leq j < p$. *Then System* (A7) *has the following unique cycle after each point on the cycle has been transformed by the map* $h(S_1, \ldots, S_k) = S_1 + 2S_2 + \cdots + 2^{n-1}S_n$.

$\mathcal{C}_0$: $\mathcal{U}_0 \to \mathcal{U}_1 \to \cdots \to \mathcal{U}_{p-1} \to \mathcal{U}_0$ *when* $r_0 = 0$,
$\mathcal{C}_1$: $\mathcal{Y}_0 \to \mathcal{U}_1 \to \cdots \to \mathcal{U}_{p-1} \to \mathcal{Y}_0$ *when* $r_0 = 1$,
$\mathcal{C}_2$: $\mathcal{Y}_0 \to \cdots \to \mathcal{Y}_{r_0-1} \to \mathcal{U}_{r_0} \to \cdots \to \mathcal{U}_{p-1} \to \mathcal{Y}_0$ *when* $r_0 > 1$.

Proof: Note that after the first k transient steps, the chain $(S_1S_2\cdots S_k)$ exhibits the periodic pattern of the controller C_1. More specifically, $(S_1S_2\cdots S_k)$ has the value of C_1 at the previous k time steps. Thus, $(S_1S_2\cdots S_k)$ must have the pattern $(\cdots 10\cdots 010\cdots)$, where there are exactly $p-1$ zeros between two consecutive 1s. Therefore, it is sufficient to construct a truth matrix that contains these states only. By (A7), $\mathbf{S}^+ = (S_1, \ldots, S_k)^+$ is just one unit shift to the right with an exception for S_1, which is determined by C_1, and the ending element that is pushed out of the chain. If j is the number of shifts to the right we can expect

$$h(\underbrace{0\cdots0}_{j}1\underbrace{0\cdots0}_{p-1}10\cdots) = 2^j(1 + 2^p + 2^{2p} + \cdots + 2^{lp}), \text{for some } j \text{ with } \quad 0 \leq j < p, \tag{A8}$$

where $l = l_0$ or $l = l_0 - 1$.

We next consider the detailed expression of $h(S^+) = h(S)^+$ on a case by case basis.

Since $l_0 = \lfloor \frac{k}{p} \rfloor$ and $r_0 = k - l_0 p$, then $0 \leq r_0 < p$.

(1) Consider first the case $r_0 = 0$. In this case, $k = l_0 p$ and there are exactly l_0 number of S_i with a value of 1.

Suppose $(S_1, \ldots, S_k) = (\overbrace{1\underbrace{0\cdots0}_{p-1}}^{l_0\,times})$ then,

$$h(S_1, \ldots, S_k) = 1 + 2^p + 2^{2p} + \cdots + 2^{(l_0-1)p} = \mathcal{U}_0.$$

If $p = 1$ then $\mathcal{U}_0^+ = \mathcal{U}_0$. If $p > 1$, then $S_1^+ = 0$ and the sequence $(S_1, \ldots, S_k)^+$ shifts by one unit to the right. Hence,

$$\mathcal{U}_0^+ = 2(1 + 2^p + 2^{2p} + \cdots + 2^{(l_0-1)p}) = \mathcal{U}_1.$$

Similarly, we can show that

$$\mathcal{U}_j^+ = \mathcal{U}_{j+1}$$

for all $1 \le j < p-1$. Note that $\mathcal{U}_{p-1} = 2^{p-1}(1 + 2^p + 2^{2p} + \cdots + 2^{(l_0-1)p})$ corresponds to the state with $C_1 = 1$. Hence, $S_1^+ = 1$. It follows that $\mathcal{U}_{p-1}^+ = \mathcal{U}_0$. Therefore, the system has a unique cycle: $\mathcal{U}_0 \to \mathcal{U}_1 \to \cdots \to \mathcal{U}_{p-1} \to \mathcal{U}_0$.

(2) Next consider the case $r_0 > 0$. In this case,

$$(S_1, \dots, S_k) = (\underbrace{0\cdots0}_{j}\overbrace{1\underbrace{0\cdots0}_{p-1}}^{l_0 times}1\underbrace{0\cdots0}_{r_0-1-j}) \text{ for } \quad 0 \le j \le r_0 - 1$$

and

$$(S_1, \dots, S_k) = (\underbrace{0\cdots0}_{j}\overbrace{1\underbrace{0\cdots0}_{p-1}}^{l_0-1, times}1\ \underbrace{0\cdots0}_{p+r_0-1-j}\ ) \text{ for } \quad r_0 \le j < p$$

Next we derive the truth matrix. If $S_1 = 1$ ($j = 0$), then

$$h(S_1, \dots, S_k) = 1 + 2^p + 2^{2p} + \cdots + 2^{l_0p} = \mathcal{Y}_0.$$

Similar to the arguments in the previous case, we can see that $\mathcal{Y}_j^+ = \mathcal{Y}_{j+1}$ for $0 \le j < r_0 - 1$, $\mathcal{Y}_{r_0-1}^+ = \mathcal{U}_{r_0}, \mathcal{U}_j^+ = \mathcal{U}_{j+1}$ for $r_0 \le j < p-1$ and $\mathcal{U}_{p-1}^+ = \mathcal{Y}_0$.

In summary, we can write the truth matrix M for this case as follows.

$$M = \begin{bmatrix} \mathcal{Y}_0 & \mathcal{Y}_1 & \cdots & \mathcal{Y}_{r_0-1} & \mathcal{U}_{r_0} & \mathcal{U}_{r_0+1} & \cdots & \mathcal{U}_{p-1} \\ \mathcal{Y}_1 & \mathcal{Y}_2 & \cdots & \mathcal{U}_{r_0} & \mathcal{U}_{r_0+1} & \mathcal{U}_{r_0+2} & \cdots & \mathcal{Y}_0 \end{bmatrix}. \tag{A9}$$

When $r_0 = 1$, the cycle is: $\mathcal{Y}_0 \to \mathcal{U}_1 \to \cdots \to \mathcal{U}_{p-1} \to \mathcal{Y}_0$ and when $r_0 > 1$ the cycle is: $\mathcal{Y}_0 \to \cdots \to \mathcal{Y}_{r_0-1} \to \mathcal{U}_{r_0} \to \cdots \to \mathcal{U}_{p-1} \to \mathcal{Y}_0$. Note that in all cases the cycle is of period p.

■

We can now prove the dynamics of X_1 in Equation A5.

Theorem A.3 ([4, Theorem 3 (wording slightly modified)]): *Consider the System* (A5) *where* $C_1 = (1\underbrace{0\cdots0}_{p-1})$. *Then the system has the following limit sets when* $N > 1$:

(1) *if* $p \le \frac{k}{N}$, *then* $X_1 = \{\overline{1}\}$
(2) *if* $p \ge \frac{k}{N-1}$, *then* $X_1 = \{\overline{0}\}$
(3) *if* $\frac{k}{N} < p < \frac{k}{N-1}$, $X_1 = \{\overline{\underbrace{1\cdots1}_{\bar{r}_0}\underbrace{0\cdots0}_{p-\bar{r}_0}}\}$ *where* $\bar{r}_0 = k - (N-1)p$.

When $N = 1$ *the limit sets are:*

(1) *when* $p \le k$, $X_1 = \{\overline{1}\}$;
(2) *when* $p > k$, $X_1 = \{\overline{\underbrace{1\cdots1}_{k}\underbrace{0\cdots0}_{p-k}}\}$.

Remark A.1: Limit sets in Boolean networks are usually classified as fixed points or cycles. However, in [4] a more sophisticated classification of limit sets has been introduced. The solution $X_1 = \{\overline{1}\}$ is a fixed point but was also classified in [4] as a periodic solution with period $p = 1$. The solution

$$X_1 = \{\overline{\underbrace{1\cdots1}_{\bar{r}_0}\underbrace{0\cdots0}_{p-\bar{r}_0}}\}, \quad \text{where } \bar{r}_0 = k - (N-1)p$$

is a cycle with period p. It was classified in [4] as "bursting" with period p (see [4] for the precise definition).

Proof: Suppose $N > 1$.

(1) First note that $p \leq \frac{k}{N}$ implies $\frac{k}{p} \geq N$ and $l_0 = \lfloor \frac{k}{p} \rfloor \geq N$. By Lemma A.2, all the three possible cycles $\mathscr{C}_0, \mathscr{C}_1$, and $\mathscr{C}_2$ which correspond to the state values of $(S_1, \ldots, S_k)$, contain at least l_0 ($\geq N$) number of 1s. It follows that X_1 has at least N number of active inputs at steady state. Hence $X_1 = (\bar{1})$.

(2) When $p \geq \frac{k}{N-1}$, $\frac{k}{p} \leq N - 1$. If $\frac{k}{p} = N - 1$ then, $\frac{k}{p} = \lfloor \frac{k}{p} \rfloor = l_0$ and $r_0 = k - l_0 p = 0$. It has been shown in Lemma A.2 that in this case, all possible cycles that correspond to the state values of $(S_1, \ldots, S_k)$ have at most l_0 number of 1s. Since $l_0 = N - 1$, $l_0 < N$. If $\frac{k}{p} < N - 1$ then $l_0 < N - 1$ since $\lfloor \frac{k}{p} \rfloor < \frac{k}{p}$. It has been shown in Lemma A.2 that all possible cycles that correspond to the state values of $(S_1, \ldots, S_k)$ have at most $l_0 + 1$ number of 1s. Since $l_0 < N - 1$, $l_0 + 1 < N$. In both cases, X_1 has at most $N-1$ active inputs at steady state and thus $X_1 = (\bar{0})$.

(3) When $\frac{k}{N} < p < \frac{k}{N-1}$, we have $N - 1 < \frac{k}{p} < N$. This implies that $l_0 = \lfloor \frac{k}{p} \rfloor = N - 1$ and $0 < r_0 < p$ where $r_0 = k - l_0 p$. It follows that $l_0 + 1 = N$. By Lemma A.2, $\mathscr{C}_1$ and $\mathscr{C}_2$ are the only possible cycles. Since $\mathcal{Y}_i$ corresponds to states of $(S_1, \ldots, S_k)$ that contain exactly $l_0 + 1$ number of 1s and $\mathcal{U}_i$ corresponds to states of $(S_1, \ldots, S_k)$ that contain exactly l_0 number of 1s, X_1 will be activated only when the states of $(S_1, \ldots, S_k)$ are in $\mathcal{Y}_i$. Since $\mathcal{Y}_i$ exist for $0 < i < r_0 - 1$ and $l_0 = N - 1$, it follows that

$$X_1 = \{\overline{\underbrace{1 \cdots 1}_{\bar{r}_0} \underbrace{0 \cdots 0}_{p - \bar{r}_0}}\},$$

where $\bar{r}_0 = k - (N - 1)p$.

Suppose now that $N = 1$.

(1) When $p \leq k$, $\frac{k}{p} \geq 1$, as in the case with $N > 1$, $(S_1, \ldots, S_k)$ contains at least one 1 in all the three possible limit cycles. Hence, $X_1 = (\bar{1})$.

(2) Note that $p > k$ implies that $0 < \frac{k}{p} < 1 = N$ and $l_0 = \lfloor \frac{k}{p} \rfloor = 0$. Hence, $r_0 = k - l_0 p = k$. By Lemma A.2, $\mathscr{C}_1$ and $\mathscr{C}_2$ are the only possible cycles. Notice that $\mathcal{Y}_i$ corresponds to the states of $(S_1, \ldots, S_k)$ containing exactly one 1 since $l_0 + 1 = 1$ and $\mathcal{U}_i$ corresponds to states of $(S_1, \ldots, S_k) = (0 \cdots 0)$ since $l_0 = 0$. Because the threshold N for activating X_1 is 1 and because there are r_0 states that correspond to $\mathcal{Y}_i$, $X_1 = \{\overline{\underbrace{1 \cdots 1}_{k} \underbrace{0 \cdots 0}_{p-k}}\}$ at steady state. ■

A.3. Simplification of an excitatory network with memory loss

In this section, we illustrate how the isomorphic representation we introduced in Section 2 can be used to simplify and analyse a Boolean network by decomposing it into subnetworks. The full network was first published in [4] and is shown in Figure A3(a). The network can be decomposed into two subnetworks shown in Figure A3(b ,c), and an equivalent network that uses the isomorphic representation is shown in Figure A3(d).

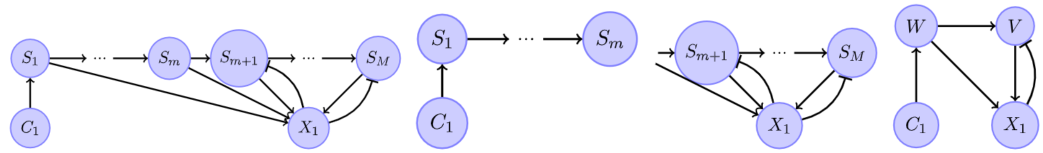
The Boolean system associated with the network in Figure A3(a) is:

$$\begin{array}{lll} S_1^+ = C_1 & & \\ S_i^+ = S_{i-1} & \text{for} & 1 < i \leq m \\ S_i^+ = S_{i-1} \wedge (\neg X_1(t)) & \text{for} & m < i \leq M \\ X_1^+ = \bigvee_{i,j \in K, i \neq j} (\bigwedge_{j=1}^{2} S_{i_j}(t)) & & \end{array} \tag{A10}$$

where $K = \{1, \ldots, M\}$.

Let $W = h(S_1, \ldots, S_m)$ and $V = h(S_{m+1}, \ldots, S_M)$. Then the isomorphic system to (A10) depicted in Figure A3(d), has the form

$$\begin{cases} W(t+1) = f_1(W(t), C_1(t)) \\ V(t+1) = f_2(V(t), X_1(t), W(t)) \\ X_1(t+1) = f_3(V(t), W(t)) \end{cases} \tag{A11}$$



(a) Full network (b) Sub-network 1 (c) Sub-network 2 (d) Equivalent network

Figure A3. Excitatory network with memory loss. (a) The full network [4]. C_1 is a periodic signal of period p, that is, it has a repeated pattern of one followed by $p-1$ zeros. $X_1 = 1$ in the next step whenever there are at least 2 ones in $(S_1, \ldots, S_M)$ at the current step. $S_i = 1$ in the next step if $C_1 = 1$ in the current step (for $i = 1$), if $S_{i-1} = 1$ in the current step (for $2 \le i \le m$) or if $S_{i-1} = 1$ and $X_1 = 0$ in the current step (for $m+1 \le i \le M$). (b) Sub-network 1 consisting of nodes $S_1, \ldots, S_m$ and the periodic input, C_1. (c) Sub-network 2 consisting of $S_{m+1}, \ldots, S_M$ and X_1. (d) An equivalent network with $W = h(S_1, \ldots, S_m)$ and $V = h(S_{m+1}, \ldots, S_M)$. (a) Full network, (b) Sub-network 1, (c) Sub-network 2 and (d) Equivalent network.

Note that the functions f_1 and f_2 have the range of the function h (see Section 2) while the function f_3 has the range of $\{0, 1\}$.

Next we prove Theorem A.4 below, which is similar to Theorem 2.6 in the supplement of [4] (there are small differences in the wording of the theorem). While the end result is the same, the proof is different and uses the various representations of the network shown in Figure A3.

Theorem A.4: *Consider the system in* (A10) *with* $C_1 = (\overline{1\underbrace{0\cdots0}_{p-1}})$. *Let* $l_0 = \lfloor \frac{m}{p} \rfloor$ *and* $r_0 = m - l_0 p$.

Then, after the first M steps,

1. *When* $p \le \frac{m}{2}$, $X_1 = \{\overline{1}\}$.
2. *When* $\frac{m}{2} < p \le m-2$, $X_1 = \{\overline{\underbrace{1\cdots1}_{r_0}\underbrace{0\cdots0}_{p-r_0}}\}$.
3. *When* $m-2 < p < M-1$, $X_1 = \{\overline{11\underbrace{0\cdots0}_{p-2}}\}$.
4. *When* $p = M-1$, $X_1 = \{\overline{1\underbrace{0\cdots0}_{p-1}}\}$.
5. *When* $p \ge M$, $X_1 = \{\overline{0}\}$.

Proof: First note the following observations, assuming that the isomorphic system (A11) was created using the map $h(S_1, \ldots, S_k) = S_1 + 2S_2 + \cdots + 2^{n-1}S_n$:

(i) $f_2(V, 1, W) = 0$ since X_1 suppresses all the nodes S_i for $m+1 \le i \le M$.
(ii) If $V(t) = 0$ for all t, then the dynamics of the network (A10) is the same as that of the network in Figure A2 with $m = k$.
(iii) The dynamics of sub-network 1 is governed by Lemma A.2.

We now prove the theorem point by point.

(1) Suppose $p \le \frac{m}{2}$. Considering only the interaction between Sub-network 1 and X_1, by Theorem A.3 with $N = 2$ and $k = m$, $X_1 = \{\overline{1}\}$. Since the inputs from $S_{m+1}, \ldots, S_M$ are activating, the additional interaction between these nodes and X_1 do not change the dynamics of X_1. Thus, when considering the full network, $X_1 = \{\overline{1}\}$.
(2) Suppose $\frac{m}{2} < p \le m-2$. First note that $m > 5$ (otherwise the statement is meaningless). Also note that $1 + \frac{2}{p} \le \frac{m}{p} < 2$. Therefore, $l_0 = \lfloor \frac{m}{p} \rfloor = 1$ and $2 \le r_0 = m - p < p$. By Lemma A.2, $W(t)$ goes around the cycle $\mathscr{C}_2$ since $r_0 \ge 2$.

Notice that independent of the initial values of X_1 and V, if $W(0) = \mathcal{Y}_0$, then $X_1(t) = 1$ for $1 \le t \le r_0$. The reason is as follows. By Lemma A.2, along $\mathscr{C}_2$,

$$W(t) = \begin{cases} \mathcal{Y}_t, & 0 \le t \le r_0 - 1 \\ \mathcal{U}_t, & r_0 \le t \le p - 1 \end{cases}$$

From the proof of Lemma A.2, Part 2, it follows that there are exactly two S_i having a value 1 when $0 \le t \le r_0 - 1$ and exactly one S_i having a value of 1 when $r_0 \le t \le p - 1$. Hence, $X_1 = 1$ for consecutive r_0 time steps, which effectively leads to $V = 0$ all the time. Hence $X_1 = \{\overline{\underbrace{1 \cdots 1}_{r_0}\underbrace{0 \cdots 0}_{p-r_0}}\}$.

To prove Points 3 and 4, we group $S_1, \dots, S_M$ as one entity and X_1 as another. Let $\widetilde{W} = h(S_1, \dots, S_M)$, then the system isomorphic to System (A10) is of the form

$$\begin{cases} \widetilde{W}(t+1) = \tilde{f}_1(\widetilde{W}(t), C_1(t), X_1(t)) \\ X_1(t+1) = \tilde{f}_2(\widetilde{W}(t)) \end{cases} \tag{A12}$$

Suppose $m - 2 < p \le M - 1$. We first show that all possible trajectories have to pass through $(\widetilde{W}, X_1) = (1 + 2^p, 0)$. We show it in two steps: (a) we show that X_1 can neither remain 0 nor remain 1 in any trajectory. (b) We show that if $X_1 = 0$, it will next be excited by $\widetilde{W} = 1 + 2^p$.

(a) X_1 *can neither remain 0 nor remain 1 in any trajectory.*
Suppose $X_1(t) = 0$ for all t, then $S_1, \dots S_M$ behave exactly as in Theorem A.3 with $k = M$. Since $p < M$, $X_1(t) = 1$ for some t. This contradicts the assumption that $X_1(t) = 0$ for all t. Hence, X_1 cannot remain 0 in any trajectory. On the other hand, suppose $X_1(t) = 1$ for all t, then $S_i(t) = 0$ for all t when $m + 1 \le i \le M$ because X_1 suppresses these nodes. Additionally, note that after the first initial M steps, the first m S_i must have a pattern of the form $(0 \cdots 010 \cdots 010 \cdots)$, where there are exactly $p{-}1$ zeros between two consecutive ones. Since the pattern shifts one unit to the right after each step, and since $p > \frac{m}{2}$ there exists time t_0 for which there is only one S_i with a value of 1. This implies $X_1(t_0 + 1) = 0$ which contradicts the assumption $X_1(t) = 1$ for all t. Hence, X_1 cannot remain 1 in any trajectory.

(b) *If* $X_1 = 0$, *it will next be excited by* $\widetilde{W} = 1 + 2^p$.
From Point (a) above, it is clear that when X_1 changes from $X_1 = 1$ to $X_1 = 0$, $S_i = (0 \cdots 010 \cdots)$. The first time two ones will appear in S_i will be when $(S_1 S_2 \cdots S_p S_{p+1} \cdots) = (10 \cdots 01 \cdots)$ and $X_1 = 0$, that is, when $(\widetilde{W}, X_1) = (1 + 2^p, 0)$. We next prove Points 3 and 4 by considering trajectories that start at $(\widetilde{W}, X_1) = (1 + 2^p, 0)$. Recall that a shift to the right in S_i is equivalent to a multiplication by 2 in $\widetilde{W}$ (see the proof of Lemma A.2).

(3) When $m{-}2 < p < M{-}1$, we can summarize the trajectories in Table A1. Note that since $p + 2 > m$, $S_i = 0$ for $i > p + 2$. Hence, $X_1 = \{\overline{11\underbrace{0 \cdots 0}_{p-2}}\}$.
(4) When $p = M{-}1$, the first time $X_1 = 0$ will be excited will be when $S_1 = 1$ and $S_M = 1$. In the next step, $S_2 = 1$ and $S_M = 0$. Hence, $X_1 = \{\overline{1\underbrace{0 \cdots 0}_{p-1}}\}$. See also Table A2.
(5) When $p \ge M$, there is at most one S_i having a value of 1 after the first M steps. Hence, $X_1 = (\bar{0})$.

■

Table A1. The state values of the trajectory through $(\widetilde{W}, X_1) = (1 + 2^p, 0)$ over one period when $m{-}2 < p < M{-}1$.

$(\widetilde{W}, X_1)$	$(1 + 2^p, 0)$	$(2 + 2^{p+1}, 1)$	$(2^2, 1)$	$(2^3, 0)$	$\cdots$	$(2^{p-1}, 0)$
$(\widetilde{W}, X_1)^+$	$(2 + 2^{p+1}, 1)$	$(2^2, 1)$	$(2^3, 0)$	$(2^4, 0)$	$\cdots$	$(1 + 2^p, 0)$

Table A2. The state values of the trajectory through $(\widetilde{W}, X_1) = (1 + 2^p, 0)$ over one period when $p = M{-}1$.

$(\widetilde{W}, X_1)$	$(1 + 2^p, 0)$	$(2, 1)$	$(2^2, 0)$	$\cdots$	$(2^{p-1}, 0)$
$(\widetilde{W}, X_1)^+$	$(2, 1)$	$(2^2, 0)$	$(2^3, 0)$	$\cdots$	$(1 + 2^p, 0)$