A Linear Representation of Dynamics of Boolean Networks*

Daizhan Cheng, Hongsheng Qi
Institute of Systems Science, Chinese Academy of Sciences, Beijing 100190, P.R.China
E-mail: dcheng@iss.ac.cn

Abstract:

A new matrix product, called semi-tensor product of matrices, is reviewed. Using it, a matrix expression of logic is proposed, where a logical variable is expressed as a vector, a logical function is expressed as a multiple linear mapping. Under this framework, a Boolean network equation is converted into an equivalent algebraic form as a conventional discrete-time linear system. Analyzing the transition matrix of the linear system, easily computable formulas are obtained to show (a) the number of fixed points; (b) the numbers of cycles of different lengths; (c) transient period, for all points to enter the set of attractors; (d) basin of each attractor. The corresponding algorithms are developed to calculate all the fixed points, cycles, transient period, and basins of attraction of all attractors.

Key Words: Boolean network, semi-tensor product, fixed point, cycle, transient period.

1 Introduction

Inspired by the Human Genome Project, a new view of biology, called the systems biology, is emerging. Systems biology does not investigate individual genes, proteins or cells, one in a time. Rather, it studies the behavior and relationships of all of the cells, proteins, DNA and RNA in a biological system, called cellular networks. The most active networks may be the genetic regulatory networks, which, reacting to the change of environment, determine the growth, replication, and death of cells. We refer to [20], [24] for a general introduction to systems biology.

The Boolean network, introduced firstly by Kauffman [21], and then developed by [1], [2], [31], [16], [3], [29], [12] and many others, becomes a powerful tool in describing, analyzing, and simulating the cellular networks. Hence, it has received the most attention, not only from the biology community, but also physics, systems science, etc. In this model, gene state is quantized to only two levels: True and False. Then the state of each gene is determined by the states of its neighborhood genes, using logical rules. It was shown that the Boolean network plays an important role in modeling cell regulation, because they can represent important features of living organisms [4], [19]. The structure of a Boolean network is described in terms of its cycles and the transient states that lead to them. Two different methods, iteration and scalar form, were developed in [17] to determine cyclic structure and the transient states that lead to them. In [13], a linear reduced scalar equation is derived from a more rudimentary nonlinear scalar equation to get immediate information about both cycle and transient structure of the network. Several useful Boolean networks have been analyzed and their cycles have been revealed (see, e.g., [17], [13] and references therein). It was pointed in [34] that finding fixed points and cycles of a Boolean network is an NP-complete problem. We refer to [5], [30], and [23] for some interesting recent developments on this topic.

* Supported partly by NNSF 60674022, 60736022, and 60221301 of China.

The analysis of the dynamics of Boolean networks focuses also on the link between the dependence between variables and the state space [32], [27].

The purpose of this paper is to propose a new method, which converts the network world into a new world, called state space, where the logical system becomes a conventional dynamic system. Then most of the tools developed in control theory for analysis and control of dynamic systems become applicable.

We first give a brief review to the semi-tensor product of matrices. It extends the conventional matrix product to two arbitrary matrices and keeps almost all properties of conventional matrix product unchanged. Using semi-tensor product, a matrix expression of logic is proposed, where a logic variable is expressed as a vector and a logical function is expressed as a multi-linear mapping. Under this algebraic expression a Boolean network equation can be expressed as a conventional discrete-time linear system

$$x(t+1) = Lx(t), \tag{1}$$

which contains complete information of the dynamics of a Boolean network. Analyzing the network transition matrix L, precise formulas are obtained to determine the number of fixed points and numbers of all possible cycles of different lengths. The minimum number of transient states that lead all states to cycles, called the transient period, is also determined by L. Then some easily computable algorithms are provided to construct all fixed points, cycles, transient period, and basins of attraction of all attractors.

The rest of the paper is organized as follows. Section 2 gives a brief review for the semi-tensor product of matrices. Some concepts and basic properties related to this paper are presented. The matrix expression of logic and its basic properties are discussed in Section 3. Using the tools developed in Section 3, a Boolean network equation is converted into a conventional discrete-time linear system in Section 4. Then in Section 5 the formulas are obtained for (a) the number of fixed points; (b) the numbers of cycles of different lengths; (c) transient period; (d) basin of attractors. The formulas for constructing them are also presented. Section 6 revisits three examples, which have been studied widely. Comparing our solutions with known results shows the advantage of our new approach. Section 7 is the concluding remarks, which provide a comparison of our algorithms with others and a clue for how to use this approach to control problems.

2 Semi-tensor Product

Throughout this paper "semi-tensor product" means the left semi-tensor product for multiplying dimensional case, which is reviewed in this session. We refer to [9] for right semi-tensor product, arbitrary dimensional case and much more details.

Definition 2.1 *1. Let X be a row vector of dimension np, and Y be a column vector with dimension p. Then we split X into p equal-size blocks as $X^1, \cdots, X^p$, which are $1 \times n$ rows. Define the STP, denoted by $\ltimes$, as*

$$\begin{cases} X \ltimes Y = \sum_{i=1}^{p} X^i y_i \in \mathbb{R}^n, \\ Y^T \ltimes X^T = \sum_{i=1}^{p} y_i (X^i)^T \in \mathbb{R}^n. \end{cases} \tag{2}$$

2. Let $A \in M_{m\times n}$ and $B \in M_{p\times q}$. If either n is a factor of p, say $nt = p$ and denote it as $A \prec_t B$, or p is a factor of n, say $n = pt$ and denote it as $A \succ_t B$, then we define the STP of A and B , denoted by $C = A \ltimes B$, as the following: C consists of $m \times q$ blocks as $C = (C^{ij})$ and each block is

$$C^{ij} = A^i \ltimes B_j, \quad i = 1, \cdots, m, \quad j = 1, \cdots, q,$$

where A^i is i-th row of A and B_j is the j-th column of B.

Some related fundamental properties of the STP are collected in the following:

Proposition 2.2 *The STP satisfies (as long as the related products are well defined)*
1. (Distributive rule)

$$\begin{array}{l} A \ltimes (\alpha B + \beta C) = \alpha A \ltimes B + \beta A \ltimes C; \\ (\alpha B + \beta C) \ltimes A = \alpha B \ltimes A + \beta C \ltimes A, \quad \alpha, \beta \in \mathbb{R}. \end{array} \tag{3}$$

2. (Associative rule)

$$A \ltimes (B \ltimes C) = (A \ltimes B) \ltimes C. \tag{4}$$

Proposition 2.3 *Assume $A \succ_k B$, then ("$\otimes$" is the Kronecker product)*

$$A \ltimes B = A(B \otimes I_k); \tag{5}$$

Assume $A \prec_k B$, then

$$A \ltimes B = (A \otimes I_k)B. \tag{6}$$

Proposition 2.4 *Assume $A \in M_{m\times n}$ is given.*
1. Let $Z \in \mathbb{R}^t$ be a row vector. Then

$$A \ltimes Z = Z \ltimes (I_t \otimes A); \tag{7}$$

2. Let $Z \in \mathbb{R}^t$ be a column vector. Then

$$Z \ltimes A = (I_t \otimes A) \ltimes Z. \tag{8}$$

Let $A \in M_{m\times n}$ and assume either m is a factor of n or n is a factor of m. Then

$$A^k := \underbrace{A \ltimes \cdots \ltimes A}_{k}$$

is well defined. Particularly, for a column (or a row) ξ, ξ^k is always well defined.

Define a delta set as $\Delta_k := \{\delta_k^i | i = 1, 2, \cdots, k\}$, where δ_k^i is the i-th column of the identity matrix I_k. A matrix $A \in M_{m\times n}$ is called a logic matrix if $m = 2^p$ and $n = 2^q$, for some $p, q \in \mathbb{Z}_+$, where $\mathbb{Z}_+$ is the set of natural numbers, and the columns of A are elements in Δ_{2^p}. Denote the set of logic matrices by $\mathcal{L}$. A straightforward computation shows the following:

Lemma 2.5 *Assume $A, B \in \mathcal{L}$, then $A \ltimes B \in \mathcal{L}$.*

Later on, one will see that the Boolean network related matrices are all in M_L. So the semi-tensor product between them is always well defined.

Assume a matrix $A = [\delta_m^{i_1}, \delta_m^{i_2}, \cdots, \delta_m^{i_n}]$, to save space we denote it as

$$A = \delta_m[i_1, i_2, \cdots, i_n].$$

Next, we define a *swap matrix*, $W_{[m,n]}$, which is an $mn \times mn$ matrix constructed in the following way: label its columns by $(11, 12, \cdots, 1n, \cdots, m1, m2, \cdots, mn)$ and its rows by $(11, 21, \cdots, m1, \cdots, 1n, 2n, \cdots, mn)$. Then its element in the position $((I, J), (i, j))$ is assigned as

$$w_{(IJ),(ij)} = \delta_{i,j}^{I,J} = \begin{cases} 1, & I = i \text{ and } J = j, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

When $m = n$ we briefly denote $W_{[n]} := W_{[n,n]}$.

Example 2.6 Let $m = 2$ and $n = 3$, the swap matrix $W_{[2,3]}$ is constructed as

$$W_{[2,3]} = \begin{array}{c} \begin{array}{cccccc} (11) & (12) & (13) & (21) & (22) & (23) \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ (11) \\ (21) \\ (12) \\ (22) \\ (13) \\ (23) \end{array} .$$

In condensed form we have

$$W_{[2,3]} = \delta_6[1, 3, 5, 2, 4, 6].$$

Proposition 2.7 *Let $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^n$ be two columns. Then*

$$W_{[m,n]} \ltimes X \ltimes Y = Y \ltimes X, \quad W_{[n,m]} \ltimes Y \ltimes X = X \ltimes Y. \tag{10}$$

Remark 2.8 *It is obvious that if $A \in M_{m\times s}$ and $B \in M_{s\times n}$, i.e., the conventional matrix product AB exists, then*

$$AB = A \ltimes B.$$

Hence the semi-tensor product is a generalization of conventional matrix product. Based on this, the notation "$\ltimes$" can be omitted. In the following all the matrix products are assumed to be semi-tensor product and the notation "$\ltimes$" is always omitted. As the conventional matrix product exists, the product turns to be conventional one automatically.

3 Matrix Expression of Logic

In this section we recall the matrix expression of logic. Under matrix expression a logical variable is expressed as a vector and an n-ary logical function is expressed by a 2×2^n matrix, called the *structure matrix* of the function. Then the logical action of the function over n logical variables becomes a matrix product of the structure matrix with n vectors. We refer to [8] for details.

First, we give some necessary notations and concerning results for logic. A logical domain, denoted by D, is defined as

$$D = \{T = 1, F = 0\}. \tag{11}$$

An n-ary logical function is a function $t : D^n \to D$. To use matrix expression we identify each element in D with a vector as $T \sim (1,0)^T$ and $F \sim (0,1)^T$, and denote

$$D_v = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}.$$

Using this vector expression, we can define the structure matrix of a logical function.

Definition 3.1 *A 2×2^n matrix M_σ is called the structure matrix of an n-ary logical function σ, if*

$$\sigma(P_1, P_2, \cdots, P_n) = M_\sigma P_1 P_2 \cdots P_n, \quad where \quad P_1, \cdots, P_n \in D_v. \tag{12}$$

If such a matrix exists, it uniquely determines the logic function. To show the existence of such a matrix for each logical function, we need some preparations. Define a matrix, called the *power-reducing matrix*, as (in a condensed form)

$$M_r = \delta_4[1, 4]. \tag{13}$$

Its name is from the following property.

Lemma 3.2 *Let $P \in D_v$. Then we have*

$$P^2 = M_r P. \tag{14}$$

The proof of the following theorem can be found in [25].

Theorem 3.3 *Any logical function $L(P_1, \cdots, P_n)$ with logical arguments $P_1, \cdots, P_n \in D_v$ can be expressed in a canonical form as*

$$L(P_1, \cdots, P_n) = M_L P_1 P_2 \cdots P_n, \tag{15}$$

where M_L is a 2×2^n matrix, called the structure matrix of L.

Next, we give some examples to illustrate the structure matrix.

Example 3.4 1. *Consider a fundamental unary logical function: Negation, $\neg P$, and four fundamental binary logical functions [26]: Disjunction, $P \vee Q$; Conjunction, $P \wedge Q$; Implication, $P \to Q$; Equivalence, $P \leftrightarrow Q$. Their structure matrices are as follows:*

$$\begin{array}{l} M_\neg := M_n = \delta_2[2, 1]; \\ M_\vee := M_d = \delta_2[1, 1, 1, 2]; \; M_\wedge := M_c = \delta_2[1, 2, 2, 2]; \\ M_\to := M_i = \delta_2[1, 2, 1, 1]; \; M_\leftrightarrow := M_e = \delta_2[1, 2, 2, 1]. \end{array} \tag{16}$$

2. Assume

$$L(P,Q) = (P \to Q) \vee (\neg P).$$

Using vector form of logic variables, Proposition 2.4, and the power-reducing matrix, we have

$$\begin{aligned} L(P,Q) &= M_d(M_iPQ)(M_nP) \\ &= M_dM_i(I_4 \otimes M_n)PQP \\ &= M_dM_i(I_4 \otimes M_n)PW_{[2]}PQ \\ &= M_dM_i(I_4 \otimes M_n)(I_2 \otimes W_{[2]})P^2Q \\ &= M_dM_i(I_4 \otimes M_n)(I_2 \otimes W_{[2]})M_rPQ \end{aligned}$$

We conclude that

$$M_L = M_dM_i(I_4 \otimes M_n)(I_2 \otimes W_{[2]})M_r = \delta_2[1,2,1,1].$$

□

In the following we use D and D_v alternatively for logical variables P, Q etc. without explanation. From the context it is easy to figure out which form is used then.

4 Dynamics of Boolean Networks

Definition 4.1 *[13] A Boolean network of a set of nodes $A_1, A_2, \cdots, A_n$ can be described as*

$$\begin{cases} A_1(t+1) = f_1(A_1(t), A_2(t), \cdots, A_n(t)), \\ A_2(t+1) = f_2(A_1(t), A_2(t), \cdots, A_n(t)), \\ \vdots \\ A_n(t+1) = f_n(A_1(t), A_2(t), \cdots, A_n(t)), \end{cases} \tag{17}$$

where f_i, $i = 1, 2, \cdots, n$ are n-ary logic functions.

We give a simple example to show the structure of a Boolean network.

Example 4.2 *Consider a Boolean network*

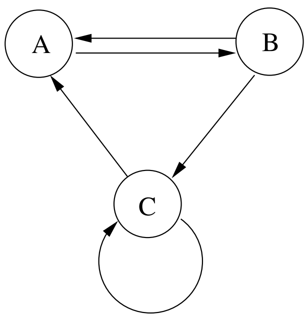

Fig. 1: Boolean network of (18)

Its dynamics is described as

$$\begin{cases} A(t+1) = B(t) \wedge C(t), \\ B(t+1) = \neg A(t), \\ C(t+1) = B(t) \vee C(t). \end{cases} \tag{18}$$

Our first purpose is to convert it into an algebraic form. Precisely, express it as a conventional discrete-time linear system. Using semi-tensor product, we define

$$x(t) = \ltimes_{i=1}^{n} A_i(t). \tag{19}$$

Remark 4.3 *Note that in (19) we defined a mapping $\ltimes_{i=1}^{n} : \Delta_2^n \to \Delta_{2^n}$. It is easy to prove that $\ltimes_{i=1}^{n}$ is a bijective mapping. In fact, Proposition 5.1 provides a precise formula to recover A_i, $1 \le i \le n$ from $x = \ltimes_{i=1}^{n} A_i$.*

Using Theorem 3.3, we can find structure matrices, $M_i = M_{f_i}$, $i = 1, \cdots, n$, such that

$$A_i(t+1) = M_i x(t), \quad i = 1, 2, \cdots, n. \tag{20}$$

Remark 4.4 *Note that usually the indegree is much less than n, that is, the right hand side of i-th equation of (17) may not have all $A_j's$, $j = 1, 2, \cdots, n$. Say, in the previous example, for node A we have*

$$A(t+1) = B(t) \wedge C(t).$$

In matrix form it is

$$A(t+1) = M_c B(t) C(t). \tag{21}$$

To get the form of (20), we can construct a dummy matrix as

$$E_d := \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

It is easy to prove that for any two logical variables X, Y,

$$E_d X Y = Y, \quad \text{or} \quad E_d W_{[2]} X Y = X.$$

Then we can rewrite (21) as

$$A(t+1) = M_c E_d A(t) B(t) C(t) = M_c E_d x(t).$$

Multiplying the equations in (20) together yields

$$x(t+1) = M_1 x(t) M_2 x(t) \cdots M_n x(t). \tag{22}$$

To simplify (22) we need some preparations:

Lemma 4.5 *Assume $P_k = A_1 A_2 \cdots A_k$, then*

$$P_k^2 = \Phi_k P_k, \tag{23}$$

where

$$\Phi_k = \prod_{i=1}^{k} I_{2^{i-1}} \otimes \left[\left(I_2 \otimes W_{[2,2^{k-i}]} \right) M_r \right].$$

Proof. We prove it by mathematical induction. When $k = 1$, using Lemma 3.2, we have

$$P_1^2 = A_1^2 = M_r A_1.$$

In above formula

$$\Phi_1 = \left(I_2 \otimes W_{[2,1]}\right) M_r.$$

Note that $W_{[2,1]} = I_2$, it follows that $\Phi_1 = M_r$. Hence (23) is true for $k = 1$. Assume (23) is true for $k = s$, then for $k = s + 1$ we have

$$\begin{array}{rcl} P_{s+1}^2 & = & A_1 A_2 \cdots A_{s+1} A_1 A_2 \cdots A_{s+1} \\ & = & A_1 W_{[2,2^s]} A_1 [A_2 \cdots A_{s+1}]^2 \\ & = & \left(I_2 \otimes W_{[2,2^s]}\right) A_1^2 [A_2 \cdots A_{s+1}]^2 \\ & = & \left[\left(I_2 \otimes W_{[2,2^s]}\right) M_r\right] A_1 [A_2 \cdots A_{s+1}]^2 \end{array}$$

Using induction assumption to the last factor of the above expression, we have

$$\begin{array}{rcl} P_{s+1}^2 & = & \left(I_2 \otimes W_{[2,2^s]}\right) M_r A_1 \left(\prod_{i=1}^s I_{2^{i-1}} \otimes \left[\left(I_2 \otimes W_{[2,2^{s-i}]}\right) M_r\right]\right) A_2 A_3 \cdots A_{s+1} \\ & = & \left[\left(I_2 \otimes W_{[2,2^s]}\right) M_r\right] \left(\prod_{i=1}^s I_{2^i} \otimes \left[\left(I_2 \otimes W_{[2,2^{s-i}]}\right) M_r\right]\right) A_1 A_2 \cdots A_{s+1}, \end{array}$$

which completes the proof. □

Lemma 4.6 *Equation (22) can be expressed as*

$$x(t+1) = Lx(t), \tag{24}$$

where

$$L = M_1 \prod_{j=2}^{n} \left[\left(I_{2^n} \otimes M_j\right) \Phi_n\right].$$

Proof. Note that from Lemma 4.5 we have

$$x(t)^2 = \Phi_n x(t).$$

Now

$$\begin{array}{rcl} x(t+1) & = & M_1 x(t) M_2 x(t) \cdots M_n x(t) \\ & = & M_1 (I_{2^n} \otimes M_2) x(t)^2 M_3 x(t) \cdots M_n x(t) \\ & = & M_1 (I_{2^n} \otimes M_2) \Phi_n x(t) M_3 x(t) \cdots M_n x(t) \\ & = & \cdots \\ & = & M_1 (I_{2^n} \otimes M_2) \Phi_n (I_{2^n} \otimes M_3) \Phi_n \cdots (I_{2^n} \otimes M_n) \Phi_n x(t). \end{array}$$

From Remark 4.3 it is easy to see that (24) is enough to describe the dynamics. □

Example 4.7 *Recall the Boolean network in Example 4.2.*

In algebraic form, we have

$$\begin{cases} A(t+1) = M_c B(t) C(t) \\ B(t+1) = M_n A(t) \\ C(t+1) = M_d B(t) C(t). \end{cases} \tag{25}$$

Setting $x(t) = A(t)B(t)C(t)$, we can calculate L as

$$
\begin{array}{rcl}
x(t+1) & = & M_c BCM_n AM_d BC \\
& = & M_c(I_4 \otimes M_n)BCAM_d BC \\
& = & M_c(I_4 \otimes M_n)(I_8 \otimes M_d)BCABC \\
& = & M_c(I_4 \otimes M_n)(I_8 \otimes M_d)W_{[2,4]}ABCBC \\
& = & M_c(I_4 \otimes M_n)(I_8 \otimes M_d)W_{[2,4]}ABW_{[2]}BCC \\
& = & M_c(I_4 \otimes M_n)(I_8 \otimes M_d)W_{[2,4]}(I_4 \otimes W_{[2]})AM_r BM_r C \\
& = & M_c(I_4 \otimes M_n)(I_8 \otimes M_d)W_{[2,4]}(I_4 \otimes W_{[2]})(I_2 \otimes M_r)(I_4 \otimes M_r)ABC
\end{array}
\tag{26}
$$

Then the system (18) is expressed in a matrix form as

$$
x(t+1) = Lx(t),
$$

where the network transition matrix is

$$
\begin{array}{rcl}
L & = & M_c(I_4 \otimes M_n)(I_8 \otimes M_d)W_{[2,4]}(I_4 \otimes W_{[2]})(I_2 \otimes M_r)(I_4 \otimes M_r) \\
& = & \delta_8[3, 7, 7, 8, 1, 5, 5, 6].
\end{array}
$$

5 Fixed Points and Cycles

To begin with, we consider how to get the logical variables $\{A_i(t)\}$ from $x(t) = A_1(t)A_2(t)\cdots A_n(t)$. It is clear that $x(t) \in \Delta_{2^n}$. It is easy to prove the following formula.

Proposition 5.1 *Assume $x(t) = \delta_{2^n}^i$. Define $B_0 := 2^n - i$, then $A_k(t)$ can be calculated inductively (in scalar form) as*

$$
\begin{cases}
A_k(t) = \left[\frac{B_{k-1}}{2^{n-k}}\right] \\
B_k = B_{k-1} - A_k(t) * 2^{n-k}, \quad k = 1, 2, \cdots, n,
\end{cases}
\tag{27}
$$

where in the first equation $[a]$ is the largest integer less than or equal to a.

Example 5.2 *Assume $x = A_1A_2A_3A_4A_5$ and $x = \delta_{32}^7$. Then $B_0 = 32 - 7 = 25$. It follows that $A_1 = [B_0/16] = 1$, $B_1 = B_0 - A_1 * (16) = 9$, $A_2 = [B_1/8] = 1$, $B_2 = B_1 - A_2 * 8 = 1$, $A_3 = [B_2/4] = 0$, $B_3 = B_2 - A_3 * 4 = 1$, $A_4 = [B_3/2] = 0$, $B_4 = B_3 - A_4 * 2 = 1$, $A_5 = [B_4/1] = 1$. We conclude that $A_1 = 1 \sim (1,0)^T$, $A_2 = 1 \sim (1,0)^T$, $A_3 = 0 \sim (0,1)^T$, $A_4 = 0 \sim (0,1)^T$, and $A_5 = 1 \sim (1,0)^T$.*

Consider the Boolean Network equation (24), and denote by L_i, $i = 1, 2, \cdots, 2^n$ the i-th column of the network transition matrix L. Then it is easy to see that $L_i \in \Delta_{2^n}, \forall\, i$.

Definition 5.3

1. *A state $x_0 \in \Delta_{2^n}$ is called a fixed point of system (24), if $Lx_0 = x_0$.*
2. *$\{x_0, Lx_0, \cdots, L^k x_0\}$ is called a cycle of system (24) with length k, if , $L^k x_0 = x_0$, and the elements in set $\{x_0, Lx_0, \cdots, L^{k-1}x_0\}$ are distinct.*

The next two theorems are main results of this paper, which show how many fixed points and cycles of different lengths a Boolean network has.

Theorem 5.4 *Consider the Boolean network system (17). $\delta_{2^n}^i$ is its fixed point, iff in its algebraic form (24) the diagonal element ℓ_{ii} of network transition matrix L equals 1. It follows that the number of equilibriums of system (17), denoted by N_e, equals the number of i, for which $\ell_{ii}=1$. Equivalently,*

$$N_e = \mathrm{Trace}(L). \tag{28}$$

Proof. Assume $\delta_{2^n}^i$ is its fixed point. Since, $L\delta_{2^n}^i = L_i$, the i-th column of L, it is clear that $\delta_{2^n}^i$ is its fixed point, iff $L_i = \delta_{2^n}^i$, which completes the proof. □

For statement ease, if $\ell_{ii}=1$, the i-th column of L is called a *diagonal nonzero column* of L.

Next, we consider the cycles of the Boolean network system (17). We need a notation: Let $k \in \mathbb{Z}_+$. A positive integer $s \in \mathbb{Z}_+$ is called a *proper factor* of k if $s<k$ and $k/s \in \mathbb{Z}_+$. The set of proper factors of k is denoted by $\mathcal{P}(k)$. For instance, $\mathcal{P}(8)=\{1,2,4\}$, $\mathcal{P}(12)=\{1,2,3,4,6\}$, etc.

Using a similar argument as for Theorem 5.4, we can have the following theorem.

Theorem 5.5 *The number of length s cycles, N_s, is inductively determined by*

$$\begin{cases} N_1 = N_e, \\ N_s = \frac{\mathrm{Trace}(L^s) - \sum\limits_{k\in\mathcal{P}(s)} kN_k}{s}, \quad 2 \le s \le 2^n. \end{cases} \tag{29}$$

Proof. First, if $\delta_{2^n}^i$ is an element of a cycle of length s, then $L^s\delta_{2^n}^i = \delta_{2^n}^i$. From the proof of Theorem 5.4 $(L^s)_i$ is a diagonal nonzero column of L^s, which adds 1 to the Trace(L^s). Note that if $\delta_{2^n}^k$ is an element of a cycle of length $k \in \mathcal{P}(s)$, we also have $L^s\delta_{2^n}^k = \delta_{2^n}^k$, and $(L^s)_k$ will also add 1 to the Trace(L^s). Such diagonal elements have to be subtracted from the Trace(L^s). Taking this into consideration, formula (29) is obvious.

As for the upper boundary of s, note that since $x(t)$ can only have at most 2^n possible values, the length of any cycle is less than or equal to 2^n. □

Next, we consider how to find the cycles. If

$$\mathrm{Trace}(L^s) - \sum_{k\in\mathcal{P}(s)} kN_k > 0, \tag{30}$$

then we call "s" a *non-trivial power*.

Assume s is a non-trivial power. Denote by ℓ_{ii}^s the (i,i)-th entrance of matrix L^s. Then we define

$$C_s = \{i | \ell_{ii}^s = 1\}, \quad s = 1, 2, \cdots, 2^n,$$

and

$$D_s = C_s \bigcap_{i\in\mathcal{P}(s)} C_i^c,$$

where C_i^c is the compliment of C_i.

From the above argument the following is obvious.

Proposition 5.6 *Let $x_0 = \delta_{2^n}^i$. Then $\{x_0, Lx_0, \cdots, L^s x_0\}$ is a cycle with length s, iff $i \in D_s$.*

Theorem 5.5 and Proposition 5.6 provide a simple algorithm for constructing cycles. We give some examples to show the algorithm.

Example 5.7 *Recall Example 4.2. It is easy to check that*

$$\mathrm{Trace}(L^t) = 0, \quad t \leq 3,$$

and

$$\mathrm{Trace}(L^t) = 4, \quad t \geq 4.$$

Using Theorem 5.5, we conclude that there is only one cycle of length 4*. Moreover, note that*

$$L^4 = \delta_8[1,3,3,1,5,7,7,3],$$

then each diagonal nonzero column can generate the cycle. Say, choosing $Z = \delta_8^1$*, then we have*

$$LZ = \delta_8^3, \quad L^2 Z = \delta_8^7, \quad L^3 Z = \delta_8^5, \quad L^4 Z = Z.$$

Using Proposition 5.1 to convert the vector forms back to the scalar form of $A(t)$*,* $B(t)$*, and* $C(t)$*, we have the cycle as* $(111) \to (101) \to (001) \to (011) \to (111)$*.*

In the following we consider the transient period, i.e., the minimum transient states that leads any point to the limit set, Ω, which is the union of all fixed points and cycles. First, it is easy to see that there are only $r := 2^n \times 2^n = 2^{2n}$ different logic matrices. Hence, if we construct a sequence of $r+1$ matrices as

$$L^0 = I_{2^n}, L, L^2, \cdots, L^r,$$

then there must be two equal matrices. Let $r_0 < r$ be the smallest i such that L^i appears again in the sequence. That is, there exists a $k > i$ such that $L^i = L^k$. Precisely,

$$r_0 = \min\left\{ i \,\middle|\, L^i \in \{L^{i+1}, L^{i+2}, \cdots, L^r\},\ 0 \leq i < r \right\}. \tag{31}$$

Then such r_0 exists. The following proposition is obvious.

Proposition 5.8 *Let* r_0 *be defined as in (31). Then starting from any state, the trajectory will enter into a cycle after* r_0 *iterations.*

For a given state x_0, the *transient period* of x_0, denoted by $T_t(x_0)$, is the smallest k, satisfying $x(0) = x_0$ and $x(k) \in \Omega$. The *transient period* of a Boolean network, denoted by T_t, is defined as

$$T_t = \max_{\forall\, x \in \Delta_{2^n}} \left(T_t(x) \right).$$

In fact, we can show that r_0 is the transient period of the system.

Theorem 5.9 *The* r_0 *defined in (31) is the transient period of the system. That is,*

$$T_t = r_0. \tag{32}$$

Proof. First, assume

$$L^{r_0} = L^{r_0+T}, \tag{33}$$

and $T > 0$ is the smallest positive number, which verifies (33). By definition, $r_0 + T \leq r$. We first claim that if there is a cycle of length t, then t is a factor of T. We prove the claim by contradiction: Assume $T(mod\ t) = s$ and $1 \leq s < t$. Let x_0 be a state on the cycle. Then $L^{r_0}x_0$ is also a state on the same cycle. Hence

$$L^{r_0}x_0 = L^{r_0+T}x_0 = L^T L^{r_0}x_0 = L^s(L^{r_0}x_0) \neq L^{r_0}x_0,$$

which is a contradiction.

From (33) and the definition of T_t it is obvious that $T_t \leq r_0$. To prove $T_t = r_0$, we assume $T_t < r_0$. By definition, for any x, $L^{T_t}x$ is on a cycle, which has length as a factor of T. Hence

$$L^{T_t}x = L^T L^{T_t}x = L^{T_t+T}x, \quad \forall\, x. \tag{34}$$

It is easy to check that if for any $x \in \Delta_{2^n}$ (34) holds, then $L^{T_t} = L^{T_t+T}$, which is a contradiction to the definition of r_0. □

Remark 5.10 *1. According to Theorem 5.9 it is clear that $r_0 \leq 2^n$, because the transient period can not be larger than 2^n.*

2. Let $r_0 = T_t$ be defined as in above, and $T > 0$ is the smallest positive number, which verifies (33). Then it is easy to see that T is the least common multiplier of the lengths of all cycles.

Finally, we consider the basin of each attractor. Denote

$$\Omega := \cup_{i=1}^k C_i,$$

where $\{C_i \,|\, i = 1, \ldots, k\}$ is the set of attractors. We give the following definition:

Definition 5.11 *1. Denote by $x(t, p)$ the trajectory with initial value $x(0, p) = p$. S_i is called the basin of attractor C_i, if S_i is the set of points, which will converge to C_i. Precisely, $p \in S_i$, iff, the trajectory satisfies $x(t, p) \in C_i$ for $t \geq T_t$;*

2. q is called the parent state of p, if $p = x(1, q)$.

Remark 5.12

- *Let $C \subset D^n$. Denote by*

$$L^{-1}(C) = \{q \,|\, Lq \in C\}.$$

Then the set of parent states of p is $L^{-1}(p)$.

- *$D^n = \cup_{i=1}^k S_i$. Moreover, since $\{S_i \,|\, i = 1, \cdots, k\}$ are disjointed, it is a partition of the state space D^n.*

What remains now is how to find S_i. Starting from each point $p \in C_i$. If we can find its parent states $L^{-1}(p)$, then for each point $p_1 \in L^{-1}(p)$, we can also find $L^{-1}(p_1)$. Continuing this process and after T_t times we get a tree of states, which converge to p. Summarizing above arguments, we have

Proposition 5.13

$$S_i = L^{-1}(C_i) \cup L^{-2}(C_i) \cup \cdots \cup L^{-T_t}(C_i) \tag{35}$$

Finally let us see how to find $L^{-1}(p)$. Denote the j-th column of L by L_j. Then it is easy to verify that

Proposition 5.14

$$\begin{cases} L^{-1}(p) = \left\{ \delta_{2^n}^j \,\middle|\, L_j = p \right\} \\ L^{-k}(p) = \left\{ \delta_{2^n}^j \,\middle|\, L_j^k = p \right\}, \quad k = 2, \cdots, T_i. \end{cases} \tag{36}$$

Example 5.15 *Recall Example 4.2. It is easy to check that* $r_0 = 3$ *and*

$$L^3 = L^7 = \delta_8[5, 1, 1, 5, 7, 3, 3, 1].$$

We then have the transient period $T_t = 3$. *Using Propositions 5.13 and 5.14, we may choose any point* $p \in C$, *where* C *is its only cycle, to find* $L^{-1}(p)$, $L^{-2}(p)$, *and* $L^{-3}(p)$.

Say, choosing $p = (011) \sim \delta_8^5$. *Then we can see* L_6 *and* L_7 *equal to* p. *So* $\delta_8^6 \sim (010)$ *and* $\delta_8^7 \sim (001)$ *form* $L^{-1}(p)$. *But* (001) *is on the cycle, so we are interested in* $p_1 = \delta_8^6 \sim (010)$. *Now since only* $L_8 = p_1$, *we have* $L^{-1}(p_1) = \{\delta_8^8\}$. *Let* $p_2 = \delta_8^8 \sim (000)$. *Only* $L_4 = p_2$, *so we have* $p_3 := \delta_8^4 \sim (100) \in L^{-1}(p_2)$. *So we have a chain* $p_3 \to p_2 \to p_1 \to p$. *Choosing* $q = (001) \sim \delta_8^7$. *Then* $L_2 = L_3 = q$. *Since* $\delta_8^3 \sim (101)$ *is on the cycle, we choose* $q_1 = \delta_8^2 \sim (110)$. *It is easy to check that* $L^{-1}(q_1) = \varnothing$, *and we have no more parent states. Finally, we get the state space graph of the network in Example 4.2 as in Fig. 2. (Note that here we use* L^{-1} *only. The iterative calculation provides whole tree. If we need only the basins* S_i, L^{-k} *are convenient.)*

$$\begin{tikzcd} 100 \arrow[r] & 000 \arrow[r] & 010 \arrow[r] & 011 \arrow[r] & 111 \arrow[d] \\ & & 110 \arrow[r] & 001 \arrow[u] & 101 \arrow[l] \end{tikzcd}$$

Fig. 2: The State Space Graph

In literatures of Boolean networks $A + B$ and AB are often used. Using standard logical notations $A + B := A \bar{\vee} B$, and $AB := A \wedge B$, where $\bar{\vee}$ is called the "exclusive or", that is, $A \bar{\vee} B$ is true whenever either A or B, but not both are true [26].

Example 5.16 *[13] Consider the following Boolean network*

$$\begin{cases} A(t+1) = B(t)C(t) \\ B(t+1) = 1 + A(t) \\ C(t+1) = B(t). \end{cases} \tag{37}$$

It is easy to calculate that

$$\begin{aligned} x(t+1) &= M_c BCM_n AB \\ &= M_c(I_4 \otimes M_n)BCAB \\ &= M_c(I_4 \otimes M_n)W_{[2,4]}ABCB \\ &= M_c(I_4 \otimes M_n)W_{[2,4]}ABW_{[2]}BC \\ &= M_c(I_4 \otimes M_n)W_{[2,4]}(I_4 \otimes W_{[2]})AM_r BC \\ &= M_c(I_4 \otimes M_n)W_{[2,4]}(I_4 \otimes W_{[2]})(I_2 \otimes M_r)x(t) \\ &:= Lx(t). \end{aligned}$$

L follows immediately as

$$L = \delta_8[3, 7, 8, 8, 1, 5, 6, 6].$$

$$\mathrm{Trace}(L^k) = 0, \quad k = 1, 2, 3, 4,$$

and

$$L^5 = \delta_8[1, 3, 3, 3, 5, 6, 8, 8],$$

$$\mathrm{Trace}(L^5) = 5.$$

Choosing any diagonal nonzero column of L^5*, say,* $X = \delta_8^1 \sim (111)$*, we can generate a length* 5 *cycle as* $X \to LX \to L^2X \to L^3X \to L^4X \to L^5X = X$*, where* $LX = \delta_8^3 \sim (101)$*,* $L^2X = \delta_8^8 \sim (000)$*,* $L^3X = \delta_8^6 \sim (010)$*,* $L^4X = \delta_8^5 \sim (011)$*,* $L^5X = \delta_8^1 \sim (111)$*.*

It is easy to check that $r_0 = 2$ *and* $L^2 = L^7$*. That is,* $T_t = 2$*. Since* $T = 5$*, there are no cycles of length longer than* 5*.*

Choosing $Z = \delta_8^2 \sim (110)$*, then*

$$LZ = \delta_8^7 \sim (001), \quad L^2Z = \delta_8^6 = L^3X;$$

Choosing $Y = \delta_8^4 \sim (100)$*, then*

$$LY = \delta_8^8 = L^2X.$$

The state space graph (Fig. 3) coincides with the one in [13].

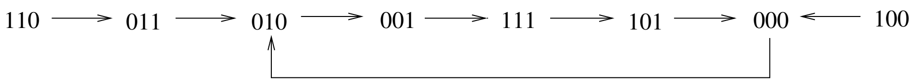

Fig. 3: State Space Graph of (37)

6 Some Useful Examples

In this section we revisit some examples which have been investigated in several literatures.

The first example is a biochemical network of coupled oscillations in the cell cycle [14].

Example 6.1 *Consider the following Boolean network*

$$\begin{cases} A(t+3) = A(t)B(t+1) + 1 \\ B(t+3) = A(t+1)B(t) + 1. \end{cases} \tag{38}$$

Converting it to matrix form, we have

$$\begin{cases} A(t+3) = M_x A(t)B(t+1) \\ B(t+3) = M_x A(t+1)B(t), \end{cases} \tag{39}$$

where

$$M_x = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

From above equation the period five and period ten orbits can be found directly [17]. In [13] it was proved that the network can contain fixed points or cycles of length two, five or ten. (It was also pointed out in [13] that there are no fixed points.) Using the approach developed in this paper, we will provide complete description for the state space graph of this system.

Note that we need initial states $\{A(0), B(0), A(1), B(1), A(2), B(2)\}$ *to determine the dynamics, so we can re-scale* $\tau = 3t$ *as new unit of time. Denote* $E(t) = A(t)$, $F(t) = A(t+1)$, $G(t) = A(t+2)$, $H(t) = B(t)$, $I(t) = B(t+1)$, $J(t) = B(t+2)$, *then by substitution we can get*

$$\begin{array}{rcl} E(t+3) & = & M_x E(t) I(t), \\ H(t+3) & = & M_x F(t) H(t), \\ F(t+3) & = & A(t+4) = M_x A(t+1) B(t+2) = M_x F(t) J(t), \\ I(t+3) & = & B(t+4) = M_x A(t+2) B(t+1) = M_x G(t) I(t), \\ G(t+3) & = & A(t+5) = M_x A(t+2) B(t+3) = M_x G(t) M_x F(t) H(t) = M_x (I_2 \otimes M_x) G(t) F(t) H(t), \\ J(t+3) & = & B(t+5) = M_x A(t+3) B(t+2) = M_x^2 E(t) B(t+1) B(t+2) = M_x^2 E(t) I(t) J(t). \end{array} \tag{40}$$

Define

$$x(t) := E(t)H(t)F(t)I(t)G(t)J(t),$$

then we have

$$x(t+3) = M_x EIM_x FHM_x FJM_x GIM_x (I_2 \otimes M_x) GFHM_x^2 EIJ. \tag{41}$$

Using Propositions 2.3, 2.4, and 2.7, a simple computation shows that

$$x(t+3) = Lx(t), \tag{42}$$

where

$$\begin{aligned} L = & M_x (I_{2^2} \otimes M_x)(I_{2^4} \otimes M_x)(I_{2^2} \otimes W_{[2]})(I_{2^3} \otimes M_r)(I_{2^5} \otimes M_x)(I_{2^2} \otimes W_{[2,2^4]})(I_2 \otimes M_r) \\ & \ltimes (I_{2^6} \otimes (M_x(I_2 \otimes M_x)))(I_{2^5} \otimes M_r)(I_{2^3} \otimes W_{[2,2^4]})(I_{2^2} \otimes M_r)(I_{2^4} \otimes W_{[2,4]})(I_{2^3} \otimes M_r) \\ & \ltimes (I_{2^6} \otimes M_x^2)(I_2 \otimes W_{[2,2^5]}) M_r (I_2 \otimes W_{[4,2]})(I_{2^4} \otimes W_{[2,4]})(I_{2^3} \otimes M_r)(I_{2^4} \otimes W_{[2]})(I_{2^5} \otimes M_r). \end{aligned}$$

It can be calculated as

$$\begin{aligned} L = \delta_{64}[\, & 61 \;\; 53 \;\; 57 \;\; 49 \;\; 26 \;\; 17 \;\; 26 \;\; 17 \;\; 39 \;\; 39 \;\; 33 \;\; 33 \;\; 4 \;\; 3 \;\; 2 \;\; 1 \\ & 47 \;\; 39 \;\; 41 \;\; 33 \;\; 12 \;\; 3 \;\; 10 \;\; 1 \;\; 39 \;\; 39 \;\; 33 \;\; 33 \;\; 4 \;\; 3 \;\; 2 \;\; 1 \\ & 30 \;\; 21 \;\; 26 \;\; 17 \;\; 26 \;\; 17 \;\; 26 \;\; 17 \;\; 8 \;\; 7 \;\; 2 \;\; 1 \;\; 4 \;\; 3 \;\; 2 \;\; 1 \\ & 16 \;\; 7 \;\; 10 \;\; 1 \;\; 12 \;\; 3 \;\; 10 \;\; 1 \;\; 8 \;\; 7 \;\; 2 \;\; 1 \;\; 4 \;\; 3 \;\; 2 \;\; 1\,]. \end{aligned}$$

Then we can check $\mathrm{Trace}(L^k)$, $k = 1, 2, \cdots, 64$ *and look for nontrivial power s. They can be easily calculated as*

$$\mathrm{Trace}(L^2) = 2, \quad \mathrm{Trace}(L^5) = 5, \quad \mathrm{Trace}(L^{10}) = 17.$$

Using Theorem 5.5, we conclude that the system doesn't have fixed point, it has one cycle of length 2, *one cycle of length* 5 *and one cycle of length* 10.

Next, we can find out the cycles. Consider L^2. *It is easy to figure out that the* 26*th column of it is a diagonal nonzero column. Then we can use it to general the cycle of length* 2. *Since* $L\delta^{26} = \delta^{39}$, *and*

$L\delta^{39} = \delta^{26}$, we have a cycle of length 2. Using formula (27) to convert δ^{26} and δ^{39} back to binary form, we have $\delta^{26} \sim (100110)$, $\delta^{39} \sim (011001)$. Note that $x(t) = A(t)B(t)A(t+1)B(t+1)A(t+2)B(t+2)$, we have

$$(10) \to (01) \to (10) \to (01) \to (10) \to (01) \to (10) \to (01) \to (10) \to \cdots .$$

which is a cycle of length 2.

Similarly, since δ^1 is a diagonal nonzero column of L^5, then δ^1, $L\delta^1 = \delta^{61}$, $L^2\delta^1 = \delta^4$, $L^3\delta^1 = \delta^{49}$, $L^4\delta^1 = \delta^{16}$ forms a cycle of length 5. Converting them to binary form yields the following cycle:

$$(11) \to (11) \to (11) \to (00) \to (00) \to (11) \to (11) \to (11) \to (00) \to (00) \to \cdots .$$

since δ^2 is a diagonal nonzero column of L^{10}, then δ^2, $L\delta^2 = \delta^{53}$, $L^2\delta^2 = \delta^{12}$, $L^3\delta^2 = \delta^{33}$, $L^4\delta^2 = \delta^{30}$, $L^5\delta^2 = \delta^3$, $L^6\delta^2 = \delta^{57}$, $L^7\delta^2 = \delta^8$, $L^8\delta^2 = \delta^{17}$, $L^9\delta^2 = \delta^{47}$ form a cycle of length 10. Converting them to binary form yields the following cycle:

$$\begin{aligned} &(11) \to (11) \to (10) \to (00) \to (10) \to (11) \to (11) \to (01) \to (00) \to (01) \to \\ &(11) \to (11) \to (10) \to (00) \to (10) \to (11) \to (11) \to (01) \to (00) \to (01) \to \cdots . \end{aligned}$$

Our result coincides the one in [17].

Finally, we consider the transient period. Since it is easy to check that the first repeating L^k is $L^2 = L^{12}$, then $r_0 = 2$. Since each iteration contains 3 time units, we can only conclude that the transient period satisfies $3 < T_t \leq 6$. By analyzing this particular system [13] pointed that $T_t \leq 4$.

The following example is the Boolean model of cell growth, differentiation, and apoptosis (programmed cell death) introduced by [18] and re-investigated in [13].

Example 6.2

$$\begin{cases} A(t+1) = K(t) + K(t)H(t) \\ B(t+1) = A(t) + A(t)C(t) \\ C(t+1) = 1 + D(t) + D(t)I(t) \\ D(t+1) = J(t)K(t) \\ E(t+1) = 1 + C(t) + C(t)F(t) \\ F(t+1) = E(t) + E(t)G(t) \\ G(t+1) = 1 + B(t)E(t) \\ H(t+1) = F(t) + F(t)G(t) \\ I(t+1) = H(t) + H(t)I(t) \\ J(t+1) = J(t) \\ K(t+1) = K(t). \end{cases} \tag{43}$$

Converting into matrix form yields

$$
\begin{cases}
A(t+1) = M_n M_i K(t) H(t) \\
B(t+1) = M_n M_i A(t) C(t) \\
C(t+1) = M_i D(t) I(t) \\
D(t+1) = M_c J(t) K(t) \\
E(t+1) = M_i C(t) F(t) \\
F(t+1) = M_n M_i E(t) G(t) \\
G(t+1) = M_n M_c B(t) E(t) \\
H(t+1) = M_n M_i F(t) G(t) \\
I(t+1) = M_n M_i H(t) I(t) \\
J(t+1) = J(t) \\
K(t+1) = K(t).
\end{cases}
\tag{44}
$$

It is easy to calculate the structure matrix L as

$$
\begin{aligned}
&L = \\
&M_nM_i(I_2 \otimes (I_2 \otimes M_nM_i(I_2 \otimes (I_2 \otimes M_i(I_2 \otimes (I_2 \otimes M_c(I_2 \otimes (I_2 \otimes M_i(I_2 \otimes (I_2 \otimes M_nM_i(I_2 \otimes (I_2 \otimes \\
&M_nM_c(I_2 \otimes (I_2 \otimes M_nM_i(I_2 \otimes (I_2 \otimes M_nM_i))))))))))))))))(I_2 \otimes W_{[2]})W_{[2]}(I_{2048} \otimes W_{[2]})(I_{1024} \otimes W_{[2]}) \\
&(I_{512} \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{128} \otimes W_{[2]})(I_{64} \otimes W_{[2]})(I_{32} \otimes W_{[2]})(I_{16} \otimes W_{[2]})(I_8 \otimes W_{[2]})(I_4 \otimes W_{[2]}) \\
&(I_2 \otimes W_{[2]})(I_8 \otimes W_{[2]})(I_4 \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{128} \otimes W_{[2]})(I_{64} \otimes W_{[2]})(I_{32} \otimes W_{[2]})(I_{16} \otimes W_{[2]}) \\
&(I_8 \otimes W_{[2]})(I_{32} \otimes W_{[2]})(I_{16} \otimes W_{[2]}))(I_{1024} \otimes W_{[2]}(I_{512} \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{128} \otimes W_{[2]})(I_{64} \otimes W_{[2]}) \\
&(I_{32} \otimes W_{[2]})(I_{4096} \otimes W_{[2]})(I_{2048} \otimes W_{[2]})(I_{1024} \otimes W_{[2]})(I_{512} \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{128} \otimes W_{[2]})(I_{64} \otimes W_{[2]}) \\
&(I_{2048} \otimes W_{[2]})(I_{1024} \otimes W_{[2]})(I_{512} \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{128} \otimes W_{[2]})(I_{8192} \otimes W_{[2]})(I_{4096} \otimes W_{[2]}) \\
&(I_{2048} \otimes W_{[2]})(I_{1024} \otimes W_{[2]})(I_{512} \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{8192} \otimes W_{[2]})(I_{4096} \otimes W_{[2]})(I_{2048} \otimes W_{[2]}) \\
&(I_{1024} \otimes W_{[2]})(I_{512} \otimes W_{[2]})(I_{16384} \otimes W_{[2]})(I_{8192} \otimes W_{[2]})(I_{4096} \otimes W_{[2]})(I_{2048} \otimes W_{[2]})(I_{1024} \otimes W_{[2]}) \\
&(I_{2048} \otimes W_{[2]})(I_{32768} \otimes W_{[2]})(I_{16384} \otimes W_{[2]})(I_{8192} \otimes W_{[2]})(I_{4096} \otimes W_{[2]})(I_{8192} \otimes W_{[2]}) \\
&(I_{65536} \otimes W_{[2]})(I_{32768} \otimes W_{[2]})(I_{16384} \otimes W_{[2]})(I_{32768} \otimes W_{[2]})(I_{131072} \otimes W_{[2]})(I_{65536} \otimes W_{[2]}) \\
&(I_2 \otimes (I_2 \otimes M_r(I_2 \otimes (I_2 \otimes M_r(I_2 \otimes M_r(I_2 \otimes M_r(I_2 \otimes M_r(I_2 \otimes M_r(I_2 \otimes M_rM_r))))))))).
\end{aligned}
$$

Since it is a $2^{11} \times 2^{11}$ matrix, even using condensed form, it is still too long to show here. But it can be easily stored in a computer. It is easy to calculate that

$$
\mathrm{Trace}(L) = 3; \quad \mathrm{Trace}(L^9) = 12,
$$

and there are no other non-trivial powers. We conclude that there are only 3 fixed points and 1 cycle of length 9. Finding diagonal nonzero columns of L and L^9 respectively, it is easy to figure out that the three fixed points are

$$
\begin{aligned}
E_1 &= (1,0,1,0,0,0,1,0,0,0,1), \\
E_2 &= (0,0,1,0,0,0,1,0,0,1,0), \\
E_3 &= (0,0,1,0,0,0,1,0,0,0,0).
\end{aligned}
$$

The only cycle of length 9 is

$$
\begin{aligned}
&(11011101011) \to (01011101111 \to (00111101011) \to (00011111111) \to (00111010011) \to \\
&(10010010011) \to (11011010011) \to (11011000011) \to (11011100011) \to (11011101011).
\end{aligned}
$$

The minimum power for repeating L^k is $L^{10} = L^{19}$, so the transient period $T_t = 10$.

Remark 6.3 *It was shown in [18] that a nontrivial growth attractor exists. Our result shows that there are exactly 3 fixed points and one cycle of length 9. As $J = K = D = 1$, both [18] and [13] showed that there exists the cycle. Our result agrees with them. In the case of $J = K = D = 1$, it is easy to check that The transient period is still $T_t = 10$. [13] claimed that $T_t \leq 7$. This is incorrect. Consider $x(0) := x_0 = (01111000011)$. It is easy to calculate that $x(10) = (10010110011)$, which is not in the cycle, and $x(11) = (11011010011)$ which is in the cycle. So $T_t(x_0) = 10$.*

The following example is from [14] and re-investigated in [17].

Example 6.4 *Consider the following system*

$$\begin{cases} A(t+1) = 1 + C(t) + F(t) + C(t)F(t) \\ B(t+1) = A(t) \\ C(t+1) = B(t) \\ D(t+1) = 1 + C(t) + F(t) + I(t) + C(t)F(t) + C(t)I(t) + F(t)I(t) + C(t)F(t)I(t) \\ E(t+1) = D(t) \\ F(t+1) = E(t) \\ G(t+1) = 1 + F(t) + I(t) + F(t)I(t) \\ H(t+1) = G(t) \\ I(t+1) = H(t). \end{cases} \tag{45}$$

The matrix form of the above equation is

$$\begin{cases} A(t+1) = M_n M_d C F \\ B(t+1) = A \\ C(t+1) = B \\ D(t+1) = M_c^2 M_n I M_n C M_n F \\ E(t+1) = D \\ F(t+1) = E \\ G(t+1) = M_n M_d F I \\ H(t+1) = G \\ I(t+1) = H. \end{cases} \tag{46}$$

Let $x(t) = A(t)B(t)C(t)D(t)E(t)F(t)G(t)H(t)I(t)$, and $x(t+1) = Lx(t)$. Then

$$\begin{aligned} &L = \\ &M_n M_d (I_2 \otimes (I_2 \otimes (I_2 \otimes (I_2 \otimes M_c M_c M_n (I_2 \otimes M_n (I_2 \otimes M_n (I_2 \otimes (I_2 \otimes (I_2 \otimes M_n M_d)))))))))(I_2 \otimes W_{[2]}) \\ &W_{[2]}(I_4 \otimes W_{[2]})(I_2 \otimes W_{[2]})(I_{16} \otimes W_{[2]})(I_8 \otimes W_{[2]})(I_{64} \otimes W_{[2]})(I_{32} \otimes W_{[2]})(I_{16} \otimes W_{[2]})(I_{128} \otimes W_{[2]}) \\ &(I_{64} \otimes W_{[2]})(I_{32} \otimes W_{[2]})(I_{128} \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{1024} \otimes W_{[2]})(I_{512} \otimes W_{[2]})(I_{2048} \otimes W_{[2]})(I_{1024} \otimes W_{[2]}) \\ &(I_2 \otimes (I_2 \otimes M_r (I_2 \otimes (I_2 \otimes (I_2 \otimes M_r M_r (I_2 \otimes (I_2 \otimes (I_2 \otimes M_r)))))))). \end{aligned}$$

The non-trivial powers are $\mathrm{Trace}(L^2) = 4$*, and* $\mathrm{Trace}(L^6 = 64)$*. It follows from Theorem 5.5 that there are only* 2 *cycles of length* 2 *and* 10 *cycles of length* 6*. Searching diagonal nonzero columns of* L^2 *yields*

$$(101101101) \to (010010010) \to (101101101)$$
$$(101000010) \to (010000101) \to (101000010)$$

Searching diagonal nonzero columns of L^6 *yields*

$$(111111111) \to (011011011) \to (001001001) \to$$
$$(000000000) \to (100100100) \to (110110110) \to$$
$$(111111111).$$

$$(111110110) \to (011011111) \to (001001011) \to$$
$$(000000001) \to (100000000) \to (110100100) \to$$
$$(111110110).$$

$$(111101101) \to (011010010) \to (001001101) \to$$
$$(000000010) \to (100100101) \to (110010010) \to$$
$$(111101101).$$

$$(111100100) \to (011010110) \to (001001111) \to$$
$$(000000011) \to (100000001) \to (110000000) \to$$
$$(111100100).$$

$$(111011011) \to (011001001) \to (001000000) \to$$
$$(000000100) \to (100100110) \to (110110111) \to$$
$$(111011011).$$

$$(111010010) \to (011001101) \to (001000010) \to$$
$$(000000101) \to (100000010) \to (110100101) \to$$
$$(111010010).$$

$$(111001001) \to (011000000) \to (001000100) \to$$
$$(000000110) \to (100100111) \to (110010011) \to$$
$$(111001001).$$

$$(111000000) \to (011000100) \to (001000110) \to$$
$$(000000111) \to (100000011) \to (110000001) \to$$
$$(111000000).$$

$$(101101111) \to (010010011) \to (101001001) \to$$
$$(010000000) \to (101100100) \to (010010110) \to$$
$$(101101111).$$

$$(101100110) \to (010010111) \to (101001011) \to$$
$$(010000001) \to (101000000) \to (010000100) \to$$
$$(101100110).$$

Finally, we can calculate that the first repeating L^k *is* $L^3 = L^9$*. So,* $T_t = 3$*.*

Remark 6.5 *In [17] it was shown that there are no fixed points, and there are* 2 *cycles of length* 2*. Our results about fixed points and cycles with length* 2 *coincide with [17]. [17] pointed out only* 6 *cycles of length* 6*. According to our result, there are exactly* 10 *cycles of length* 6*.*

7 Concluding Remarks

In this paper the topological structure of Boolean networks has been investigated. Four major objects were considered: (1) fixed points; (2) cycles of different lengths; (3) transient period; (4) basin of each attractor. A systematic solution is obtained by providing precise formulas and algorithms.

Using semi-tensor product of matrices, the Boolean network dynamics is converted into a standard discrete-time linear dynamics. This approach yields the above mentioned results.

Here we would like to give a comparison of our algorithm with some existing methods. The main interest of this paper is its theoretical aspect: How to convert the dynamics of Boolean networks into a linear dynamic system. As a byproduct, the algorithms are obtained. There are several existing numerical methods for Boolean problems, such as discrete iteration (DI)[28], satisfiability (SAT)[33] etc. The main advantage of both DI and SAT is they can be used for large scale Boolean problems. DI is mainly used to find the fixed points. It is said in [28] (page 153) that "If the algorithm ends at a cycle of length ≥ 2 then this corresponds to a failure of the iteration". It has been used in [17], and one sees that some auxiliary treatments are necessary for finding different length cycles. SAT is mainly for static Boolean equations. Fixed points can be considered as static problem and SAT can be used. Using SAT for finding cycles is at least not straightforward. In addition, neither DI nor SAT can assure finding all solutions.

The major disadvantage of our algorithms is complexity: the dimension of the system is exponential in the number of the nodes. So it can only be used for small networks (say, $n \leq 20$). The advantage is that it provide a complete solution to fixed points, cycles, transient time and basins.

Finally, let's see how to use linear expression of Boolean dynamics to Boolean control problems. A Boolean control system can be expressed as

$$\begin{cases} A_1(t+1) = f_1(A_1(t), A_2(t), \cdots, A_n(t), u_1(t), \cdots, u_m(t)), \\ A_2(t+1) = f_2(A_1(t), A_2(t), \cdots, A_n(t), u_1(t), \cdots, u_m(t)), \\ \vdots \\ A_n(t+1) = f_n(A_1(t), A_2(t), \cdots, A_n(t), u_1(t), \cdots, u_m(t)), \end{cases} \tag{47}$$

and

$$y_j(t) = h_j(A_1(t), A_2(t), \cdots, A_n(t)), \quad j = 1, 2, \cdots, p, \tag{48}$$

where f_i, $i = 1, 2, \cdots n$, h_j, $j = 1, 2, \cdots p$ are logical functions; y_j, $j = 1, 2, \cdots p$ are outputs; u_i, $i = 1, 2, \cdots m$, are inputs (or controls).

Let $x(t) = \ltimes_{i=1}^n A_i(t)$, $u(t) = \ltimes_{i=1}^m u_i(t)$, and $y(t) = \ltimes_{i=1}^p y_i(t)$. Using the linear expression to the Boolean control network, it can be expressed as

$$\begin{cases} x(t+1) = Lu(t)x(t), & u \in D^m, \ x \in D^n, \\ y(t) = Hx(t), & y \in D^p. \end{cases} \tag{49}$$

Let $x(t) = \ltimes_{i=1}^{n} A_i(t)$, $u(t) = \ltimes_{i=1}^{m} u_i(t)$, and $y(t) = \ltimes_{i=1}^{p} y_i(t)$. Using the linear expression to the Boolean control network, it can be expressed as

$$\begin{cases} x(t+1) = Lu(t)x(t), & u \in D^m,\ x \in D^n, \\ y(t) = Hx(t), & y \in D^p. \end{cases} \tag{50}$$

If we want to calculate the control-depending transition matrix, the so called "direct computation" can do nothing for this. Using semi-tensor product, we have

$$x(t+1) = L(t)x(t),$$

where $L(t) = L \ltimes u(t)$ is straightforward computable. Moreover, this analytic form can be used for control design. Similarly, the input-output mapping can also be easily calculated. We refer to [10, 11] for further discussion.

Acknowledgment

A toolbox in Matlab is provided in `http://lsc.amss.ac.cn/~dcheng/stp/STP.zip` for the related computations.

References

[1] T. Akutsu, S. Miyano, S. Kuhara, Inferring qualitative relations in genetic networks and metabolic pathways, *Bioinformatics*, Vol. 16, 727-73, 2000.

[2] R. Albert, A-L Barabasi, Dynamics of complex systems: scaling laws or the period of Boolean networks, *Phys. Rev. Lett.*, Vol. 84, 5660-5663, 2000.

[3] M. Aldana, Boolean dynamics of networks with scale-free topology, *Physica D*, Vol. 185, 45-66, 2003.

[4] R. Albert, H.G. Othmer, The topology and signature of the regulatory interactions predict the expression pattern of the segment polarity genes in Drosophila melanogaster, *J. Theor. Bio.*, Vol. 223, No. 1, 1-18, 2003.

[5] J. Aracena, J. Demongeot, E. Goles, On limit cycles of monotone functions with symmetric connection graph, *Theoretical Computer Science*, Vol. 322, 237-244, 2004.

[6] F. Baccelli, G. Cohen, G.J. Olsder, J.P. Quadrat, *Synchronization and linearity, An Algebra for Discrete Event Systems*, Wiley, Chichester, 1992

[7] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica*, Vol. 35, 407-427, 1999.

[8] D. Cheng, H. Qi, Matrix Expression of Logic and Fuzzy Control, *Proc. 44th IEEE CDC*, Seville, 3273-3278, 2005.

[9] D. Cheng, Semi-tensor product of matrices and its applications — A survey, *ICCM 2007*, Vol. 3, 641-668, 2007.

[10] D. Cheng, Input-state approach to Boolean networks, *IEEE Trans. Neural Network*, to appear, (preprint: http://lsc.amss.ac.cn/∼dcheng).

[11] D. Cheng, H. Qi, Controllability and observability of Boolean control networks, sumbitted to Automatica, under revision, (preprint: http://lsc.amss.ac.cn/∼dcheng).

[12] B. Drossel, T. Mihaljev, F. Greil, Number and length of attractors in a critical Kauffman model with connectivity one, *Phys. Rev. Lett.*, Vol 94, 088701, 2005.

[13] C. Farrow, J. Heidel, H. Maloney, J. Rogers, Scalar equations for synchronous Boolean networks with biological applications, *IEEE Trans. Neural Networks*, Vol. 15, No. 2, 348-354, 2004.

[14] B.C. Goodwin, *Temporal Organization in Cells*, Academic, New York, 1963.

[15] A.G. Hamilton, *Logic for Mathematicians*, Revised Ed., Cambridge Univ. Press, Cambridge, 1988.

[16] S.E. Harris, B.K. Sawhill, A. Wuensche, S. Kauffman, A model of transcriptional regulatory networks based on biases in the observed regulation rules, *Complexity*, Vol. 7, 23-40, 2002.

[17] J. Heidel, J. Maloney, J. Farrow, and J. Rogers, Finding cycles in synchronous Boolean networks with applications to biochemical systems, *Int. J. Bifurcat. Chaos*, Vol. 13, No. 3, 535-552, 2003.

[18] S. Huang, I. Ingber, Shape-dependent control of cell growth, differentiation, and apotosis: Switching between attractors in cell regulatory networks, *Exp. Cell Res.*, Vol. 261, 91-103, 2000.

[19] S. Huang, Regulation of cellular states in mammalian cells from a genomewide view, in *Gene Regulation and Metabolism*, J. Collado-Vodes and R. Hofestadt, Eds., Cambridge, MA: MIT Press, 181-220, 2002.

[20] T. Ideker, T. Galitski, L. Hood, A new approach to decoding life: systems biology, *Annu. Rev. Genomics Hum. Genet.*, Vol. 2, 343-372, 2001.

[21] S.A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, *J. Theoretical Biology*, Vol. 22, 437-467, 1969.

[22] S.A. Kauffman, At Home in the Universe, Oxford Univ. Press, Oxford, 1995.

[23] M. Kaufman, C. Soulé, R. Thomas, A new necessary condition on interaction graphs for multistationarity, *J. Theoretical Biology*, Vol. 248, 675-685, 2007.

[24] H. Kitano, Systems Biology: A brief overview, *Science*, Vol. 259, 1662-1664, 2002.

[25] H. Qi, D. Cheng, Logic and logic-based control, *J. Contr. Theory and Appl.*, Vol. 6, No. 1, pp. 123-133, 2008.

[26] L. Rade, B. Westergren, *Mathematics Handbook for Science and Engineering*, 4th ed, Studentlitteratur, Sweden, 1998.

[27] A. Richard, J-P. Comet, Necessary condition for multistationarity in discrete dynamical systems, *Dis. Appl. Math.*, Vol. 155, 2403-2413, 2007.

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

[29] B. Samuelsson, C. Troein, Superpolynomial growth in the number of attractors in Kauffman networks, *Phys. Rev. Lett.*, Vol. 90, 90098701, 2003.

[30] M.-H. Shih, J.-L. Dong, A combinatorial analogue of the Jacobian problem in automata networks, *Advances in Applied Mathematics*, Vol. 34, 30-46, 2005.

[31] I. Shmulevich, R. Dougherty, S. Kim, W. Zhang, Probabilistic Boolean neworks: A rule-based uncertainty model for gene regulatory networks, *Bioinformatics*, Vol. 2, No. 18, 261-274, 2002

[32] C. Soulé, Mathematical approaches to differentiation and gene regulation, *Biologies*, Vol. 329, 13-20, 2006.

[33] K. Truemper, *Design of Logic-based Intelligent Systems*, Wiley & Sons, New Jersey, 2004.

[34] Q. Zhao, A remark on 'Scalar equations for synchronous Boolean networks with biological applications' by C. Farrow, J. Heidel, J. Maloney, and J. Rogers, *IEEE Trans. Neural Networks*, Vol. 16, No. 6, 1715-1716, 2005.