# A Linear Representation of Dynamics of Boolean Networks

Daizhan Cheng, *Fellow, IEEE*, and Hongsheng Qi

***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, 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; and d) basin of each attractor. The corresponding algorithms are developed and used to some examples.**

***Index Terms*—Boolean network, cycle, fixed point, semi-tensor product, transient period.**

## I. 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 at a time. Rather, it studies the behavior and relationships of all of the cells, proteins, DNAs and RNAs in a biological system, called a cellular network. 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 [27].

The Boolean network, first introduced by Kauffman [25], and then developed by [2]–[4], [15], [19], [22], [32], [33] 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. Say, there is a one to one relationship between model attractors and observed phenotypes [23].

The structure of a Boolean network is described in terms of its cycles and the transient states that lead to them. Several useful algorithms have been developed to detect attractors. For instance, Bounded Model Checking using All Satisfactory [11], [28], Iteration and Scalar Form [20], Linear Reduced Scalar Equation [16], Constraint Programming [14], Reduced Ordered Binary Decision Diagram [17], and some other new algorithms [24], [35]. It was pointed out in [1], [36] that finding fixed points and cycles of a Boolean network is an NP-complete problem. Searching for parent states (basin of attractor) has been discussed by [5], and its complexity by [12]. Matrix representation of a Boolean control network has been discussed, e.g., [13].

The analysis of the dynamics of Boolean networks focuses also on the link between the dependence between variables and the state space [31], [34]. We refer to [26] for some interesting recent developments on this topic.

The purpose of this paper is to use semi-tensor product ($\ltimes$) to convert the logical dynamic equations of a Boolean network into a linear discrete-time dynamic equation

$$x(t+1) = L \ltimes x(t) \tag{1}$$

where $x = \ltimes_{i=1}^{n} x_i$ is the semi-tensor product of logical variables. Unlike usual transition matrix expression, (1) contains complete information of the logical equations of the Boolean network. Using this transition matrix $L$, formulas for cycles, transient time, and basins can be easily obtained. Furthermore, based on the semi-tensor product based framework, the subspace structure and the relationship between subspace cycles and the state space cycles can be revealed [8]. This form can easily be extended to the Boolean control networks, and certain control problems, such as controllability and observability [9], realization [10] etc., can be investigated.

The rest of the paper is organized as follows. Section II 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 III. Using the tools developed in Section III, the logical equations of a Boolean network are converted into a semi-tensor product based linear representation in Section IV. Then in Section V the formulas are obtained for a) the number of fixed points; b) the numbers of cycles of different lengths; c) transient period; and d) basin of attractors. Corresponding algorithms are also developed and used to some examples to compare our results with existing ones. Section VI is the concluding remarks.

## II. Semi-Tensor Product

This section introduces the semi-tensor product (STP) of matrices [7].

Manuscript received September 04, 2008; revised February 26, 2009; accepted February 02, 2010. First published February 17, 2010; current version published October 06, 2010. This work was supported in part by the National Natural Science Foundation (NNSF) of China under Grants 60736022, 60821091, and SIC 07010201. Recommended by Associate Editor M. Egerstedt.

The authors are with the Key Laboratory of Systems and Control, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: dcheng@iss.ac.cn; qihongsh@amss.ac.cn).

Digital Object Identifier 10.1109/TAC.2010.2043294

*Definition II.1:*

1) Let $X$ be a row vector of dimension $np$, and $Y$ be a column vector of dimension $p$. Then we split $X$ into $p$ equal-size blocks as $X^1,\dots,X^p$, 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

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

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

*Example II.2:*

1) Let $A=\begin{bmatrix}1&3&0&-2\\2&1&-2&3\end{bmatrix}$, $B=\begin{bmatrix}1&-1\\2&3\end{bmatrix}$. Then

$$\begin{aligned} &A\ltimes B\\ &=\begin{bmatrix}(1\ 3)\times 1+(0\ -2)\times 2 & (1\ 3)\times(-1)+(0\ -2)\times 3\\ (2\ 1)\times 1+(-2\ 3)\times 2 & (2\ 1)\times(-1)+(-2\ 3)\times 3\end{bmatrix}\\ &=\begin{bmatrix}1&-1&-1&-9\\-2&7&-8&8\end{bmatrix}. \end{aligned}$$

2) Let $x=[x_1,\dots,x_m]^T\in\mathbb{R}^m$, $y=[y_1,\dots,y_n]^T\in\mathbb{R}^n$. Then

$$x\ltimes y=[x_1y_1,\dots,x_1y_n,\dots,x_my_1,\dots,x_my_n]^T\in\mathbb{R}^{mn}.$$

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

*Proposition II.3:* The STP satisfies (as long as the related products are well defined)

1) (Distributive rule)

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

2) (Associative rule)

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

*Proposition II.4:* Assume $A\succ_t B$, then (where "$\otimes$" is the Kronecker product, $I_t$ is the identity matrix)

$$A\ltimes B=A(B\otimes I_t). \tag{5}$$

Assume $A\prec_t B$, then

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

*Proposition II.5:* 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 that 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) $\xi$, $\xi^k$ is always well defined.

Define a delta set as $\Delta_k:=\{\delta_k^i\mid i=1,2,\dots,k\}$, where $\delta_k^i$ is the $i$-th column of $I_k$. For example

$$\Delta_3=\left\{\delta_3^1=\begin{bmatrix}1\\0\\0\end{bmatrix},\delta_3^2=\begin{bmatrix}0\\1\\0\end{bmatrix},\delta_3^3=\begin{bmatrix}0\\0\\1\end{bmatrix}\right\}.$$

A matrix $A\in M_{m\times n}$ is called a logical matrix if the columns of $A$, denoted by $\mathrm{Col}(A)$, satisfy $\mathrm{Col}(A)\subset\Delta_m$. The set of $m\times n$ logical matrices is denoted by $\mathcal{L}_{m\times n}$.

Assume $A\in\mathcal{L}_{m\times n}$, which means $A=[\delta_m^{i_1},\delta_m^{i_2},\dots,\delta_m^{i_n}]$, to save space we denote it as

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

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 $(11,12,\dots,1n,\dots,m1,m2,\dots,mn)$ and its rows by $(11,21,\dots,m1,\dots,1n,2n,\dots,mn)$. 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 $W_{[n]}:=W_{[n,n]}$. We refer to [21] for an alternative definition of swap matrix, where it is called commutation matrix.

*1) Example II.6:* Let $m=2$ and $n=3$, the swap matrix $W_{[2,\,3]}$ is

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

□

*Proposition II.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 II.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. Through this paper all the matrix products are assumed to be semi-tensor product and the notation "$\ltimes$" is mostly omitted.

## III. Matrix Expression of Logic

In this section we recall the matrix expression of logic. We refer to [6], [7] for details.

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

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

A logical function with $n$ arguments is a mapping $f : \mathcal{D}^n \to \mathcal{D}$. To use matrix expression we identify each element in $\mathcal{D}$ with a vector as $T \sim \delta_2^1$ and $F \sim \delta_2^2$, and denote

$$\Delta := \Delta_2 = \{\delta_2^1, \delta_2^2\} \sim \mathcal{D}.$$

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

*Definition III.1:* A $2 \times 2^n$ matrix $M_\sigma$ is called the structure matrix of a logical function $\sigma : \mathcal{D}^n \to \mathcal{D}$, if

$$\sigma(A_1, A_2, \ldots, A_n) = M_\sigma A_1 A_2 \cdots A_n \tag{12}$$

where $A_1, \ldots, A_n \in \Delta$.

If such a matrix exists, it uniquely determines the logical function. To show the existence of such a matrix for each logical function, we define a power-reducing matrix as $M_r = \delta_4[1, 4]$. It is easy to prove that $P^2 = M_r P$, $\forall P \in \Delta$.

Using swap matrix and the power-reducing matrix, the following theorem can be proved easily [29].

*Theorem III.2:* Any logical function $L(A_1, \ldots, A_n)$ with logical arguments $A_1, \ldots, A_n \in \Delta$ can be expressed in a multi-linear form as

$$L(A_1, \ldots, A_n) = M_L A_1 A_2 \cdots A_n \tag{13}$$

where $M_L \in \mathcal{L}_{2\times 2^n}$ is unique, called the structure matrix of $L$.

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

*Example III.3:*

1) Consider a fundamental unary logical function: Negation, $\neg P$, and four fundamental binary logical functions [30]: Disjunction, $P \vee Q$; Conjunction, $P \wedge Q$; Implication, $P \to Q$; Equivalence, $P \leftrightarrow Q$. Setting the logical variables to either $\delta_2^1$ or $\delta_2^2$, it is ready to check that their structure matrices are as follows:

$$\begin{aligned} M_\neg &:= M_n = \delta_2[2, 1];\\ M_\vee &:= M_d = \delta_2[1, 1, 1, 2];\\ M_\wedge &:= M_c = \delta_2[1, 2, 2, 2];\\ M_\to &:= M_i = \delta_2[1, 2, 1, 1];\\ M_\leftrightarrow &:= M_e = \delta_2[1, 2, 2, 1]. \end{aligned} \tag{14}$$

2) Assume that

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

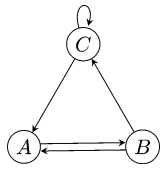


Fig. 1. Boolean network of (16).

Using formulas in (14) and Proposition II.5, we have

$$\begin{aligned} L(u, v) &= M_d(M_i uv)(M_n u)\\ &= M_d M_i(I_4 \otimes M_n)uvu\\ &= M_d M_i(I_4 \otimes M_n)uW_{[2]}uv\\ &= M_d M_i(I_4 \otimes M_n)(I_2 \otimes W_{[2]})u^2 v\\ &= M_d M_i(I_4 \otimes M_n)(I_2 \otimes W_{[2]})M_r uv. \end{aligned}$$

It follows that:

$$\begin{aligned} M_L &= M_d M_i(I_4 \otimes M_n)(I_2 \otimes W_{[2]})\\ M_r &= \delta_2[1, 2, 1, 1]. \end{aligned}$$

□

## IV. Dynamics of Boolean Networks

*Definition IV.1 [16]:* A Boolean network with a set of nodes $A_1,\ A_2, \ldots, A_n$ can be described as

$$\begin{cases} A_1(t+1) = f_1(A_1(t), A_2(t), \ldots, A_n(t))\\ A_2(t+1) = f_2(A_1(t), A_2(t), \ldots, A_n(t))\\ \vdots\\ A_n(t+1) = f_n(A_1(t), A_2(t), \ldots, A_n(t)) \end{cases} \tag{15}$$

where $f_i, i = 1, 2, \ldots, n$ are logical functions.

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

*Example IV.2:* Consider a Boolean network as shown in Fig. 1. Its dynamics is described as

$$\begin{cases} A(t+1) = B(t) \wedge C(t)\\ B(t+1) = \neg A(t)\\ C(t+1) = B(t) \vee C(t). \end{cases} \tag{16}$$

□

Our first purpose is to convert system (15) into an algebraic form. Precisely, express it as a conventional discrete-time linear system. Using semi-tensor product, we define

$$x(t) = \ltimes_{i=1}^n A_i(t). \tag{17}$$

*Remark IV.3:* Note that in (17) 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 V.1 provides a precise formula to recover $A_i$, $1 \le i \le n$ from $x = \ltimes_{i=1}^n A_i$.

Using Theorem III.2, we can find structure matrices, $M_i = M_{f_i}$, $i = 1, \ldots, n$, such that

$$A_i(t+1) = M_i x(t), \quad i = 1, 2, \ldots, n. \tag{18}$$

*Remark IV.4:* Note that usually the indegree is much less than $n$, that is, the right hand side of the $i$-th equation of (15) may not have all $A'_j s,\ j = 1, 2, \ldots, n$. Say, in the previous example, for node $A$ we have

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

In matrix form it is

$$A(t+1) = M_c B(t) C(t). \tag{19}$$

To get the form of (18), we can construct a *dummy matrix* as $E_d := \delta_2[1, 2, 1, 2]$. It is easy to prove that for any two logical variables $u$, $v$

$$E_d uv = v, \quad \text{or} \quad E_d W_{[2]} uv = u.$$

Then we can rewrite (19) as

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

Multiplying the equations in (18) together yields

$$x(t+1) = M_1 x(t) M_2 x(t) \cdots M_n x(t). \tag{20}$$

To simplify (20) we need some preparations:

*Lemma IV.5:* Assume $P_k = A_1 A_2 \cdots A_k$, then

$$P_k^2 = \Phi_k P_k \tag{21}$$

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 power-reducing matrix, 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 (21) is true for $k = 1$. Assume that (21) is true for $k = s$, then for $k = s + 1$ we have

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

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

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

which completes the proof. ■

*Theorem IV.6:* Equation (20) can be expressed as

$$x(t+1) = L x(t) \tag{22}$$

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 IV.5 we have

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

Now

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

From Remark IV.3 it is easy to see that (22) is enough to describe the dynamics. ■

*Example IV.7:* Recall the Boolean network in Example IV.2. In algebraic form, we have

$$\begin{cases} A(t+1) = M_c B(t) C(t) \\ B(t+1) = M_n A(t) \\ C(t+1) = M_d B(t) C(t). \end{cases} \tag{23}$$

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

$$\begin{aligned} x(t+1) &= M_c B C M_n A M_d B C \\ &= M_c (I_4 \otimes M_n) B C A M_d B C \\ &= M_c (I_4 \otimes M_n)(I_8 \otimes M_d) B C A B C \\ &= M_c (I_4 \otimes M_n)(I_8 \otimes M_d) W_{[2,4]} A B C B C \\ &= M_c (I_4 \otimes M_n)(I_8 \otimes M_d) \\ &\quad \otimes W_{[2,4]} A B W_{[2]} B C C \\ &= M_c (I_4 \otimes M_n)(I_8 \otimes M_d) W_{[2,4]} \\ &\quad (I_4 \otimes W_{[2]}) A M_r B M_r C \\ &= M_c (I_4 \otimes M_n)(I_8 \otimes M_d) W_{[2,4]} \\ &\quad (I_4 \otimes W_{[2]})(I_2 \otimes M_r)(I_4 \otimes M_r) A B C. \end{aligned} \tag{24}$$

Then system (16) is expressed in a matrix form as

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

where the network transition matrix is

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

□

Linear representation of logical mappings is very useful in control of Boolean networks. For instance, let $y_1, \ldots, y_p$ be a set of logical functions of $x_1, \ldots, x_n$. Setting $y = \ltimes_{i=1}^{p} y_i$, $x = \ltimes_{i=1}^{n} x_i$, we can find a unique $H \in \mathcal{L}_{2^p \times 2^n}$, such that

$y = Hx$. The properties of subspace $\mathcal{Y}$ generated by $y$ is completely determined by $H$. For instance, if $p = n$, $x \to y$ is a logical coordinate transformation, iff $H$ is nonsingular [9]. $\mathcal{Y}$ is a regular subspace (i.e., $y$ is part of coordinate variables), iff $H$ satisfies certain algebraic conditions [10].

## V. Fixed Points and Cycles

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

*Proposition V.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

$$\begin{cases} A_k(t) = \left[\frac{B_{k-1}}{2^{n-k}}\right] \\ B_k = B_{k-1} - A_k(t) * 2^{n-k}, \quad k = 1, 2, \ldots, n \end{cases} \tag{25}$$

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

*Example V.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 $A_5 = 1 \sim (1,0)^T$. □

*Definition V.3:* Consider system (15) with its algebraic form (22).

1) A state $x_0 \in \Delta_{2^n}$ is called a fixed point of system (15), if $Lx_0 = x_0$.
2) $\{x_0, Lx_0, \ldots, L^k x_0\}$ is called a cycle of system (15) with length $k$, if , $L^k x_0 = x_0$, and the elements in set $\{x_0, Lx_0, \ldots, L^{k-1}x_0\}$ are distinct.

To present the following theorem for numbers of fixed points and cycles of different lengths, we need some preparations: (i) A fixed point is a cycle of length 1; (ii) 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.

*Theorem V.4:* Consider Boolean network (15). The number of length $s$ cycles, $N_s$, is inductively determined by

$$\begin{cases} N_1 = \mathrm{Trace}(L), \\ N_s = \dfrac{\mathrm{Trace}(L^s) - \sum\limits_{k \in \mathcal{P}(s)} kN_k}{s}, \quad 2 \le s \le 2^n. \end{cases} \tag{26}$$

*Proof:* Assume that $\delta_{2^n}^i$ is its fixed point. Then, $L\delta_{2^n}^i = L_i$, the $i$-th column of $L$. Hence $L_i = \delta_{2^n}^i$. That is, each diagonal nonzero (precisely 1) column is a fixed point, which proves $N_1 = \mathrm{Trace}(L)$.

As for the second equation, note that a point on a cycle of length $s$ is a fixed point of $L^s$. Subtracting number of cycles of factor length and taking into consideration of multiplicity caused by $s$ elements in the cycle, the conclusion follows. ■

Next, we consider how to find the cycles. If

$$\mathrm{Trace}(L^s) - \sum_{k \in \mathcal{P}(s)} kN_k > 0 \tag{27}$$

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

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

$$C_s = \{i \mid \ell_{ii}^s = 1\}, \quad s = 1, 2, \ldots, 2^n,$$
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 V.5:* Let $x_0 = \delta_{2^n}^i$. Then $\{x_0, Lx_0, \ldots, L^s x_0\}$ is a cycle with length $s$, iff $i \in D_s$.

Theorem V.4 and Proposition V.5 provide a simple algorithm for calculating cycles. We give some examples to show the algorithm.

*Example V.6:* Recall Example IV.2. It is easy to check that

$$\mathrm{Trace}(L^t) = 0, \quad t \le 3,$$
and
$$\mathrm{Trace}(L^t) = 4, \quad t \ge 4.$$

Using Theorem V.4, 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

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

Using Proposition V.1 to convert the vector forms back to the scalar form of $A(t)$, $B(t)$, and $C(t)$, we have the cycle as $(111) \to (101) \to (001) \to (011) \to (111)$. □

Both fixed point and cycle are called attractor. The attracting set, denoted by $\Omega$, is the union of all attractors. In the following we consider the transient period, i.e., the minimum transient states that leads any point to the attracting set $\Omega$. First, it is easy to see that there are only $r := 2^n \times 2^n = 2^{2n}$ different logical matrices. Hence, if we construct a sequence of $r+1$ matrices as

$$L^0 (= I_{2^n}),\ L,\ L^2,\ \ldots,\ L^r$$

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

$$r_0 = \min\left\{i \middle| L^i \in \{L^{i+1}, L^{i+2}, \ldots, L^r\},\ 0 \le i < r\right\}. \tag{28}$$

Then such $r_0$ exists. The following proposition is obvious.

*Proposition V.7:* Let $r_0$ be defined as in (28). 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 V.8:* The $r_0$ defined in (28) is the transient period of the system. That is

$$T_t = r_0. \tag{29}$$

*Proof:* First, assume that

$$L^{r_0} = L^{r_0+T} \tag{30}$$

and $T > 0$ is the smallest positive number, which verifies (30). 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 $T(\mathrm{mod}\, t) = s$ 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

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

which is a contradiction.

From (30) and the definition of $T_t$ it is obvious that $T_t \leq r_0$. To prove $T_t = r_0$, we assume that $T_t < r_0$. By definition, for any $x$, $L^{T_t}x$ is on a cycle, which has length as a factor of $T$. Hence

$$L^{T_t}x = L^T L^{T_t}x = L^{T_t+T}x, \ \forall x. \tag{31}$$

It is easy to check that if for any $x \in \Delta_{2^n}$ (31) holds, then $L^{T_t} = L^{T_t+T}$, which is a contradiction to the definition of $r_0$. ■

*Remark V.9:*

1) According to Theorem V.8 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 (30). 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 V.10:*

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 V.11:*

- Let $C \subset \mathcal{D}^n$. Denote by

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

  Then the set of parent states of $p$ is $L^{-1}(p)$.
- $\mathcal{D}^n = \cup_{i=1}^{k} S_i$. Moreover, since $\{S_i \,|\, i = 1, \ldots, k\}$ are disjointed, it is a partition of the state space $\mathcal{D}^n$.

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 V.12:*

$$S_i = L^{-1}(C_i) \cup L^{-2}(C_i) \cup \cdots \cup L^{-T_t}(C_i). \tag{32}$$

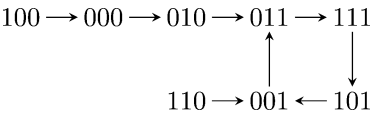


Fig. 2. State Space Graph of (16).

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 V.13:*

$$\begin{cases} L^{-1}(p) = \left\{ \delta_{2^n}^j \,\middle|\, L_j = p \right\} \\ L^{-k}(p) = \left\{ \delta_{2^n}^j \,\middle|\, L_j^k = p \right\}, \quad k = 2, \ldots, T_t. \end{cases} \tag{33}$$

*Example V.14:* Recall Example IV.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 V.12 and V.13, 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 IV.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.) □

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 [30].

*Example V.15 ([16]):* Consider the following Boolean network

$$\begin{cases} A(t+1) = B(t)C(t) \\ B(t+1) = 1 + A(t) \\ C(t+1) = B(t). \end{cases} \tag{34}$$

It is easy to calculate that

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

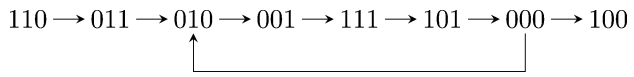


Fig. 3. State space graph of (34).

$L$ follows immediately as:

$$\begin{aligned} L &= \delta_8[3,7,8,8,1,5,6,6]. \\ \mathrm{Trace}(L^k) &= 0, \quad k = 1,2,3,4, \\ &\text{and} \\ L^5 &= \delta_8[1,3,3,3,5,6,8,8], \\ \mathrm{Trace}(L^5) &= 5. \end{aligned}$$

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 [16]. □

The last example is from [18] and re-investigated in [20].

*1) Example V.16:* Consider the following system:

$$\begin{cases} A(t+1) = 1 + C(t) + F(t) + C(t)F(t) \\ B(t+1) = A(t) \\ C(t+1) = B(t) \\ D(t+1) = 1 + C(t) + F(t) + I(t) + C(t)F(t) \\ \qquad\qquad + C(t)I(t) + F(t)I(t) + C(t)F(t)I(t) \\ E(t+1) = D(t) \\ F(t+1) = E(t) \\ G(t+1) = 1 + F(t) + I(t) + F(t)I(t) \\ H(t+1) = G(t) \\ I(t+1) = H(t). \end{cases} \tag{35}$$

We skip the detailed computation and present the result directly:

The non-trivial powers are $\mathrm{Trace}(L^2) = 4$, and $\mathrm{Trace}(L^6) = 64$. It follows from Theorem V.4 that there are only two cycles of length 2 and ten cycles of length 6.

Searching diagonal nonzero columns of $L^2$ yields

$$\begin{aligned} &\to (101101101) \to (010010010) \to; \\ &\to (101000010) \to (010000101) \to . \end{aligned}$$

Searching diagonal nonzero columns of $L^6$ yields

$$\begin{aligned} &\to (111111111) \to (011011011) \to (001001001) \\ &\to (000000000) \to (100100100) \to (110110110) \to; \\ &\to (111110110) \to (011011111) \to (001001011) \\ &\to (000000001) \to (100000000) \to (110100100) \to; \\ &\to (111101101) \to (011010010) \to (001001101) \\ &\to (000000010) \to (100100101) \to (110010010) \to; \\ &\to (111100100) \to (011010110) \to (001001111) \\ &\to (000000011) \to (100000001) \to (110000000) \to; \\ &\to (111011011) \to (011001001) \to (001000000) \\ &\to (000000100) \to (100100110) \to (110110111) \to; \\ &\to (111010010) \to (011001101) \to (001000010) \\ &\to (000000101) \to (100000010) \to (110100101) \to; \\ &\to (111001001) \to (011000000) \to (001000100) \\ &\to (000000110) \to (100100111) \to (110010011) \to; \\ &\to (111000000) \to (011000100) \to (001000110) \\ &\to (000000111) \to (100000011) \to (110000001) \to; \\ &\to (101101111) \to (010010011) \to (101001001) \\ &\to (010000000) \to (101100100) \to (010010110) \to; \\ &\to (101100110) \to (010010111) \to (101001011) \\ &\to (010000001) \to (101000000) \to (010000100) \to . \end{aligned}$$

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

It was shown that there are no fixed points, and there are two cycles of length 2. Our results about fixed points and cycles with length 2 coincide with [20]. Reference [20] pointed out only six cycles of length 6. According to our result, there are exactly ten cycles of length 6.

## VI. Conclusion

In this paper, the topological structure of Boolean networks has been investigated. Using semi-tensor product of matrices, the dynamics of Boolean networks has been converted into an algebraic form $x(t+1) = Lx(t)$. Since in our expression between $L$ and $x = \ltimes_{i=1}^n x_i$, all the products are semi-tensor product, it gives a clear insight for the mapping with respect to each arguments. In this paper such structure provides formulas for 1) fixed points; 2) cycles of different lengths; 3) transient period; 4) basin of each attractor. Unlike many existing powerful algorithms, our main interest is on the theoretical part. For instance, using this semi-tensor product based structure, the invariant subspace and the cycles on such subspaces have been discussed [8]. If we consider a Boolean control network, its algebraic form can be obtained as $x(t+1) = Lu(t)x(t)$. Then, using the associativity of semi-tensor product, we have the control-depending transition matrix as $Lu(t)$. Several control properties can be investigated via this form [9].

## References

[1] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions," *Genome Inform.*, vol. 9, pp. 151–160, 1998.

[2] T. Akutsu, S. Miyano, and S. Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways," *Bioinformatics*, vol. 16, pp. 727–734, 2000.

[3] R. Albert and A.-L. Barabasi, "Dynamics of complex systems: Scaling laws or the period of Boolean networks," *Phys. Rev. Lett.*, vol. 84, pp. 5660–5663, 2000.

[4] M. Aldana, "Boolean dynamics of networks with scale-free topology," *Physica D*, vol. 185, pp. 45–66, 2003.
[5] C. Barrett, H. B. Hunt, III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, and M. Thakur, "Predecessor existence problems for finite discrete dynamical systems," *Theor. Comp. Sci.*, vol. 386, pp. 3–37, 2007.
[6] D. Cheng and H. Qi, "Matrix expression of logic and fuzzy control," in *Proc. 44th IEEE CDC*, Seville, Spain, 2005, pp. 3273–3278.
[7] D. Cheng, "Semi-tensor product of matrices and its applications—A survey," in *Proc. ICCM'07*, 2007, vol. 3, pp. 641–668.
[8] D. Cheng, "Input-state approach to Boolean networks," *IEEE Trans. Neural Networks*, vol. 20, no. 3, pp. 512–521, May 2009.
[9] D. Cheng and H. Qi, "Controllability and observability of Boolean control networks," *Automatica*, vol. 45, no. 7, pp. 1659–1667, 2009.
[10] D. Cheng, Z. Q. Li, and H. Qi, "Realization of Boolean control networks," *Automatica*, vol. 46, no. 1, pp. 62–69, 2010.
[11] E. Clarke, D. Kröning, J. Ouaknine, and O. Strichman, *Completeness and Complexity of Bounded Model Checking, in Verification, Model Checking, and Abstract Interpretation LNCS 2937*. Venice, Italy: Springer, 2004, pp. 85–96.
[12] S. N. Coppersmith, "Complexity of the predecessor problem in Kauffman networks," *Phys. Rev. E*, vol. 75, p. 051108, 2007.
[13] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, pp. 169–181, 2003.
[14] Y. Devloo, P. Hansen, and M. Labbé, "Identification of all steady states in large networks by logical analysis," *Bull. Math. Bio.*, vol. 65, pp. 1025–1051, 2003.
[15] B. Drossel, T. Mihaljev, and F. Greil, "Number and length of attractors in a critical Kauffman model with connectivity one," *Phys. Rev. Lett.*, vol. 94, p. 088701, 2005.
[16] C. Farrow, J. Heidel, H. Maloney, and J. Rogers, "Scalar equations for synchronous Boolean networks with biological applications," *IEEE Trans. Neural Networks*, vol. 15, no. 2, pp. 348–354, Mar. 2004.
[17] A. Garg, A. D. Cara, L. Xenarios, L. Mendoza, and G. D. Micheli, "Synchronous versus asynchronous modeling of gene regulatory networks," *Bioinformatics*, vol. 24, no. 17, pp. 1917–1925, 2008.
[18] B. C. Goodwin, *Temporal Organization in Cells*. New York: Academic, 1963.
[19] S. E. Harris, B. K. Sawhill, A. Wuensche, and S. Kauffman, "A model of transcriptional regulatory networks based on biases in the observed regulation rules," *Complexity*, vol. 7, pp. 23–40, 2002.
[20] 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, pp. 535–552, 2003.
[21] R. Horn and C. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.
[22] S. Huang and I. Ingber, "Shape-dependent control of cell growth, differentiation, and apotosis: Switching between attractors in cell regulatory networks," *Exper. Cell Res.*, vol. 261, pp. 91–103, 2000.
[23] 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, 2002, pp. 181–220.
[24] D. J. Irons, "Improving the efficiency of attractor cycle identificationin Boolean network," *Physica D*, vol. 217, pp. 7–21, 2006.
[25] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, pp. 437–467, 1969.
[26] M. Kaufman, C. Soulé, and R. Thomas, "A new necessary condition on interaction graphs for multistationarity," *J. Theor. Biol.*, vol. 248, pp. 675–685, 2007.
[27] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 259, pp. 1662–1664, 2002.
[28] C. J. Langmead, S. Jha, and E. M. Clarke, Temporal-Logics as Query Languages for Dynamic Bayesian Networks: Application to D. Melanogaster Embryo Development CMU-CS-06-159, 2010 [Online]. Available: http://reports-archive.adm.cs.cmu.edu/anon/2006/CMU-CS-06-159.pdf
[29] H. Qi and D. Cheng, "Logic and logic-based control," *J. Contr. Theory Appl.*, vol. 6, no. 1, pp. 123–133, 2008.
[30] L. Rade and B. Westergren, *Mathematics Handbook for Science and Engineering*, 4th ed. Lund, Sweden: Studentlitteratur, 1998.
[31] A. Richard and J.-P. Comet, "Necessary condition for multistationarity in discrete dynamical systems," *Dis. Appl. Math.*, vol. 155, pp. 2403–2413, 2007.
[32] B. Samuelsson and C. Troein, "Superpolynomial growth in the number of attractots in Kauffman networks," *Phys. Rev. Lett.*, vol. 90, p. 90098701, 2003.
[33] I. Shmulevich, R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean neworks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 2, no. 18, pp. 261–274, 2002.
[34] C. Soulé, "Mathematical approaches to differentiation and gene regulation," *Biologies*, vol. 329, pp. 13–20, 2006.
[35] S. Q. Zhang, M. Hayashida, T. Akutsu, W. K. Ching, and M. K. Ng, "Algorithms for finding small attractors in Boolean networks," *EURASIP J. Bioinf. Syst. Bio.*, vol. 2007, p. 20180, 2007.
[36] Q. Zhao, "A remark on ′Scalar equations for synchronous Boolean networks with biological applications′ by C. farrow, J. Heidel, J. Maloney, J. Rogers," *IEEE Trans. Neural Networks*, vol. 16, no. 6, pp. 1715–1716, Nov. 2005.

**Daizhan Cheng** (F'06) received the Ph.D. degree from Washington University, St. Louis, MO, in 1985.

He is currently a Professor with Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing. His research interests include nonlinear systems, numerical method, complex systems etc.

Dr. Cheng is a Fellow of the IFAC and the Chairman of Technical Committee on Control Theory, Chinese Association of Automation (2003-).

**Hongsheng Qi** received the Ph.D. degree in systems theory from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, in 2008.

He is currently an Assistant Professor with the Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences. His research interests include nonlinear control, complex systems, etc.