Analysis of Boolean Networks using An Optimized Algorithm of Structure Matrix based on Semi-Tensor Product

Jinyu Zhan
University of Electronic Science and Technology of China, Chengdu, China
Email: zhanjy@uestc.edu.cn

Shan Lu and Guowu Yang
University of Electronic Science and Technology of China, Chengdu, China
Email: 617999242@qq.com, guowu@uestc.edu.cn

***Abstract*—The structure matrix based on semi-tensor product can provide formulas for analyzing the characteristics of a Boolean network, such as the number of fixed points, the number of circles of different lengths, transient period for all points to enter the set of attractors and basin of each attractor. However, the conventional method of semi-tensor product gains the structure matrix through complex matrix operations with high computation complexity. This paper proposes an optimized algorithm which gains the structure matrix through the truth table reflecting the state transformation of Boolean networks. The effectiveness and feasibility of our optimized approach are demonstrated through the analysis of a practical Boolean network of the mammalian cell.**

***Index Terms*—semi-tensor product, Boolean network, structure matrix, truth table**

I. Introduction

The Boolean network, introduced firstly by Kauffman [1], and then developed by [2][3][4][5][6][7][8] and many others, becomes a powerful tool in describing, analyzing, and simulating the cell network. It was shown that the Boolean network plays an important role in modeling cell regulation, because they can represent important features of living organisms [9][10]. It has received the most attention, not only from the biology community, but also physics, system science, etc.

The structure of a Boolean network is described in terms of its cycles and the transient states that lead to them. Several useful Boolean networks have been analyzed and their circles have been revealed [11][12]. It was pointed in [13] that finding fixed points and circles of a Boolean network is an NP hard problem. Semi-tensor product of matrix (STP), presented by Cheng [14]. Using STP, a Boolean network equation can be expressed as a conventional discrete time linear system which contains complete information of the dynamics of a Boolean network. Analyzing the structure matrix of a Boolean network, precise formulas are obtained to determine the number of fixed points and numbers of all possible circles of different lengths.

But the conventional method to calculate the structure matrix of a Boolean network, presented in [15][18][19][21], is very complex. In this paper, a optimized algorithm is proposed to calculate the structure matrix. Unlike existing methods, our approach gets the structure matrix of a Boolean network not through the complex matrix operations but through the truth table which reflects the state transformation of the Boolean network. Compared with the conventional method, our approach can greatly reduce the calculation complexity. The methods for analyzing the characteristics of a Boolean network are given. The analysis of a practical Boolean network of the mammalian cell shows that our approach is effective and efficient.

The rest of the paper is organized as follows. Section II gives a brief introduction to semi-tensor product of matrices, matrix expression of logic and dynamics of Boolean network. The conventional method to calculate the structure Matrix of a Boolean Network is given in Section III and our approach is proposed in Section IV. Section V gives the methods to analyze the characteristics of Boolean networks through the structure matrix. Section VI gives a practical Boolean network of the mammalian cell to show the effectiveness and feasibility of our approach. Finally, some conclusions are drawn in Section VII.

II. Expression of Boolean Networks in Semi-tensor Product

A. Semi-tensor Product

This section is a brief introduction to semi-tensor product (STP) of matrices. STP of matrices is a

Manuscript received May 24, 2012; revised June 1, 2012; accepted July 1, 2012.

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant No. ZYGX2009J062 and the National Natural Science Foundation of China under Grant No. 60973016.

Corresponding author: Jinyu Zhan, email: zhanjy@uestc.edu.cn

doi:10.4304/jcp.8.6.1441-1448

generalization of conventional matrix product, which extends the conventional matrix product to any two matrices. It plays a fundamental rule in the following discussion. We restrict it to some concepts and basic properties used in this paper. In addition, only left semi-tensor product for multiplying dimension case is involved in the paper. We refer to [14][15][16][17] for right semi-tensor product, arbitrary dimensional case and much more details. Throughout this paper "semi-tensor product" means the left semi-tensor product for multiplying dimensional case.

Definition 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 $X^1, X^2, \cdots, X^p$, which are $1\times n$ rows. Define the STP, denoted by $\times$, as in (1).

$$\begin{cases} X \times Y = \sum_{i=1}^{p} X^i y_i \in R^n \\ Y^T \times X^T = \sum_{i=1}^{p} y_i (X^i)^T \in R^n \end{cases} \quad (1)$$

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 is as $A \succ_t B$, then we define the STP of A and B, denoted by $C = A \times B$, as the following: C consists of $m \times q$ blocks as $C = (C^{ij})$ and each block is in (2).

$$C^{ij} = A^i \times B_j,\ i = 1, \cdots, m,\ j = 1, \cdots, q. \quad (2)$$

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

We use some simple numerical examples to describe it.

Example 1. Let $X = [1 \quad 2 \quad 3 \quad -1]$ and $Y = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$. Then

$$X \times Y = [1 \quad 2] \cdot 1 + [3 \quad -1] \cdot 2 = [7 \quad 0]$$

Example 2. Let

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

$$A \times B = \begin{bmatrix} (1 \quad 2 \quad 1 \quad 1)\begin{pmatrix} 1 \\ 2 \end{pmatrix} & (1 \quad 2 \quad 1 \quad 1)\begin{pmatrix} -2 \\ -1 \end{pmatrix} \\ (2 \quad 3 \quad 1 \quad 2)\begin{pmatrix} 1 \\ 2 \end{pmatrix} & (2 \quad 3 \quad 1 \quad 2)\begin{pmatrix} -2 \\ -1 \end{pmatrix} \\ (3 \quad 2 \quad 1 \quad 0)\begin{pmatrix} 1 \\ 2 \end{pmatrix} & (3 \quad 2 \quad 1 \quad 0)\begin{pmatrix} -2 \\ -1 \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 3 & 4 & -3 & -5 \\ 4 & 7 & -5 & -8 \\ 5 & 2 & -7 & -4 \end{bmatrix}$$

B. Matrix Expression of Logic

In this section, the matrix expression of logic will be given. In a logical domain, we usually set "true" as "1" and "false" as "0". Then a logical variable is defined as $x \in D = \{0,1\}$. There are several fundamental binary functions such as $\neg$, $\wedge$, $\vee$, $\leftrightarrow$, $\rightarrow$, $\overline{\vee}$, $\uparrow$ and $\downarrow$. Their truth table is as TABLE I.

To use matrix expression each element can be identified in D with a vector as $1 \sim \delta_2^1$ and $0 \sim \delta_2^2$, where $\delta_n^i = Col(I_n)$. Therefore, That a *n*-ary logical operator (or function) is a mapping: $f: D^n \to D$ can be formed as $f: \Delta_{2^n} \to \Delta$.

Theorem 1: Let $f(x_1, \cdots, x_n)$ be a logical function in vector form as $f: \Delta_{2^n} \to \Delta$. Then there exists a unique $M_f \in \mathbf{L}_{2\times 2^n}$, called the structure matrix of f, such that in (3).

$$f(x_1, \cdots, x_n) = M_f \times x, \text{ where } x = \times_{i=1}^{n} x_i \quad (3)$$

Therefore, the structure matrix of Negation, Conjunction, Disjunction, Equivalence and Implication are as in (4) - (11).

$$M_\neg = \delta_2[2 \quad 1] \quad (4)$$

$$M_\wedge = \delta_2[1 \quad 2 \quad 2 \quad 2] \quad (5)$$

$$M_\vee = \delta_2[1 \quad 1 \quad 1 \quad 2] \quad (6)$$

$$M_\leftrightarrow = \delta_2[1 \quad 2 \quad 2 \quad 1] \quad (7)$$

$$M_\rightarrow = \delta_2[1 \quad 2 \quad 1 \quad 1] \quad (8)$$

TABLE I.
TRUTH TABLE OF $\neg$, $\wedge$, $\vee$, $\leftrightarrow$, $\rightarrow$, $\overline{\vee}$, $\uparrow$ AND $\downarrow$

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \leftrightarrow q$	$p \rightarrow q$	$p \overline{\vee} q$	$p \uparrow q$	$p \downarrow q$
0	0	1	0	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	0
1	0	0	0	1	0	0	1	1	0
1	1	0	1	1	1	1	0	0	0

$$M_{\bar{\vee}} = \delta_2[2 \quad 1 \quad 1 \quad 2] \tag{9}$$

$$M_{\uparrow} = \delta_2[2 \quad 1 \quad 1 \quad 1] \tag{10}$$

$$M_{\downarrow} = \delta_2[2 \quad 2 \quad 2 \quad 1] \tag{11}$$

Theorem 2: Let $F(x_1,\cdots,x_n): D^n \to D^k$ be a logical mapping: $F:\Delta_{2^n} \to \Delta_{2^k}$ .Then there exists a unique $M_F \in \mathbf{L}_{2^k\times 2^n}$ called the structure matrix of F, such in (12).

$$F(x_1,\cdots,x_n) = M_F \times x \tag{12}$$

C. Dynamics of Boolean Networks

The Boolean networks play an important role in modeling cell regulation, because they can represent important features of living organisms. The dynamics of the Boolean networks will be given in this section.

Definition 2[15][18][20]: A Boolean network is a set of nodes $A_1, A_2, \cdots, A_n$, which interact with each other in a synchronous manner. At each given time t=0, 1, 2, ⋯, a node has only one of two different values: 1 or 0. Thus the network can be described by a set of equations as in (13).

$$\begin{cases} A_1(t+1) = f_1(A_1(t), A_2(t), \cdots, A_n(t)) \\ A_2(t+1) = f_2(A_1(t), A_2(t), \cdots, A_n(t)) \\ \qquad\qquad \vdots \\ A_n(t+1) = f_n(A_1(t), A_2(t), \cdots, A_n(t)) \end{cases} \tag{13}$$

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

Note that in Boolean networks each function f_i has only constant, linear, or product terms [12].

III. Conventional Calculation of Structure Matrix

Using Theorem 1 and 2, the dynamics of Boolean networks can be expressed as in (14).

$$A(t+1) = LA(t) \tag{14}$$

where $A(t+1) = \ltimes_{i=1}^{l} A_i(t+1)$, $A(t) = \ltimes_{i=1}^{l} A_i(t)$, L is the structure matrix of F, $L \in \mathbf{L}_{2^l\times 2^l}$.

By means of the STP, the dynamics of Boolean networks can be converted into the equivalent algebraic forms. Through the analysis of the structure matrix L, we can get the characteristics of the Boolean networks such as: (1) fixed points; (2) circles of different lengths; (3) transient period; (4) basin of each attractor[15][18]. Therefore, how to get the structure matrix L easily is very important. The conventional method to get the structure matrix L is as follows.

Firstly, a simple example is given to show the structure of a Boolean network.

Example 3: Consider a Boolean network which dynamics is described as in (15).

$$\begin{cases} A_1(t+1) = \overline{A_2(t) \wedge A_3(t)} \wedge A_1(t) \\ A_2(t+1) = (A_1(t) \wedge \overline{A_2(t)}) \vee (\overline{A_1(t)} \wedge \overline{A_3(t)} \wedge A_2(t)) \\ A_3(t+1) = (A_1(t) \wedge A_2(t) \wedge \overline{A_3(t)}) \vee (\overline{A_2(t)} \wedge A_3(t)) \end{cases} \tag{15}$$

In algebraic form (the notation "×" is omitted), we can have as in (16).

$$\begin{cases} A_1(t+1) = M_{\wedge}(M_{\uparrow}A_2(t)A_3(t))A_1(t) \\ A_2(t+1) = M_{\vee}((M_{\wedge}A_1(t)(M_{\neg}A_2(t)) \\ \qquad\qquad (M_{\wedge}(M_{\downarrow}A_1(t)A_3(t))A_2(t))) \\ A_3(t+1) = M_{\vee}((M_{\wedge}(M_{\wedge}A_1(t)A_2(t)) \\ \qquad\qquad (M_{\neg}A_3(t)))(M_{\wedge}(M_{\neg}A_2(t)) \\ \qquad\qquad A_3(t)) \end{cases} \tag{16}$$

There are some propositions in [15][18] to calculate the structure matrix *L*.

Proposition 1: Let $Z \in R^t$ be a column. Then there exists in (17).

$$ZA = (I_t \otimes A)Z \tag{17}$$

Proposition 2: There exists an unique matrix $W_{[m,n]} \in M_{mn\times mn}$, called the swap matrix, such that for any two column vectors . $X \in R^m$. and $Y \in R^n$.

$$W_{[m,n]}XY = YX \tag{18}$$

We refer to [15][18] for constructing swap matrix.

Proposition 3: Let $X \in \Delta$. Then we have (19).

$$X^2 = M_r X, \quad M_r = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{19}$$

Where M_r is the power-reducing matrix.

$$\begin{aligned} L = {} & M_{\vee}M_{\wedge}M_{\wedge}(I_2 \otimes (I_2 \otimes M_{\neg}(I_2 \otimes \\ & M_{\wedge}M_{\neg}(I_2 \otimes (I_2 \otimes M_{\vee}M_{\wedge}(I_2 \otimes M_{\neg} \\ & (I_2 \otimes M_{\wedge}M_{\downarrow}(I_2 \otimes (I_2 \otimes (I_2 \otimes M_{\downarrow} \\ & M_{\wedge}))))))))W_{[2]}(I_2 \otimes W_{[2]})(I_4 \otimes W_{[2]}) \\ & (I_8 \otimes W_{[2]})(I_{32} \otimes W_{[2]})(I_{128} \otimes W_{[2]}) \\ & (I_{256} \otimes W_{[2]})(I_{512} \otimes W_{[2]})(I_{1024} \otimes W_{[2]}) \\ & W_{[2]}(I_4 \otimes W_{[2]})(I_{16} \otimes W_{[2]})(I_{64} \otimes W_{[2]}) \\ & (I_{128} \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{512} \otimes W_{[2]}) \\ & (I_2 \otimes W_{[2]})(I_{32} \otimes W_{[2]})(I_{64} \otimes W_{[2]}) \\ & (I_{128} \otimes W_{[2]})(I_{256} \otimes W_{[2]})(I_{16} \otimes W_{[2]}) \\ & (I_{128} \otimes W_{[2]})(I_8 \otimes W_{[2]})(I_{64} \otimes W_{[2]}) \\ & (I_4 \otimes W_{[2]})(I_{32} \otimes W_{[2]})(I_{16} \otimes W_{[2]}) \\ & (I_8 \otimes W_{[2]})M_rM_rM_r(I_2 \otimes (M_rM_r \\ & M_rM_r(I_2 \otimes M_rM_rM_r)) \end{aligned} \tag{20}$$

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

Using some theorems in [15][18][23][24] and $A(t+1) = LA(t)$, the structure matrix L is as in (20).

We can get the structure matrix by the conventional method. But the process is very complex and the biggest order of the matrices in the equation (20) is more than 1024.

IV. New Method for Calculation of Struture Matrix

The conventional method to calculate the structure matrix L is very complex. A new method will be proposed in this section.

Definition 3: Form a square matrix by all the present-state vectors $A(t) = \times_{i=1}^{l} A_i(t)$, the matrix is called present-state matrix, denoted by $Q(t)$ .There is another matrix correspond to $Q(t)$, called next-state matrix, denoted by $Q(t+1)$.

As $A(t+1) = LA(t)$, we can derive $Q(t+1) = LQ(t)$. It is easy to know that $Q(t) \in \mathbf{L}_{2^l \times 2^l}$, and $Q(t)$ is a invertible matrix. Then the structure matrix $L = Q(t+1)[Q(t)]^{-1}$. Further simplify the calculation, $Q(t)$ can be arrayed to 2^l-order identity matrix. Therefore, $L = Q(t+1)$.

For the example 3, we have the truth table as TABLE II.

TABLE II.
TRUTH TABLE OF EXAMPLE 3

A_3 (t)	A_2 (t)	A_1 (t)	A_3 (t+1)	A_2 (t+1)	A_1 (t+1)
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	0

The state vectors' table is as TABLE III.

TABLE III.
STATE VECTORS' TABLE

A_3 (t)	A_2 (t)	A_1 (t)	A (t)
0	0	0	$[0\,0\,0\,0\,0\,0\,0\,1]^T$
0	0	1	$[0\,0\,0\,0\,0\,0\,1\,0]^T$
0	1	0	$[0\,0\,0\,0\,0\,1\,0\,0]^T$
0	1	1	$[0\,0\,0\,0\,1\,0\,0\,0]^T$
1	0	0	$[0\,0\,0\,1\,0\,0\,0\,0]^T$
1	0	1	$[0\,0\,1\,0\,0\,0\,0\,0]^T$
1	1	0	$[0\,1\,0\,0\,0\,0\,0\,0]^T$
1	1	1	$[1\,0\,0\,0\,0\,0\,0\,0]^T$

Then, we can get the present-state matrix and next-state matrix as in (21) and (22).

$$Q(t) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

$$Q(t+1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (22)$$

Therefore, we can get the structure matrix $L = Q(t+1)$ in (23).

$$L = Q(t+1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (23)$$

We can compare the two structure matrix L gained in Section 3 and our method. And the structure matrix which is gained by our approach is correct.

By our method, it is easy to get the structure matrix through the truth table which reflects the transformation of the states. Our method to get the structure matrix is simpler than the conventional method.

V. Application of Stucture Matrix on Analysis of Boolean Networks

Since a Boolean network has only finite states, a trajectory will eventually enter into a fixed point or a cycle. The fixed points and cycles form the most important topological structure of a Boolean network. Therefore, there are many methods to analyze the fixed points and cycles of Boolean networks.

Analyzing the structure matrix of the system, easily computable formulas are obtained to show the number of fixed points, the numbers of circles of different lengths

and the states in the circles. The following is a general result based on the algebraic form.

Consider the Boolean network equation $A(t+1)=LA(t)$, and denote by L_i, 1, 2, …, 2^n the i-th column of the network matrix L. Then there are $L_i \in \Delta_{2^n}$ [15][18].

Definition 4: 1. A state $x_0 \in \Delta_{2^n}$ is called a fixed point of Boolean network $A(t+1)=LA(t)$, if $Lx_0 = x_0$.

2. $\{x_0, Lx_0, \cdots, L^{k-1}x_0\}$ is called a circle of Boolean network $A(t+1)=LA(t)$ 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.

L can be used for the matrix and its linear mapping. So x_0 may be in an L-invariant subspace. In this way, a circle (or a fixed point) can be defined on an L-invariant subspace.

The next two theorems [15][18] show how many fixed points and circles of different lengths a Boolean network has.

Theorem 3: Consider the Boolean network system (13). $\delta_{2^n}^i$ is its fixed point, iff in its algebraic form (14) the diagonal element l_{ii} of network matrix L equals 1. It follows that the number of equilibriums of system (13), denoted by N_e, equals the number of i, for which l_{ii} = 1. Equivalently, in (24).

$$N_e = Trace(L) \tag{24}$$

Theorem 4: The number of length s circles, N_s, is inductively determined by (25).

$$\begin{cases} N_1 = N_e, \\ N_s = \dfrac{Trace(L^s) - \sum\limits_{k \in P(s)} kN_k}{s}, \quad 2 \le s \le 2^n. \end{cases} \tag{25}$$

where $P(s)$ is the set of proper factors of s, $s \in Z^+$. For instance, $P(6)=\{1,2,3\}$.

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

Consider the Boolean network of the example 3, $N_1 = N_2 = N_3 = N_4 = N_5 = N_6 = N_8 = 0$, $N_7 = 1$. Therefore, there is no fixed point in this network, and there is only one circle which length is 7. Moreover, note in (26).

$$L^7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{26}$$

Then each diagonal nonzero column can generate the circle. Choosed $Z = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T$, then

$$LZ = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^T$$

$$L^2Z = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]^T$$

$$L^3Z = [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$$

$$L^4Z = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$$

$$L^5Z = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$L^6Z = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$L^2Z = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T = Z$$

The vector forms in the circle can be got in TABLE IV.

TABLE IV.
VECTOR FORMS IN THE CIRCLE

A (t)	A_3 (t)	A_2 (t)	A_1 (t)
$[0\,0\,0\,0\,0\,0\,0\,1]^T$	0	0	0
$[0\,0\,0\,0\,0\,0\,1\,0]^T$	0	0	1
$[0\,0\,0\,0\,0\,1\,0\,0]^T$	0	1	0
$[0\,0\,0\,0\,1\,0\,0\,0]^T$	0	1	1
$[0\,0\,0\,1\,0\,0\,0\,0]^T$	1	0	0
$[0\,0\,1\,0\,0\,0\,0\,0]^T$	1	0	1
$[0\,1\,0\,0\,0\,0\,0\,0]^T$	1	1	0

The vector forms can be converted back to the scalar form of $A_1(t)$, $A_2(t)$, and $A_3(t)$. The circle is as $000 \to 001 \to 010 \to 011 \to 100 \to 101 \to 110 \to 000$. Finally, the state space graph of the network in Example 3 can be gained as in Figure 1.

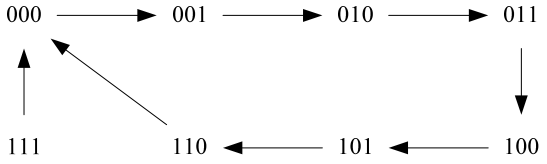

Figure 1. The state space graph

VI. CASE STUDY: MAMMALIAN CELL

In this section, a useful example of mammalian cell[22] is given to show that our new approach is effective and feasible.

A proper understanding of the structure and temporal behaviors of biological regulatory networks requires the integration of regulatory data into a formal dynamical model. A logical framework enables a more systematic and extensive characterization of all the behaviors compatible with a given regulatory graph. Furthermore, this framework offers enumerative or analytical means to identify relevant asymptotical behaviors (stable states, state transition cycles).

The cell cycle involves a succession of molecular events leading to the reproduction of the genome of a cell. Here, the logical regulatory graph for a mammalian cell

cycle network and logical rules associated with the regulatory graph (in Figure 2) are given.

Each node represents the activity of a key regulatory element, whereas the edges represent cross-regulations. Blunt arrows stand for inhibitory effects, normal arrows for activations.

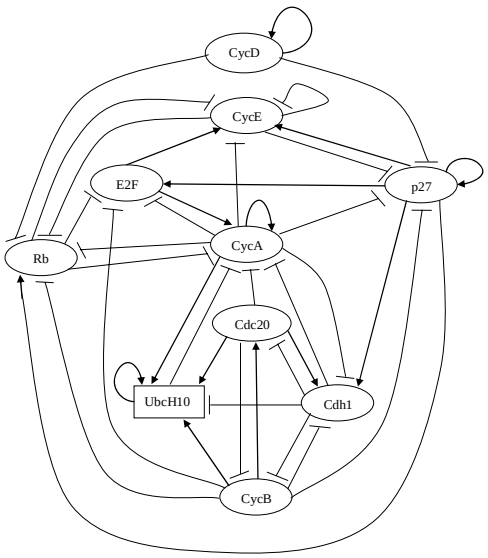

Figure 2. The state space graph

The logical equations, which are called dynamics of Boolean network in STP, are as in (27).

$$\begin{cases} CycD = CycD \\ Rb = (\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \\ \quad \vee (p27 \wedge \overline{(CycE \wedge CycA)} \wedge \overline{CycB} \\ \quad \wedge \overline{CycD}) \\ E2F = (\overline{Rb} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \\ \quad \overline{Rb} \wedge \overline{CycB}) \\ CycE = (E2F \wedge \overline{Rb}) \\ CycA = (E2F \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge \\ \quad \overline{(Cdh1 \wedge UbcH10)}) \vee (CycA \wedge \\ \quad \overline{Rb} \wedge \overline{Cdc20} \wedge \overline{(Cdh1 \wedge UbcH10)}) \\ p27 = (\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \\ \quad \vee (p27 \wedge \overline{(CycE \wedge CycA)} \wedge \\ \quad \overline{CycB} \wedge \overline{CycD}) \\ Cdc20 = CycB \\ Cdh1 = (\overline{CycA} \wedge \overline{CycB}) \vee (Cdc20) \vee \\ \quad (p27 \wedge \overline{CycB}) \\ UbcH10 = \overline{(Cdh1)} \vee (Cdch1 \wedge UbcH10 \\ \quad \wedge (Cdc20 \vee CycA \vee CycB)) \\ CycB = (\overline{Cdc20} \wedge \overline{Cdh1}) \end{cases} \quad (27)$$

Then we use the methods mentioned above to get stable states and state transition cycles.

There are 10 nodes, so the complete state transition graph contains 1024 vertices. The structure matrix L can be gained by algorithm1.

Algorithm1 Algorithm for computing structure matrix L

```
L=zeros[1024][1024]
A=zeros[10]
begin
 for k=0 to 1023 do
  A=int_to_binary (k, 10)
    /*convert k to binary number of 10 bit, */
  CycD=A[1];
  Rb=A[2];
  E2F=A[3];
  CycE=A[4];
  CycA=A[5];
  p27=A[6];
  Cdc20=A[7];
  Cdh1=A[8];
  UbcH10=A[9];
  CycB=A[10];
  /*assignment each bit to the variables,
   from high bit to low bit, A[1] is the highest bit*/
  CycD=CycD;
   Rb= ( (!CycD) && (!CycE) && (!CycA) && (!CycB))
       || (p27&& (!CycD) && (!CycB)) ;
   E2F= ( (!Rb) && (!CycA) && (!CycB)) || (p27&& (!Rb)
        && (!CycB)) ;
  CycE= (E2F&& (!Rb)) ;
  CycA= (E2F&& (!Rb) && (!Cdc20) && (! (Cdh1
           &&UbcH10))) || (CycA&& (!Rb) && (!Cdc20)
           && (! (Cdh1&&UbcH10))) ;
  p27= ( (!CycD) && (!CycE) && (!CycA) && (!CycB)) || (p27
       && (! (CycE&&CycA)) && (!CycD) && (!CycB)) ;
  Cdc20=CycB;
  Cdh1= ( (!CycA) && (!CycB)) || (Cdc20) || (p27&& (!CycB)) ;
  UbcH10= (!Cdh1) || ( (Cdh1) && (UbcH10) && ( (Cdc20) ||
  (CycA)
           || (CycB))) ;
  CycB= ( (!Cdc20) && (!Cdh1)) ;
  /* substitute into the logical equations */
  i=1024- (CycD*512 + Rb*256 + E2F*128 + CycE*64 +
CycA*32
     + p27*16 + Cdc20*8+Cdh1*4 + UbcH10*2 + CycB*1) ;
  j=1024-k;
  L[i][j]=1;
 return L
end
```

According to *Theorem 3*, the characteristics of the mammalian cell example, such as the number of fixed points, the numbers of circles of different lengths and the states in the circles, can be analyzed by algorithms2.

Through Algorithm1 and Algorithm2, the number of stable states or fixed point in the example of mammalian cell is 1. The state or fixed point is 0100010100, which means only Rb, p27 and Cdh1 active, in the absence of CycD. And there is 1 circle with the length of 7 in the example of mammalian cell. The states on the circle include 1011100100, 1001100000, 1000100011, 1000101011, 1000001110, 1010000110, and 1011000100.

Algorithm2 Algorithm for computing the number of stable states; the numbers of circles of different lengths;

```
Input: structure matrix L
N=zeros[1024]
begin
 N[1]=trace (L)
 if N[1]≠0 then
  report  the number of stable states is N[1]
 for i=2 to 1024 do
     T=0
     for j=1 to i/2 do
       if mod (i, j) =0 then
         T=T+j*N[j];
```

```
        return T
    N[i]= (trace (L^i) -T) /i;
    if N[i]>0 then
     report  the number of circle length of i is N[i]
end
```

The fixed point and the circle of the mammalian cell example are as in Figure 3.

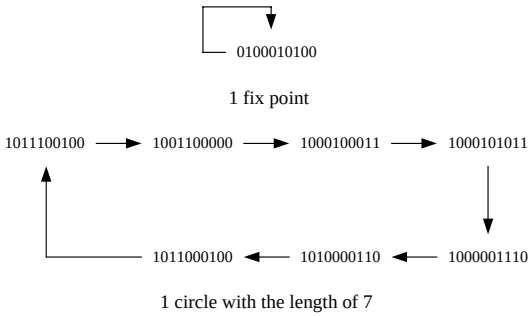

Figure 3. The state space graph

The algorithms in the mammalian cell example in this section seem difficult. In fact, it can be easily done in computer. We have created a program to handle them.

VII. Conclusion

Semi-tensor product is an efficient tool for analyzing the characteristics of Boolean networks which are determined by the structure matrix. Unlike existing methods which calculates the structure matrix through matrix operations with high computation complexity, an optimized approach is proposed in this paper. The approach gets the structure matrix of Boolean network through the truth table which reflects the state transformation of the Boolean network. Compared with the conventional methods, our method can greatly reduce the calculation complexity. A practical Boolean network of mammalian cell shows our approach is effective and efficient.

The structure matrix which is gained in semi-tensor product is a sparse matrix. On the other hand, with the number of variables in a Boolean network increasing, the size of the structure matrix will become larger and larger. These cause higher computation complexity. To optimize the algorithms in section VI, we need to solve the following problems: How to express the structure matrix in sparse matrix? How to analyze the fixed points and circles in the sparse matrix? They are our future work.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant No. ZYGX2009J062 and the National Natural Science Foundation of China under Grant No. 60973016.

References

[1] S. A. Kauffman, “Metabolic stability and epigenesist in randomly constructed genetic nets”, Journal of Theoretical Biology, vol. 22, pp. 437–467, 1969.

[2] T. Akustsu, S. Miyano, and S. Kuhara, “Inferring qualitative relations in genetic networks and metabolic pathways”, Bioinformatics, vol. 16, pp.727–773, 2000.

[3] 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.

[4] I. Shumlevich, R. Dougherty, S. Kim, and W. Zhang, “Probabilistic Boolean network: a rule-based uncertainty model for gene regulatory networks”, Bioinformatics, no. 2, vol. 18, 261–274, 2002.

[5] 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, 23–40, 2002.

[6] M. Aldana, “Boolean dynamics of networks with scale-free topology”, Physica D, vol. 185, 45–66, 2003.

[7] B. Samuelsson, C. Troein, “Superpolynomial growth in the number of attractors in kayffman networks.” Phys. Rev. Lett., vol. 90, pp. 90098701, 2003.

[8] 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, pp. 088701, 2005.

[9] R. Albert, H. G. Othmer, “The topology and signature of the regulatory interactions predict the expression pattern of the segment polarity genes in Drospphila melanogaster”, Journal of Theory Biology, vol. 223, no. 1, pp. 1–18, 2003.

[10] 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.

[11] 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.

[12] 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, 348–354, 2004.

[13] Q. Zhao, “A remark on ‘Scalar Equations for synchronous Boolean Networks with biologicapplications’ by C.Farrow, J.Heidel, J.Maloney, and J.Rogers”, IEEE Trans. Neural Networks, vol. 16, no. 6, 1715–1716, 2005.

[14] D. Cheng, H. Qi, Semi-tensor Product of Matrices-Theory and Application (second edition), Science Press: Beijing, 2011.

[15] D. Cheng, H. Qi, and Z. Li, Analysis and Control of Boolean Networks: A Semi-tensor Product Approach, Springer Press: London, 2011.

[16] D. Cheng, Matrix and Polynomial Approach to Dynamic Control Systems, Science Press: Beijing, 2002.

[17] D. Cheng, “Semi-tensor product of matrices and its applications – A survey”, Proceeding of ICCM, vol. 3, 641–668, 2007.

[18] D. Cheng, H. Qi, and Y. Zhao, “Analysis and control of Boolean networks: a semi-tensor product approach”, ACTA Automatica SINICA, vol. 37, no. 5, 529–539, 2011.

[19] D. Cheng, H. Qi, and Z. Li, Model construction of Boolean network via observed data. IEEE Transactions on Neural Networks, vol. 22, no. 4, pp. 525–536, 2011.

[20] D. Cheng, H. Qi, “A linear representation of dynamics of boolean networks”, IEEE Transactions on Automatic Control, vol. 55, no. 10, pp. 2251–2258, 2011.

[21] D. Cheng, H. Qi, “State-space analysis of Boolean networks”, IEEE Transaction on Neural Networks, vol. 21, no. 4, pp. 584–594, 2010.

[22] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle", Bioinformatics, vol. 22, no. 14, pp. 124–131, 2006.
[23] D. Cheng, Y. Zhao, X. Xu, "Matrix approach to boolean calculus", IEEE Conference on Decision and Control and European Control, pp. 6950–6955, 2011.
[24] D. Cheng, H. Qi, Y. Zhao, "Synthesis of Boolean networks via semi-tensor product", 30th Chinese Control Conference, pp. 6–17, 2011.

Jinyu Zhan was born in Heilongjiang Province of China in 1978. She received the Ph.D. degree in computer applications from the University of Electronic Science and Technology of China in 2006.

Currently, she is an associate professor at the University of Electronic Science and Technology of China. Her research interests include formal co-verification of SoC, model checking, real time embedded systems, and VLSI design and verification.

Shan Lu was born in Hunan Province of China in 1989. He received the Bachelor degree in information and computation science from Chongqing Three Gorges University in 2011.

Currently, he is a graduate student at the University of Electronic Science and Technology of China. His research interests include formal verification and model checking.

Guowu Yang was born in Hubei Province of China in 1966. He received the Ph.D. degree at Electrical and Computer Engineering department of Portland State University in USA in 2005. He was a research associate at Computer Science department of Portland State University from 2005 to 2006.

Currently, he is a professor of College of Computer Science and Engineering at University of Electronic Science and Technololgy of China. His research interests include formal verification and developing corresponding program package, theoretical study of synthesis algorithms in quantum computing, and non-linear control theory.