Automatica 45 (2009) 1659–1667

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Controllability and observability of Boolean control networks ☆

Daizhan Cheng *, Hongsheng Qi

Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, PR China

ARTICLE INFO

Article history:
Received 2 June 2008
Received in revised form
6 December 2008
Accepted 12 March 2009
Available online 11 April 2009

Keywords:
Boolean control network
Network transition matrix
Controllability
Observability

ABSTRACT

The controllability and observability of Boolean control networks are investigated. After a brief review on converting a logic dynamics to a discrete-time linear dynamics with a transition matrix, some formulas are obtained for retrieving network and its logical dynamic equations from this network transition matrix. Based on the discrete-time dynamics, the controllability via two kinds of inputs is revealed by providing the corresponding reachable sets precisely. Then the problem of observability is also solved by giving necessary and sufficient conditions.

1. Introduction

A Boolean network is a network with nodes and directed edges, denoted by $(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is a finite set of nodes and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is the edge set. A node can take a logic value from $\{0, 1\}$ at a discrete time $0, 1, 2, \ldots$. Assume that $A, B \in \mathcal{N}$ and $(A, B) \in \mathcal{E}$, then it means that in the network dynamics $B(k+1)$ depends on $A(k)$.

We give a simple example to describe it.

Example 1. In Fig. 1 we have a Boolean network with two nodes A and B. Its dynamics is described as

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

where the disjunction $\vee$ (conjunction $\wedge$) can be considered as $\max(A(t), B(t))$ $(\min(A(t), B(t)))$.

In 1960s, Jacob and Monod (Nobel Prize winners) found that "Any cell contains a number of 'regulatory' genes that act as switches and can turn one another on and off.... If genes can turn one another on and off, then you can have genetic circuits." (Waldrop, 1992) Based on these Boolean-type actions in genetic circuits, Kauffman proposed using the Boolean network to describe

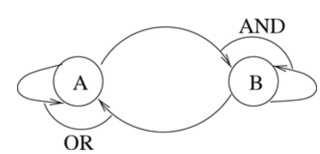


Fig. 1. A network.

the genetic circuits (Kauffman, 1969). Some general descriptions of the Boolean network and its applications to biological systems can be found in Kauffman (1993, 1995). Since then the Boolean network has been investigated widely and has become a power tool in analyzing and manipulating genetic circuits.

The first interesting problem concerns the topological structure of a Boolean network, including its fixed points, its cycles, basin of attractors, and transient times, etc. (Albert & Barabasi, 2000; Albert & Othmer, 2003; Aldana, 2003; Drossel, Mihaljev, & Greil, 2005; Harris, Sawhill, Wuensche, & Kauffman, 2002). The applications of Boolean network to analysis of genetic regulation networks are of particular interest (Akutsu, Miyano, & Kuhara, 2000; Heidel, Maloney, Farrow, & Rogers, 2003; Huang, 2002; Huang & Ingber, 2000).

The control of Boolean networks is also a challenging problem. There are some recent papers concerning this problem (Data, Choudhary, Bittner, & Dougherty, 2003, 2004; Pal, Datta, Bittner, & Dougherty, 2005, 2006). When the random Boolean network is considered, the main interest lies on the stationary distribution of the system. Only for the deterministic network, the reachability problem as in control theory becomes a common concern (Akutsu, Hayashida, Ching, & Ng, 2007).

Recently, a new matrix product, namely, semi-tensor product of matrices, has been introduced. Consider an $m \times n$ matrix A and

☆ This work is supported partly by NNSF 60674022, 60736022, and 60821091 of China. This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Lihua Xie under the direction of Editor Roberto Tempo.

* Corresponding address: Chinese Academy of Sciences, Institute of Systems Science, Beijing 100190, PR China. Tel.: +86 10 6265 1445; fax: +86 10 6258 7343.
E-mail addresses: dcheng@iss.ac.cn (D. Cheng), qihongsh@amss.ac.cn (H. Qi).

0005-1098/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2009.03.006

a $p \times q$ matrix B. We defined a semi-tensor product of A and B, denoted as $A \ltimes B$. We refer to Cheng and Wang (2004) for a brief introduction. When $n = p$, $A \ltimes B = AB$. So it is a generalization of the conventional matrix product, and hence $\ltimes$ can be omitted. Moreover, all the main properties of the conventional matrix product remain true for this generalization. Throughout this paper the matrix product is assumed to be the semi-tensor product.

Using semi-tensor product, a logical function can be converted into an algebraic function (Cheng, 2007). To do this we give logical values a vector form as: $T = 1 \sim \delta_2^1, F = 0 \sim \delta_2^2$, where δ_n^i is the ith column of the identity matrix I_n. Then the logical variable $A(t)$ takes value from these two vectors, i.e.,

$$A(t) \in D := \left\{\delta_2^1, \delta_2^2\right\}.$$

According to Cheng (2007), for each logical function $\xi = \xi(A_1, \ldots, A_n)$ there exists a structure matrix of ξ, say $M_\xi \in M_{2\times 2^n}$, such that as A_i takes vector values, we have

$$\xi(A_1, \ldots, A_n) = M_\xi A_1 \cdots A_n. \tag{2}$$

For instance, for conjunction $\wedge$ and disjunction $\vee$, we can find their structure matrices M_c and M_d as

$$M_c = \delta_2[1, 2, 2, 2]; \qquad M_d = \delta_2[1, 1, 1, 2].$$

Where and hereafter we use the following compact notation: Assume that a matrix L is of the following form

$$L = \left[\delta_q^{i_1}, \delta_q^{i_2}, \ldots, \delta_q^{i_s}\right]. \tag{3}$$

The L is expressed as

$$L = \delta_q[i_1, i_2, \ldots, i_s]. \tag{4}$$

Using the structure matrices, the logical dynamics (1) can be expressed in the following algebraic form:

$$\begin{cases} A(t+1) = M_d A(t)B(t), \\ B(t+1) = M_c A(t)B(t). \end{cases} \tag{5}$$

Using this form, Cheng and Qi (in press) further convert the algebraic form into a standard discrete-time dynamics and then using its transition matrix to provide formulas for fixed points, cycles, transient time and basin of attractions etc. In the next section, we will briefly review it. In Cheng (2009) the control Boolean network was considered. Using the input-state approach, a general structure of Boolean network, called the "rolling gears", is proposed to explain why in a cellular network the smallest cycle(s) plays fundamental role for the properties of overall cellular network as described in Kauffman (1995).

This paper considers two fundamental problems: controllability and observability of a Boolean control network. The paper is organized as follows. Section 2 briefly reviews how to convert a logical dynamics to a discrete-time dynamics proposed by Cheng and Qi (in press). Section 3 provides a systematic procedure to reconstruct the network with its logical dynamics of a Boolean network from its network transition matrix. The controllability via two types of controls is considered in Section 4. Necessary and sufficient conditions are proved for each case by constructing reachable sets for each case. In Section 5 the observability of a Boolean control network with outputs of logical functions is discussed and necessary and sufficient conditions are also proved. Section 6 is a brief concluding remark.

2. Converting a logical dynamics to a discrete-time dynamics

A Boolean network with n nodes A_i, $i = 1, 2, \ldots, n$ can be expressed as

$$\begin{cases} A_1(t+1) = \xi_1(A_1(t), A_2(t), \ldots, A_n(t)), \\ \vdots \\ A_n(t+1) = \xi_n(A_1(t), A_2(t), \ldots, A_n(t)), \end{cases} \tag{6}$$

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

Using (2), for each logical function ξ_i we can find its structure matrix W_i such that the equations in (6) can be converted into an algebraic form as

$$A_i(t+1) = W_i A_1(t) \cdots A_n(t), \quad i = 1, \ldots, n. \tag{7}$$

Define $x(t) = A_1(t)A_2(t)\cdots A_n(t)$. Multiplying all the equations in (7) together yields

$$x(t+1) = W_1 x(t) W_2 x(t) \cdots W_n x(t). \tag{8}$$

Using the properties of semi-tensor product and the power reducing matrix $M_r = \delta_4[1, 4]$ (Cheng, 2007), (6) can be converted to a standard discrete-time dynamic system as

$$x(t+1) = Lx(t), \tag{9}$$

where L is called the network transition matrix of (6). It was proved in Cheng and Qi (in press) that (9) is equivalent to (6).

For example, consider the system (1) in Example 1. Setting $x(t) = A(t)B(t)$, it is easy to show that $x(t+1) = Lx(t)$ with $L = \delta_4[1, 2, 2, 4]$.

Next, we consider a control Boolean network as (Cheng, 2009)

$$\begin{cases} A_1(t+1) = f_1(A_1(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)) \\ \vdots \\ A_n(t+1) = f_n(A_1(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)), \end{cases} \tag{10}$$

$$y_j(t) = h_j(A_1(t), \ldots, A_n(t)), \quad j = 1, 2, \ldots, p, \tag{11}$$

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

Two kinds of controls are considered:

(1) The controls are logical variables satisfying certain logical rule, called the input network, as

$$\begin{cases} u_1(t+1) = g_1(u_1(t), \ldots, u_m(t)), \\ \vdots \\ u_m(t+1) = g_m(u_1(t), \ldots, u_m(t)). \end{cases} \tag{12}$$

(2) The control is a free Boolean sequence. Precisely, set $u(t) = u_1(t)u_2(t)\cdots u_m(t)$. Then the control is a designed sequence $u(0), u(1), \ldots \in D^m$.

Using the structure matrix approach to the Boolean control network, it is easy to obtain the algebraic form of the network (10)–(12) as

$$\begin{cases} u(t+1) = Gu(t), \quad u \in D^m \\ x(t+1) = Lu(t)x(t) := L_u(t)x(t), x \in D^n \end{cases}$$
$$y(t) = Hx(t), \quad y \in D^p, \tag{13}$$

where $L_u(t) = Lu(t)$ is the control-depending network transition matrix, G is the network transition matrix of the input network, H is the transition matrix from x to y (calculated exactly in the same way as for L and G).

3. Reconstructing networks

From a set of input–output data we may identify the structure matrix L. Particularly, in the case of large or huge networks, we may find an L to approximate the original system or a particular input–output response of the original network. We leave the identification problem for further investigation. Since L is the coefficient matrix of a standard discrete-time linear system it seems that many known methods can be used for this purpose.

In this section we consider how to reconstruct the Boolean network from its network matrix L. This is important because we will work on state space and try to design a network matrix. Then we have to convert it back to the network and give its logical relations for design purposes.

Assume that L is known, we will try to retrieve (6) and the network.

First, we have to reconstruct the structure matrices W_i of the logical operators f_i. We define a set of 2×2^n matrices, S_i^n, called retrievers, in the following way. Divide columns, labeled by $1, 2, \ldots, 2^n$, into 2^i equal parts, where $1 \le i \le n$. Then put δ_2^1 into the first segment of columns, and put δ_2^2 into the second segment of columns, then the δ_2^1 again, and continue this process to define S_i^n. In this way we have defined

$$\begin{aligned} S_1^n &= \delta_2[\underbrace{1,\ldots,1}_{2^{n-1}}, \underbrace{2,\ldots,2}_{2^{n-1}}]; \\ S_2^n &= \delta_2[\underbrace{1,\ldots,1}_{2^{n-2}}, \underbrace{2,\ldots,2}_{2^{n-2}}, \underbrace{1,\ldots,1}_{2^{n-2}}, \underbrace{2,\ldots,2}_{2^{n-2}}]; \\ &\vdots \\ S_n^n &= \delta_2[1, 2, 1, 2, \ldots, 1, 2]. \end{aligned} \tag{14}$$

We need the swap matrix $W_{[m,n]}$ (with $W_{[n]} := W_{[n,n]}$), which is the unique $mn \times mn$ matrix, such that for any $X \in \mathbb{R}^m$, $Y \in \mathbb{R}^n$ (Cheng & Wang, 2004)

$$W_{[m,n]}XY = YX.$$

To construct W_i we have

Proposition 2. *The structure matrices W_i of f_i can be retrieved as follows:*

$$W_i = S_i^n L, \quad i = 1, 2, \ldots, n. \tag{15}$$

Proof. We prove (15) for $i = 1$. The proof for other i is similar (using the swap matrix to change the order of factors first). Denote

$$P = A_2(t+1)A_3(t+1)\cdots A_n(t+1) \in D^{n-1}.$$

Then

$$x(t+1) = A_1(t+1)P.$$

If $A_1(t+1) = \delta_2^1$, $x(t+1) = [P^{\mathrm{T}}\ \underbrace{0,\ldots,0}_{2^{n-1}}]^{\mathrm{T}}$, if $A_1(t+1) = \delta_2^2$, $x(t+1) = [\underbrace{0,\ldots,0}_{2^{n-1}}\ P^{\mathrm{T}}]^{\mathrm{T}}$. Note that $P = \delta_{2^{n-1}}^i$, for some i, it follows immediately that $A_1(t+1) = S_1^n x(t+1)$. Equivalently, $W_1 x(t) = S_1^n L x(t)$. Since $x(t) \in D^n$ is arbitrary, $W_1 = S_1^n L$. □

Note that the neighborhood of node i (equivalently, edges, starting from other nodes, toward i), called the in-degree of node i, is usually much smaller than n. We have to find which node is connected to i. We have the following:

Proposition 3. *Consider system (6) with its algebraic form (7). j is not in the neighborhood of i, (i.e., the edge $j \to i$ does not exist), iff W_i satisfies*

$$W_i W_{[2,2^{j-1}]}(M_n - I_2) = 0, \tag{16}$$

where M_n is the structure matrix of negation $\neg$ (Cheng, 2007).

Moreover, as long as (16) holds, the equation of A_i can be replaced by

$$A(t+1) = W_i' A_1(t)\cdots A_{j-1}(t)A_{j+1}(t)\cdots A_n(t), \tag{17}$$

where

$$W_i' = W_i W_{[2,2^{j-1}]}\delta_2^1.$$

Proof. Note that we can rewrite the ith equation of (7) as

$$A_i(t+1) = W_i W_{[2,2^{j-1}]} A_j(t) \prod_{i=1, i\neq j}^{n} A_i(t).$$

Now we replace $A_j(t)$ by $\neg A_j(t)$, if it does not affect the overall structure matrix, it means $A_i(t+1)$ is independent of $A_j(t)$. The invariance of replacement is depicted by (16). As for (17), since $A_j(t)$ does not affect $A_i(t+1)$, we can simply set $A_j(t) = \delta_2^1$ (equally, you can set $A_j(t) = \delta_2^2$ if you wish,) to simplify the expression. □

Repeating the verification of (16), all the redundant dummy variables can be removed from the equation. We give an example to show this.

Example 4. Assume that we have a Boolean network with 5 nodes A, B, C, D, E. Let $x = ABCDE$. We have $x(t+1) = Lx(t)$ with

$$\begin{aligned} L = \delta_{32}[&3, 6, 7, 6, 19, 22, 31, 30, 19, 22, 23, 22, 3, 6, 15, 14, \\ &3, 5, 7, 5, 19, 21, 31, 29, 19, 21, 23, 21, 3, 5, 15, 13]. \end{aligned}$$

We try to recover the logic dynamic system from L. We know that $W_i = S_i^5 L$, $i = 1, 2, 3, 4, 5$, which yield

$$\begin{aligned} W_1 = \delta_2[&1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, \\ &1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1]; \\ W_2 = \delta_2[&1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, \\ &1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2]; \\ W_3 = \delta_2[&1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, \\ &1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2, 1, 2, 2, 2]; \\ W_4 = \delta_2[&2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, \\ &2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1]; \\ W_5 = \delta_2[&1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, \\ &1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]. \end{aligned}$$

Next, considering W_1 it is easy to verify that

$$\begin{aligned} &W_1 M_n = W_1, \qquad W_1 W_{[2]} M_n \neq W_1, \\ &W_1 W_{[2,2^2]} M_n \neq W_1, \qquad W_1 W_{[2,2^3]} M_n = W_1, \\ &W_1 W_{[2,2^4]} M_n = W_1. \end{aligned}$$

We conclude that $A(t+1)$ depends on $B(t)$ and $C(t)$ only. Then we can remove the dummy variables $A(t), D(t), E(t)$ from the first equation $A(t+1) = W_1 A(t)B(t)C(t)D(t)E(t)$ by replacing $A(t)$, $D(t)$, $E(t)$ by $A(t) = D(t) = E(t) = \delta_2^1$, which yields

$$\begin{aligned} A(t+1) &= W_1 \delta_2^1 B(t)C(t)\delta_2^1\delta_2^1 \\ &= W_1 W_{[4,8]}(\delta_2^1)^3 B(t)C(t) \\ &= \delta_2[1, 2, 2, 1]B(t)C(t). \end{aligned} \tag{18}$$

Its logical form is: $A(t+1) = B(t) \leftrightarrow C(t)$. Similarly, we can get the logical equations for other nodes. Finally, we have

$$\begin{cases} A(t+1) = B(t) \leftrightarrow C(t) \\ B(t+1) = C(t) \vee D(t) \\ C(t+1) = D(t) \wedge E(t) \\ D(t+1) = \neg E(t) \\ E(t+1) = A(t) \to E(t). \end{cases} \tag{19}$$

Then we can reconstruct the network as shown in Fig. 2.

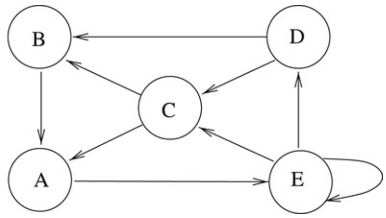


Fig. 2. Reconstructed graph from system matrix.

In general, converting an algebraic form back to logical form is not an easy job. The following proposition provides a mechanical procedure for this.

Proposition 5. *Assume that a logical variable E has an algebraic expression as*

$$E = L(A_1, A_2, \ldots, A_n) = W_L A_1 A_2 \cdots A_n, \tag{20}$$

where W_L is the structure matrix of L. Then

$$E = [A_1 \wedge L_1(A_2, \ldots, A_n)] \vee [\neg A_1 \wedge L_2(A_2, \ldots, A_n)], \tag{21}$$

where $W_L = \left(W_{L_1} \mid W_{L_2}\right)$, i.e., the structure matrix of L_1 (L_2) is the first (last) half of W_L.

Proof. Using (20), when $A_1 = \delta_2^1$

$$E = W_L \delta_2^1 A_2 \cdots A_n = W_{L_1} A_2 \cdots A_n,$$

and when $A_1 = \delta_2^2$

$$E = W_L \delta_2^2 A_2 \cdots A_n = W_{L_2} A_2 \cdots A_n.$$

Then (21) follows. □

Example 6. Assume that

$$E = \delta_2[1, 2, 2, 1, 2, 1, 2, 1, 1, 1, 2, 2, 2, 1, 1, 2]ABCD. \tag{22}$$

Then

$$E = [A \wedge L_1(B, C, D)] \vee [\neg A \wedge L_2(B, C, D)],$$

and

$$W_{L_1} = \delta_2[1, 2, 2, 1, 2, 1, 2, 1],$$
$$W_{L_2} = \delta_2[1, 1, 2, 2, 2, 1, 1, 2].$$

Next,

$$L_1(B, C, D) = [B \wedge L_{11}(C, D)] \vee [\neg B \wedge L_{12}(C, D)],$$

where

$$W_{L_{11}} = \delta_2[1, 2, 2, 1] \quad \Rightarrow \quad L_{11}(C, D) = C \leftrightarrow D;$$
$$W_{L_{12}} = \delta_2[2, 1, 2, 1] \quad \Rightarrow \quad L_{12}(C, D) = \neg D.$$

L_2 can be calculated similarly. Finally, we have

$$\begin{aligned} E = {} & [A \wedge B \wedge (C \leftrightarrow D)] \vee [A \wedge (\neg B) \wedge (\neg D)] \vee \\ & [(\neg A) \wedge B \wedge C] \vee [(\neg A) \wedge (\neg B) \wedge (\neg (C \leftrightarrow D))]. \end{aligned}$$

4. Controllability

The known results on controllability of Boolean control networks is very limited (Akutsu et al., 2007). In this section we consider the problem via two different kinds of controls.

4.1. Control via input Boolean network

Definition 7. Consider system (10) with control (12). Given initial state $x(0) = x_0$ and destination state x_d, x_d is said to be controllable from x_0 (at s steps) with fixed (designable) input structure G, if we can find u_0 (and G), such that $x(u, 0) = x_0$ and $x(u, s) = x_d$ (for a fixed $s \geq 1$).

Note that according to the above definition we may consider four cases: (i) fixed s and fixed G; (ii) fixed s and designable G; (iii) free $s > 0$ and fixed G; (iv) free $s > 0$ and designable G.

Definition 8. For a fixed G the input-state transfer matrix $\Theta^G(t, 0)$ is defined as follows: for any $u_0 \in D^m$ and any $x(0) = x_0 \in D^n$, we have $x(t) = \Theta^G(t, 0)u_0 x_0$, $t > 0$.

It is obvious that $\Theta^G(t, 0)$ depends on G. In the following we will find the input-state transfer matrix. Since

$$x_1 = Lu_0 x_0,$$

we have $\Theta^G(1, 0) = L$. Next, we calculate $x_2 = x(2)$, which is

$$x_2 = Lu_1 x_1 = LGu_0 Lu_0 x_0 = LG(I_{2^m} \otimes L)\Phi_m u_0 x_0,$$

where Φ_m is defined as

$$\Phi_m = \ltimes_{i=1}^{m} I_{2^{i-1}} \otimes \left[\left(I_2 \otimes W_{[2, 2^{m-i}]}\right) M_r\right];$$

$M_r = \delta_4[1, 4]$ is defined in Cheng (2007), and $\otimes$ is the Kronecker product. Then we have $\Theta^G(2, 0) = LG(I_{2^m} \otimes L)\Phi_m$. Using mathematical induction, it is easy to prove that

$$\begin{aligned} \Theta^G(t, 0) = {} & LG^{t-1}(I_{2^m} \otimes LG^{t-2})(I_{2^{2m}} \otimes LG^{t-3}) \cdots \\ & (I_{2^{(t-1)m}} \otimes L)(I_{2^{(t-2)m}} \otimes \Phi_m) \\ & (I_{2^{(t-3)m}} \otimes \Phi_m) \cdots (I_{2^m} \otimes \Phi_m)\Phi_m. \end{aligned} \tag{23}$$

We start from case (i). From the above argument the following result is obvious:

Theorem 9. *Consider system (10) with control (12), equivalently, (13), where G is fixed. x_d is s step reachable from x_0, iff*

$$x_d \in \mathrm{Col}\left\{\Theta^G(s, 0) W_{[2^n, 2^m]} x_0\right\}, \tag{24}$$

where and hereafter Col *is the column set.*

We give an example to describe this result.

Example 10. Consider the following system

$$\begin{cases} A(t+1) = B(t) \leftrightarrow C(t) \\ B(t+1) = C(t) \vee u_1(t) \\ C(t+1) = A(t) \wedge u_2(t), \end{cases} \tag{25}$$

with controls satisfying

$$\begin{cases} u_1(t+1) = g_1(u_1(t), u_2(t)) = \neg u_2(t), \\ u_2(t+1) = g_2(u_1(t), u_2(t)) = u_1(t). \end{cases} \tag{26}$$

Assume that $A(0) = 1$, $B(0) = 0$, and $C(0) = 1$ and $s = 5$. Denote $u(t) = u_1(t)u_2(t)$, then

$$u(t+1) = u_1(t+1)u_2(t+1) = M_n u_2(t) u_1(t) = M_n W_{[2]} u(t).$$

So $G = M_n W_{[2]} = \delta_4[3, 1, 4, 2]$.

$$x(t+1) = M_e B(t) C(t) M_d C(t) u_1(t) M_c A(t) u_2(t) = Lx(t),$$

where

$$\begin{aligned} L = {} & \delta_8[1, 5, 5, 1, 2, 6, 6, 2, 2, 6, 6, 2, 2, 6, 6, 2, \\ & 1, 7, 5, 3, 2, 8, 6, 4, 2, 8, 6, 4, 2, 8, 6, 4]. \end{aligned}$$

$$\Phi_2 = (I_2 \otimes W_{[2]})M_r(I_2 \otimes M_r) = \delta_{16}[1, 6, 11, 16].$$

Finally, using formula (23) yields $\Theta(5, 0) \in M_{8\times 32}$ as

$$\begin{aligned} \Theta(5, 0) = {} & LG^4(I_{2^6} \otimes LG^3)(I_{2^4} \otimes LG^2)(I_{2^6} \otimes LG) \\ & (I_{2^8} \otimes L)(I_{2^6} \otimes \Phi_2)(I_{2^4} \otimes \Phi_2) \\ & (I_{2^2} \otimes \Phi_2)(I_2 \otimes \Phi_2)\Phi_2 \\ = {} & \delta_8[6, 5, 5, 6, 6, 5, 5, 6, 2, 2, 2, 2, 2, 2, 2, 2, \\ & 8, 8, 8, 8, 2, 2, 2, 2, 4, 8, 4, 8, 4, 8, 4, 8]. \end{aligned}$$

Now assume that $(A(0), B(0), C(0)) = (1, 0, 1)$, then $x_0 = A(0)B(0)C(0) = [0, 0, 1, 0, 0, 0, 0, 0]^{\mathrm{T}}$. Using Theorem 9, we have that the reachable set is

$$\Theta(5, 0)W_{[8,4]}x_0 = \delta_8[5, 2, 8, 4].$$

We conclude that the reachable set at step 5 is

$$\left\{\delta_8^5, \delta_8^2, \delta_8^8, \delta_8^4\right\}.$$

Converting them to binary form, we have

$$(A(5), B(5), C(5)) \in \{(0, 1, 1), (1, 1, 0), (0, 0, 0), (1, 0, 0)\}.$$

Finally, we have to find the initial control u_0, which drives the trajectory to the assigned x_d. Since

$$x_d = \Theta(5, 0)W_{[8,4]}x_0u_0 = \delta_8[5, 2, 8, 4]u_0,$$

it is obvious that to reach, say, $5 \sim (0, 1, 1)$, the $u_0 = [1, 0, 0, 0]^{\mathrm{T}}$, i.e., $u_1(0) = 1$ and $u_2(0) = 1$. Similarly, to reach the four points $\{(0, 1, 1), (1, 1, 0), (0, 0, 0), (1, 0, 0)\}$ the corresponding controls should be $(u_1(0), u_2(0)) = \{(1, 1), (1, 0), (0, 1), (0, 0)\}$.

Next, we consider case (ii).

Since there are $m_0 = (2^m)^{2^m}$ possible distinct G's, we may express each G in the condensed form and order them in "increasing order". Say, when $m = 2$ we have $G_1 = \delta_4[1111]$, $G_2 = \delta_4[1112], \ldots, G_{256} = \delta_4[4444]$. In general, we may consider a subset $\Lambda \subset \{1, 2, \ldots, m_0\}$, and allow G be chosen from the admissible set $\{G_\lambda | \lambda \in \Lambda\}$. The following result is an immediate consequence of Theorem 9.

Corollary 11. *Consider system (10) with control (12), where $G \in \{G_\lambda | \lambda \in \Lambda\}$. Then x_d is reachable from x_0, iff*

$$x_d \in \mathrm{Col}\left\{\Theta^{G_\lambda}(s, 0)W_{[2^n,2^m]}x_0 | \lambda \in \Lambda\right\}. \tag{27}$$

Example 12. Consider the system (25) again. We still assume that $A(0) = 1$, $B(0) = 0$, and $C(0) = 1$ (equivalently, $x(0) = \delta_8^3$) and $s = 5$. Assume that the admissible set of G is nonsingular G's. Denote $G_1 = \delta_4[1234]$, $G_2 = \delta_4[1243]$, $G_3 = \delta_4[1324], \ldots, G_{24} = \delta_4[4, 3, 2, 1]$, the corresponding $V_i = \mathrm{Col}\left\{\Theta^i(5, 0)W_{[2^n,2^m]}x_0\right\}$ are

$$\begin{aligned}&\delta_8[5684], \delta_8[5686], \delta_8[5684], \delta_8[5742], \delta_8[5824], \delta_8[5288],\\ &\delta_8[5684], \delta_8[5686], \delta_8[6824], \delta_8[6274], \delta_8[1248], \delta_8[6882],\\ &\delta_8[8564], \delta_8[5284], \delta_8[5684], \delta_8[7681], \delta_8[5288], \delta_8[2678],\\ &\delta_8[6272], \delta_8[8582], \delta_8[8614], \delta_8[5688], \delta_8[2278], \delta_8[5688].\end{aligned}$$

So the reachable set at 5 steps is

$$\{\delta_8^1, \delta_8^2, \delta_8^4, \delta_8^5, \delta_8^6, \delta_8^7, \delta_8^8\}.$$

It is interesting that starting from $(A(0), B(0), C(0)) = (1, 0, 1)$, the only unreachable point in 5 steps is δ_8^3, which is the starting point. Now assume that we want to reach $(A(5), B(5), C(5)) = (1, 1, 1)$, which is δ_8^1. Since the first component of V_{11} is 1, (we have some other choices such as V_{16}, V_{21},) we can choose G_{11} and $u_1(0)u_2(0) = \delta_4^1$ to drive $(1, 0, 1)$ to $(1, 1, 1)$ in 5 steps. It is easy to figure out that $G_{11} = \delta_4[2413]$.

From $u_1(0)u_2(0) = \delta_4^1$, we have $u_1(0) = 1$ and $u_2(0) = 1$.

To reconstruct the control dynamics, we need retrievers

$$S_1^2 = \delta_2[1, 1, 2, 2]; \qquad S_2^2 = \delta_2[1, 2, 1, 2].$$

Then we have the structure matrices as

$$W_1 = S_1^2G = \delta_2[1, 2, 1, 2]; \qquad W_2 = S_2^2G = \delta_2[2, 2, 1, 1].$$

It follows that

$$\begin{aligned}u_1(t + 1) &= W_1u_1(t)u_2(t) = u_2(t);\\ u_2(t + 1) &= W_2u_1(t)u_2(t) = \neg u_1(t).\end{aligned}$$

Finally, we consider cases (iii) and (iv), i.e., for free s.

First we give a lemma, which itself is interesting.

Lemma 13. *For a Boolean network, if its network transition matrix is nonsingular, then every point is on a cycle.*

Before proving this lemma, we need some preparation. The transient period T_t is the smallest time, such that starting from any x_0 and after T_t time the trajectory will enter an attractor.

Lemma 14 (Cheng & Qi, in press). *The transient period T_t is the smallest $k \ge 0$ such that there exists a $T > 0$ such that*

$$L^k = L^{k+T}.$$

Proof of Lemma 13. According to Lemma 14, it suffices to show that the transient period T_t is zero. Let the network matrix be L. Consider the sequence $L, L^2, \ldots$. Since there are only finite distinct $2^n \times 2^n$ logical matrices, there must be two integers $p < q$ such that $L^p = L^q$. It follows that $L^{p-q} = I$, which means the transient period is zero. □

In the following we assume that

A1 G is nonsingular.

According to Lemma 13, we, starting from u_0, can find a minimum $T_0 > 0$ such that $G^{T_0}u_0 = u_0$. Hence $u_0, Gu_0, \ldots, G^{T_0}u_0$ is a cycle of length T_0. Following the procedure in Cheng (2009), we can construct a mapping

$$\Psi := (LG^{T_0-1}u_0)(LG^{T_0-2}u_0)\cdots(LGu_0)(Lu_0). \tag{28}$$

Then for x_0 we consider the sequence $x_0, \Psi x_0, \ldots$, and find the transient period r_1 and a minimum $T_1 > 0$ such that

$$\Psi^{r_1}x_0 = \Psi^{r_1+T_1}x_0. \tag{29}$$

Then the reachable set starting from x_0 with u_0, can be constructed easily. We give the following algorithm:

- Step 1. Find T_0 such that $u_0, Gu_0, \ldots, G^{T_0}u_0$ is a cycle in the input space.
- Step 2. Find the transient period r_1 and minimum $T_1 > 0$, satisfying (29).
- Step 3. Construct a sequence

$$x_0^i = \Psi^i x_0, \quad i = 0, 1, 2, \ldots, r_1 + T_1 - 1. \tag{30}$$

- Step 4. For each x_0^i construct inductively a sequence

$$x_j^i = LG^{j-1}u_0x_{j-1}^i, \quad j = 1, \ldots, T_0 - 1. \tag{31}$$

Note that the above construction is the special case of the general one discussed in Cheng (2009) for constructing input-state product cycles. So it is easily seen that $\{x_j^i\}$ is the set of reachable points starting from x_0 using u_0 and fixed G. We write it as the following theorem.

Theorem 15. *Consider system (10) with control (12). Assume A1 and use the above algorithm, then*

(1) *for given u_0 and G_i, the set of reachable states is*

$$R_{u_0}^i = \left\{x_j^i | i = 0, 1, \ldots, r_1 + T_1 - 1; j = 0, 1, \ldots, T_0 - 1\right\};$$

where $\{x_j^i\}$ are constructed by (30)–(31) and the steady state reachable set is

$$RS_{u_0}^i = \left\{x_j^i \in R_{u_0}^i | i \ge r_1\right\};$$

(2) *for fixed $G = G_i$, the reachable set from x_0 is*

$$R^i = \bigcup_{u_0} R_{u_0}^i;$$

(3) *for admissible $\{G_\lambda | \lambda \in \Lambda\}$, the reachable set is*

$$R = \bigcup_{\lambda\in\Lambda}\bigcup_{u_0} R_{u_0}^\lambda.$$

Table 1
Reachable set for $G_1 = \delta_4[1, 2, 3, 4]$.

$u(0)$	T_0	r_1	T_1	R^{G_1}
1	1	2	2	(2, 3, 5)
2	1	2	1	(3, 6)
3	1	1	7	(3, 4, 8)
4	1	4	1	(3, 4, 6, 8)

Table 2
Reachable set for $G_2 = \delta_4[2, 4, 3, 1]$.

$u(0)$	T_0	r_1	T_1	R^{G_2}
1	3	2	1	(1, 2, 3, 4, 5, 8)
2	3	2	1	(2, 3, 5, 6, 8)
3	1	1	7	(2, 3, 4, 5, 6, 7, 8)
4	3	2	1	(3, 6, 8)

Example 16. Consider the system (25) again with $x(0) = \delta_8^3$. It is easy to get the reachable set for each G and each $u(0)$. We give two special G's.

- $G_1 = \delta_4[1, 2, 3, 4]$.
  So the overall reachable set for G_1 is $\{2, 3, 4, 5, 6, 8\}$ (Table 1).
- $G_2 = \delta_4[2, 4, 3, 1]$.
  So the overall reachable set for G_2 is D^3 (Table 2), which means the system is G_2-controllable from $(1, 0, 1)$, (equivalently, $x(0) = \delta_8^3$).

4.2. Controllability via free Boolean sequence

In the following we consider the case when the controls are free Boolean sequences. The following definition is from Akutsu et al. (2007) with our notation.

Definition 17 (*Akutsu et al., 2007*). Given $x_0, x_e \in D^n$. The Boolean control network (10) is said to be controllable from x_0 to x_e (by free Boolean sequence) at the s steps, if we can find control $u(t) \in D^m$, $t = 0, 1, \ldots, s-1$, such that the state $\ltimes_{i=1}^n A_i(0) = x_0$ and $\ltimes_{i=1}^n A_i(s) = x_e$, $i = 1, \ldots, n$.

Define $\tilde{L} = LW_{[2^n,2^m]}$, then the second equation in (13) can be expressed as

$$x(t+1) = \tilde{L}x(t)u(t). \tag{32}$$

Using it repetitively yields

$$x(s) = \tilde{L}^s x(0)u(0)u(1)\cdots u(s-1). \tag{33}$$

So the answer to this kind of control problem is obvious.

Theorem 18. *x_e is reachable from x_0, at the sth time step by controls of Boolean sequences of length s, iff*

$$x_s \in \mathrm{Col}\{\tilde{L}^s x_0\}. \tag{34}$$

Example 19 (*Akutsu et al., 2007*). Consider the Boolean control system depicted in Fig. 3.

Its logical equation is

$$\begin{cases} A(t+1) = B(t) \wedge u_1(t) \\ B(t+1) = \neg u_2(t) \\ C(t+1) = A(t) \vee B(t). \end{cases} \tag{35}$$

Denote $x(t) = A(t)B(t)C(t)$, $u(t) = u_1(t)u_2(t)$. Then we can express the system by

$$x(t+1) = \tilde{L}x(t)u(t) \tag{36}$$

where $\tilde{L}$ is

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

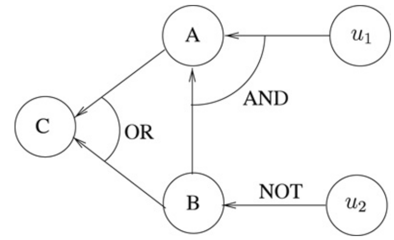


Fig. 3. A Boolean control network.

As in Akutsu et al. (2007) we assume that $(A(0), B(0), C(0)) = (0, 0, 0)$. We want to know if a design state can be reached at the sth step. Say, $s = 3$. Using Theorem 18, we calculate $\tilde{L}^3 x_0 \in M_{8\times 64}$ as

$$\tilde{L}^3 x_0 = \delta_8[8, 6, 8, 6, 3, 1, 7, 5, 8, 6, 8, 6, 3, 1, 7, 5 \\ 7, 5, 7, 5, 3, 1, 7, 5, 8, 6, 8, 6, 3, 1, 7, 5 \\ 8, 6, 8, 6, 3, 1, 7, 5, 8, 6, 8, 6, 3, 1, 7, 5 \\ 7, 5, 7, 5, 3, 1, 7, 5, 8, 6, 8, 6, 3, 1, 7, 5].$$

It is clear that at the 3rd step all states, but δ_{16}^2 δ_{16}^4, can be reached. Now we choose one state, say, 5, which means $\delta_8^5 \sim (0, 1, 1)$. Note that in 8th, 16th, 18th, 20th $\cdots$ columns we have 5, which means controls δ_{64}^8, or δ_{64}^{16}, or δ_{64}^{18}, or δ_{64}^{20}, or $\cdots$ can drive the initial state $(0, 0, 0)$ to the destination state $(0, 1, 1)$. we choose, for example,

$$u_1(0)u_2(0)u_1(1)u_2(1)u_1(2)u_2(2) = \delta_{64}^8.$$

Converting $64-8 = 56$ to binary form yields 111000, which means the corresponding controls are: $u_1(0) = 1$, $u_2(0) = 1$, $u_1(1) = 1$, $u_2(1) = 0$, $u_1(2) = 0$, $u_2(2) = 0$. It is easy to check directly that this set of controls works. We may check some others. Say, choosing δ_{64}^{24}, similar calculation yields the controls as: $u_1(0) = 1$, $u_2(0) = 0$, $u_1(1) = 1$, $u_2(1) = 0$, $u_1(2) = 0$, $u_2(2) = 0$, which also works.

In general, it is easy to calculate that when $s = 1$ the reachable set from $(0, 0, 0)$ is $\{(0, 1, 0), (0, 0, 0)\}$. When $s > 1$ the reachable set is $\{(1, 1, 1), (1, 0, 1), (0, 1, 1), (0, 1, 0), (0, 0, 1), (0, 0, 0)\}$.

A generalization for the controllability via controls of Boolean sequences is when the length of sequences, s, is free. An immediate consequence of Theorem 18 is

Corollary 20. *x_d is reachable from x_0, iff*

$$x_d \in \mathrm{Col}\left\{\bigcup_{i=1}^{\infty} \tilde{L}^i x_0\right\}. \tag{37}$$

Denote by $R(x_0, s)$ the reachable set from x_0 at time s, and $R(x_0) = \bigcup_{s\geq 0} R(x_0, s)$. The following proposition makes (37) verifiable.

Proposition 21. (1) *The reachable set, $R(x_0)$, is a subset of* $\mathrm{Col}\{\tilde{L}\}$*;*
(2) *Assume that k^* is the smallest $k > 0$, such that*

$$\mathrm{Col}\{\tilde{L}^{k+1}x_0\} \subset \mathrm{Col}\left\{\tilde{L}^s x_0 \,\middle|\, s = 1, 2, \ldots, k\right\},$$

then the reachable set

$$R(x_0) = \mathrm{Col}\left\{\bigcup_{i=1}^{k^*} \tilde{L}^i x_0\right\}. \tag{38}$$

Proof. (1) A straightforward computation shows that $\tilde{L}^k x_0 \in M_{2^n \times 2^{km}}$. Since $\tilde{L} \in M_{2^n \times 2^{n+m}}$ by the property of semi-tensor product we have (cf Cheng and Wang (2004))

$$\tilde{L}^{k+1}x_0 = \tilde{L} \ltimes \tilde{L}^k x_0 = \tilde{L} \cdot [\tilde{L}^k x_0 \otimes I_{2^m}],$$

where $\cdot$ is the conventional matrix product. The conclusion follows immediately.

(2) We use the notation

$$\mathrm{Col}\{\tilde{L}^k\} \otimes I_m := \left\{ X \otimes I_m | X \in \mathrm{Col}\{\tilde{L}^k\} \right\}.$$

Assume that

$$\mathrm{Col}\{\tilde{L}^{k+1}x_0\} \subset \mathrm{Col}\{\tilde{L}^s x_0 \Big| s = 1, 2, \ldots, k\}.$$

Then

$$\begin{aligned}\mathrm{Col}\{\tilde{L}^{k+2}x_0\} &= \left\{\tilde{L}\eta \Big| \eta \in \mathrm{Col}\{\tilde{L}^{k+1}x_0\} \otimes I_m \right\} \\ &\subset \left\{\tilde{L}\eta \Big| \eta \in \mathrm{Col}\{\tilde{L}^s x_0\} \otimes I_m, s = 1, 2, \ldots, k \right\} \\ &= \mathrm{Col}\left\{\tilde{L}^s x_0 | \ s = 2, 3, \ldots, k+1\right\} \\ &\subset \mathrm{Col}\left\{\tilde{L}^s x_0 \otimes I_m | \ s = 1, 2, 3, \ldots, k\right\}.\end{aligned}$$

This inequality shows that after k there are no more new columns. From part 1 we know that such k^* does exist. □

Example 22. Consider Example 19 again. We denote the 8 possible initial points by (in decreasing order) $x_0^1 = (1, 1, 1)$, $x_0^2 = (1, 1, 0), \ldots, x_0^8 = (0, 0, 0)$. Then it is easy to see that for all of them the first degenerate steps are the same, which is $s_0 = 3$. For $x_0^1, x_0^2, x_0^5, x_0^6$, the first step reachable set is:

$$\begin{aligned}R(x_0^1, 1) &= R(x_0^2, 1) = R(x_0^5, 1) = R(x_0^6, 1) \\ &= \{(1, 1, 1), (1, 0, 1), (0, 1, 1), (0, 0, 1)\}.\end{aligned}$$

For x_0^3, x_0^4, the first step reachable set is:

$$R(x_0^3, 1) = R(x_0^4, 1) = \{(0, 1, 1), (0, 0, 1)\}.$$

For $x_0^7, x_0^8 8$, the first step reachable set is:

$$R(x_0^7, 1) = R(x_0^8, 1) = \{(010), (000)\}.$$

They have the same second step reachable set

$$\begin{aligned}R(x_0^i, 2) = \{&(1, 1, 1), (1, 0, 1), (0, 1, 1), (0, 1, 0), \\ &(0, 0, 1), (0, 0, 0)\}, \quad i = 1, 2, \ldots, 8.\end{aligned}$$

Note that since $R(x_0^i, 2) = \mathrm{Col}\{\tilde{L}\}$, according to part 1 of Proposition 21, no more states can be reached.

Definition 23. System (10) is said to be globally reachable from x_0 (by controls of free length Boolean sequence) if $R(x_0) = D^n$. System (10) is called globally controllable (by controls of free length Boolean sequence) if $R(x_0) = D^n$, $\forall x_0 \in D^n$.

Example 24. Consider the following system

$$\begin{cases} A(t+1) = B(t) \wedge u_1(t) \\ B(t+1) = C(t) \leftrightarrow (\neg u_2(t)) \\ C(t+1) = A(t) \vee u_2(t). \end{cases} \tag{39}$$

It is easy to check that from point $x_0 = (1, 0, 0)$ the first three steps' reachable sets are:

$$\begin{aligned}R(x_0, 1) &= \{(0, 1, 1), (0, 0, 1)\}; \\ R(x_0, 2) &= \{(1, 1, 0), (1, 0, 1), (0, 1, 0), (0, 0, 1)\}; \\ R(x_0, 3) &= \{(1, 1, 1), (1, 0, 1), (1, 0, 0), (0, 1, 1), \\ &\quad (0, 1, 0), (0, 0, 1), (0, 0, 0)\}.\end{aligned}$$

So system (39) is globally reachable from $(1, 0, 0)$.

It is obvious that control by free length Boolean sequences is the strongest way of control. It was pointed out by some literature that in some Boolean network problems the controls can only be generated by a Boolean system of controls. The control of free length Boolean sequences could destroy the cycle structure of the systems, which could be very important, such as deciding the type of cells.

5. Observability

It is obvious that for a Boolean network the observability is control depending. We first give a definition.

Definition 25. System (10) with outputs (11) is said to be observable if for any initial state x_0 there exists at least a Boolean sequence of control, such that the initial state can be determined by the output sequence.

We give an algorithm for observability.

- Step 1. Construct a sequence Γ_i, $i = 1, 2, \ldots$, which are sets of $2^n \times 2^n$ matrices as follows:

  $\Gamma_1 = \left\{L\delta_{2^m}^i | i = 1, 2, \ldots, 2^m\right\}$;

  $\Gamma_{k+1} = \left\{L\delta_{2^m}^i \gamma | \gamma \in \Gamma_k; i = 1, 2, \ldots, 2^m\right\}, \quad k \geq 1$;

  If $\mathrm{Col}\{\Gamma_{k^*+1}\} \subset \mathrm{Col}\{\Gamma_i | i \leq k^*\}$, k^*+1 is called the degenerated step. Let $k^* > 0$ be the first degenerated step, the sequence will stop at k^*. (Since there are at most 2^n different columns, $k^* \leq 2^n$.
- Step 2. Construct a sequence of sets of $2^p \times 2^n$ matrices as $H_0 = H$, $H_i = H\Gamma_i = \{H\gamma | \gamma \in \Gamma_i\}$.
- Step 3. Using condensed form, each matrix in H_i becomes a 2^n dimensional row.

  Choosing $h^0 \sim H$ and linearly independent rows $h_j^i \in H_i$, $i = 1, 2, \ldots, k^*$ to form a matrix as

  $$\mathcal{C} = \left[(h^0)^{\mathrm{T}} \ (h_1^1)^{\mathrm{T}} \ \cdot \ (h_{i_1}^1)^{\mathrm{T}} \ \cdot \ (h_1^{k^*})^{\mathrm{T}} \ \cdot \ (h_{i_{k^*}}^{k^*})^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{40}$$

Theorem 26. *Assume that system (10) is globally controllable, then with outputs (11) it is observable, iff $\mathcal{C}$ has all distinct columns.*

Proof. Starting from one point x_0 we can observe Hx_0. Using different controls $\delta_{2^n}^i$, we can observe $HL\delta_{2^n}^i$. Using different $\delta_{2^n}^i$ is allowed because the system is globally controllable. Hence we can start from the same point as many times as we wish. Continuing this process, one sees that

$$HL\delta_{2^n}^{i_1} L\delta_{2^n}^{i_2} \cdots L\delta_{2^n}^{i_s} x_0, \quad s \geq 0$$

are observable. Since $s \geq k_0$ adds no linearly independent rows to the previous set, and linearly dependent row is useless in distinguishing initial values, the initial values can be distinguished, iff $\mathcal{C}$ contains all distinct columns. □

Next, we consider the controllability and observability with control of sequence of $1-0-\varnothing$, where $\varnothing$ means the input channel is disconnected. This is reasonable. For instance, in cellular network the active cycles determine the type of cells. Now the genetic regulation network can change the active cycles in the cellular network to change the type of cells. But it acts only over a very short time period like a pulse. So the control becomes a sequence of $1-0-\varnothing$.

When an input u_i is disconnected, we should ask what is the *nominal network* dynamics? Principally, it is reasonable to ask the network graph being a subgraph of the original one by removing u_i related edges. In this way the nominal network graph is unique. But the nominal network dynamics could be different. To specify it, we assume that it has a network matrix $L_\varnothing$. For convenience, we assume that there is a *frozen control* $u_i^{\varnothing} = constant$ such that the ith input disconnected system has the form as $u_i = u_i^{\varnothing}$. When $u_i = u_i^{\varnothing}$, $\forall i$, the control-free system is the nominal network of the original Boolean control network. That is,

$$L_{\varnothing} = Lu_1^{\varnothing}u_2^{\varnothing}\cdots u_m^{\varnothing}.$$

In many cases we are only interested in the steady state case. For the nominal Boolean network, let C^i, $i = 1, 2, \ldots, k$ be its cycles (attractors), and denote by $S = \bigcup_{i=1}^{k} C^i$ its set of steady states, B^i denotes the region of attraction of C^i.

Definition 27. A Boolean network is globally steady state controllable by control of sequences of $1 - 0 - \varnothing$, if for any two points $x, y \in S$ there is a control of sequences of $1 - 0 - \varnothing$, which drives the trajectory from x to y. A Boolean network is steady state observable, if for any $x_0, y_0 \in S$, there is a control sequence of $1 - 0 - \varnothing$, such that x_0, y_0 are distinguished from outputs.

The following result is a direct consequence of the definition and Theorem 26.

Proposition 28. (1) *Consider a Boolean control network, its nominal system has cycles C^i, $i = 1, 2, \ldots, k$. The system is globally steady state controllable, iff for any $1 \le i, j \le k$ there exist at least one $x \in C^i$, one $y \in B_j$ and a $1 - 0 - \varnothing$ sequence of control, which drives x to y.*
(2) *If a Boolean control network is steady state controllable, then it is steady state observable, iff $\mathcal{C}$, defined in (40), has all distinct columns.*

Proof. Note that a point on a cycle of the nominal system can be reached infinity times as $\varnothing$ is used. Then the conclusions are trivial. □

We give an example.

Example 29. Consider system (25) in Example 10. It is natural to assume its nominal system to be (by using frozen controls $u_1^{\varnothing} = 0$ and $u_2^{\varnothing} = 1$)

$$\begin{cases} A(t+1) = B(t) \leftrightarrow C(t) \\ B(t+1) = C(t) \\ C(t+1) = A(t). \end{cases} \tag{41}$$

Using the technique developed in Cheng and Qi (in press), it is easy to calculate that there are two cycles: equilibrium C^1 : $(1, 1, 1)$ and length 7 cycle

$$C^2 : (1,1,0) \to (0,0,1) \to (0,1,0) \to (0,0,0) \\ \to (1,0,0) \to (1,0,1) \to (0,1,1) \to (1,1,0).$$

Since there are no transient states, globally steady state controllable is the same as globally controllable. To prove global steady state controllability, we have to find a control to drive a point in one cycle to the other and vise versa.

Let $(A(0), B(0), C(0)) = (1, 1, 1) \in C^1$ and use $u_1(0) = 0$, $u_2(0) = 0$. Then $(A(1), B(1), C(1)) = (1, 1, 0) \in C^2$. Let $(A(0), B(0), C(0)) = (1, 0, 0) \in C^2$ and use $u_1(0) = 1, u_2(0) = 1$. Then $(A(1), B(1), C(1)) = (1, 1, 1) \in C^1$. By Proposition 28, system (25) is globally steady state controllable.

Now we assume that the outputs are

$$\begin{aligned} y_1(t) &= A(t) \\ y_2(t) &= B(t) \vee C(t). \end{aligned} \tag{42}$$

Then we have

$$y(t) := y_1(t)y_2(t) = A(t)M_dB(t)C(t) = Hx(t),$$

where $H \in M_{4\times 8}$ is

$$H = \delta_4[1, 1, 1, 2, 3, 3, 3, 4].$$

For system (25), it is easy to calculate that

$$L = \delta_8[1, 5, 5, 1, 2, 6, 6, 2, 2, 6, 6, 2, 2, 6, 6, 2, \\ 1, 7, 5, 3, 2, 8, 6, 4, 2, 8, 6, 4, 2, 8, 6, 4].$$

Then we can calculate that

$$HL\delta_4^1 = \delta_4[1, 3, 3, 1, 1, 3, 3, 1];$$
$$HL\delta_4^2 = \delta_4[1, 3, 3, 1, 1, 3, 3, 1];$$
$$HL\delta_4^3 = \delta_4[1, 3, 3, 1, 1, 4, 3, 2];$$
$$HL\delta_4^4 = \delta_4[1, 4, 3, 2, 1, 4, 3, 2].$$

We need only to construct part of $\mathcal{C}$. Choosing linearly independent rows, we have

$$\mathcal{C} = \begin{bmatrix} H \\ HL\delta_4^1 \\ HL\delta_4^2 \\ HL\delta_4^3 \\ HL\delta_4^4 \\ \vdots \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 2 & 3 & 3 & 3 & 4 \\ 1 & 3 & 3 & 1 & 1 & 3 & 3 & 1 \\ 1 & 3 & 3 & 1 & 1 & 4 & 3 & 2 \\ 1 & 4 & 3 & 2 & 1 & 4 & 3 & 2 \\ \vdots & & & & & & & \end{bmatrix}.$$

From part of $\mathcal{C}$ it is enough to see that there are no equal columns in $\mathcal{C}$. So the system is observable.

6. Conclusion

The paper considered the controllability and observability of Boolean control networks. As a necessary tool, we first discussed how to reconstruct a Boolean network from its known network matrix. Then the controllability via two kinds of controls has been investigated. First, assume that the controls are generated by a control Boolean network. Second, assume that the controls are free Boolean sequences (with control-disconnected moments). In both cases, necessary and sufficient conditions have been obtained to show the reachable sets precisely. The observability problem has also been solved for the controls of free Boolean sequences.[1]

Overall, the paper provided a framework for using system and control techniques to analyze and manipulate Boolean networks.

Since the dimension of state space is 2^n, where n is the number of nodes, as n is large, the complexity of computation is a series problem in this approach. It is not discussed in this paper. As mentioned at the beginning of Section 3, a large network or its some particular input–output responses may be approximated by a smaller network.

There are many control related problems for Boolean control systems, such as realization, stabilization and optimal control etc., which remain for further study.

References

Akutsu, T., Hayashida, M., Ching, W., & Ng, M. K. (2007). Control of Boolean networks: hardness results and algorithms for tree structured networks. *Journal of Theoretical Biology*, *244*, 670–679.

Akutsu, T., Miyano, S., & Kuhara, S. (2000). Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, *16*, 727–773.

Albert, R., & Barabasi, A. L. (2000). Dynamics of complex systems: Scaling laws or the period of Boolean networks. *Physical Review Letters*, *84*, 5660–5663.

Albert, R., & Othmer, H. G. (2003). The topology and signature of the regulatory interactions predict the expression pattern of the segment polarity genes in Drospphila melanogaster. *Journal of Theoretical Biology*, *223*(1), 1–18.

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

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Physica D*, *185*, 45–66.

Cheng, D. (2007). Semi-tensor product of matrices and its applications — A survey. In: *Proc. ICCM 2007*. *Vol. 3* (641–668).

Cheng, D. (2009). Input-state approach to Boolean networks. *IEEE Transactions on Neural Network*, *20*(3), 512–521.

Cheng, D., Hu, X., & Wang, Y. (2004). Non-regular feedback linearization of nonlinear systems via a normal form algorithm. *Automatica*, *40*(3), 439–447.

Cheng, D., & Qi, H. (2009). A linear representation of dynamics of Boolean networks. *IEEE Transactions on Automatic Control*, in press (preprint: http://lsc.amss.ac.cn/~dcheng).

Data, A., Choudhary, A., Bittner, M. L., & Dougherty, E. (2003). External control in Markovian genetic regulatory networks. *Machine Learning*, *52*, 169–191.

Data, A., Choudhary, A., Bittner, M. L., & Dougherty, E. (2004). External control in Markovian genetic regulatory networks: The imperfect information case. *Bioinformatics*, *20*, 924–930.

Drossel, B., Mihaljev, T., & Greil, F. (2005). Number and length of attractors in a critical Kauffman model with connectivity one. *Physical Review Letters*, *94*, 088701.

Harris, S. E., Sawhill, B. K., Wuensche, A., & Kauffman, S. (2002). A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complexity*, *7*, 23–40.

Heidel, J., Maloney, J., Farrow, J., & Rogers, J. (2003). Finding cycles in synchronous Boolean networks with applications to biochemical systems. *International Journal of Bifurcation and Chaos*, *13*(3), 535–552.

Huang, S. (2002). Regulation of cellular states in mammalian cells from a genomewide view. In C.-V. J. R. Hofestadt (Ed.), *Gene Regulation and Metabolism* (pp. 181–220). Cambridge, MA: MIT Press.

Huang, S., & Ingber, I. (2000). Shape-dependent control of cell growth, differentiation, and apotosis: Switching between attractors in cell regulatory networks. *Exp. Cell Res.*, *261*, 91–103.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, *22*, 437–467.

Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. New York: Oxford University Press.

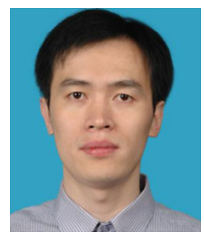
Kauffman, S. A. (1995). *At home in the universe*. Oxford: Oxford Univ. Press.

Pal, R., Datta, A., Bittner, M. L., & Dougherty, E. R. (2005). Intervention in context-sensitive probabilistic Boolean networks. *Bioinformatics*, *21*, 1211–1218.

Pal, R., Datta, A., Bittner, M. L., & Dougherty, E. R. (2006). Optimal infinitehorizon control for probabilistic Boolean networks. *IEEE Transactions on Signal Process*, *54*, 2375–2387.

Waldrop, M. M. (1992). *Complexity*. New York: Touchstone.

Daizhan Cheng received the Ph.D. degree from Washington University, St. Louis, in 1985. He is currently a professor with Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Chairman of Technical Committee on Control Theory (2003–), Chinese Association of Automation, Fellow of IEEE and Fellow of IFAC. His research interests include nonlinear systems, numerical method, switched systems, systems biology etc.



Hongsheng Qi received the Ph.D. degree in systems theory from Academy of Mathematics and Systems Science, Chinese Academy of Sciences in 2008. He is currently a post doctoral fellow at the Key Laboratory of Systems and Control, Chinese Academy of Sciences. His research interests include nonlinear control, systems biology etc.