Input-State Approach to Boolean Networks
Daizhan Cheng, Fellow, IEEE

Abstract—This paper investigates the structure of Boolean networks via input-state structure. Using the algebraic form proposed by the author, the logic-based input-state dynamics of Boolean networks, called the Boolean control networks, is converted into an algebraic discrete-time dynamic system. Then the structure of cycles of Boolean networks is obtained as compounded cycles. Using the obtained input-state description, the structure of Boolean networks is investigated, and their attractors are revealed as nested compounded cycles, called rolling gears. This structure explains why small cycles mainly decide the behaviors of cellular networks. Some illustrative examples are presented.

Index Terms—Algebraic form, input-state structure, invariant subspace, network transition matrix.

I. INTRODUCTION

INPUT–OUTPUT structure is essential in systems and control theory. How about the cellular networks? It was pointed out by [11] that “Gene-regulatory networks are defined by trans and cis logic. . . . Both of these types of regulatory networks have input and output.” Ignoring outputs, this paper focuses on input-state structure only.

A Boolean network could be a description of genetic circuits, an explanation of self-organization in organisms, and the structure causing order in the evolution, which leads to life [12]. In Boolean network model, gene expression is quantized to only two levels: “T”(True) and “F”(False), or “1” and “0,” respectively, denoted by $D = \{ T, F \}$, or $D = \{ 1, 0 \}$. We refer to [7] for logical notations, concepts, and operators used in this paper, and refer to [2] for some related works in neural networks.

Denote the nodes of a network graph by $A_1(t), A_2(t), \ldots$. Each node is functionally related to the expression states of some other nodes. If $A_i$ is affected directly by $A_j$, there is a directed edge from $A_j$ to $A_i$, and it is said that $A_j$ is in the neighborhood of $A_i$. It can also be understood as the intracellular signal transduction from $j$th cellular to $i$th cellular. Throughout this paper, we consider only the networks that have fixed graph topologies. The actions between genes are described by logical rules, which are described by a logical dynamic equation [6]

$$
\begin{align*}
A_1(t+1) &= f_1(A_1(t), A_2(t), \ldots, A_n(t)) \\
A_2(t+1) &= f_2(A_1(t), A_2(t), \ldots, A_n(t)) \\
&\vdots \\
A_n(t+1) &= f_n(A_1(t), A_2(t), \ldots, A_n(t))
\end{align*}
$$

(1)

Fig. 1. Graph of network.

where $f_i, i = 1, 2, \ldots, n$, are logical functions (also called $n$-ary logical operators[1]).

We use an example to illustrate the graph and dynamics of a network. It will be used again in the sequel.

Example 1.1: The graph of a Boolean network is depicted in Fig. 1.

Its dynamic model is assumed to be

$$
\begin{align*}
A(t+1) &= A(t) \\
B(t+1) &= A(t) \to C(t) \\
C(t+1) &= B(t) \lor D(t) \\
D(t+1) &= \neg B(t) \\
E(t+1) &= \neg C(t).
\end{align*}
$$

(2)

If the number of nodes in a Boolean network is $n$, then it is obvious that the state space considers of $2^n$ states, which is a finite set described as $D^n$. So as a dynamic process on $D^n$, there must be at least a fixed point or a cycle, and eventually a trajectory starting from any initial state must enter a cycle (a fixed point can be considered as a cycle of length one). So a cycle is also called an attractor. For convenience, we briefly denote by $\Omega$ the set of attractors. For a state $x_0$, the smallest number of steps to enter $\Omega$ is called its transient period, denoted by $T_t(x_0)$. That is, let $x(t, x_0)$ be the trajectory starting from $x_0$ (i.e., $x(0, x_0) = x_0$). Then

$$
T_t(x_0) = \min \{ k \mid x(k, x_0) \in \Omega \}.
$$

For overall network, the transient period is defined as

$$
T_t = \max_{x \in D^n} T_t(x).
$$

One of the most important issues in investigating a Boolean network is to find its cycles and transient period. These topics have been studied widely, e.g., in [6] and [10]. But there was no reported technique, which solves the problem systematically so far.

To investigate the structure of a Boolean network, Cheng and Qi [5] proposed a way to convert system (1) into a standard discrete-time dynamic system. The key tool for this approach is the matrix expression of logic, based on semitensor product of matrices. We give a very brief introduction here and refer to

Manuscript received May 11, 2008; revised November 04, 2008; accepted December 02, 2008. Current version published February 27, 2009. This work was supported in part by the National Natural Science Foundation (NNSF) of China under Grants 60674022, 60736022, and 60221301.

The author is with the Institute of Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: dcheng@iss.ac.cn).

Digital Object Identifier 10.1109/TNN.2008.2011359
[3] and [4] for details. Investigating the structure of a network via its dynamics can also be found in [8], [13], and [16].

Definition 1.2:
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 $X^1, \ldots, X^p$, which are $1 \times n$ rows. Define the semitensor product (STP), denoted by $\ltimes$, as

$$
\begin{align*}
X \ltimes Y &= \sum_{i=1}^{p} X^i y_i \in \mathbb{R}^n \\
Y^T \ltimes X^T &= \sum_{i=1}^{p} y_i (X^i)^T \in \mathbb{R}^n.
\end{align*}
$$

2) Let $A \in M_{m \times n}$ and $B \in M_{p \times q}$. If either $n$ is a factor of $p$, say $n = pt$, denoted as $A \ltimes_t B$, or $p$ is a factor of $n$, say $n = pt$, denoted as $A \gtimes_t B$, 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

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

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

STP of matrices is a generalization of the conventional matrix product, so the notation $\ltimes$ can be omitted. Moreover, all the basic properties of the conventional matrix product remain true for this extension.

Set

$$
\Delta_k := \{ \delta_k^i | i = 1,2, \ldots, k \}
$$

where $\delta_k^i$ is the $i$th column of identity matrix $I_k$. In the framework of STP, the logical variables are expressed as a vector in $\Delta_2$ by identifying $T$ with $[1, 0]^T$ and $F$ with $[0, 1]^T$. Then the region $D$ is replaced by

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

In this way, for a logical operator, we can always find a matrix, called its structure matrix, and the action of a logical operator on logical variables becomes a (semiconductor) product of the structure matrix with its arguments’ vectors. For instance, consider conjunction “$\wedge$” [7], then its structure matrix is

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

Hence, (in vector form) we have

$$
P \wedge Q = M_c P Q, \quad P, Q \in D_v.
$$

In general, we have the following proposition.

Proposition 1.3: Let $f$ be a logical function (operator) of $X_1, X_2, \ldots, X_n$, and

$$
Y = f(X_1, X_2, \ldots, X_n).
$$

Then we can find a matrix, called the structure matrix of $f$ and denoted by $M_f$, such that

$$
Y = M_f X_1 X_2 \cdots X_n.
$$

Note that $X_i$ may appear in (4) many times, but in (5), $Y$ is multilinear with respect to $X_1, \ldots, X_n$.

Using this vector expression, Cheng and Qi [5] convert a logical equation of a Boolean network into a discrete time linear system, called its algebraic form as $x(t+1) = L x(t)$. Analyzing the structure of $L$, precise formulas have been obtained to reveal the structure of the network.

This paper considers the Boolean control networks, which have input-state structure. We first propose a framework for Boolean control networks, and the structure of attractors of the networks is investigated. The input-state approach is then applied to the analysis of the structure of general Boolean networks and a structure of nested compounded cycles is obtained. We call such a structure “rolling gears” structure and will discuss some interesting properties of this kind of structures. We guess it could be used to reveal the hidden order in lives.

This paper is organized as follows. Section II reviews the converting technique from logical dynamic equation to algebraic one. An example is used to depict it. In Section III, a framework of Boolean control networks is proposed and the structure of attractors of input-state type of networks is investigated. In Section IV, the input-state approach is implemented to analyze the structure of general Boolean networks and the structure of nested compounded cycles, called “rolling gears,” is revealed. Section V contains two illustrative examples. Section VI is a brief conclusion.

II. FROM LOGICAL EQUATION TO ALGEBRAIC EQUATION

In this section, we briefly review the technique, developed in [5], which provides a systematic tool to treat Boolean networks. Assume a logical variable is expressed in a vector form. That is, $A_i(t) \in D_v \sim D$. (We will use scalar form $D$ and vector form $D_v$ alternatively without explanation, and use $D$ for both. From the text, it is very easy to figure out what form is used there.) Consider system (1). Since $f_i, i = 1, 2, \ldots, n$, are logical functions, according to Proposition 1.3, we can convert (1) into an algebraic dynamic form as

$$
\begin{align*}
A_1(t+1) &= M_1 A_1(t) A_2(t) \cdots A_n(t) \\
A_2(t+1) &= M_2 A_1(t) A_2(t) \cdots A_n(t) \\
& \vdots \\
A_n(t+1) &= M_n A_1(t) A_2(t) \cdots A_n(t)
\end{align*}
$$

where $M_i, i = 1, \ldots, n$, are the structure matrices of $f_i$. Define

$$
x(t) = A_1(t) A_2(t) \cdots A_n(t) \in \Delta_{2^n}.
$$

Multiplying $A_i(t+1)$ together yields

$$
x(t+1) = \prod_{i=1}^{n} [M_i A_1(t) A_2(t) \cdots A_n(t)].
$$

Using the properties of STP of matrices, (8) can be converted into an algebraic form as

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

where $L$ is called the network transition matrix of (1).

The following result reveals all the attractors from $L$. 

Theorem 2.1: Consider system (1) with its network transition matrix $L$.

1) The number of length $s$ cycles $N_s$ is inductively determined by

$$
\begin{align*}
N_1 &= \text{Trace}(L) \\
N_s &= \frac{\text{Trace}(L^s)}{\text{Trace}(L)} - \sum_{k \in \mathcal{P}(s)} kN_k, \\
2 \leq s \leq 2^m
\end{align*}
$$

(10)

where $\mathcal{P}(s)$ is the set of proper factors of $s$.

2) The elements on cycles of length $s$, denoted by $C_s$, is

$$
C_s = D_a(L^s) \setminus \cup_{k \in \mathcal{P}(s)} D_a(L^k)
$$

(11)

where $D_a(L)$ is the set of diagonal nonzero columns of $L$. Note that $a \in \mathcal{P}(k)$, iff $a \in \mathbb{Z}_k$, $a < k$, and $k/a \in \mathbb{Z}_k$. For instance, $\mathcal{P}(8) = \{1, 2, 4\}$, $\mathcal{P}(12) = \{1, 2, 3, 4, 6\}$, etc.

Denote

$$
r_0 = \min \left\{ k \left| L^k \in \{ L^{k+1}, L^{k+2}, \ldots, L^{2^m} \} \right. \right\}.
$$

Then, we have the following theorem.

Theorem 2.2: For system (1), the transient period

$$
T_t = r_0 \min \left\{ k \left| L^k \in \{ L^{k+1}, L^{k+2}, \ldots, L^{2^m} \} \right. \right\}.
$$

(12)

The following theorem provides an easy way to construct the regions of attraction.

Theorem 2.3: Given an $\eta \in \Delta_{2^m}$, denote the columns of $L$, which equal to $\eta$, by $L_{ij} = \eta$, $j = 1, 2, \ldots, k$. Then the set of parent points of $\eta$ is

$$
L^{-1}(\eta) = \{ \delta_{i1}^{(2)}, \delta_{i2}^{(2)}, \ldots, \delta_{ik}^{(2)} \}.
$$

(13)

We use an example to illustrate these results.

Example 2.4: Consider Example 1.1. Equation (2) can be converted into algebraic form as

$$
\begin{align*}
A(t+1) &= A(t) \\
B(t+1) &= M_iA(t)C(t) \\
C(t+1) &= M_iB(t)D(t) \\
D(t+1) &= M_iB(t) \\
E(t+1) &= M_iC(t).
\end{align*}
$$

(14)

Let $x(t) := A(t)B(t)C(t)D(t)E(t)$. Then

$$
x(t+1) = A(t)M_iA(t)C(t)M_iB(t)D(t)M_iB(t)M_iC(t) = \cdots
$$

(15)

Starting from the second row of (15), the front constant coefficient matrix in the previous row is replaced in the next row by “...” to save space.

Note that now in the left-hand side of (15) there is no $E(t)$. To get $x(t)$ we have to add it. We can use a dummy operator [5]

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

which satisfies

$$
E_dPQ = Q \quad \forall P, Q \in \Delta_2.
$$

Hence, (15) can be converted further as

$$
x(t+1) = \cdots (I_4 \otimes M_i)A(t)B(t)C(t)E_dW_{[2]}D(t)E(t)
$$

$$
= \cdots (I_8 \otimes (E_dW_{[2]}))x(t).
$$

(16)

From (16), we have

$$
L = (I_2 \otimes M_i)(I_4 \otimes M_i)(I_4 \otimes M_i)(I_8 \otimes W_{[2]})
$$

$$
(14 \otimes M_i)(I_8 \otimes (E_dW_{[2]})) (I_4 \otimes M_i)(I_8 \otimes (E_dW_{[2]})).
$$

(17)

Note that $L \in M_{32 \times 32}$. It is easy to calculate $L$. We express it into a condensed form as

$$
L = \delta_{32}^{[4, 4, 4, 4, 4, 1, 1, 1, 1, 1, 2, 2, 2, 6, 6, 9, 9, 9, 13, 13, 13, 13, 20, 20, 20, 20, 19, 19, 19, 18, 18, 22, 22, 17, 17, 21, 21, 11100 \quad \text{and} \quad 01100
$$

and a cycle of length 2, which is

$$
11010 \rightarrow 10101 \rightarrow 11010.
$$

We can also check that $r_0 = 4$ and $L^4 = L^5$, so the transient period is $T_t = 4$.

III. BOOLEAN CONTROL NETWORKS

A Boolean control network is defined as

$$
\begin{align}
A_1(t+1) &= f_1(A_1(t), A_2(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)) \\
A_2(t+1) &= f_2(A_1(t), A_2(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t)) \\
& \vdots \\
A_n(t+1) &= f_n(A_1(t), A_2(t), \ldots, A_n(t), u_1(t), \ldots, u_m(t))
\end{align}
$$

(18)

where $u_i, i = 1, 2, \ldots, m$, are inputs (or controls), which are logical variables satisfying certain logical rule, called the input network, described as

$$
\begin{align}
u_1(t+1) &= g_1(u_1(t), u_2(t), \ldots, u_m(t)) \\
u_2(t+1) &= g_2(u_1(t), u_2(t), \ldots, u_m(t)) \\
& \vdots \\
u_m(t+1) &= g_m(u_1(t), u_2(t), \ldots, u_m(t)).
\end{align}
$$

(19)
In an algebraic form, a Boolean control network can be expressed as

\[
\begin{align*}
\begin{cases}
u(t + 1) &= G\nu(t), & \nu \in D^m \\
x(t + 1) &= L(\nu)x(t), & x \in D^n
\end{cases}
\end{align*}
\]

(20)

where \(L(\nu) = L\nu(t)\) is the control-depending network transition matrix.

**Example 3.1:** Consider Example 1.1 again. It is very natural to take \(A(t)\) as input. Ignoring \(E(t)\), which is considered as an output, the system can be rewritten as

\[
\begin{align*}
B(t + 1) &= u(t) \rightarrow C(t) \\
C(t + 1) &= B(t) \lor D(t) \\
D(t + 1) &= -B(t)
\end{align*}
\]

(21)

and the control network is

\[
u(t + 1) := A(t + 1) = A(t).
\]

Converting this system into an algebraic form, we have

\[
\begin{align*}
\begin{cases}
u(t + 1) &= u(t) \\
x(t + 1) &= L(\nu)x(t)
\end{cases}
\end{align*}
\]

(22)

where \(L(\nu)\) can be easily calculated as

\[
L(\nu) = M_0 u(I_2 \otimes M_0 d)(I_3 \otimes M_0 n)W_2[I_2 \otimes I_2 \otimes I_2]M_r
\]

and \(u = (\alpha, 1-\alpha)^T\), where \(\alpha = 0 \text{ or } 1\).

Now both \(\delta^2_2\) and \(\delta^2_3\) are fixed points of the control network.

Using Theorem 2.1, it is easy to figure out that for \(u = \delta^2_2\), there is a fixed point for the system, which is \(x = (01000000) \approx 110\), and there is also a cycle of length 2, which is 101 \(\rightarrow 010 \rightarrow 101\).

While \(u = \delta^2_3\), there is only a fixed point 110.

In general, we consider the structure of the Boolean control system (18), where the controls are varying, according to its own logical evolution rule (19).

Denote by \(U = D^m\) the input space, by \(X = D^n\) the state space, and let \(W = U \times X\) be the input-state (product) space. Let \(w \in W\). It is easy to prove that there exist unique \(u \in U\) and \(x \in X\), such that \(w = ux\). Now assume there is a cycle of length \(k\) in the input-state space \(W\). Say it is

\[C^k_w : u(0) = u_0 = u_0 x_0 \rightarrow w(1) = u_1 = u_1 x_1 \rightarrow \cdots \rightarrow w(k) = u_k = u_k x_k = w_0.\]

First, one sees easily that since \(u_0 = u_k\), \([u_0, u_1, \ldots, u_k]\) contains, say, \(j\) folds of a cycle of length \(\ell\), say, \(j \ell = k\). Hence \(w_0 = u_0\). Now let us see what condition the \(\{x_i\}\) in the cycle \(C^k_w\) should satisfy. Define a network transition matrix as

\[\Psi : = L(u_{\ell-1})L(u_{\ell-2}) \cdots L(u_1)L(u_0).\]

(23)

Starting from \(u_0 = u_0 x_0\), we have \(x\) component of the cycle \(C^k_w\) as

\[
\begin{align*}
x_0 &\rightarrow x_1 = L(u_0)x_0 \rightarrow x_2 = L(u_1)L(u_0)x_0 \\
&\rightarrow \cdots \rightarrow x_{\ell} = \Psi x_0 \rightarrow x_{\ell+1} = L(u_0)\Psi x_0 \\
&\rightarrow x_{\ell+2} = L(u_1)L(u_0)\Psi x_0 \rightarrow \cdots \rightarrow x_{2\ell} = \Psi^2 x_0 \rightarrow \\
&\vdots \\
&\rightarrow x_{(j-1)\ell + 1} = L(u_0)\Psi^{j-1}x_0 \rightarrow x_{(j-1)\ell + 2} = L(u_1)L(u_0)\Psi^{j-1}x_0 \\
&\rightarrow \cdots \rightarrow x_{j\ell} = \Psi^j x_0 = x_0.
\end{align*}
\]

(24)

We conclude that \(x_0 \in D^n\) is a fixed point of the equation (with \(j > 0\) being the smallest one)

\[x(t + 1) = \Psi^j x(t).\]

(25)

Conversely, if \(x_0 \in D^n\) is a fixed point of (25) and \(u_0\) is a point on a cycle of control space \(C^\ell_w\), then it is obvious that we have the cycle (24).

Summarizing above arguments yields the following theorem.

**Theorem 3.2:** Consider the Boolean control network (20). A set \(C^k_w \subset D^{k(n+m)}\) is a cycle of the control system with length \(k\), if for any point \(u_0 = u_0 x_0 \in C^k_w\), there exists an \(\ell \leq k\) as a factor of \(k\), such that \(u_0, u_1 = G(u_0), u_2 = G^2(u_0), \ldots, u_\ell = G^\ell = \Psi = u_0\) is a cycle in the control space, and \(x_0\) is a fixed point of (25) (with \(j > 0\) being the smallest one).

Theorem 3.2 shows how to find all the cycles in the input-state space. First, we can find the cycles in the input space. Pick a cycle in the input space, say \(C^\ell_w\), then for each point \(u_0 \in C^\ell_w\), we can construct an auxiliary system

\[x(t + 1) = \Psi^j x(t).\]

(26)

Now, say \(C^\ell_w = (u_0, u_1, \ldots, u_\ell = u_0)\) is a cycle in \(U\), and \(C^\ell_x = (x_0, x_1, \ldots, x_j = x_0)\) is a cycle of (26). Then, a cycle \(C^k_w, k = j \ell\), can be constructed by

\[
\begin{align*}
u_0 &= u_0 x_0 \rightarrow w_1 = u_1 L(u_0)x_0 \rightarrow w_2 = u_2 L(u_1)L(u_0)x_0 \\
&\rightarrow \cdots \rightarrow w_\ell = u_\ell x_\ell \rightarrow w_{\ell+1} = u_1 L(u_0)x_1 \\
&\rightarrow w_{\ell+2} = u_2 L(u_1)L(u_0)x_1 \rightarrow \cdots \\
&\vdots \\
&\rightarrow w_{(j-1)\ell} = u_{(j-1)\ell} x_{(j-1)} \rightarrow w_{(j-1)\ell+1} = u_1 L(u_0)x_{(j-1)} \\
&\rightarrow w_{(j-1)\ell+2} = u_2 L(u_1)L(u_0)x_{(j-1)} \rightarrow \cdots \\
&\rightarrow w_{j\ell} = u_0 x_j = u_0 x_0 = w_0.
\end{align*}
\]

(27)

We call this \(C^k_w\) the compounded cycle of \(C^\ell_w\) and \(C^\ell_x\), denoted by \(C^k_w = C^\ell_w \circ C^\ell_x\).

Note that from a cycle in the input space \(C^\ell_w\), we can choose any point as the starting point \(u_0\). Then, in (26), we have different \(\Psi\), which produces different \(C^\ell_x\). It is reasonable to guess that the final \(C^k_w = C^\ell_w \circ C^\ell_x\) is independent of the choice of \(u_0\). Otherwise, the picture will be incorrect. In the following, we will prove this is true.

**Definition 3.3:** Let \(C^k_w = \{u(t) | t = 0, 1, \ldots, k\}\) be a cycle in the input-state space, and \(C^\ell_x\) be a cycle in the input space.
Splitting \( u(t) = u(t)x(t) \), we said that \( C^k_w \) is attached to \( C^k_u \) at \( u_0 \), if \( u(0) = u_0 x_0 \), and

1) \( u(t) \in C^k_u \) with \( u(0) = u_0 \);
2) \( x(0) = x_0 \) is a fixed point of (25) with \( j = k/\ell \in \mathbb{Z}_+ \).

**Remark 3.4:** According to Theorem 3.2, each cycle \( C^k_u \) in the input-state space must be attached precisely to one cycle in the input space. In fact, the following argument shows that \( C^k_u \) attaches \( C^k_u \) at \( u_0 \) at moment \( t = 0 \) (and the attaching point of \( C^k_u \) is \( u_0 = u_0 x_0 \)) and will attach it at \( u_1 \) at moment \( t = 1 \) (with the attaching point of \( C^k_u \) being \( u_1 = u_1 x_1 \)) and so on. So \( C^k_w \) and \( C^k_u \) are moving as two assembled gears.

**Proposition 3.5:** The sets of the cycles in the input-state space, attached to any point of a given cycle \( C^k_u \), are the same.

**Proof:** Let \( C^k_u = \{u_0, u_1, \ldots, u_\ell = u_0\} \) be the cycle we are concerned with. Let \( S_0, S_1, \ldots, S_{\ell - 1} \) be the set of cycles attached to \( u_0, u_1, \ldots, u_{\ell - 1} \), respectively. First, we show that

\[
S_0 \subset S_i, \quad i = 1, 2, \ldots, \ell - 1.
\]

Let \( C^k_u \) be \( \{u_0, u_1, \ldots, u_\ell\} \in S_0 \), i.e., it is a cycle attached to \( C^k_u \) at \( u_0 \). Using the elements of a control cycle, we can define

\[
L_i := L(u_i), \quad i = 0, 1, \ldots, \ell - 1.
\]

Then, we construct \( \ell \) system matrices as

\[
\begin{align*}
\Psi_0 &:= L_{\ell - L_{\ell - 2} \cdots L_0}, \\
\Psi_1 &:= L_0 L_{\ell - L_{\ell - 2} \cdots L_1}, \\
\vdots &:= \vdots, \\
\Psi_{\ell - 1} &:= L_{\ell - 2} L_{\ell - 3} \cdots L_0 L_{\ell - 1}.
\end{align*}
\]

Correspondingly, we then construct \( \ell \) auxiliary systems as

\[
x(t + 1) = \Psi_i x(t), \quad i = 0, 1, \ldots, \ell - 1.
\]

(28)

Since \( u_0 = u_0 x_0 \in C^k_w \subset S_0 \), then \( x_0 \) satisfies

\[
(\Psi_0)^j x_0 = x_0.
\]

Note that \( u(1) := u_1 = u_0 L_0 x_0 \). To see that \( u_1 \in C^k_w \in S_1 \), we have to show that \( L_0 x_0 \) satisfies

\[
(\Psi_1)^j L_0 x_0 = L_0 x_0.
\]

(30)

This is true because

\[
\begin{align*}
L_0 x_0 &= L_0(\Psi_0)^j x_0 = L_0(L_{\ell - 1} \cdots L_0)^j x_0 \\
&= L_0 \left( \frac{L_{\ell - 1} \cdots L_0}{L_0} \right)^j x_0 \\
&= \left( \frac{L_0 L_{\ell - 1} \cdots L_0}{L_0} \right)^j L_0 x_0 \\
&= \left( L_0 L_{\ell - 1} \cdots L_0 \right)^j L_0 x_0 = L_0 x_0.
\end{align*}
\]

Similarly, we can show that

\[
u_0 \ell_{\ell - 1} \cdots L_0 x_0 \in C^k_w \subset S_s, \quad s = 1, 2, \ldots, \ell - 1.
\]

Note that, precisely speaking, (30) can only assure that there is a cycle of length \( \ell \times j \) attached to the cycle at \( u_1 \), where \( j \) is a factor of \( j \). But since the above definition of \( \{\Psi_i\} \) is on a rotating style, starting from a point \( u_0 = u_1 x_0 \), the same argument shows \( j \leq j' \). So \( j = j' \) and

\[
\max \{\nu, \ell, (\nu - 1)\} \leq T_i \leq \max \{\nu, \ell, (\nu')\}
\]

(32)

where \( i = 1, \ldots, p \).
Define
\[ V_i := \max \{ r_0, \ell_i(r^i - 1) \} \]
\[ U_i := \max \{ r_0, \ell_i(r^{2i}) \}, \quad i = 1, \ldots, p. \]

Then, the following is obvious.

**Proposition 3.7:** The transient period of the control network satisfies
\[ \max_{1 \leq i \leq p} \{ V_i \} < T_i \leq \max_{1 \leq i \leq p} \{ U_i \}. \] (33)

**IV. CASCADeD BOOLEAN NETWORKS**

The input-state structure proposed in the previous section is very useful in analyzing the structure of Boolean networks with the cascading structure.

**Definition 4.1:** Consider system (1), where \( x \in X = D^n \).

A subspace \( V = D^k \subset X \) is called an invariant subspace, if \( x_0 \in V \) implies \( x(t, x_0) \in V, \forall t > 0 \).

From Section III, one sees easily that the control space is an invariant subspace of the control-state (product) space. Conversely, an invariant subspace can also be considered as a control space.

To testify if a subspace is a control invariant subspace, we can use either network graph or network equation. We use the following two examples to illustrate this.

Let \( \{ A_{i_1}, \ldots, A_{i_s} \} \) be a subset of the nodes of a network.

\( V = \text{Span}\{ A_{i_1}, \ldots, A_{i_s} \} \) means that \( V \subset X \) is the subspace of the states of the subnetwork with nodes \( \{ A_{i_1}, \ldots, A_{i_s} \} \) and edges between them, inherited from original graph.

**Example 4.2:** Consider Fig. 3. One sees easily that \( V_1 = \text{Span}\{ A \} \) and \( V_2 = \text{Span}\{ A, B, C, D \} \) are two invariant subspaces. We have the nested invariant subspaces as
\[ V_1 \subset V_2 \subset X. \]

Note that \( V = \text{Span}\{ A, B, C \} \) is not an invariant subspace, because it will be affected by \( D \). (If you are familiar with graph theory, it is easy to see that a subspace is invariant iff the subgraph has in-degree zero.)

The structure of nested invariant subspaces can also be discovered from network equations. Consider the following example.

\[
\begin{align*}
A_1(t+1) &= f^1_1(A_1(t), \ldots, A_6(t)) \\
& \vdots \\
A_6(t+1) &= f^1_6(A_1(t), \ldots, A_6(t)) \\
B_1(t+1) &= f^2_1(A_1(t), A_6(t), B_1(t), \ldots, B_5(t)) \\
& \vdots \\
B_5(t+1) &= f^2_5(A_1(t), A_6(t), B_1(t), \ldots, B_5(t)) \\
C_1(t+1) &= f^3_1(A_1(t), A_6(t), B_1(t), \ldots, B_5(t), C_1(t), \ldots, C_6(t)) \\
& \vdots \\
C_6(t+1) &= f^3_6(A_1(t), A_6(t), B_1(t), \ldots, B_5(t), C_1(t), \ldots, C_6(t)).
\end{align*}
\] (34)
cycles), i.e., the perimeter of $U_1^3$ is a multiplier of the perimeter of $U_2^2$, and the perimeter of $U_4^3$ is a multiplier of the perimeter of $U_1^1$.

- In each chair, the smaller gears affect the larger gears, and the larger gears do not affect the smaller gears.
- Smallest gears look like steering gears, which steer the other gears to run.

Kauffman claimed that [12] in a cellular network the tiny attractors decide the vast order. The “rolling gears” structure explained why small cycles decide the order of the whole network. We guess that the structure of “rolling gears” may be used to explain the “hidden order” in lives.

Finally, one may ask: Why should there be an invariant subspace? In fact, if a large or huge network has small cycles, then the small cycles with the elements in their region of attraction form small invariant subspaces. If there are no such small cycles, the system is in chaos [12]. So an ordered large scale network should have the structure of nested invariant subspaces.

V. TWO ILLUSTRATIVE EXAMPLES

First example is from [14]. It is used for two purposes: 1) showing the standard algorithm; and 2) demonstrating that the “small cycles” have decisive importance for the structure of the overall network.

**Example 5.1**: Consider a system with five nodes, as

$$A_i = f_i(A_{j1}, A_{j2}, A_{j3}), \quad i = 1, 2, 3, 4, 5 \quad (35)$$

where the logical functions $f_i$, $i = 1, \ldots, 5$, are determined by the truth table (Table I).

<table>
<thead>
<tr>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>$f_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$j_1$</th>
<th>$j_2$</th>
<th>$j_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

To get the structure matrix, note that the first row of the structure matrix of $f_i$ is exactly the same as its values in truth table.

To convert the matrix form back to logical form, mod(2) algebra is more convenient. Using mod(2) algebra, system (35) can be expressed as

$$A(t + 1) = B(t) + D(t)$$
$$B(t + 1) = D(t) + C(t) X_2 E(t)$$
$$C(t + 1) = A(t) + D(t) X_2 E(t)$$
$$D(t + 1) = C(t) X_2 E(t)$$
$$E(t + 1) = A(t) X_2 D(t) X_2 E(t)$$

(37)

It is easy to see that the structure matrix of $\text{mod}(2)$ times $X_2$ is $M_e$ and $\text{mod}(2)$ plus “+2” is

$$M_+ = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Let $x(t) = A(t) B(t) C(t) D(t) E(t)$. Then

$$x(t + 1) = M_e^2 BDM_e BDM_e^2 DCE M_e^3 ACEDM_e^2 ACEDM_e^2 ADE.$$ 

Now there is a normal routine to figure out $L$. In fact

$L = \delta_{22}[1, 6, 4, 16, 13, 2, 8, 12, 1, 6, 20, 32, 13, 2, 24, 28, 2, 2, 4, 12, 10, 6, 4, 16, 2, 2, 20, 28, 10, 6, 20, 32].$

Then, one can check that the nontrivial powers are 1 and 2, and

$\text{Trace}(L) = 4 \quad \text{Trace}(L^2) = 6.$

We conclude that there are four fixed points and one cycle of length 2. Using Theorem 2.1, one sees easily that the fixed points are

$$E_1 = 11111 \quad E_2 = 10011 \quad E_3 = 00100 \quad E_4 = 00000$$

and the cycle of length 2 is

11110 $\rightarrow$ 11010 $\rightarrow$ 11110.

The smallest repeating $L^k$ is $L^3 = L^5$, so the transient period $T_1 = 3$.

Finally, we use Theorem 2.3 to get the whole picture of the state space.

- Starting from $E_1 = 11111$, we calculate its parent states, its grand parent states, and so on. We have (in the following, $[x]$ is used to show that $x$ is already on the cycle, so we remove it from the retrieving chain)

$$E_1 = 11111 \sim \delta^3 \Rightarrow [L_1 \rightarrow \delta^3]$$

$$L_9 \sim \delta^9 \sim 10111 \Rightarrow \emptyset.$$

$$E_2 = 10011 \sim \delta^{33} \Rightarrow [L_2 \rightarrow \delta^{13}]$$

$$L_5 \sim \delta^{15} \sim 11011 \Rightarrow \emptyset.$$

- $E_3 = 00100 \sim \delta^{28} \Rightarrow [L_28 \rightarrow \delta^{28}]$

$L_{16} \sim \delta^{16} \sim 10000$

$$E_4 \sim \delta^4 \sim 11100 \Rightarrow \emptyset.$$
Part 1. \( D = 0 \)

\[
E_4 = 00000 \sim \delta^{32} \Rightarrow \left[ L_{32} \sim \delta^{32} \right] \\
L_{22} \sim \delta^{12} \sim 10100 \\
\Rightarrow \begin{cases} 
L_{20} \sim \delta^{20} \sim 01100 \Rightarrow \emptyset \\
L_{27} \sim \delta^{27} \sim 00101 \Rightarrow \emptyset \\
L_{31} \sim \delta^{31} \sim 00001 \Rightarrow \emptyset 
\end{cases} \\
L_8 \sim \delta^8 \sim 11000 \Rightarrow L_7 \sim \delta^7 \sim 11001 \Rightarrow \emptyset.
\]

- Next, we consider two points on a cycle: \( C_1 = 11010 \) and \( C_2 = 11110 \). For \( C_1 \)

\[
C_1 = 11010 \sim \delta^6 \\
\Rightarrow \begin{cases} 
L_2 \sim \delta^2 \sim 11110 \\
L_{10} \sim \delta^{10} \sim 10110 \Rightarrow \emptyset \\
L_{22} \sim \delta^{22} \sim 01010 \Rightarrow \emptyset \\
L_{31} \sim \delta^{31} \sim 00010 \Rightarrow \emptyset 
\end{cases}
\]

- \( C_2 = 11110 \sim \delta^2 \Rightarrow \begin{cases} 
L_6 \sim \delta^6 \sim 11010 \\
L_{14} \sim \delta^{14} \sim 11010 \Rightarrow \emptyset \\
L_{17} \sim \delta^{17} \sim 01111 \Rightarrow \emptyset \\
L_{18} \sim \delta^{18} \sim 01110 \Rightarrow \emptyset \\
L_{25} \sim \delta^{25} \sim 00111 \Rightarrow \emptyset \\
L_{26} \sim \delta^{26} \sim 00110 \Rightarrow \emptyset 
\end{cases}
\]

The following state transition diagram from [14] verifies our conclusion.

What is significant in this example is the following observation. There is a smallest “cycle”: fixed point \( D \). From Fig. 5, one sees easily that for \( D = 0 \) and \( D = 1 \) the topological structures of the state-space graphs are completely different.

Next, we analyze a system, which is used to simulate gene and protein signaling activity patterns [9].

Example 5.2: The network depicted in Fig. 6 and Table II is presented in [9] to simulate gene and protein signaling activity patterns within a small model Boolean network. For notational brevity, we use \( A \) for “Erk,” \( B \) for “cyclin D1,” \( C \) for “p27,” \( D \) for “cyclin E,” \( E \) for “E2F,” \( F \) for “pRb,” \( G \) for “S phase genes,” \( U \) for “growth factors,” \( V \) for cell “shape(spreading),” and \( W \) for “X.” We refer to [9] for the biological meanings of the notations.

Then, the logical equation is expressed as

\[
\begin{aligned}
A(t + 1) &= \neg(A(t) \rightarrow G(t)) \\
B(t + 1) &= \neg(A(t) \rightarrow C(t)) \\
C(t + 1) &= H(t) \rightarrow W(t) \\
D(t + 1) &= E(t) \rightarrow C(t) \\
E(t + 1) &= \neg(D(t) \rightarrow F(t)) \\
F(t + 1) &= \neg(B(t) \land D(t)) \\
G(t + 1) &= \neg(F(t) \rightarrow E(t)) \\
H(t + 1) &= \neg(H(t) \rightarrow G(t)).
\end{aligned}
\]

As for the control network, we have

\[
\begin{aligned}
U(t + 1) &= g_1(U(t)) \\
V(t + 1) &= g_2(V(t)) \\
W(t + 1) &= g_3(U(t), V(t)).
\end{aligned}
\]

In a matrix form, we have an algebraic equation as

\[
\begin{aligned}
A(t + 1) &= M_p M_i U(t) G(t) \\
B(t + 1) &= M_p M_i A(t) C(t) \\
C(t + 1) &= M_p H(t) W(t) \\
D(t + 1) &= M_i E(t) C(t) \\
E(t + 1) &= M_i M_d D(t) F(t) \\
F(t + 1) &= M_i M_b B(t) D(t) \\
G(t + 1) &= M_i M_f E(t) G(t) \\
H(t + 1) &= M_i M_h H(t) G(t).
\end{aligned}
\]

As in [9], we first set the control network as

\[
\begin{aligned}
U(t + 1) &= U(t) \\
V(t + 1) &= V(t) \\
W(t + 1) &= U(t) \land V(t).
\end{aligned}
\]

Case 1: \( U(0) = V(0) = [1, 0]^T, \sigma_1 = \sigma_2 = \text{identity}, \) i.e., \( U(t) \) and \( V(t) \) equal to \([1, 0]^T \) constantly.
Table II
LOGICAL RELATIONS

<table>
<thead>
<tr>
<th>network element</th>
<th>W</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>input 1</td>
<td>U</td>
<td>U</td>
<td>A</td>
<td>H</td>
<td>E</td>
<td>D</td>
<td>B</td>
<td>F</td>
<td>H</td>
</tr>
<tr>
<td>input 2</td>
<td>V</td>
<td>G</td>
<td>C</td>
<td>W</td>
<td>C</td>
<td>F</td>
<td>D</td>
<td>E</td>
<td>G</td>
</tr>
<tr>
<td>boolean function</td>
<td>and</td>
<td>not if</td>
<td>not if</td>
<td>implicate</td>
<td>implicate</td>
<td>not if</td>
<td>nand</td>
<td>not if</td>
<td>not if</td>
</tr>
</tbody>
</table>

Fig. 7. Chaired circles. (a) Circles in $U \subset I - S$ spaces. (b) Circles in $V \subset V \times W \subset I - S$ spaces.

Plugging them into (39) yields the system

$$L(U(t), W(t)) = L \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$  

In calculation, a control can be treated as a logical operator, so the routine for calculating the network transition matrix remains available. Then, it is easy to get the following result:

- the only attractor is a fixed point $00110110$;
- $L^{10} = L^{11}$ and the transient period is $T_1 = 10$.

**Case 2:** $U(0) = [1, 0]^T$ and $V(0) = [0, 1]$. In this case, we have the same conclusion.

**Case 3:** $U(0) = [0, 1]^T$, then we always have $W(t) = [0, 1]^T$, $t \geq 1$. The conclusion is as follows:

- the only attractor is a fixed point $00110110$;
- $I^{10} = I^{11}$ and the transient period is $T_1 = 6$. (Taking $W(0)$ into consideration, $T_1$ should be 7.)

Next, we assume that the control network is

$$\begin{cases} 
U(t + 1) = -U(t) \\
V(t + 1) = -V(t) \\
W(t + 1) = V(t) \land W(t). 
\end{cases}$$  

Then, we have two sequences of nested invariant subspaces. We consider them separately. Consider the first chair, which is

$$V_1 = \text{Span}\{U\} \subset V_2 = \text{Span}\{U, V\}.$$  

In $V_1$, we have an obvious cycle: $0 \rightarrow 1 \rightarrow 0$. For $U = 0$, a routine computation shows that there is only a cycle of length 2, which is

$$00110110 \rightarrow 001101100 \rightarrow 0011011010.$$  

$L(0)$ is a $1024 \times 1024$ matrix. We omit it here. But we can calculate that $L(0)^{10} = L(0)^{11}$ and $T_1 = 7$. Similarly, for $U = 1$, we have the same cycle. $L(1)^{11} = L(1)^{13}$ and $T_1 = 11$.

Finally, let $\Psi = L(1)L(0)$. Then, $\Psi$ has only one fixed point $0011011010$. We conclude that overall in $U$ space we have only one cycle $0 \rightarrow 1 \rightarrow 0$; and in the whole space, we have only one product cycle

$$0 \times 0011011010 \rightarrow 1 \times 0011011000 \rightarrow 0 \times 0011011010.$$  

They are depicted in Fig. 7(a), where $I - S$ is the overall input-state space.

Next, we consider the second chair, which is

$$V_1 = \text{Span}\{V\} \subset V_2 = \text{Span}\{V, W\} \subset V_3 = \text{Span}\{A, B, C, D, E, F, G, H, U, V, W\}.$$  

First, there is a trivial cycle in $V$ space as: $0 \rightarrow 1 \rightarrow 0$. Then, in $V \times W$ space, it is easy to calculate that

$$L(0) = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad L(1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$
So

\[ \Psi = L(1)L(0) = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \]

which has a unique fixed point \((0, 1) \sim 0\). We conclude that in \(V \times W\) space we have only one cycle: \(0 \times 0 \rightarrow 1 \times 0 \rightarrow 0 \times 0\). Finally, we consider the space \(V \times W \times ABCDEFGH\). Calculating \(\Psi = L(0 \times 0)L(1 \times 0)\), it is easy to find that the only cycle is a fixed point: \(001101101\). We conclude that there is only one cycle of length 2 in the overall product space, which is \(0 \times 0 \times 001101101 \rightarrow 1 \times 0 \times 001101100\). Circles in nested subspaces are depicted in Fig. 7(b).

VI. CONCLUSION

In this paper, we first reviewed the main results of [5]: how to convert a logical form of Boolean equation into an algebraic form. Then, we proposed a framework for control Boolean networks. The structure of the cycles in the input-state space was obtained. Using the input-state technique, we investigated the general structure of Boolean networks. Then, as the network has a cascade structure, a structure of chained cycles, called rolling gears, has been revealed. It was shown that the rolling gears structure is the explanation that tiny attractors decide the order of networks.\(^1\)

REFERENCES


Daizhan Cheng (M’01–SM’01–F’06) received the Ph.D. degree in systems science from Washington University, St. Louis, in 1985. Currently, he is a Professor with Institute of Systems Science, Chinese Academy of Sciences, Beijing, China. His research interests include nonlinear systems, numerical method, systems biology, etc.

Dr. Cheng is Chairman of the Technical Committee on Control Theory (since 2003), Chinese Association of Automation, and the International Federation of Automatic Control (IFAC) Fellow.