International Journal of Bifurcation and Chaos, Vol. 20, No. 3 (2010) 561–582

DOI: 10.1142/S0218127410025892

ALGEBRAIC APPROACH TO DYNAMICS OF MULTIVALUED NETWORKS

ZHIQIANG LI* and DAIZHAN CHENG†
Key Laboratory of Systems and Control, AMSS,
Chinese Academy of Sciences, Beijing 100190, P. R. China
**lizhiqiang@amss.ac.cn*
†*dcheng@iss.ac.cn*

Received October 14, 2008; Revised July 20, 2009

Using semi-tensor product of matrices, a matrix expression for multivalued logic is proposed, where a logical variable is expressed as a vector, and a logical function is expressed as a multilinear mapping. Under this framework, the dynamics of a multivalued logical network is converted into a standard discrete-time linear system. Analyzing the network transition matrix, easily computable formulas are obtained to show (a) the number of equilibriums; (b) the numbers of cycles of different lengths; (c) transient period, the minimum time for all points to enter the set of attractors, respectively. A method to reconstruct the logical network from its network transition matrix is also presented. This approach can also be used to convert the dynamics of a multivalued control network into a discrete-time bilinear system. Then, the structure and the controllability of multivalued logical control networks are revealed.

Keywords: Logical network; semi-tensor product; attractor; transient period; controllability.

1. Introduction

When the genetic circuits are described as Boolean networks, the gene state is quantized to only two levels: True and False. This approach was first proposed by [Kauffman, 1969] and then studied by many authors, e.g. [Huang & Ingber, 2000; Huang, 2002; Ideker *et al.*, 2001].

A Boolean network is a set of nodes $A_1, A_2, \ldots, A_n$, which interact with each other in a synchronous manner [Farrow *et al.*, 2004]. At each given time $t = 0, 1, 2, \ldots$, a node can take only one of two different values: 1 or 0. In a Boolean network with n nodes, since there are only finite possible states, starting from any initial state, the Boolean network will evolve into either a fixed point or a cycle. Both fixed points and cycles are called attractors. The attractors of a Boolean network have two different interpretations [Zhang *et al.*, 2007]: one is that the attractors should correspond to cell types [Kauffman, 1995] and the other is that they should correspond to cell states of growth, differentiation and apoptosis [Huang, 1999]. The attractors form the limit sets of a Boolean network. So it is important to find the attractors in the system. The problem of finding attractors is widely studied, for instance, [Cheng, 2009; Farrow *et al.*, 2004; Heidel *et al.*, 2003; Zhang *et al.*, 2007].

A Boolean network is commonly described by a directed graph, called the network graph. The graph consists of a set of nodes, denoted by $\mathcal{N} = \{1, 2, \ldots, n\}$, which represents n nodes, and the set of edges, denoted by $E \subset \mathcal{N} \times \mathcal{N}$. If $(i, j) \in E$, there is a directed edge from node i to node j, which

*Address for Correspondence: No. 55, Zhongguancun East Road, Haidian District, Beijing 100190, P. R. China.

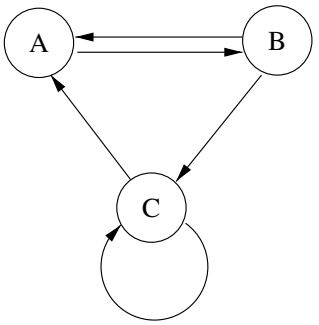

Fig. 1. Boolean network with three nodes.

means node j is affected by node i directly. The network graph shows only incidence relation. In fact, it is equivalent to incidence matrix [Robert, 1986]. To describe the dynamics of a network, we need a logical dynamic equation. We give a simple example to illustrate these concepts.

Example 1.1. Consider a Boolean network, depicted in Fig. 1. It consists of three nodes A, B, C.

Its dynamics is described by a set of discrete-time logical dynamic equations, for instance, as

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

From (1), it is easy to calculate that its state-transition diagram is depicted in Fig. 2.

From Fig. 2, one sees that the cycle $(0\ 1\ 1) \to (1\ 1\ 1) \to (1\ 0\ 1) \to (0\ 0\ 1) \to (0\ 1\ 1)$ is the only attractor.

The aforementioned Boolean network is a deterministic model. Random Boolean network means there are several models and the real dynamics takes one of them as the active model at a time with certain probability. We refer to [Aldana *et al.*, 2003; Shmulevich *et al.*, 2002a, 2002b] for detailed description.

For some complex physical and biological phenomena, we give some abstractions and idealizations in modeling. [Harvey & Bossomaier, 1997] mentioned that "... *In fitting such complex systems into a framework such as RBNs, simplifications and abstractions must be made; two of these are the Boolean idealization and the synchrony idealization*". [Huberman & Glace, 1993] pointed out that synchronous updating and asynchronous updating can lead to different conclusions in the simulation of Prisoner's Dilemma interactions. The result leads to a debate on whether the synchronous updating was reasonable for the RBNs model to study gene regulatory network [May *et al.*, 1995]. Gershenson studied the properties of asynchronous random Boolean networks in [Gershenson, 2002, 2003, 2004]. It was pointed by [Gershenson, 2004] that synchronous RBNs are justifiable theoretical models of biological networks.

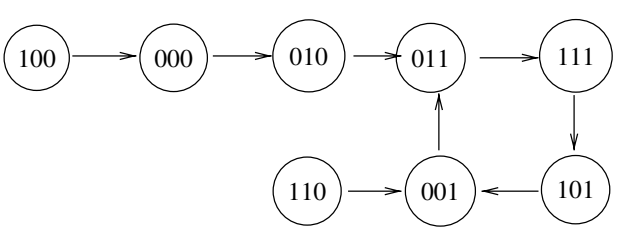

Fig. 2. State-transition diagram.

This paper focuses on deterministic logic dynamics. For the Boolean idealization, it is obvious that the binary approach is an approximation. Even if two-valued status is precisely correct, since the heredity is group-wise, a group of genes bounded by chromosome may also be considered as a multivalue logical variable. Multivalued networks also appear in some other complex systems, for instance in chemical reactions [Adamatzky, 2003], cognitive sciences [Volker & Conrad, 1998], etc. When the gene state is not limited to true and false, such as the inference of one gene to the other one is not strong, we should modify the model. It was pointed by [Kitano, 2001] that binary values may lose the precision.

In [Martin *et al.*, 1984], a class of cellular automata (CA) which have additivity are studied by using algebraic technique. The global properties of CA are determined by a fixed polynomial. All the results give by [Martin *et al.*, 1984] can be used for p-valued CAs, where p is a prime integer. But when p is a composite number or the CA is non-additive, the algebraic technique given in [Martin *et al.*, 1984] seems not suitable. It was pointed by [Martin *et al.*, 1984]: "...*the possibility of universal computation with sufficiently complex non-additive cellar automata demonstrates that a complete analysis of these systems is fundamental impossible* ...". In this paper, we give a new way to study k-valued networks, and k can be any positive integer. The method given in this paper may be used not only to deal with the above *impossible* problem (while the system has limited size), but also deal with the more general multivalued cellular automata, multivalued

network. In the following sections, the dynamics of k-valued networks are discussed.

First, we give a rigorous definition.

Definition 1.2

(1) A k-valued logical variable A takes a value from D_k, which is

$$D_k = \left\{1, \frac{k-2}{k-1}, \ldots, \frac{1}{k-1}, 0\right\}. \tag{2}$$

(2) A k-valued network, $G(V, F, k)$, is defined by a set of nodes $V = \{A_1, A_2, \ldots, A_n\}$, and a list of logical functions $F = \{f_1, f_2, \ldots, f_n\}$, which provide a rule for each node to take values from D_k at each time $t = 0, 1, 2, \ldots$.

For the ith node $A_i(t) \in D_k, i = 1, 2, \ldots, n$, its value at $t+1$ is determined by the k-valued logical function f_i, called the network dynamics. The dynamics of the k-valued network can be described as

$$\begin{cases} A_1(t+1) = f_1(A_1(t), A_2(t), \ldots, A_n(t)) \\ A_2(t+1) = f_2(A_1(t), A_2(t), \ldots, A_n(t)) \\ \qquad \vdots \\ A_n(t+1) = f_n(A_1(t), A_2(t), \ldots, A_n(t)), \end{cases} \tag{3}$$

where f_i, $i = 1, 2, \ldots, n$ are n-ary k-valued logical functions.

One of the main tools we used is semi-tensor product, which is a generalization of conventional matrix product [Cheng & Qi, 2007]. For the sake of completeness, we will review it briefly in Sec. 2. When $k = 2$, the network becomes a Boolean network. Using semi-tensor product and the matrix expression of logic, the topological structure, including the fixed points, cycles and transient period, has been revealed by providing easily computable formulas for corresponding issues. This approach can also be used to the synthesis of Boolean control networks. Some related topics such as controllability and observability, etc. are also studied. We refer to [Cheng, 2009; Cheng & Qi, 2009a, 2009b] for details.

This paper considers the logical network, whose nodes can take more than two values. The paper extends the results obtained in [Cheng, 2009; Cheng & Qi, 2009a, 2009b] to multivalued networks. Using matrix approach, the topological structure of k-valued networks is revealed by providing precise formulas. Then some characteristics of the networks are discussed. Finally, the k-valued control networks are investigated and necessary and sufficient conditions for controllability are presented.

The paper is organized as follows. Section 2 gives a brief review on the semi-tensor product and the algebraic expression of logical operator. In Sec. 3, the dynamics of a k-valued network is converted into a standard discrete-time linear system. In Sec. 4 formulas are obtained to calculate (a) the number of equilibriums; (b) the numbers of cycles of different lengths; (c) transient time. The formulas for constructing them are also presented. In Sec. 5, it is proved that the k-valued network can be reconstructed from its network transition matrix L, and an algorithm is presented. In Sec. 6, the controllability of k-valued control networks is investigated. Section 7 is a brief conclusion.

2. Preliminaries

2.1. *Semi-tensor product*

This section is a brief review on semi-tensor product (STP) of matrices. STP was first introduced by [Cheng & Qi, 2007], and it plays a fundamental role in the following discussion. We restrict this review to the concepts and some basic properties, which are useful in the sequel. In addition, only left semi-tensor product for multiple-dimension case is considered in the paper. We refer to [Cheng, 2007; Cheng & Qi, 2007] for right semi-tensor product, general dimension case and more details. Throughout this paper "semi-tensor product" means the left semi-tensor product.

Definition 2.1

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

$$\begin{cases} X \ltimes Y = \displaystyle\sum_{i=1}^{p} X^i y_i \in \mathbb{R}^n, \\ Y^T \ltimes X^T = \displaystyle\sum_{i=1}^{p} y_i (X^i)^T \in \mathbb{R}^n. \end{cases} \tag{4}$$

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

consists of $m \times q$ blocks as $C = (C^{ij})$ and each block is

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

where A^i is ith row of A and B_j is the jth column of B.

We use some simple numerical examples to describe it.

Example 2.2

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

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

(2) Let

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

Then

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

Remark 2.3. Note that when $n = p$ the STP coincides with the conventional matrix product. Therefore, the STP is only a generalization of traditional matrix product.

Some fundamental properties of the STP are collected in the following, which coincide with that of conventional matrix product.

Proposition 2.4. *The STP satisfies* (*as long as the related products are well defined*)

(1) (*Distributive rule*)

$$A \ltimes (\alpha B + \beta C) = \alpha A \ltimes B + \beta A \ltimes C,$$
$$(\alpha B + \beta C) \ltimes A = \alpha B \ltimes A + \beta C \ltimes A, \quad \alpha, \beta \in \mathbb{R}. \tag{5}$$

(2) (*Associative rule*)

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

Proposition 2.5. *Let $A \in M_{p \times q}$ and $B \in M_{m \times n}$. If $q = km$, then*

$$A \ltimes B = A(B \otimes I_k), \tag{7}$$

If $kq = m$, then

$$A \ltimes B = (A \otimes I_k)B. \tag{8}$$

Proposition 2.6

(1) *Assume A and B are of proper dimensions such that $A \ltimes B$ is well defined, then*

$$(A \ltimes B)^T = B^T \ltimes A^T, \tag{9}$$

(2) *In addition, assume both A and B are invertible, then*

$$(A \ltimes B)^{-1} = B^{-1} \ltimes A^{-1}. \tag{10}$$

Proposition 2.7. *Assume $A \in M_{m \times n}$ is given.*

(1) *Let $Z \in \mathbb{R}^t$ be a row vector, then*

$$A \ltimes Z = Z \ltimes (I_t \otimes A), \tag{11}$$

(2) *Let $Z \in \mathbb{R}^t$ be a column vector, then*

$$Z \ltimes A = (I_t \otimes A) \ltimes Z. \tag{12}$$

Next we define the concept of swap matrix, which is also called the permutation matrix and is defined implicitly in [Magnus & Neudecker, 1999]. Many properties can be found in [Cheng, 2002; Cheng & Qi, 2007]. The swap matrix $W_{[m,n]}$ is an $mn \times mn$ matrix constructed in the following way: label its columns by $(11, 12, \ldots, 1n, \ldots, m1, m2, \ldots, mn)$ and its rows by $(11, 21, \ldots, m1, \ldots, 1n, 2n, \ldots, mn)$. Then its element in the position $((I, J), (i, j))$ is assigned as

$$w_{(IJ),(ij)} = \delta^{I,J}_{i,j} = \begin{cases} 1, & I = i \text{ and } J = j, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

When $m = n$, we simply denote by $W_{[n]}$ for $W_{[n,n]}$.

Example 2.8. Let $m = 2$ and $n = 3$, the swap matrix $W_{[2,3]}$ is constructed as

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

Let $A \in M_{m\times n}$, i.e. A is an $m \times n$ matrix. Denote by $V_r(A)$ the row stacking form of A, that is,

$$V_r(A) = (a_{11} \cdots a_{1n} \cdots a_{m1} \cdots a_{mn})^T,$$

and by $V_c(A)$ the column stacking form of A, that is,

$$V_c(A) = (a_{11} \cdots a_{m1} \cdots a_{1n} \cdots a_{mn})^T.$$

The following "swap" property shows the meaning of the name.

Proposition 2.9

(1) *Let $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^n$ be two columns, then*

$$\begin{aligned} W_{[m,n]} \ltimes X \ltimes Y &= Y \ltimes X, \\ W_{[n,m]} \ltimes Y \ltimes X &= X \ltimes Y. \end{aligned} \tag{14}$$

(2) *Let $A \in M_{m\times n}$, then*

$$\begin{aligned} W_{[m,n]} V_r(A) &= V_c(A), \\ W_{[n,m]} V_c(A) &= V_r(A). \end{aligned} \tag{15}$$

(3) *Let $X_i \in \mathbb{R}^{n_i}$, $i = 1, \ldots, m$. Then*

$$\begin{aligned} &(I_{n_1+\cdots+n_{k-1}} \otimes W_{[n_k,n_{k+1}]} \otimes I_{n_{k+2}+\cdots+n_m}) \\ &\quad \times X_1 \ltimes \cdots \ltimes X_k \ltimes X_{k+1} \ltimes \cdots \ltimes X_m \\ &\quad = X_1 \ltimes \cdots \ltimes X_{k+1} \ltimes X_k \ltimes \cdots \ltimes X_m. \end{aligned} \tag{16}$$

Proposition 2.10

$$W_{[m,n]}^T = W_{[m,n]}^{-1} = W_{[n,m]}. \tag{17}$$

Proposition 2.11

$$W_{[pq,r]} = (W_{[p,r]} \otimes I_q)(I_p \otimes W_{[q,r]}). \tag{18}$$

Taking transpose on both sides of (18) yields

$$W_{[r,pq]} = (I_p \otimes W_{[r,q]})(W_{[r,p]} \otimes I_q). \tag{19}$$

The swap matrix can be constructed by the following method: Denote the ith canonical basic element in $\mathbb{R}^n$ by δ_i^n. That is, δ_n^i is the ith column of I_n. Then we have

Proposition 2.12

$$\begin{aligned} W_{[m,n]} = (&\delta_n^1 \ltimes \delta_m^1 \cdots \delta_n^n \ltimes \delta_m^1 \cdots \delta_n^1 \\ &\ltimes \delta_m^m \cdots \delta_n^n \ltimes \delta_m^m). \end{aligned} \tag{20}$$

In [Magnus & Neudecker, 1999], (20) is used as the definition.

Using swap matrix, we can prove that

Proposition 2.13. *Let $A \in M_{m\times n}$ and $B \in M_{s\times t}$. Then*

$$\begin{aligned} A \otimes B &= W_{[s,m]} \ltimes B \ltimes W_{[m,t]} \ltimes A \\ &= (I_m \otimes B) \ltimes A. \end{aligned} \tag{21}$$

If $X \in \mathbb{R}^n$, $Y^T \in \mathbb{R}^m$, then

$$X \ltimes Y = Y \ltimes W_{[n,m]} \ltimes X. \tag{22}$$

Remark 2.14

(1) Let $X \in \mathbb{R}^m, Y \in \mathbb{R}^n$ be two column (or row) vectors. Then $X \ltimes Y$ is always well defined.
(2) Throughout this paper, the matrix product is assumed to be semi-tensor product. For notational compactness, hereafter in most cases the symbol "$\ltimes$" is omitted.

2.2. *Matrix expression of k-valued logic*

Now we consider the matrix expression of k-valued logic. Similar to Boolean logic, we will show that as the k-valued logical variables are expressed as vectors, the logical operators can also be determined by their structure matrix. Then the action of an operator can be performed as the product of its structure matrix with argument vectors. We refer to [Cheng & Qi, 2007] for this matrix expression.

We need some necessary notations and preresults for the matrix expression of k-valued logic.

Definition 2.15

(1) The domain of a k-valued logic, denoted by D_k, is

$$D_k = \left\{ T = 1, \frac{k-2}{k-1}, \ldots, \frac{1}{k-1}, F = 0 \right\}. \tag{23}$$

(2) An n-ary k-valued logical operator is a function $f : D_k^n \to D_k$, where n is called the arity of f, denoted by $ar(f) = n$ [Barnes, 1975].

To use matrix expression we identify the elements in D_k with a k-dimensional vector as $T = 1 \sim \delta_k^1, (k-2/k-1) \sim \delta_k^2, \ldots, F = 0 \sim \delta_k^k$, where δ_k^i is the ith column of the identity matrix I_k. Denote the set of such vectors as

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

We use $(k - i/k - 1)$ or δ_k^i alternatively for the value of a k-valued logical variables without explanation. The D_k and Δ_k will be used freely according to the variable types.

Using this vector expression, we can define the structure matrix of a k-valued logical operator.

First, we define some basic logical operators.

Definition 2.16. The following operators are defined using scalar value expression of logical variables.

- An anary logical operator $\neg : D_k \to D_k$, called the negation, is defined as

$$\neg(P) = \neg P := 1 - P;$$

- A binary logical operator $\wedge : D_k^2 \to D_k$, called the conjunction, is defined as

$$\wedge(P, Q) = P \wedge Q := \min(P, Q);$$

- A binary logical operator $\vee : D_k^2 \to D_k$, called the disjunction, is defined as

$$\vee(P, Q) = P \vee Q := \max(P, Q);$$

- An anary logical operator $\nabla_i : D_k \to D_k, i = 1, 2, \ldots, k$, is defined as

$$\nabla_i(P) = \begin{cases} 1, & \text{when } P = \dfrac{k-i}{k-1}, \\ 0, & \text{otherwise.} \end{cases}$$

Definition 2.17. An anary operator $\oslash : D_k \to D_k$, called the rotator, is defined as

$$\oslash(P) := \begin{cases} P + \dfrac{1}{k-1}, & P \neq 1, \\ 0, & P = 1. \end{cases}$$

Remark 2.18. In general, there are k^{k^n} n-ary k-valued logical operators. In Definitions 2.16 and 2.17, we give only a few of them. They are commonly used. Moreover, we can prove that they form a complete set [Luo, 1992]. Precisely speaking, all other k-valued logical operators can be expressed as combinations of $\{\oslash, \wedge, \vee\}$. Roughly speaking, they are enough to express any k-valued logical operators.

Definition 2.19. A $k \times k^s$ matrix M_σ is called the structure matrix of an s-ary k-valued logical operator σ, if

$$\sigma(P_1, P_2, \ldots, P_s) = M_\sigma P_1 P_2 \cdots P_s, \qquad (24)$$

where $P_1, \ldots, P_s \in \Delta_k$. Like in the Boolean logic case, if a structure matrix exists, it uniquely determines the k-valued logical operator [Cheng & Qi, 2007].

It is easy to check by direct computation that, for each operator we can construct its structure matrix. For notational ease, we let $k = 3$ and give the truth table for the operators in Tables 1 and 2.

Table 1. Truth table of 3-valued anary operators.

P	$\neg(P)$	$\oslash(P)$	$\nabla_1(P)$	$\nabla_2(P)$	$\nabla_3(P)$
1	0	0	1	0	0
0.5	0.5	1	0	1	0
0	1	0.5	0	0	1

Converting the values in truth table into vector forms yields their structure matrices. For instance, for negative operator $\neg$, we have

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

Example 2.20. When $P = \delta_3^1$,

$$\neg(P) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \delta_3^1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Definition 2.21. A $k \times s$ matrix is called a logical matrix, if its columns, Col(A), are of the form δ_k^i, i.e.

$$\mathrm{Col}(A) \subset \Delta_k.$$

The set of $k \times s$ logical matrixes is denoted by $\mathcal{L}_{k\times s}$. Let $A \in \mathcal{L}_{k\times s}$. Then A has its columns of δ_k^{ij}, $j = 1, \ldots, s$. For notational compactness, we denote it as

$$A = \delta_k[i_1 \quad i_2 \quad \cdots \quad i_s].$$

Using this notation, we have $M_n := M_\neg = \delta_3[3 \quad 2 \quad 1]$. Similarly, we have

$$M_o := M_\oslash = \delta_3[3 \quad 1 \quad 2],$$
$$M_c := M_\wedge = \delta_3[1 \quad 2 \quad 3 \quad 2 \quad 2 \quad 3 \quad 3 \quad 3 \quad 3],$$
$$M_d := M_\vee = \delta_3[1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 1 \quad 2 \quad 3].$$

Table 2. Truth table of 3-valued binary operators.

P	Q	$P \wedge Q$	$P \vee Q$
1	1	1	1
1	0.5	0.5	1
1	0	0	1
0.5	1	0.5	1
0.5	0.5	0.5	0.5
0.5	0	0	0.5
0	1	0	1
0	0.5	0	0.5
0	0	0	0

Similar to the Boolean case, for k-valued logic, we also need the power-reducing matrix, defined as

$$M_{r_k} = \begin{bmatrix} \delta_k^1 & 0_k & \cdots & 0_k \\ 0_k & \delta_k^2 & \cdots & 0_k \\ \vdots & & & \\ 0_k & 0_k & \cdots & \delta_k^k \end{bmatrix}, \tag{25}$$

where $0_k \in \mathbb{R}^k$, $M_{r_k} \in \mathcal{L}_{k^2 \times k}$.

As its name implies, it is easy to check that

Lemma 2.22. *Let $P \in \Delta_k$. Then we have*

$$P^2 = M_{r_k} P. \tag{26}$$

For being compact, denote by $M_{r_k} := M_r$. Using the above lemma, we can prove the following [Cheng & Qi, 2007]:

Theorem 2.23 [Cheng & Qi, 2007]. *Any k-valued s-ary logical operator $L(P_1, \ldots, P_s)$, with logical variables $P_1, \ldots, P_s \in \Delta_k$, can be expressed in a canonical form as*

$$L(P_1, \ldots, P_s) = M_L P_1 P_2 \cdots P_s, \tag{27}$$

where $M_L \in \mathcal{L}_{k \times k^s}$, called the structure matrix of L.

Remark 2.24. In Boolean logic, we have $A \to B = (\neg A) \vee B$ and $A \leftrightarrow B = (A \to B) \wedge (B \to A)$. If we use them as the definitions of implication and equivalence for k-valued logic, when $k = 3$, we have

$$M_i = M_d M_n = \delta_3[1 \quad 2 \quad 3 \quad 1 \quad 2 \quad 2 \quad 1 \quad 1 \quad 1],$$

and

$$\begin{aligned} M_e &= M_c M_i (I_9 \otimes M_i)(I_3 \otimes M_r)(I_3 \otimes W_{[3]}) M_r \\ &= \delta_3[1 \quad 2 \quad 3 \quad 2 \quad 2 \quad 2 \quad 3 \quad 2 \quad 1]. \end{aligned}$$

In Table 3, we list the structure matrixes for some basic logical operators (when $k = 3$), which are used in the sequel.

Table 3. Some matrix of operators ($k = 3$).

Operator	Structure Matrix
$\neg$	$M_n = \delta_3[3 \quad 2 \quad 1]$
$\oslash$	$M_o = \delta_3[3 \quad 1 \quad 2]$
∇_1	$M_{\nabla_1} = \delta_3[1 \quad 1 \quad 1]$
∇_2	$M_{\nabla_2} = \delta_3[2 \quad 2 \quad 2]$
∇_3	$M_{\nabla_3} = \delta_3[3 \quad 3 \quad 3]$
$\vee$	$M_d = \delta_3[1 \quad 1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 1 \quad 2 \quad 3]$
$\wedge$	$M_c = \delta_3[1 \quad 2 \quad 3 \quad 2 \quad 2 \quad 3 \quad 3 \quad 3 \quad 3]$
$\to$	$M_c = \delta_3[1 \quad 2 \quad 3 \quad 1 \quad 2 \quad 2 \quad 1 \quad 1 \quad 1]$
$\leftrightarrow$	$M_e = \delta_3[1 \quad 2 \quad 3 \quad 2 \quad 2 \quad 2 \quad 3 \quad 2 \quad 1]$

3. Algebra Form of Multivalued Networks

In this section, using semi-tensor-product (STP), we convert the dynamics of a multivalued network (3) into an algebraic form. The technique is the same as the one for Boolean networks. We use some examples to depict this.

Example 3.1. Consider the following k-valued network

$$\begin{cases} A(t+1) = A(t) \\ B(t+1) = A(t) \to C(t) \\ C(t+1) = B(t) \vee D(t) \\ D(t+1) = \neg E(t) \\ E(t+1) = \neg C(t). \end{cases} \tag{28}$$

In algebraic form, we have

$$\begin{cases} A(t+1) = A(t) \\ B(t+1) = M_i A(t) C(t) \\ C(t+1) = M_d B(t) D(t) \\ D(t+1) = M_n E(t) \\ E(t+1) = M_n C(t). \end{cases} \tag{29}$$

where M_i, M_d, M_n are structure matrixes for the corresponding logical operators defined in the above section.

Define $x(t) = A(t)B(t)C(t)D(t)E(t)$. Then we have

$$\begin{aligned} x(t+1) = A(t) M_i A(t) C(t) M_d B(t) D(t) \\ \times M_n E(t) M_n C(t). \end{aligned} \tag{30}$$

Using the pseudo-commutative property of semi-tensor product, say Proposition 2.7 etc., we can move $A(t), B(t), \ldots, E(t)$ to the last part of the product in the right-hand side of (30). Then use the power-reducing matrix to reduce the powers of $A(t), B(t), \ldots, E(t)$ to 1. Finally, we have

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

where

$$\begin{aligned} L = {} & (I_k \otimes M_i) M_r (I_k \otimes (I_k \otimes M_d (I_k \\ & \otimes (I_k \otimes M_n (I_k \otimes M_n))))) (I_k \otimes W_{[k]}) \\ & \times (I_{k^4} \otimes W_{[k]}) (I_{k^3} \otimes W_{[k]}) (I_k \otimes (I_k \otimes M_r)). \end{aligned}$$

When k is decided, we can calculate the network matrix L. Say, $k=3$, L is a 243×243 matrix as

$$
\begin{aligned}
L=\delta_{243}[&9 \quad 6 \quad 3 \quad 9 \quad 6 \quad 3 \quad 9 \quad 6 \quad 3 \quad 35\\
&32 \quad 29 \quad 35 \quad 32 \quad 29 \quad 35 \quad 32 \quad 29 \quad 61 \quad 58\\
&55 \quad 61 \quad 58 \quad 55 \quad 61 \quad 58 \quad 55 \quad 9 \quad 6 \quad 3\\
&18 \quad 15 \quad 12 \quad 18 \quad 15 \quad 12 \quad 35 \quad 32 \quad 29 \quad 44\\
&41 \quad 38 \quad 44 \quad 41 \quad 38 \quad 61 \quad 58 \quad 55 \quad 70 \quad 67\\
&4 \quad 70 \quad 67 \quad 64 \quad 9 \quad 6 \quad 3 \quad 18 \quad 15 \quad 12\\
&27 \quad 24 \quad 21 \quad 35 \quad 32 \quad 29 \quad 44 \quad 41 \quad 38 \quad 53\\
&50 \quad 47 \quad 61 \quad 58 \quad 55 \quad 70 \quad 67 \quad 64 \quad 79 \quad 76\\
&73 \quad 90 \quad 87 \quad 84 \quad 90 \quad 87 \quad 84 \quad 90 \quad 87 \quad 84\\
&116 \quad 113 \quad 110 \quad 116 \quad 113 \quad 110 \quad 116 \quad 113 \quad 110 \quad 115\\
&112 \quad 109 \quad 115 \quad 112 \quad 109 \quad 115 \quad 112 \quad 109 \quad 90 \quad 87\\
&84 \quad 99 \quad 96 \quad 93 \quad 99 \quad 96 \quad 93 \quad 116 \quad 1131 \quad 10\\
&125 \quad 122 \quad 119 \quad 125 \quad 122 \quad 119 \quad 115 \quad 112 \quad 109 \quad 124\\
&121 \quad 118 \quad 124 \quad 121 \quad 118 \quad 90 \quad 87 \quad 84 \quad 99 \quad 96\\
&93 \quad 108 \quad 105 \quad 102 \quad 116 \quad 113 \quad 110 \quad 125 \quad 122 \quad 119\\
&134 \quad 131 \quad 128 \quad 115 \quad 112 \quad 109 \quad 124 \quad 121 \quad 118 \quad 133\\
&130 \quad 127 \quad 171 \quad 168 \quad 165 \quad 171 \quad 168 \quad 165 \quad 171 \quad 168\\
&165 \quad 170 \quad 167 \quad 164 \quad 170 \quad 167 \quad 164 \quad 170 \quad 167 \quad 164\\
&169 \quad 166 \quad 163 \quad 169 \quad 166 \quad 163 \quad 169 \quad 166 \quad 163 \quad 171\\
&168 \quad 165 \quad 180 \quad 177 \quad 174 \quad 180 \quad 177 \quad 174 \quad 170 \quad 167\\
&164 \quad 179 \quad 176 \quad 173 \quad 179 \quad 176 \quad 173 \quad 169 \quad 166 \quad 163\\
&178 \quad 175 \quad 172 \quad 178 \quad 175 \quad 172 \quad 171 \quad 168 \quad 165 \quad 180\\
&177 \quad 174 \quad 189 \quad 186 \quad 183 \quad 170 \quad 167 \quad 164 \quad 179 \quad 176\\
&173 \quad 188 \quad 185 \quad 182 \quad 169 \quad 166 \quad 163 \quad 178 \quad 175 \quad 172\\
&187 \quad 184 \quad 181].
\end{aligned}
$$

Example 3.2. Consider the following k-valued network

$$
\begin{cases}
A(t+1)=B(t)\\
B(t+1)=(D(t)\to\neg C(t))\vee A(t)\\
C(t+1)=B(t)\leftrightarrow D(t)\\
D(t+1)=\neg A(t)\\
E(t+1)=\neg C(t).
\end{cases}
\tag{31}
$$

In algebraic form, we have

$$
\begin{cases}
A(t+1)=B(t)\\
B(t+1)=M_dM_iD(t)M_nC(t)A(t)\\
C(t+1)=M_eB(t)D(t)\\
D(t+1)=M_nA(t)\\
E(t+1)=M_nC(t).
\end{cases}
\tag{32}
$$

Define $x(t)=A(t)B(t)C(t)D(t)E(t)$. We have

$$
\begin{aligned}
x(t+1)=&B(t)(M_dM_iDM_nC(t)A(t))(M_eB(t)D(t))\\
&\times(M_nA(t))(M_nC(t)).
\end{aligned}
\tag{33}
$$

Since there is no $E(t)$ on the left-hand side, we have to formally add it.

As in the Boolean case, we need a dummy operator to add some fabricated variables into the right-hand side of Eq. (33) when these variables do not appear. Define

$$
E_d:=\underbrace{[I_k\,I_k\,\cdots\,I_k]}_{k}:=\delta_k[\underbrace{1\cdots k}_{k}\cdots\underbrace{1\cdots k}_{k}].
\tag{34}
$$

A straightforward computation shows that

Proposition 3.3

$$
E_dPQ=Q,\quad P,Q\in\Delta_k.
\tag{35}
$$

Similar to the Boolean case [Cheng, 2009], we can prove the following power-reducing formula by mathematical induction.

Lemma 3.4. *Assume* $P_\ell = A_1 A_2 \cdots A_\ell$, *where* $A_i \in \Delta_k$, $i = 1, 2, \ldots, \ell$, *then*

$$P_\ell^2 = \Phi_\ell P_\ell, \tag{36}$$

where

$$\Phi_\ell = \prod_{i=1}^{\ell} \left(I_{k^{i-1}} \otimes \left[I_k \otimes W_{[k,k^{\ell-i}]} M_{r_k} \right] \right).$$

In the following, we give a universal method to convert the network dynamics to a discrete system. For a network described by Eq. (3), using semi-tensor product, we can convert Eq. (3) into an algebraic form. Define

$$x(t) = A_1(t) A_2(t) \cdots A_n(t).$$

Using Theorem 2.23, we can find structure matrixes, $M_i = M_{f_i}$, $i = 1, \ldots, n$, such that

$$A_i(t+1) = M_i x(t), \quad i = 1, 2, \ldots, n. \tag{37}$$

Using Lemma 3.4, a straightforward computation shows the following:

Proposition 3.5. *Equation (3) can be expressed as*

$$x(t+1) = Lx(t), \quad x \in \Delta_{k^n}, \tag{38}$$

where the system coefficient matrix L *is*

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

Proof. Note that from Lemma 3.4 we have

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

Now

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

So

$$\begin{aligned} L &= M_1 (I_k \otimes M_2)\Phi_n (I_k \otimes M_3)\Phi_n \cdots (I_k \otimes M_n)\Phi_n \\ &= M_1 \prod_{j=2}^{n} [(I_k \otimes M_j)\Phi_n]. \end{aligned}$$

■

We call the system coefficient matrix L of Eq. (38) the network transition matrix. For a particular system, we may get the network transition matrix by a direct computation.

Example 3.6. Reconsider Example 3.2, we add $E(t)$ by using dummy matrix E_d.

$$\begin{aligned} x(t+1) &= B(t)(M_d M_i D M_n C(t) A(t)) \\ &\quad \times (M_e B(t) D(t))(M_n A(t))(M_n C(t)) \\ &= B(t)(M_d M_i D M_n C(t) A(t))(M_e B(t) D(t)) \\ &\quad \times (M_n A(t))(M_n E_d E(t) C(t)). \end{aligned}$$

Using the same technique as in Example 3.1, we have

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

where

$$\begin{aligned} L = (&I_k \otimes M_d M_i (I_k \otimes M_n (I_k \otimes (I_k \otimes M_e \\ &\times (I_k \otimes (I_k \otimes M_n (I_k \otimes M_n E_d)))))))(I_{k^2} \otimes W_{[k]}) \\ &\times (I_k \otimes W_{[k]}) W_{[k]} (I_{k^5} \otimes W_{[k]})(I_{k^4} \otimes W_{[k]}) \\ &\times (I_{k^3} \otimes W_{[k]})(I_{k^2} \otimes W_{[k]})(I_k \otimes W_{[k]})(I_{k^4} \otimes W_{[k]}) \\ &\times (I_{k^3} \otimes W_{[k]})(I_{k^4} \otimes W_{[k]})(I_{k^7} \otimes W_{[k]})(I_{k^6} \otimes W_{[k]}) \\ &\times (I_{k^5} \otimes W_{[k]}) M_r (I_k \otimes M_r (I_k \otimes M_r))). \end{aligned}$$

When $k = 3$, we can compute

$$\begin{matrix} L = \delta_{243}[9 & 9 & 9 & 18 & 18 & 18 & 27 & 27 & 27 & 8 \\ 8 & 8 & 17 & 17 & 17 & 26 & 26 & 26 & 7 & 7 \\ 7 & 16 & 16 & 16 & 25 & 25 & 25 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 98 & 98 & 98 & 98 \\ 98 & 98 & 98 & 98 & 98 & 97 & 97 & 97 & 97 & 97 \\ 97 & 97 & 97 & 97 & 189 & 189 & 189 & 180 & 180 & 180 \\ 171 & 171 & 171 & 188 & 188 & 188 & 179 & 179 & 179 & 170 \\ 170 & 170 & 187 & 187 & 187 & 178 & 178 & 178 & 169 & 169 \end{matrix}$$

169	33	33	33	42	42	42	24	24	24
32	32	32	41	41	41	23	23	23	4
4	4	13	13	13	22	22	22	123	123
123	123	123	123	96	96	96	122	122	122
122	122	122	95	95	95	94	94	94	94
94	94	94	94	94	213	213	213	204	204
204	168	168	168	212	212	212	203	203	203
167	167	167	184	184	184	175	175	175	166
166	166	57	57	57	39	39	39	21	21
21	29	29	29	38	38	38	20	20	20
1	1	1	10	10	10	19	19	19	147
147	147	120	120	120	93	93	93	119	119
119	119	119	119	92	92	92	91	91	91
91	91	91	91	91	91	237	237	237	201
201	201	165	165	165	209	209	209	200	200
200	164	164	164	181	181	181	172	172	172
163	163	163].							

4. Attractors and Transient Period

This section considers the attractors and transient period of k-valued networks. As in the study of Boolean network, the attractors in k-valued networks are also important. We will use the algebraic form [Eq. (38)] to investigate these. Now we have to answer such a question: Is the original logical system equivalent to its algebraic form? Using the properties of semi-tensor product, it is easy to prove the following proposition, which shows how to calculate $A_i(t)$ from $x(t)$. Then one sees easily that $x(t)$ contains the same information as the set $\{A_1(t), A_2(t), \ldots, A_n(t)\}$.

Proposition 4.1

(1) *The state,* $x(t)$*, is of the form of* $\delta^i_{k^n}$*. Precisely,*

$$x(t) \in \Delta_{k^n}, \quad \forall\, t \geq 0. \tag{39}$$

(2) *Assume* $x(t) = \delta^i_{k^n}$*. Define* $b_0 := k^n - i$*, then* $A_j(t)$ *can be calculated inductively (in scalar form) as*

$$\begin{cases} a_j(t) = \left[\dfrac{b_{j-1}}{k^{n-j}}\right], \\ b_j = b_{j-1} - a_j * k^{n-j}, \\ A_j = a_j/(k-1), \quad j = 1, 2, \ldots, n \end{cases} \tag{40}$$

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

We give an example to explain how to use the above algorithm.

Example 4.2. Assume $x = A_1A_2A_3A_4A_5$ and $x = \delta^{17}_{243}$. Then $b_0 = 243 - 17 = 226$. It follows that

$$\begin{cases} a_1 = \left[\dfrac{b_0}{3^4}\right] = 2, \\ A_1 = 1, \\ b_1 = b_0 - a_1 * (3^4) = 64, \end{cases}$$

$$\begin{cases} a_2 = \left[\dfrac{b_1}{3^3}\right] = 2, \\ A_2 = 1, \\ b_2 = b_1 - a_2 * 3^3 = 10, \end{cases}$$

$$\begin{cases} a_3 = \left[\dfrac{b_2}{3^2}\right] = 1, \\ A_3 = 0.5, \\ b_3 = b_2 - a_3 * 3^2 = 1, \end{cases}$$

$$\begin{cases} a_4 = \left[\dfrac{b_3}{3}\right] = 0, \\ A_4 = 0, \\ b_4 = b_3 - a_4 * 3 = 1, \end{cases}$$

$$\begin{cases} a_5 = \left[\dfrac{b_4}{1}\right] = 1, \\ A_5 = 0.5. \end{cases}$$

We conclude that $A_1 = 1 \sim (1,0,0)^T$, $A_2 = 1 \sim (1,0,0)^T$, $A_3 = 0.5 \sim (0,1,0)^T$, $A_4 = 0 \sim (0,0,1)^T$, and $A_5 = 0.5 \sim (0,1,0)^T$.

Using the same technique developed for Boolean network, we can obtain the following results for cycles and transient period.

Theorem 4.3. *Consider the k-valued network (3).*

(1) *$\delta_{k^n}^i$ is its fixed point, iff in its algebraic form (38) the diagonal element ℓ_{ii} of network matrix L equals to 1. It follows that the number of equilibriums of system (3), denoted by N_e, is equal to the number of i, for which $\ell_{ii} = 1$. Equivalently,*

$$N_e = \mathrm{Trace}(L). \tag{41}$$

(2) *The number of length s cycles, N_s, is inductively determined by*

$$\begin{cases} N_1 = N_e, \\ N_s = \dfrac{\mathrm{Trace}(L^s) - \sum\limits_{t \in \mathcal{P}(s)} t N_t}{s}, \quad 2 \le s \le k^n, \end{cases} \tag{42}$$

where $\mathcal{P}(s)$ is set of proper factors of s, that is, a positive integer $t \in \mathcal{P}(s)$ iff $t < s$ and s/t is an integer. For example, $\mathcal{P}(6) = \{1,2,3\}$, $\mathcal{P}(10) = \{1,2,5\}$.

(3) *The elements on cycles of length s, denoted by $\mathcal{C}_s$, is*

$$\mathcal{C}_s = \mathcal{D}_a(L^s) \backslash \cup_{t \in \mathcal{P}(s)} \mathcal{D}_a(L^t), \tag{43}$$

where $\mathcal{D}_a(L)$ is the set of diagonal nonzero columns of L.

Theorem 4.4. *For system (3) the transient period is*

$$T_t = r_0 = \min\{r \,|\, L^r \in \{L^{r+1}, L^{r+2}, \ldots, L^{k^n}\}\}. \tag{44}$$

Moreover, let $T > 0$ be the smallest positive number, which implies $L^{r_0} = L^{r_0+T}$. Then T is the least common multiplier of the lengths of all cycles.

We give an example to illustrate how to find the fixed points and cycles in k-valued network by using the above theorems.

Example 4.5. Consider the following 3-valued network

$$\begin{cases} A(t+1) = C(t) \wedge (\neg D(t)) \\ B(t+1) = (A(t) \leftrightarrow B(t)) \wedge D(t) \\ C(t+1) = \neg A(t) \\ D(t+1) = B(t) \vee C(t). \end{cases} \tag{45}$$

Define $x(t) = A(t)B(t)C(t)D(t)$. It is easy to calculate that

$$\begin{aligned} x(t+1) &= [M_c C(t) M_n D(t)][M_c M_e A(t) B(t) D(t)] \\ &\quad \times [M_n A(t)][M_d B(t) C(t)] \\ &= L x(t). \end{aligned}$$

where $L \in \mathcal{L}_{3^4 \times 3^4}$. L can be calculated as

$$\begin{aligned} L = \delta_{81}[&61 \quad 43 \quad 25 \quad 61 \quad 43 \quad 52 \quad 61 \\ &70 \quad 79 \quad 70 \quad 43 \quad 25 \quad 71 \quad 44 \\ &53 \quad 71 \quad 71 \quad 80 \quad 79 \quad 52 \quad 25 \\ &80 \quad 53 \quad 53 \quad 81 \quad 81 \quad 81 \quad 67 \\ &40 \quad 22 \quad 67 \quad 40 \quad 49 \quad 67 \quad 67 \\ &76 \quad 67 \quad 40 \quad 22 \quad 68 \quad 41 \quad 50 \\ &68 \quad 68 \quad 77 \quad 67 \quad 40 \quad 22 \quad 68 \\ &41 \quad 50 \quad 69 \quad 69 \quad 78 \quad 73 \quad 46 \\ &19 \quad 73 \quad 46 \quad 46 \quad 73 \quad 73 \quad 73 \\ &64 \quad 37 \quad 19 \quad 65 \quad 38 \quad 47 \quad 65 \\ &65 \quad 74 \quad 55 \quad 37 \quad 19 \quad 56 \quad 38 \\ &47 \quad 57 \quad 66 \quad 75]. \end{aligned}$$

Then it can be readily checked that

$$\begin{cases} \mathrm{Trace}(L^{6k+1}) = 2, & k = 0,1,2,\ldots \\ \mathrm{Trace}(L^{6k+2}) = 4, & k = 0,1,2,\ldots \\ \mathrm{Trace}(L^{6k+3}) = 11, & k = 0,1,2,\ldots \\ \mathrm{Trace}(L^{6k+4}) = 4, & k = 0,1,2,\ldots \\ \mathrm{Trace}(L^{6k+5}) = 2, & k = 0,1,2,\ldots \\ \mathrm{Trace}(L^{6k}) = 13, & k = 1,2,\ldots. \end{cases} \tag{46}$$

From Theorem 4.3, we conclude that the network has two fixed points, one cycle of length 2, three cycles of length 3.

Denote the ith column of L as L_i, because $L_{41} = \delta_{81}^{41}$ and $L_{64} = \delta_{81}^{64}$. Hence, the two fixed points are

$$\delta_{81}^{41} \sim (0.5 \quad 0.5 \quad 0.5 \quad 0.5), \quad \delta_{81}^{64} \sim (0 \quad 0.5 \quad 1 \quad 1).$$

Next, we can find the cycles. Consider L^2. It is easy to figure out that $(L^2)_{41} = \delta_{81}^{41}$, $(L^2)_{64} = \delta_{81}^{64}$, $(L^2)_{55} = \delta_{81}^{55}$, $(L^2)_{73} = \delta_{81}^{73}$. We should note that

δ_{81}^{41} and δ_{81}^{64} are two fixed points. For δ_{81}^{55} and δ_{81}^{73}, it follows that

$$L\delta_{81}^{55} = \delta_{81}^{73}, \quad L\delta_{81}^{73} = \delta_{81}^{55}.$$

Using formula (40) to convert δ_{81}^{55} and δ_{81}^{73} back to ternary form, we have $\delta_{81}^{55} \sim (0\ 1\ 1\ 1)$, $\delta_{81}^{73} \sim (0\ 0\ 1\ 1)$.

So the cycle of length 2 is

$$\delta_{81}^{55} \to \delta_{81}^{73} \to \delta_{81}^{55},$$

and the corresponding state expression is

$$(0\ 1\ 1\ 1) \to (0\ 0\ 1\ 1) \to (0\ 1\ 1\ 1).$$

Similarly, since δ_{81}^{19} is a diagonal nonzero column of L^3, then δ_{81}^{19}, $L\delta_{81}^{19} = \delta_{81}^{79}$, $L^2\delta_{81}^{19} = \delta_{81}^{57}$, $L^3\delta_{81}^{19} = \delta_{81}^{19}$ form a cycle of length 3. Converting them to ternary form yields the following cycle:

$$(1\ 0\ 1\ 1) \to (0\ 0\ 0\ 1) \to (0\ 1\ 1\ 0) \to (1\ 0\ 1\ 1).$$

We can also use δ_{81}^{79} or δ_{81}^{57} to generate the above cycle of length 3. Using the same method, the other two cycles of length 3 are obtained as

$$\begin{aligned}
&(0.5\quad 0.5\quad 1\quad 1) \to (0\quad 0.5\quad 0.5\quad 1)\\
&\to (0\quad 0.5\quad 1\quad 0.5) \to (0.5\quad 0.5\quad 1\quad 1),\\
&(0.5\quad 0.5\quad 1\quad 0.5) \to (0.5\quad 0.5\quad 0.5\quad 1)\\
&\to (0\quad 0.5\quad 0.5\quad 0.5) \to (0.5\quad 0.5\quad 1\quad 0.5).
\end{aligned}$$

Finally, we consider the transient period. Since it is easy to check that the first repeating L^t is $L^4 = L^{10}$, then $r_0 = 4$. This ensures that, from any initial state, the trajectory will enter into an attractor after at most four steps.

Example 4.6. Recall Example 3.1. It is easy to check that

$$\mathrm{Trace}(L^t) = 6, \quad t = 1, 2, \ldots$$

Using Theorem 4.3, we conclude that there are six fixed points and no other attractors. Moreover, we can find out the fixed points in the following way.

Consider the transition matrix L. It is easy to figure out that the 3rd, 41st, 79th, 84th, 122ed, 165th columns of L are diagonal nonzero columns. So the six fixed points are $\delta_{3^5}^{3}, \delta_{3^5}^{41}, \delta_{3^5}^{79}, \delta_{3^5}^{84}, \delta_{3^5}^{122}, \delta_{3^5}^{165}$. Using Proposition 4.1, we can convert the fixed points back to standard form as

$$\begin{aligned}
E_1 &= (1\quad 1\quad 1\quad 1\quad 0),\\
E_2 &= (1\quad 0.5\quad 0.5\quad 0.5\quad 0.5),\\
E_3 &= (1\quad 0\quad 0\quad 0\quad 1),\\
E_4 &= (0.5\quad 1\quad 1\quad 1\quad 0),\\
E_5 &= (0.5\quad 0.5\quad 0.5\quad 0.5\quad 0.5),\\
E_6 &= (0\quad 1\quad 1\quad 1\quad 0).
\end{aligned}$$

It is easy to check that the first repeating L^k is $L^6 = L^7$, then $r_0 = 6$. That is $T_t = 6, T = 1$. So the transient period is 6, which means that from any initial state, the trajectory will enter an attractor after at most six steps.

5. Network Reconstruction

Assume for a k-valued logical system the network matrix L is given. We have to reconstruct the logical network and its dynamics from the network matrix. Denoting a set of column vectors as

$$\Omega_i^k = [\underbrace{1, \ldots, 1}_{i}, \underbrace{2, \ldots, 2}_{i}, \ldots, \underbrace{k, \ldots, k}_{i}].$$

Define a set of matrices, called the retrievers, as

$$\begin{aligned}
S_1^n &= \delta_k[\Omega^k_{k^{n-1}}];\\
S_2^n &= \delta_k[\underbrace{\Omega^k_{k^{n-2}}, \ldots, \Omega^k_{k^{n-2}}}_{k}];\\
&\vdots\\
S_n^n &= \delta_k[\underbrace{\Omega_1^k, \ldots, \Omega_1^k}_{k^{n-1}}].
\end{aligned} \tag{47}$$

For example, when $n = 2$ and $k = 3$, $\Omega^3_{3^{2-1}} = \Omega_3^3 = [1, 1, 1, 2, 2, 2, 3, 3, 3]$, $\Omega^3_{3^{2-2}} = \Omega_1^3 = [1, 2, 3]$. Hence,

$$\begin{aligned}
S_1^2 &= \delta_3[1\quad 1\quad 1\quad 2\quad 2\quad 2\quad 3\quad 3\quad 3];\\
S_2^2 &= \delta_3[1\quad 2\quad 3\quad 1\quad 2\quad 3\quad 1\quad 2\quad 3].
\end{aligned}$$

Using them, we have

Proposition 5.1. *Assume the network matrix L of system (38) is known. Then the structure matrix of f_i is*

$$M_i = S_i^n L, \quad i = 1, 2, \ldots, n. \tag{48}$$

Next, we have to find which node is connected to i. That is, to remove fabricated variables from

ith logical equation. We have the following:

Proposition 5.2. *Consider system (38). Assume M_i satisfies*

$$\begin{aligned} &M_i W_{[k,k^{j-1}]}(M_o - I_k) = 0,\\ &M_i W_{[k,k^{j-1}]}((M_o)^2 - I_k) = 0,\\ &\qquad\vdots\\ &M_i W_{[k,k^{j-1}]}((M_o)^{k-1} - I_k) = 0, \end{aligned} \tag{49}$$

where M_o is the structure matrix of $\oslash$. Then the node j is not in the neighborhood of node i. In other words, the edge $j \to i$ does not exist. Then the equation of A_i can be replaced by

$$A_i(t+1) = M_i' A_1(t) \cdots A_{j-1}(t) A_{j+1}(t) \cdots A_n(t), \tag{50}$$

where

$$M_i' = M_i W_{[k,k^{j-1}]} \delta_k^1.$$

Proof. Using the property of semi-tensor product, we can rewrite the ith equation of (3) as

$$\begin{aligned} A_i(t+1) = M_i W_{[k,k^{j-1}]} A_j(t) A_1(t) \cdots\\ A_{j-1}(t) A_{j+1}(t) \cdots A_n(t). \end{aligned}$$

Now we replace $A_j(t)$ by $\nabla_1(A_j(t))$, $\nabla_2(A_j(t))$, ..., $\nabla_{k-1}(A_j(t))$, 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 Eq. (49). As for Eq. (50), since $A_j(t)$ does not affect $A_i(t+1)$, we can simply set $A_j(t) = [1, 0, \ldots, 0]^T$ (you can set $A_j(t) = [0, \ldots, 0, 1]^T$ or any other elements in Δ_k) to simplify the expression. ■

Example 5.3. Given a 3-valued network with four nodes. Assume its network transition matrix $L \in M_{81\times 81}$ is

$$\begin{array}{rccccccccc} L = \delta_{81}[3 & 6 & 9 & 29 & 41 & 44 & 55 & 67 & 79\\ 3 & 3 & 6 & 9 & 29 & 41 & 44 & 28 & 40\\ 52 & 3 & 6 & 9 & 2 & 14 & 17 & 1 & 13\\ 25 & 6 & 6 & 9 & 32 & 41 & 44 & 58 & 67\\ 79 & 6 & 6 & 9 & 32 & 41 & 44 & 31 & 40\\ 52 & 6 & 6 & 9 & 5 & 14 & 17 & 4 & 13\\ 25 & 9 & 9 & 9 & 35 & 44 & 44 & 61 & 70\\ 79 & 9 & 9 & 9 & 35 & 44 & 44 & 34 & 43\\ 52 & 9 & 9 & 9 & 8 & 17 & 17 & 7 & 16\\ 25]. \end{array}$$

We can reconstruct the system as follows. Using retrievers S_i^3, we have

$$M_i = S_i^3 L, \quad i = 1, 2, 3, 4.$$

A straightforward computation shows that

$$\begin{array}{rcccccccccccccccccccc} M_1 = \delta_3[1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 1 & 1 & 1 & 2 & 2 & 2\\ 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3\\ 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1], & & & & & & & & & & & & & & & & & & \\ M_2 = \delta_3[1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 3 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 3 & 1 & 1 & 1\\ 1 & 2 & 2 & 1 & 2 & 3 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 3 & 1 & 1 & 1 & 1 & 2 & 2\\ 1 & 2 & 3 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 3 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 3\\ 1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 3 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 2 & 3], & & & & & & & & & & & & & & & & & & \\ M_3 = \delta_3[1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3\\ 1 & 2 & 3 & 1 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3\\ 2 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3\\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3], & & & & & & & & & & & & & & & & & & \\ M_4 = \delta_3[3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 3 & 3 & 3\\ 2 & 2 & 2 & 1 & 1 & 1 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 3 & 3 & 3 & 2 & 2 & 2\\ 1 & 1 & 1 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1\\ 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1]. & & & & & & & & & & & & & & & & & & \end{array}$$

Next, to remove fabricated variables, it is easy to verify that

$$M_1M_o - M_1 = 0, \quad M_1(M_o)^2 - M_1 = 0,$$
$$M_1W_{[3]}(M_o - I_3) \neq 0, \quad M_1W_{[3]}((M_o)^2 - I_3) \neq 0,$$
$$M_1W_{[3,3^2]}(M_o - I_3) \neq 0,$$
$$M_1W_{[3,3^2]}((M_o)^2 - I_3) \neq 0,$$
$$M_1W_{[3,3^3]}(M_o - I_3) = 0,$$
$$M_1W_{[3,3^3]}((M_o)^2 - I_3) = 0.$$

So we conclude that $A(t+1)$ depends on $B(t)$ and $C(t)$ only. Using the same procedure, we know that $B(t+1)$ depends only on $C(t)$ and $D(t)$; $C(t+1)$ depends only on $A(t)$ and $D(t)$; $D(t+1)$ depends only on $C(t)$. To remove the fabricated variables $A(t)$ and $D(t)$ from the first equation, we set $A(t) = D(t) = \delta_3^1$ and get

$$\begin{aligned} A(t+1) &= M_1\delta_3^1B(t)C(t)\delta_3^1 \\ &= M_1\delta_3^1W_{[3,9]}\delta_3^1 \\ &= \delta_3[1\ \ 2\ \ 3\ \ 1\ \ 2\ \ 2\ \ 1\ \ 1\ \ 1] \\ &\quad \times B(t)C(t). \end{aligned} \tag{51}$$

In a similar way, we can remove the fabricated variables from the other equations. And finally we get

$$\begin{aligned} B(t+1) &= \delta_3[1\ \ 1\ \ 1\ \ 1\ \ 2\ \ 2\ \ 1\ \ 2\ \ 3] \\ &\quad \times C(t)D(t), \\ C(t+1) &= \delta_3[1\ \ 2\ \ 3\ \ 2\ \ 2\ \ 3\ \ 3\ \ 3\ \ 3] \\ &\quad \times D(t)A(t), \\ D(t+1) &= \delta_3[3\ 2\ 1]C(t). \end{aligned}$$

Converting back to logical equations, we have

$$\begin{cases} A(t+1) = B(t) \to C(t), \\ B(t+1) = C(t) \vee D(t), \\ C(t+1) = D(t) \wedge A(t), \\ D(t+1) = \neg C(t). \end{cases} \tag{52}$$

In general, converting an algebraic form back to its logical form is not an easy job. We give a mechanical procedure to do this.

Recall Definition 2.16, it is easy to see that,

$$M_{\nabla_i} = \delta_k[\underbrace{k, \ldots, k}_{i-1}, 1, \underbrace{k, \ldots, k}_{k-i}], \quad i = 1, 2, \ldots, k. \tag{53}$$

Proposition 5.4. *Assume a k-valued logical variable L has an algebraic expression as*

$$L = L(A_1, A_2, \ldots, A_n) = M_LA_1A_2 \cdots A_n, \tag{54}$$

where M_L is the structure matrix of logical variable L. Then

$$\begin{aligned} M_L = &[\nabla_1(A_1) \wedge L_1(A_2, \ldots, A_n)] \\ &\vee [\nabla_2(A_1) \wedge L_2(A_2, \ldots, A_n)] \vee \cdots \\ &\vee [\nabla_k(A_1) \wedge L_k(A_2, \ldots, A_n)], \end{aligned}$$

where

$$M_L = (M_{L_1}|M_{L_2}|\cdots|M_{L_k}).$$

Precisely, if we divide the columns of matrix M_L into k equal length blocks, then the structure matrix of L_i is the ith block of M_L. That is,

$$L_i(A_2, \ldots, A_n) = M_{L_i}A_2 \cdots A_n.$$

Using Proposition 5.4 we can get the logical expression of L recursively. We give an example to describe this.

Example 5.5. Let L be a logical variable, and

$$L = M_LABCD,$$

where $A, B, C, D \in \Delta_3$ and

$$\begin{aligned} M_L = \delta_3[&1\ \ 2\ \ 3\ \ 2\ \ 2\ \ 2\ \ 3\ \ 2\ \ \ 1\ \ 2\ \ 2\ \ 2 \\ &2\ \ 2\ \ 2\ \ 3\ \ 2\ \ 2\ \ 3\ \ 2\ \ \ 1\ \ 3\ \ 2\ \ 1 \\ &3\ \ 2\ \ 1\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ \ 2\ \ 3\ \ 2\ \ 2 \\ &2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ \ 2\ \ 3\ \ 2\ \ 2 \\ &2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 1\ \ 1\ \ \ 1\ \ 2\ \ 2\ \ 2 \\ &3\ \ 3\ \ 3\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ \ 2\ \ 2\ \ 2\ \ 3 \\ &3\ \ 2\ \ 1\ \ 2\ \ 2\ \ 2\ \ 1\ \ 2\ \ 3], \end{aligned} \tag{55}$$

Then

$$\begin{aligned} M_L = &[\nabla_1(A) \wedge L_1(B, C, D)] \\ &\vee [\nabla_2(A) \wedge L_2(B, C, D)] \\ &\vee [\nabla_3(A) \wedge L_3(B, C, D)], \end{aligned} \tag{56}$$

and

$$\begin{aligned} M_{L_1} = \delta_3[&1\ \ 2\ \ 3\ \ 2\ \ 2\ \ 2\ \ \ 3\ \ 2\ \ 1\ \ 2 \\ &2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 3\ \ \ 2\ \ 2\ \ 3\ \ 2 \\ &1\ \ 3\ \ 2\ \ 1\ \ 3\ \ 2\ \ 1], \end{aligned} \tag{57}$$

$$\begin{aligned} M_{L_2} = \delta_3[&2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ \ 3\ \ 2\ \ 2\ \ 2 \\ &2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ \ 2\ \ 2\ \ 3\ \ 2 \\ &2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2], \end{aligned} \tag{58}$$

$$M_{L_3} = \delta_3[\begin{matrix}1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 2\\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 2\\ 1 & 2 & 2 & 2 & 1 & 2 & 3\end{matrix}]. \tag{59}$$

Next, consider L_1

$$\begin{aligned} L_1(B,C,D) &= M_{L_1}BCD \\ &= [\nabla_1(B) \wedge L_{11}(C,D)] \vee [\nabla_2(B) \\ &\quad \wedge L_{12}(C,D) \vee [\nabla_3(B) \wedge L_{13}(C,D)], \end{aligned} \tag{60}$$

where

$$\begin{aligned} M_{L_{11}} &= \delta_3[1 \quad 2 \quad 3 \quad 2 \quad 2 \quad 2 \quad 3 \quad 2 \quad 1], \\ M_{L_{12}} &= \delta_3[2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 3 \quad 2 \quad 2], \\ M_{L_{13}} &= \delta_3[3 \quad 2 \quad 1 \quad 3 \quad 2 \quad 1 \quad 3 \quad 2 \quad 1]. \end{aligned}$$

Hence, we have

$$\begin{aligned} L_{11}(C,D) &= C \leftrightarrow D, \\ L_{12}(C,D) &= M_{L_{12}}CD, \\ L_{13}(C,D) &= \neg D. \end{aligned}$$

In the same way, we have the following expression.

$$\begin{aligned} L_2(B,C,D) &= [\nabla_1(B) \wedge L_{21}(C,D)] \vee [\nabla_2(B) \\ &\quad \wedge L_{22}(C,D) \vee [\nabla_3(B) \wedge L_{23}(C,D)], \end{aligned} \tag{61}$$

$$\begin{aligned} L_3(B,C,D) &= [\nabla_1(B) \wedge L_{31}(C,D)] \vee [\nabla_2(B) \\ &\quad \wedge L_{32}(C,D) \vee [\nabla_3(B) \wedge L_{33}(C,D)]. \end{aligned} \tag{62}$$

Putting all together, we have

$$\begin{aligned} L = {} & [\nabla_1(A) \wedge [\nabla_1(B) \wedge L_{11}(C,D)] \\ & \vee [\nabla_2(B) \wedge L_{12}(C,D)] \vee [\nabla_3(B) \wedge L_{13}(C,D)]] \\ & \vee [\nabla_2(A) \wedge \nabla_1(B) \wedge L_{21}(C,D)] \\ & \vee [\nabla_2(B) \wedge L_{22}(C,D)] \vee [\nabla_3(B) \wedge L_{23}(C,D)]] \\ & \vee [\nabla_3(A) \wedge \nabla_1(B) \wedge L_{31}(C,D)] \\ & \vee [\nabla_2(B) \wedge L_{32}(C,D)] \vee [\nabla_3(B) \wedge L_{33}(C,D)]] \end{aligned} \tag{63}$$

Remark 5.6. We can also write down the split form of all binary operators. For instance,

$$\begin{aligned} L_{12}(C,D) &= \delta_3[2\ 2\ 2\ 2\ 2\ 2\ 3\ 2\ 2]CD, \\ &= [\nabla_1(C) \wedge \delta_3[2\ 2\ 2]D] \\ &\quad \vee [\nabla_2(C) \wedge \delta_3[2\ 2\ 2]D] \\ &\quad \vee [\nabla_3(C) \wedge \delta_3[3\ 2\ 2]D], \\ &= [\nabla_1(C) \wedge \delta_3^2] \vee [\nabla_2(C) \wedge \delta_3^2] \\ &\quad \vee [\nabla_3(C) \wedge \psi(D)], \end{aligned} \tag{64}$$

where the structure matrix of the anary logical operator ψ is $\delta_3[3 \quad 2 \quad 2]$.

6. *k*-valued Logical Control Networks

A k-valued logical control network is defined as

$$\begin{cases} A_1(t+1) = f_1(A_1(t), \ldots, A_n(t), \\ \qquad u_1(t), \ldots, u_m(t)), \\ A_2(t+1) = f_2(A_1(t), \ldots, A_n(t), \\ \qquad u_1(t), \ldots, u_m(t)), \\ \vdots \\ A_n(t+1) = f_n(A_1(t), \ldots, A_n(t), \\ \qquad u_1(t), \ldots, u_m(t)); \end{cases} \tag{65}$$

and

$$y_j(t) = h_j(A_1(t), A_2(t), \ldots, A_n(t)), \quad j = 1, 2, \ldots, p. \tag{66}$$

where $A_i(t) \in D_k$, $u_i \in D_k$, f_i, $i = 1, 2, \ldots n$, h_j, $j = 1, 2, \ldots p$ are k-valued logical functions; y_j, $j = 1, 2, \ldots p$ are outputs; u_i, $i = 1, 2, \ldots m$ are controls. We may either assume u_i are logical variables satisfying certain logical rule, say

$$\begin{cases} u_1(t+1) = g_1(u_1(t), \ldots, u_m(t)), \\ u_2(t+1) = g_2(u_1(t), \ldots, u_m(t)), \\ \vdots \\ u_m(t+1) = g_m(u_1(t), u_2(t), \ldots, u_m(t)), \end{cases} \tag{67}$$

or assume $\{u_i\}$ are a sequence of k-valued variables.

In the first case, system (65)–(67) can also be expressed into an algebraic form as

$$\begin{cases} u(t+1) = Gu(t), \quad u \in \Delta_k^m, \\ x(t+1) = Lu(t)x(t)x(t), \quad x \in \Delta_k^n, \\ y(t) = Hx(t), \quad y \in \Delta_k^p. \end{cases} \tag{68}$$

We give an example.

Example 6.1. Consider the following system

$$\begin{cases} A(t+1) = u(t) \to B(t), \\ B(t+1) = A(t) \wedge C(t), \\ C(t+1) = \neg A(t), \\ y(t) = \neg B(t), \end{cases} \tag{69}$$

with input network

$$u(t+1) := u(t).$$

We set $x(t) = A(t)B(t)C(t)$, then

$$\begin{aligned} x(t+1) &= A(t+1)B(t+1)C(t+1) \\ &= M_i u(t)B(t)M_c A(t)C(t)M_n A(t) \\ &:= Lu(t)x(x). \end{aligned}$$

Then L can be calculated easily as

$$\begin{array}{rcccccccccc} L = \delta_{27}[& 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 \\ & 5 & 5 & 8 & 14 & 14 & 17 & 23 & 23 & 26 \\ & 7 & 7 & 7 & 16 & 16 & 16 & 25 & 25 & 25 \\ & 3 & 6 & 9 & 12 & 15 & 18 & 12 & 15 & 18 \\ & 5 & 5 & 8 & 14 & 14 & 17 & 14 & 14 & 17 \\ & 7 & 7 & 7 & 16 & 16 & 16 & 16 & 16 & 16 \\ & 3 & 6 & 9 & 3 & 6 & 9 & 3 & 6 & 9 \\ & 5 & 5 & 8 & 5 & 5 & 8 & 5 & 5 & 8 \\ & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7]. \end{array}$$

Note that

$$y(t) = M_n B(t) = M_n E_d^2 (I_3 \otimes W_{[3]})x(t) := Hx(t),$$

then

$$\begin{array}{rcccccccccccc} H = \delta_3[& 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 3 & 3 & 3 \\ & 2 & 2 & 2 & 1 & 1 & 1 & 3 & 3 & 3 & 2 & 2 & 2 \\ & 1 & 1 & 1]. & & & & & & & & & \end{array}$$

Finally, the algebraic form of (69) is obtained as

$$\begin{cases} u(t+1) = u(t), \\ x(t+1) = Lu(t)x(t), \\ y(t) = Hx(t). \end{cases} \tag{70}$$

Next, we consider the controllability of (65).

Definition 6.2. Consider a k-valued network (65) with control (67). Given the initial state $x(0) = x_0$ and destination state x_d. x_d is said to be reachable from x_0 (in 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 some $s \geq 1$).

We use $\Theta^G(t, 0)$ to denote the input-state transfer matrix in k-valued network, which can be calculated as

$$\begin{aligned} \Theta^G(t,0) = {} & LG^{t-1}(I_{k^m} \otimes LG^{t-2})(I_{k^{2m}} \otimes LG^{t-3}) \cdots \\ & (I_{k^{(t-1)m}} \otimes L)(I_{k^{(t-2)m}} \otimes \Phi_m) \cdots \\ & (I_{k^m} \otimes \Phi_m)\Phi_m, \end{aligned} \tag{71}$$

where Φ_m is defined in Lemma 3.4 as

$$\Phi_m = \prod_{i=1}^{m} I_{k^{i-1}} \otimes [(I_k \otimes W_{[k,k^{m-i}]})M_{r_k}].$$

Then, it is easy to prove that for system (68),

$$x(t) = \Theta^G(t, 0)u(0)x(0). \tag{72}$$

In fact, Eq. (72) provides a tool for investigating the control problems. To avoid a similar argument, we discuss only the following two cases:

Case 1. Fixed s and fixed G.

From the definition of transfer matrix, the following result is obvious.

Theorem 6.3. *Consider system (65) with control (67), where G is fixed. x_d is s-step reachable from x_0, iff*

$$x_d \in Col\left\{\Theta^G(s, 0)W_{[k^n,k^m]}x_0\right\}, \tag{73}$$

where Col means the set of columns.

We give an example to describe this result.

Example 6.4. Consider the following system

$$\begin{cases} A(t+1) = B(t) \leftrightarrow C(t), \\ B(t+1) = C(t) \vee u_1(t), \\ C(t+1) = A(t) \wedge u_2(t); \end{cases} \tag{74}$$

with controls satisfying

$$\begin{cases} u(t+1) = g_1(u_1(t), u_2(t)), \\ v(t+1) = g_2(u_1(t), u_2(t)). \end{cases} \tag{75}$$

Assume g_1 and g_2 are fixed as

$$\begin{cases} g_1(u_1(t), u_2(t)) = \neg u_2(t), \\ g_2(u_1(t), u_2(t)) = u_1(t). \end{cases} \tag{76}$$

Assume $A(0) = 0.5$, $B(0) = 0$, and $C(0) = 0.5$ and $s = 5$. Denote by $u(t) = u_1(t)u_2(t)$, then

$$u(t+1) = M_n u_2(t)u_1(t) = M_n W_{[3]}u(t).$$

Then

$$G = M_n W_{[3]} = \delta_9[7 \quad 4 \quad 1 \quad 8 \quad 5 \quad 2 \quad 9 \quad 6 \quad 3].$$

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

where

$$L = \delta_{27}[\begin{array}{cccccccccccccc}
1 & 10 & 19 & 10 & 10 & 10 & 19 & 10 & 1 & 2 & 11 & 20 & 11 \\
11 & 11 & 20 & 11 & 2 & 3 & 12 & 21 & 12 & 12 & 12 & 21 & 12 \\
3 & 2 & 11 & 20 & 11 & 11 & 11 & 20 & 11 & 2 & 2 & 11 & 20 \\
11 & 11 & 11 & 20 & 11 & 2 & 3 & 12 & 21 & 12 & 12 & 12 & 21 \\
12 & 3 & 3 & 12 & 21 & 12 & 12 & 12 & 21 & 12 & 3 & 3 & 12 \\
21 & 12 & 12 & 12 & 21 & 12 & 3 & 3 & 12 & 21 & 12 & 12 & 12 \\
21 & 12 & 3 & 1 & 13 & 22 & 10 & 13 & 13 & 19 & 13 & 4 & 2 \\
14 & 23 & 11 & 14 & 14 & 20 & 14 & 5 & 3 & 15 & 24 & 12 & 15 \\
15 & 21 & 15 & 6 & 2 & 14 & 23 & 11 & 14 & 14 & 20 & 14 & 5 \\
2 & 14 & 23 & 11 & 14 & 14 & 20 & 14 & 5 & 3 & 15 & 24 & 12 \\
15 & 15 & 21 & 15 & 6 & 3 & 15 & 24 & 12 & 15 & 15 & 21 & 15 \\
6 & 3 & 15 & 24 & 12 & 15 & 15 & 21 & 15 & 6 & 3 & 15 & 24 \\
12 & 15 & 15 & 21 & 15 & 6 & 1 & 13 & 25 & 10 & 13 & 16 & 19 \\
13 & 7 & 2 & 14 & 26 & 11 & 14 & 17 & 20 & 14 & 8 & 3 & 15 \\
27 & 12 & 15 & 18 & 21 & 15 & 9 & 2 & 14 & 26 & 11 & 14 & 17 \\
20 & 14 & 8 & 2 & 14 & 26 & 11 & 14 & 17 & 20 & 14 & 8 & 3 \\
15 & 27 & 12 & 15 & 18 & 21 & 15 & 9 & 3 & 15 & 27 & 12 & 15 \\
18 & 21 & 15 & 9 & 3 & 15 & 27 & 12 & 15 & 18 & 21 & 15 & 9 \\
3 & 15 & 27 & 12 & 15 & 18 & 21 & 15 & 9]. & & & &
\end{array}$$

$$\Phi_2 = (I_3 \otimes W_{[3]}) M_{r_3} (I_3 \otimes M_{r_3}) = \delta_{81}[1, 11, 21, 31, 41, 51, 61, 71, 81].$$

Finally, using formula (71) yields $\Theta(5,0)$ as

$$\Theta(5,0) = LG^4 (I_{3^2} \otimes LG^3)(I_{3^4} \otimes LG^2)(I_{3^6} \otimes LG)(I_{3^8} \otimes L)(I_{3^6} \otimes \Phi_3)(I_{3^4} \otimes \Phi_3)(I_{3^2} \otimes \Phi_3)(I_3 \otimes \Phi_3)\Phi_3. \tag{77}$$

It is calculated as

$$\delta_{27}[\begin{array}{cccccccccccccc}
21 & 20 & 19 & 20 & 20 & 20 & 19 & 20 & 21 & 21 & 20 & 19 & 20 & 20 \\
20 & 19 & 20 & 21 & 21 & 20 & 19 & 20 & 20 & 20 & 19 & 20 & 21 & 11 \\
11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 11 \\
11 & 11 & 11 & 12 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 12 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 14 & 14 & 23 \\
14 & 14 & 14 & 23 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 \\
14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 \\
14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 \\
14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 15 & 15 & 15 & 15 & 15 \\
15 & 6 & 15 & 15 & 15 & 15 & 15 & 15 & 15 & 15 & 6 & 15 & 15 & 15 \\
15 & 15 & 15 & 15 & 15 & 6 & 15 & 15 & 27 & 27 & 27 & 27 & 27 & 27
\end{array}$$

$$\begin{array}{cccccccccccccc} 27 & 27 & 27 & 15 & 15 & 15 & 15 & 15 & 15 & 15 & 15 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 & 3 & 3 & 14 & 14 & 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 & 15 & 14 & 14 \\ 15 & 14 & 14 & 15 & 14 & 14 & 9 & 18 & 27 & 9 & 18 & 27 & 9 & 18 \\ 27 & 9 & 18 & 27 & 9 & 18 & 27 & 9 & 18 & 27 & 9 & 18 & 27 & 9 \\ 18 & 27 & 9 & 18 & 27]. \end{array}$$

Now let $(A(0), B(0), C(0)) = (0.5, 1, 1)$. Then

$$\begin{aligned} x_0 = \delta_{27}[0 \quad & 0 \quad 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \\ & 0 \quad 0 \quad 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\ & 0 \ 0]^T. \end{aligned}$$

Using Theorem 6.3, we have the reachable set as

$$\begin{aligned} &\Theta(5,0)W_{[27,9]}x_0 \\ &\quad = \delta_{27}[21 \ \ 11 \ \ 3 \ \ 14 \ \ 14 \ \ 15 \ \ 15 \ \ 14 \ \ 9]. \end{aligned}$$

We conclude that the reachable set at step 5 is

$$\left\{\delta_{27}^{21}, \delta_{27}^{11}, \delta_{27}^{3}, \delta_{27}^{14}, \delta_{27}^{15}, \delta_{27}^{9}\right\}.$$

Converting them to ternary form, we have

$$\begin{aligned} (A(5), B(5), C(5)) \in \{&(0,1,0), (0.5,1,0.5), (1,1,0), \\ &(0.5,0.5,0.5), (0.5,0.5,0), \\ &(1,0,0)\}. \end{aligned}$$

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

$$\begin{aligned} x_d &= \Theta(5,0)W_{[27,9]}x_0u_0 \\ &= \delta_{27}[21 \ \ 11 \ \ 3 \ \ 14 \ \ 14 \ \ 15 \ \ 15 \ \ 14 \ \ 9]u_0, \end{aligned}$$

it is obvious that to reach, say, $\delta_{27}^{21} \sim (0,1,0)$, the control should be $u_0 = [1,0,0,0,0,0,0,0,0]^T$, i.e. $u_1(0) = \delta_3^1 \sim 1$ and $u_2(0) = \delta_3^1 \sim 1$. Similarly, to reach all the six points $(0,1,0)$, $(0.5,1,0.5)$, $(1,1,0)$, $(0.5,0.5,0.5)$, $(0.5,0.5,0)$, $(1,0,0)$ at step 5, the corresponding control initials are given in the Table 4.

Remark 6.5. The $\Theta^G(s,0)$ can be calculated inductively, and the algorithm is similar to the one in [Cheng & Qi, 2009a].

Case 2. Fixed s and constrained G.

It is easy to estimate that there are $m_0 = (k^m)^{k^m}$ possible distinct $G's$. For statement ease, we may express each G in a condensed form and order them in "increasing order". Say, when $m = 2, k = 3$ we have $G_1 = \delta_9[1,1,1,1,1,1,1,1,1]$, $G_2 = \delta_9[1,1,1,1,1,1,1,1,2], \ldots, G_{9^9} = \delta_9[9,9,9,9,9,9,9,9,9]$. In general, we may consider a subset $\Lambda \subset \{1,2,\ldots,m_0\}$, and allow G to be chosen from an admissible set: $\{G_\lambda | \lambda \in \Lambda\}$.

Corollary 6.6. *Consider system* (*65*) *with control* (67), *where* $G \in \{G_\lambda | \lambda \in \Lambda\}$. *Then* x_d *is reachable from* x_0, *iff*

$$x_d \in \bigcup_{\lambda \in \Lambda} Col\left\{\Theta^{G_\lambda}(s,0)W_{[k^n,k^m]}x_0\right\}. \qquad (78)$$

Example 6.7. Consider the system (71) again, and also assume $k = 3$. We still assume $A(0) = 1$, $B(0) = 0$, and $C(0) = 1$ and $s = 5$. Assume $\Xi = \{G_1, G_2, G_3, G_4\}$, where $G_1 = \delta_9[1,2,3,4,5,6,7,8,9]$, $G_2 = \delta_9[1,5,8,9,7,4,6,3,2]$, $G_3 = \delta_9[1,8,9,6,5,7,3,2,4]$, $G_4 = \delta_9[9,8,5,6,4,2,3,1,7]$, the corresponding $V_i = \mathrm{Col}\{\Theta^i(5,0)W_{[3^n,3^m]}x_0\}$ are

$$\begin{aligned} &\delta_{27}[2 \ \ 11 \ \ 21 \ \ 14 \ \ 14 \ \ 15 \ \ 14 \ \ 14 \ \ 9], \\ &\delta_{27}[2 \ \ 14 \ \ 12 \ \ 14 \ \ 15 \ \ 15 \ \ 11 \ \ 26 \ \ 15], \\ &\delta_{27}[2 \ \ 11 \ \ 17 \ \ 27 \ \ 14 \ \ 6 \ \ 15 \ \ 14 \ \ 12], \\ &\delta_{27}[23 \ \ 17 \ \ 11 \ \ 11 \ \ 15 \ \ 15 \ \ 15 \ \ 21 \ \ 15]. \end{aligned}$$

Table 4. Destinations and the corresponding controls, $x(0) = (0.5, 1, 1)$.

x_d	u(0)	$u_1(0)$	$u_2(0)$
$(0,1,0)$	$\delta_9^1 \sim (0,1,0)$	$\delta_3^1 \sim 1$	$\delta_3^1 \sim 1$
$(0.5,1,0.5)$	$\delta_9^2 \sim (0,1,0)$	$\delta_3^1 \sim 1$	$\delta_3^1 \sim 0.5$
$(1,1,0)$	$\delta_9^3 \sim (0,1,0)$	$\delta_3^1 \sim 1$	$\delta_3^1 \sim 0$
	$\delta_9^4 \sim (0,1,0)$	$\delta_3^2 \sim 0.5$	$\delta_3^1 \sim 1$
$(0.5,0.5,0.5)$	$\delta_9^5 \sim (0,1,0)$	$\delta_3^2 \sim 0.5$	$\delta_3^2 \sim 0.5$
	$\delta_9^8 \sim (0,1,0)$	$\delta_3^3 \sim 0$	$\delta_3^2 \sim 0.5$
$(0.5,0.5,0)$	$\delta_9^6 \sim (0,1,0)$	$\delta_3^2 \sim 0.5$	$\delta_3^3 \sim 0$
	$\delta_9^7 \sim (0,1,0)$	$\delta_3^3 \sim 0$	$\delta_3^1 \sim 1$
$(1,0,0)$	$\delta_9^9 \sim (0,0,0)$	$\delta_3^3 \sim 0$	$\delta_3^3 \sim 0$

So the reachable set at five steps is

$$\bigcup_{i=1}^{4} V_i = \{\delta_{27}^2, \delta_{27}^6, \delta_{27}^9, \delta_{27}^{11}, \delta_{27}^{12}, \delta_{27}^{14}, \delta_{27}^{15}, \delta_{27}^{17}, \delta_{27}^{21}, \delta_{27}^{23}, \delta_{27}^{26}, \delta_{27}^{27}\}$$

Now assume we want to reach $(A(5), B(5), C(5)) = (0.5, 1, 0)$, which is δ_{27}^{12}. The 3rd component of V_2 is 12. (We have some other choices such as the 9th component of V_3.) So we can choose G_2 and $u(0) = u_1(0)u_2(0) = \delta_9^3$ to drive $(0.5, 0, 0.5)$ to $(0.5, 1, 0)$ at five steps.

We can reconstruct the control dynamics from the logical matrix, G_2. Converting $G_2 = [1, 5, 8, 9, 7, 4, 6, 3, 2]$ back to standard form, we have

$$G_2 = \delta_9[1 \quad 5 \quad 8 \quad 9 \quad 7 \quad 4 \quad 6 \quad 3 \quad 2].$$

From $u_1(0)u_2(0) = \delta_9^3$, we have $u_1(0) = \delta_3^1$ and $u_2(0) = \delta_3^3$.

To reconstruct control dynamics, we need retrievers

$$S_1 = \delta_3[1 \quad 1 \quad 1 \quad 2 \quad 2 \quad 2 \quad 3 \quad 3 \quad 3],$$
$$S_2 = \delta_3[1 \quad 2 \quad 3 \quad 1 \quad 2 \quad 3 \quad 1 \quad 2 \quad 3].$$

Then, we have the structure matrixes as

$$M_1 = S_1 G = \delta_3[1 \quad 2 \quad 3 \quad 3 \quad 3 \quad 2 \quad 2 \quad 1 \quad 1],$$
$$M_2 = S_2 G = \delta_3[1 \quad 2 \quad 2 \quad 3 \quad 1 \quad 1 \quad 3 \quad 3 \quad 1].$$

It follows that

$$u_1(t+1) = M_1 u_1(t) u_2(t),$$
$$u_2(t+1) = M_2 u_1(t) u_2(t).$$

Similar to [Cheng & Qi, 2009a], some other cases can also be investigated.

Next, we consider the controllability via the control of k-valued sequence. We give the following definition.

Definition 6.8 [Akutsu *et al.*, 2007]. Consider k-valued logical system (65), assume an initial state of the network A_I^i, $i = 1, \ldots, n$ and a desired state of the network A_D^i, $i = 1, \ldots, n$ at the sth time step are given. Then the problem is to find a sequence of δ_k^i vectors $u(0), \ldots, u(s-1)$ such that $A_i(0) = A_I^i$, $A_i(s) = A_D^i$, $i = 1, \ldots, n$.

Define $\tilde{L} = LW_{[k^n, k^m]}$, then the second equation in (68) can be expressed as

$$x(t+1) = \tilde{L} x(t) u(t). \tag{79}$$

Using it repetitively yields

$$x(s) = \tilde{L}^s x(0) u(0) u(1) \cdots u(s-1). \tag{80}$$

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

Theorem 6.9. *A_D^i is reachable from $A_I^i, i = 1, \ldots, n$ at sth time step by controls of k-valued sequences of length s, iff*

$$x_s \in \mathrm{Col}\{\tilde{L}^s x_0\}, \tag{81}$$

where $x_s = \ltimes_{i=1}^n A_D^i, x_0 = \ltimes_{i=1}^n A_I^i$.

Remark 6.10. Note that (81) means x_s is equal to a column of $\tilde{L}^s x_0$. Say, x_s is equal to the kth column of $\tilde{L}^s x_0$, then the controls should be

$$u(0)u(1)\cdots u(s-1) = \delta_{m^s}^k, \tag{82}$$

which uniquely determines all $u_i, i = 0, 1, \ldots, s-1$.

The following example is taken from [Akutsu *et al.*, 2007], but here we allow the values of the notes in the network to be three different values $\{0, 0.5, 1\}$.

Example 6.11. Consider a 3-valued logical control network depicted in Fig. 3.

Its logical equation is

$$\begin{cases} A(t+1) = C(t) \wedge u_1(t), \\ B(t+1) = \neg u_2(t), \\ C(t+1) = A(t) \vee B(t). \end{cases} \tag{83}$$

Its algebraic form is

$$\begin{cases} A(t+1) = M_c C(t) u_1(t), \\ B(t+1) = M_n u_2(t), \\ C(t+1) = M_d A(t) B(t). \end{cases} \tag{84}$$

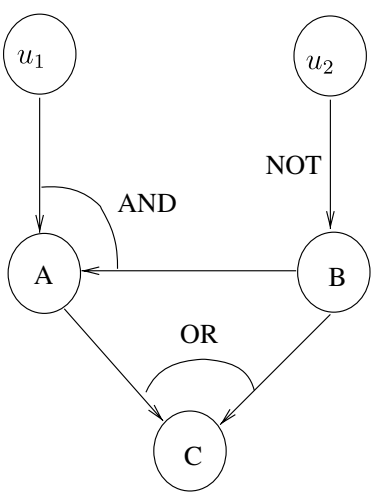

Fig. 3. A 3-valued control network.

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

$$x(t+1) = \tilde{L}x(t)u(t), \tag{85}$$

where

$$\tilde{L} = \delta_{27}[\ 7\ \ 4\ \ 1\ \ 16\ \ 13\ \ 10\ \ 25\ \ 22\ \ 19\ \ 16\ \ 13\ \ 10\ \ 16\ \ 13$$
$$10\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 7$$
$$4\ \ 1\ \ 16\ \ 13\ \ 10\ \ 25\ \ 22\ \ 19\ \ 16\ \ 13\ \ 10\ \ 16\ \ 13\ \ 10$$
$$25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 7\ \ 4$$
$$1\ \ 16\ \ 13\ \ 10\ \ 25\ \ 22\ \ 19\ \ 16\ \ 13\ \ 10\ \ 16\ \ 13\ \ 10\ \ 25$$
$$22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 7\ \ 4\ \ 1$$
$$16\ \ 13\ \ 10\ \ 25\ \ 22\ \ 19\ \ 16\ \ 13\ \ 10\ \ 16\ \ 13\ \ 10\ \ 25\ \ 22$$
$$19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 8\ \ 5\ \ 2\ \ 17$$
$$14\ \ 11\ \ 26\ \ 23\ \ 20\ \ 17\ \ 14\ \ 11\ \ 17\ \ 14\ \ 11\ \ 26\ \ 23\ \ 20$$
$$26\ \ 23\ \ 20\ \ 26\ \ 23\ \ 20\ \ 26\ \ 23\ \ 20\ \ 8\ \ 5\ \ 2\ \ 17\ \ 14$$
$$11\ \ 26\ \ 23\ \ 20\ \ 17\ \ 14\ \ 11\ \ 17\ \ 14\ \ 11\ \ 26\ \ 23\ \ 20\ \ 26$$
$$23\ \ 20\ \ 26\ \ 23\ \ 20\ \ 26\ \ 23\ \ 20\ \ 7\ \ 4\ \ 1\ \ 16\ \ 13\ \ 10$$
$$25\ \ 22\ \ 19\ \ 16\ \ 13\ \ 10\ \ 16\ \ 13\ \ 10\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22$$
$$19\ \ 25\ \ 22\ \ 19\ \ 25\ \ 22\ \ 19\ \ 8\ \ 5\ \ 2\ \ 17\ \ 14\ \ 11\ \ 26$$
$$23\ \ 20\ \ 17\ \ 14\ \ 11\ \ 17\ \ 14\ \ 11\ \ 26\ \ 23\ \ 20\ \ 26\ \ 23\ \ 20$$
$$26\ \ 23\ \ 20\ \ 26\ \ 23\ \ 20\ \ 9\ \ 6\ \ 3\ \ 18\ \ 15\ \ 12\ \ 27\ \ 24$$
$$21\ \ 18\ \ 15\ \ 12\ \ 18\ \ 15\ \ 12\ \ 27\ \ 24\ \ 21\ \ 27\ \ 24\ \ 21\ \ 27$$
$$24\ \ 21\ \ 27\ \ 24\ \ 21].$$

Now we assume $(A(0), B(0), C(0)) = (0, 0, 0)$. We wish to know if a designed state can be reached by the sth step. Say, $s = 3$, using Theorem 6.9, we calculate $\tilde{L}^3 x_0 \in M_{3^3 \times 3^6}$ as follows. (It is too large and we cite only a few columns.).

$$\delta_{27}[27\ \ 24\ \ 21\ \ 27\ \ 24\ \ 21\ \ 27\ \ 24\ \ 21\ \ 26\ \ 23\ \ 20\ \ 26\ \ 23\ \ 20$$
$$26\ \ 23\ \ 20\ \ 25\ \ 22\ \ 19\ \ 27\ \ 24\ \ 21\ \ 27\ \ 24\ \ 21\ \ 27\ \ 24\ \ 17$$
$$\vdots$$
$$14\ \ 11\ \ 26\ \ 23\ \ 20\ \ 7\ \ 4\ \ 1\ \ 16\ \ 13\ \ 10\ \ 25\ \ 22\ \ 19].$$

It is clear that at third step all states can be reached (via computer searching). Choose one state, say $\delta_{27}^{25} \sim (0, 0, 1)$. The program reported that in 19th, 22nd, 25th, ... columns we have 25, which means controls δ_{729}^{19}, or δ_{729}^{22}, or δ_{729}^{25}, or $\cdots$ can drive the initial state $(0, 0, 0)$ to the destination state $(0, 0, 1)$. we choose, for example,

$$u_1(0)u_2(0)u_1(1)u_2(1)u_1(2)u_2(2) = \delta_{729}^{19}.$$

Converting $729 - 19 = 710$ to ternary form yields $(1, 1, 1, 0, 1, 1)$, which means the corresponding controls are $u_1(0) = 1, u_2(0) = 1$; $u_1(1) = 1, u_2(1) = 0; u_1(2) = 1, u_2(2) = 1$. It is easy to check directly that this set of controls work. We may check some others. Say, choosing δ_{729}^{22}, and converting $729 - 22 = 707$ to ternary form as $(1, 1, 1, 0, 0.5, 1)$, we have $u_1(0) = 1, u_2(0) = 1$; $u_1(1) = 1, u_2(1) = 0; u_1(2) = 0.5, u_2(2) = 1$. It also works.

In general, it is easy to calculate that when $s = 1$ the reachable set from $(0, 0, 0)$ is

$$\{(0, 0, 0), (0, 0.5, 0), (0, 1, 0)\}.$$

when $s = 2$ the reachable set is

$$\{(0, 0, 0), (0, 0.5, 0), (0, 1, 0), (0, 0, 0.5), (0, 0.5, 0.5),$$
$$(0, 1, 0.5), (0, 0.5, 1), (0, 0, 1), (0, 1, 1)\}.$$

7. Conclusion

In this paper, we first review some recent developments of the semi-tensor product approach to Boolean networks, and then provide a framework

to study the dynamics of multivalued networks. For a k-valued network, the network equation was converted into an algebraic form as a standard discrete-time linear system. By analyzing the network transition matrix, easily computable formulas were obtained to show (a) the number of fixed points; (b) the numbers of cycles of different lengths; (c) transient period, i.e. the minimum time for all points to enter the set of attractors. Then formulas were obtained to recover the network and its dynamics from its transition matrix L. Finally, the controllability of k-valued logical control network via two kinds of controls has been discussed. Formulas were obtained to construct the reachable set.

Several examples were included to illustrate the results. The computations involved seem complicated, but they can be easily performed with a computer.

Acknowledgments

A toolbox for all the related computations is available at http://lsc.amss.ac. cn/dcheng/.

This work was supported in part by the National Natural Science Foundation (NNSF) of China under Grants 60674022, 60736022 and 60821091.

References

Adamatzky, A. [2003] "On dynamically non-trivial three-valued logics: Oscillatory and bifurcatory species," *Chaos Solit. Fract.* **18**, 917–936.

Aerts, D., Gershenson, C. & Broekaert, J. [2003] "Contextual random Boolean networks," *Advances in Artificial Life, 7th European Conf.* (Springer-Verlag).

Akutsu, T., Hayashida, M., Ching, W. & Ng, M. K. [2007] "Control of Boolean networks: Hardness results and algorithms for tree structured networks," *J. Theoret. Biol.* **244**, 670–679.

Aldana, M., Coppersmith, S. & Kadanofff, L. P. [2003] "Booleean dynamics with random couplings," *Perspectives and Problems in Nonlinear Science*, eds. Kaplan, E., Marsden, J. E. & Sreenivasan, K. R. (Springer, New York).

Barnes, D. W. [1975] *An Algebraic Introduction to Mathematical Logic* (Springer-Verlag, New York).

Cheng, D. [2002] *Matrix and Polynomial Approach to Dynamic Control Systems* (Science Press, Beijing).

Cheng, D. [2007] "Semi-tensor product of matrices and its applications — a survey," in *ICCM*, Vol. 3, pp. 641–668.

Cheng, D. & Qi, H. [2007] *Semi-tensor Product of Matrices — Theory and Applications* (Science Press, Beijing).

Cheng, D. [2009] "Input-state approach to boolean networks," *IEEE Trans. Neural Networks* **20**, 512–521.

Cheng, D. & Qi, H. [2009a] "Linear representation of dynamics of boolean," *IEEE Trans. Autom. Cont.*

Cheng, D. & Qi, H. [2009b] "Controllability and observability of Boolean control networks," *Automatica* **45**, 1659–1667.

Farrow, C., Heidel, J., Maloney, H. & Rogers, J. [2004] "Scalar equations for synchronous boolean networks with biological applications," *IEEE Trans. Neural Networks* **15**, 348–354.

Gershenson, C. [2002] "Classification of random Boolean networks," *Artificial Life VIII: Proc. Eight Int. Conf. Artificial Life*, eds. Standish, R. K., Bedau, M. A. & Abbass, H. A. (MIT Press).

Gershenson, C. "Phase transitions in random boolean networks with different updating schemes," arXiv:nlin/0311008.

Gershenson, C. [2004] "Updating schemes in random boolean networks: Do they really matter?" *Artificial Life IX, Proc. Ninth Int. Conf. Simulation and Synthesis of Living Systems*, eds. Pollack, J., Bedau, M., Husbands, P., Ikegami, T. & Watson, R. A. (MIT Press), pp. 238–243.

Harvey, I. & Bossomaier, T. [1997] "Time out of joint: Attractors in asynchronous random boolean networks," *Proc. Fourth European Conf. Artificial Life (ECAL97)*, eds. Husbands, P. & Harvey, I. (MIT Press), pp. 67–75.

Heidel, J., Maloney, J., Farrow, J. & Rogers, J. [2003] "Finding cycles in synchronous boolean networks with applications to biochemical systems," *Int. J. Bifurcation and Chaos* **13**, 535–552.

Huang, S. [1999] "Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery," *J. Mol. Med.* **77**, 469–480.

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

Huang, S. [2002] "Regulation of cellular states in mammalian cells from a genomewide view," *Gene Regulation and Metabolism*, eds. Collado-Vodes, J. & Hofestadt, R. (MIT Press, Cambridge, MA), pp. 181–220.

Huberman, B. A. & Glace, N. S. [1993] "Evolutionary games and computer simulations," *Proc. Nat. Acad. Sci.* **90**, 7716–7718.

Ideker, T., Galitski, T. & Hood, L. [2001] "A new approach to decoding life: systems biology," *Ann. Rev. Genom. Hum. Gen.* **2**, 343–372.

Kauffman, S. A. [1969] "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.* **22**, 437–467.

Kauffman, S. A. [1995] *At Home in the Universe* (Oxford University Press).

Kitano, H. [2001] *Foundations of Systems Biology* (MIT Press Cambridge, Massachusetts London).

Luo, C. K. [1992] *The Theory of Multi-valued Logic and Its Application* (Science Press, Beijing).

Magnus, J. R. & Neudecker, H. [1999] *Matrix Differential Calculus with Applications in Statistics and Econometrics* (John Wiley).

Martin, O., Odlyzko, A. M. & Wolfram, S. [1984] "Algebraic propertices of cellular automata," *Commun. Math. Phys.* **93**, 219–258.

May, R. M., Bonhoeffer, S. & Nowak, M. A. [1995] "Spatial games and evolution of cooperation," *Advance in Artificial Life*, eds. Moran, F., Moreno, A., Mereloand, J. J. & Chacon, P., pp. 749–759.

Robert, F. [1986] *Discrete Iterations: A Metric Study* (Springer-Verlag, Berlin Heidelberg).

Shmulevich, I., Dougherty, R., Kim, S. & Zhang, W. [2002a] "Probabilistic boolean neworks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics* **18**, 261–274.

Shmulevich, I., Dougherty, R. & Zhang, W. [2002b] "From boolean to probabilistic boolean neworks as models of gene regulatory networks," *Proc. IEEE* **90**, 1778–1791.

Volker, L. G. & Conrad, M. [1998] "The role of weak interactions in biological systems: The dual dynamic model," *J. Theor. Biol.* **193**, 287–306.

Zhang, S.-Q., Hayashida, M., Akutsu, T., Ching, W. & Ng, M. K. "Algorithms for finding small attractors in boolean networks," *EURASIP J. Bioinformatics and Systems Biology.*