SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2025, Vol. 68, Iss. 9, 192201:1–192201:16
https://doi.org/10.1007/s11432-024-4252-7

On control networks over finite lattices

Zhengping JI[1,2*] & Daizhan CHENG[1]

[1]Key Laboratory of Systems and Control, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

Received 4 August 2024/Revised 11 October 2024/Accepted 3 December 2024/Published online 25 March 2025

Abstract The modeling and control of networks over finite lattices are studied via the algebraic state space approach. Using the semi-tensor product of matrices, we obtain the algebraic state space representation (ASSR) of the dynamics of (control) networks over finite lattices. Basic properties concerning networks over sublattices and product lattices are investigated, which shows the application of the analysis of lattice structure in the model reduction and control design of networks. Then, algorithms are developed to recover the lattice structure from the structure matrix of a network over a lattice, and to construct comparability graphs over a finite set to verify whether a multiple-valued logical network is defined over a lattice. Finally, numerical examples are presented to illustrate the results.

Keywords multiple-valued control networks, semi-tensor product of matrices, finite lattices, system realization, constraint satisfaction problems

Citation Ji Z P, Cheng D Z. On control networks over finite lattices. Sci China Inf Sci, 2025, 68(9): 192201, https://doi.org/10.1007/s11432-024-4252-7

1 Introduction

The Boolean networks were first introduced by Kauffman [1] to model genetic regulatory networks. They were later generalized as multiple-valued logical networks to improve the accuracy of logical variables. The theory of Boolean algebra [2], where the value of a logical node is quantized to zero or one, and the theory of n-valued logic [3] can be unified into general classes of algebras. These notions were generalized previously [4] to the so-called Post algebra, which, by revealing itself as a special class of finite lattices, provided a way of treating multiple-valued algebra from the perspective of lattice theory [5, 6].

Lattices are common objects in combinatorics, and networks over lattices can be commonly found in the modeling and control of communication networks [7, 8], as well as systems biology [9]. Since multiple-valued logical networks can be considered as defined over Post algebras, which are special cases of lattices, one can generally consider a network where each node takes value in a finite lattice, and the evolution depends on the sup and inf operations between the states. These networks are common in various applications. For example, distant data verification and breakdown restoration in multi-agent systems with a logically linked list of entries and distributed ledger can be modeled using Allen-Givone algebra, which can be viewed as a special class of lattices [10, 11]. Meanwhile, the fuzzy bisimulation of nondeterministic transition systems is modeled using residuated lattices [12,13], whereas the asynchronous dynamics of a gene network with multiple expression levels can be modeled using operators defined on a finite lattice [14]. Moreover, if the nodes of networks are defined over lattices such as the finite sets of Eisenstein integers [15], revealing the underlying partial order or dimensions of the lattice will benefit the analysis of the network structure.

On the other hand, networks over lattices have simple logical expressions because of their properties such as the commutativity and idempotency of the generating operators, making them mathematically easier to present. From the underlying algebraic structure of these networks, one would expect that the following properties hold: restricting such a network to a sublattice provides an invariant subnetwork, and the control properties of networks over product lattices are completely determined by their decomposed

* Corresponding author (email: jizhengping@amss.ac.cn)

subnetworks over sublattices. These properties reduce the complexity of analysis and control design of such systems, as we show in this paper.

However, to the authors' knowledge, investigations on general (control) networks over lattices are scarce. Two problems are the possible reasons for this scarcity. First, given the input-output relation of a finite-valued network, we cannot always recognize whether it is defined over a lattice. Second, when a network is indeed expressed through the composition of basic operators on a lattice, the expression of the nonlinear dynamics is not easily simplified for deriving a transition law, impeding the analysis of the network.

We now consider the first problem. Since networks over lattices may encode information of the underlying partial order structure in its algebraic expression, from the dynamics of a given multiple-valued logical network, one may recognize the basic operators serving as "bricks" or "building blocks" that generate the system and hence determine if it is actually defined over a lattice. The problem of verifying whether a network can be considered as one defined over a lattice (and further reconstructing this lattice structure if possible) is of practical importance as it is a special case of the NP-hard constraint satisfaction problems [16]. Expressing a k-valued network over a lattice is not only practically useful but also theoretically challenging. It is well known that even expressing a 3-valued logical network into a 3-valued logical form is difficult in general, and the k-valued case is even more complicated [17]. There have been investigations on the generating systems of function algebras (see, e.g., Part II of [18]), but as far as the authors are concerned, only a few research works have been done for designing a lattice structure on a finite set to make a (control) network be generated by the basic operators on the lattice, which is an important issue as we sometimes only have the desired input-output relation of a system and we would like to realize it using simple generating functions [19]. For example, if the desired transition law of a network can be written explicitly in terms of operators on a lattice, then the system design can be executed on the basis of formal calculations over these operators. This can be viewed as a discrete-state analog of the circuit realization problem [20, 21].

Concerning the second problem, we introduce the semi-tensor product (STP) of matrices as a tool. Since 2009, it has been applied to the study of Boolean networks [22], providing a convenient way to represent the dynamics into algebraic equations and promoting the development of the theory of finite-valued networks. In this respect, investigations have emerged mainly in two directions. One is the control problems of Boolean networks, such as controllability [23], observability [24, 25], stabilization [26], tracking [27], and decoupling [28]. Another direction is extending the values of states in networks from the Boolean case (where they are binary) to k-valued, mixed [29], finite ring [30, 31], and finite field cases [32–34]. The STP enables simplifying the expression of a finite-valued control network to a unified bilinear form and thus the analysis of its control properties. This method has been applied to general finite state machines as well [35]. However, a main obstacle to the application of STP is that its computational complexity increases exponentially with the number of nodes in the networks. There have been several approaches proposed for this problem, such as aggregation [36] and pinning control [37].

This paper aims to provide a framework for (control) networks over finite lattices to solve the above problems. Using STP, the algebraic state space representation (ASSR) of such networks is derived, allowing us to analyze a system with existing results in multiple-valued logical networks. We show that when the lattice is the product of some finite lattices, the control properties are determined by the subnetworks defined over factor lattices. Further, from a network over a lattice, we recover the underlying lattice structure from its ASSR. When the existence of underlying partial order relations is unknown for an arbitrarily given network, we give the necessary conditions for it to be a network over a lattice and try to construct such a partial order to allow the network to be generated by some classes of basic operators over a lattice. Compared with existing methods of analyzing networks over finite rings [30] and finite fields [32, 34], our algorithm, for the first time, provides a method for reconstructing algebraic structures over dynamics, whereas there have been no existing results derived for recovering ring or field operations such as addition and multiplication that builds network dynamics. Meanwhile, our algorithm provides a way to reduce the computational complexity of systems by decomposing networks into subnetworks over factor lattices. Compared with existing methods for model reduction of logical networks, such as aggregation [38], our method has no restrictions on the topological structures of a network and is applicable for general networks over product lattices.

The rest of this paper is organized as follows. Section 2 provides preliminaries about the STP and lattice theory. Section 3 studies networks over finite lattices under the framework of algebraic state space and vector expressions. First, it gives criteria for a finite set endowed with an operator to be a lattice; then, it investigates the basic properties of networks over product lattices. Section 4 is devoted to constructing and recovering the underlying lattice structures for finite-valued networks, designing algorithms to construct partial order so that a given network can be viewed as generated by the basic operators of a lattice. These results are illustrated by numerical examples in Section 5. Section 6 consists of conclusion and further problems.

Before ending this introduction, we give a list of notations used in this sequel:

(1) $\mathcal{M}_{m\times n}$: set of $m\times n$-dimensional real matrices;

(2) $\mathrm{Col}(A)$ ($\mathrm{Row}(A)$): set of columns (rows) of A; $\mathrm{Col}_i(A)$ ($\mathrm{Row}_i(A)$): i-th column (row) of A;

(3) δ_n^i: i-th column of the identity matrix I_n;

(4) $\mathcal{D}_k := \{0,\dots,k-1\}$;

(5) $\Delta_k := \mathrm{Col}(I_k)$;

(6) $\mathcal{L}_{m\times n}$: set of logical matrices (a matrix $L\in\mathcal{M}_{m\times n}$ is called a logical matrix if $\mathrm{Col}(L)\subset\Delta_m$);

(7) $\delta_m[i_1,\dots,i_n]$: brief notation for logical matrices; that is, $\delta_m[i_1,\dots,i_n] := \left[\delta_m^{i_1},\dots,\delta_m^{i_n}\right]$;

(8) $A\times_{\mathcal{B}} B$: Boolean product of $A\in B_{m\times n}$, $B\in B_{n\times p}$, that is, $[A\times_{\mathcal{B}} B]_{i,j}=0$ if $[AB]_{i,j}=0$, and $[A\times_{\mathcal{B}} B]_{i,j}=1$ if $[AB]_{i,j}>0$, $i=1,\dots,m$, $j=1,\dots,p$.

2 Preliminaries

2.1 STP and algebraic expression of multiple-valued networks

We first give a brief review of the STP of matrices, which is the main tool in this paper. We refer to Cheng et al. [39] for details.

Definition 1. Let $A\in\mathcal{M}_{m\times n}$, $B\in\mathcal{M}_{p\times q}$ be real matrices and the least common multiple of n and p be $t=\mathrm{lcm}\{n,p\}$. The STP of A and B, denoted by $A\ltimes B$, is defined as

$$\left(A\otimes I_{t/n}\right)\left(B\otimes I_{t/p}\right), \tag{1}$$

where I_k is the $k\times k$ identity matrix and $\otimes$ is the Kronecker product.

Throughout this paper, all products are assumed to be STPs and the symbol $\ltimes$ is usually omitted.

When the dimensions of two matrices are compatible, the STP is the same as the conventional matrix product, and the laws concerning associativity, distributivity, transpose, and inverse hold for it as well. Further, it has the following properties concerning commutativity.

Proposition 1 ([39])**.** Let $X\in\mathbb{R}^m$ be a column vector and M be a matrix. Then, $X\ltimes M=(I_m\otimes M)\ltimes X$.

Given two column vectors $X\in\mathbb{R}^m$ and $Y\in\mathbb{R}^n$, then

$$W_{[m,n]}\ltimes X\ltimes Y = Y\ltimes X, \tag{2}$$

where $W_{[m,n]}\in\mathcal{M}_{mn\times mn}$ is the (m,n)-order swap matrix defined as

$$W_{[m,n]}=\delta_{mn}[1,m+1,\dots,(n-1)m+1,2,m+2,\dots,(n-1)m+2,\dots,m,2m,\dots,nm].$$

Definition 2. (1) Let $x_i\in\mathcal{D}_{k_i}$, $i=1,\dots,n$. A map $f:\prod_{i=1}^n\mathcal{D}_{k_i}\to\mathcal{D}_{k_0}$ is called a multiple-valued logical function. If $k_1=k_2=\cdots=k_n=k_0$, f is called a k_0-valued logical function.

(2) Let $x_i(t)\in\mathcal{D}_{k_i}$, $f_i:\prod_{i=1}^n\mathcal{D}_{k_i}\to\mathcal{D}_{k_i}$, $i=1,\dots,n$. The system

$$\begin{cases} x_1(t+1)=f_1(x_1(t),\dots,x_n(t)),\\ \quad\vdots\\ x_n(t+1)=f_n(x_1(t),\dots,x_n(t)) \end{cases} \tag{3}$$

is called a multiple-valued logical dynamic system.

If $k_1=k_2=\cdots=k_n=k_0$, the system is called a k_0-valued logical dynamic system.

Identify $i\sim\delta_k^i$, $i=1,\dots,k-1$, and $0\sim\delta_k^k$. Then, $x\in\mathcal{D}_k$ can be expressed as $x\in\Delta_k$. The latter is called the vector expression of a logical variable. Using this and the above properties of STP, one can express a multiple-valued logical function and a network in the algebraic state space.

Theorem 1 ([40]). Given a multiple-valued logical function $f : \prod_{i=1}^{n} \mathcal{D}_{k_i} \to \mathcal{D}_{k_0}$ denoted as $y = f(x_1, \ldots, x_n)$, there exists a unique $M_f \in \mathcal{L}_{k_0 \times k}$, where $k = \prod_{i=1}^{n} k_i$, such that, in vector expression, we have

$$y = M_f \ltimes_{i=1}^{n} x_i.$$

M_f is called the structure matrix of f.

Applying Theorem 1 to each equation of (3), we have

$$\begin{cases} x_1(t+1) = M_1 \ltimes_{i=1}^{n} x_i(t), \\ \quad\quad \vdots \\ x_n(t+1) = M_n \ltimes_{i=1}^{n} x_i(t), \end{cases} \tag{4}$$

where M_i is the structure matrix of f_i, $i = 1, \ldots, n$.

Theorem 2 ([40]). Denote $x(t) = \ltimes_{i=1}^{n} x_i(t)$. Then, Eq. (4) can be expressed as

$$x(t+1) = Mx(t), \tag{5}$$

where $M = M_1 * \cdots * M_n$ and $*$ is the Khatri-Rao product of matrices. Then, Eq. (5) is called the ASSR of the multiple-valued logical network (3), and M is called its structure matrix.

2.2 Finite lattices

Next, we review the basic notions of lattice theory.

A lattice is a partially ordered set $(L, \leqslant)$, where each pair of elements $x, y \in L$ has a least upper bound (denoted by $x \vee y$ or $\sup(x, y)$) and a greatest lower bound (denoted by $x \wedge y$ or $\inf(x, y)$). For example, define a partial order on $\mathbb{N}$ as $a \leqslant b \Leftrightarrow a|b$, and let $a \vee b$ be the least common multiple and $a \wedge b$ the greatest common divisor of two integers a and b; then, $(\mathbb{N}, \leqslant)$ is a lattice. Let $L := \{0, 1\}$; then, L is a lattice under a natural order if we define the operators $\vee$ and $\wedge$ as disjunction and conjunction of the Boolean variables, respectively.

An alternative definition is that a lattice is a tuple $(L, \wedge, \vee)$, where $\wedge, \vee$ are binary operators on L satisfying commutativity, associativity, idempotency, and the absorption property. As only finite lattices are considered in this paper, we adopt Definition 3.

Definition 3 ([41]). A finite set L is called a lattice if there exists a binary operator $\vee$ on L satisfying

(i) $x \vee x = x$;

(ii) $x \vee y = y \vee x$;

(iii) $(x \vee y) \vee z = x \vee (y \vee z)$;

(iv) $\exists \mathbf{0} \in L$, s.t. $w \vee \mathbf{0} = w$, $\forall w \in L$,

where x, y, and z are arbitrary elements in L.

Remark 1. If we define a new binary operator $\wedge$ on L as $a \wedge b := \bigvee_{u \in S} u$, where $S := \{u \in L | u \leqslant a, u \leqslant b\}$, then the triple $(L, \wedge, \vee)$ coincides with the conventional definition of a lattice. This means that there exists a partial order $\leqslant$ on L such that $\vee, \wedge$ are the least upper bound and the greatest lower bound operators under this order and $(L, \leqslant)$ admits unique maximal and minimal elements. The proofs can be found in previous studies [41, 42].

A lattice L, like other partially ordered sets, can be completely characterized by its Hasse diagram [42]: a graph whose vertices are the elements in the lattice. In the Hasse diagram of a lattice, there is an edge between two vertices a, b if and only if the elements represented by these two vertices are comparable ($a \leqslant b$ or $b \leqslant a$) and if b covers a, which means $a \leqslant b$ and there is no element $c \in L$ such that $a \leqslant c \leqslant b$. Then, b is drawn "above" a (i.e., with a higher vertical coordinate).

For example, consider a lattice $L = \{P_1, P_2, P_3, P_4\}$ whose Hasse diagram is as in Figure 1. One can see that P_1 is the greatest element and P_4 the least, whereas P_2 and P_3 are incomparable, $P_2 \vee P_3 = P_1$, $P_2 \wedge P_3 = P_4$.

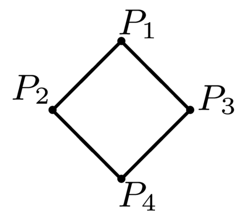

Figure 1 Hasse diagram of a four-element lattice L.

3 Networks over finite lattices

We first use the STP to give an example of how the algebraic state space can help solve the problems in lattice theory.

Because an operator over a finite set is a logical function (according to Definition 2), we can easily adapt the STP framework to networks over finite lattices. For example, Proposition 2 gives criteria for a finite set endowed with a binary operator to be a lattice.

Proposition 2. Let L be a finite set of cardinal k and f a binary operator over L with a structure matrix $M = [M_1, \dots, M_k]$, where $M_i \in L_{k\times k}$, $i = 1, \dots, k$. Then, (L, f) is a lattice if and only if M satisfies the following conditions:

(i) $M\text{diag}(\delta_k^1, \dots, \delta_k^k) = I_k$;

(ii) $MW_{[k,k]} = M$;

(iii) $M^2 = M(I_k \otimes M)$;

(iv) $\exists i \in \mathcal{D}_k$, s.t. $M_i = I_k$.

The proof follows directly from the basic properties of STP stated in the previous section. This gives a convenient way to verify using the structure matrix whether a binary function over a finite set defines a lattice.

3.1 ASSR of control networks over lattices

Let $F : L^n \to L$ be an n-ary function on L. If F is obtained from the composition of $\vee$ and $\wedge$ on L, we call it a lattice function. That is, up to inserting parentheses, F is written in the following form:

$$F(x_1, \dots, x_n) = x_{i_1} \bigcirc x_{i_2} \bigcirc \cdots \bigcirc x_{i_N}, \quad i_1, \dots, i_N \in \{1, \dots, n\}, \quad N > 0,$$

where the operator $\bigcirc$ is either $\vee$ or $\wedge$. Because lattice functions are special cases of multiple-valued logical functions, we may apply the STP method to model networks over finite lattices.

In general, a network over a finite lattice L has the form of (3), where $x_i(t) \in L$ are state variables and the maps $f_i : L^n \to L$ are lattice functions, $i = 1, \dots, n$. When $L = \mathcal{D}_2$, the system becomes a standard Boolean network.

When there are controls $u_1(t), \dots, u_m(t) \in L$ in the network, its dynamics can be expressed as

$$\begin{cases} x_1(t+1) = g_1(x_1(t), \dots, x_n(t); u_1(t), \dots, u_m(t)), \\ \quad\vdots \\ x_n(t+1) = g_n(x_1(t), \dots, x_n(t); u_1(t), \dots, u_m(t)), \end{cases} \tag{6}$$

where g_i are lattice functions, $i = 1, \dots, n$.

Assume that card$(L) = \kappa$. Since the systems over finite lattices belong to the class of multiple-valued logical systems, by Theorem 2, using STP, Eq. (3) can be converted into its ASSR as

$$x(t+1) = Mx(t), \tag{7}$$

where $x(t) = \ltimes_{j=1}^n x_j(t)$, $M \in \mathcal{L}_{\kappa^n \times \kappa^n}$, and Eq. (6) becomes

$$x(t+1) = Pu(t)x(t), \tag{8}$$

where $u(t) = \ltimes_{s=1}^m u_s(t)$, $P \in \mathcal{L}_{\kappa^n \times \kappa^{n+m}}$.

The STP provides a way to rewrite the nonlinear control system (6) to a simplified and unified form, giving it a mathematically neat expression. We give a pedagogical example to illustrate how ASSR is derived.

Example 1. Consider a four-element lattice L with a Hasse diagram as shown in Figure 1. Setting $P_i \sim \delta_4^i$, $i = 1, 2, 3, 4$, and using the formula of the operators $\vee$, $\wedge$ derived from the diagram as

$$P_1 \vee P_2 = P_1,\ P_1 \vee P_3 = P_1,\ P_1 \vee P_4 = P_1,\ P_2 \vee P_3 = P_1,\ P_2 \vee P_4 = P_2,\ P_3 \vee P_4 = P_3,$$
$$P_1 \wedge P_2 = P_2,\ P_1 \wedge P_3 = P_3,\ P_1 \wedge P_4 = P_4,\ P_2 \wedge P_3 = P_4,\ P_2 \wedge P_4 = P_4,\ P_3 \wedge P_4 = P_4,$$

we can solve the structure matrix of the operators $\vee$ and $\wedge$ respectively (for details of the calculation of the structure matrices of finite-valued functions, one may refer to Cheng et al. [40]); that is, $\forall x, y \in \{P_1, \ldots, P_4\}$,

$$\begin{aligned} & x \vee y = M_\vee xy, \quad x \wedge y = M_\wedge xy, \\ & M_\vee = \delta_4[1,1,1,1,1,2,1,2,1,1,3,3,1,2,3,4], \\ & M_\wedge = \delta_4[1,2,3,4,2,2,4,4,3,4,3,4,4,4,4,4]. \end{aligned}$$

Here, we make no distinction between an element in the lattice and its vector expression. Next, we assume that a network over L is defined as

$$\begin{cases} x_1(t+1) = x_1(t) \vee (x_2(t) \wedge u(t)), \\ x_2(t+1) = x_1(t) \vee x_2(t) \vee u(t). \end{cases} \tag{9}$$

Then, using the commutative properties in Proposition 1, the following component-wise ASSR is easily constructed from the structure matrices of the operators.

$$\begin{cases} x_1(t+1) = M_1 u(t) x(t), \\ x_2(t+1) = M_2 u(t) x(t), \end{cases}$$

where $x(t) = x_1(t)x_2(t)$, and

$$\begin{aligned} & M_1 = M_\wedge (I_4 \otimes M_\vee) W_{[4,16]}, \\ & M_2 = M_\vee (I_4 \otimes M_\wedge) W_{[4,16]}. \end{aligned}$$

Then, by a construction similar with (5), the ASSR is $x(t+1) = Mu(t)x(t)$, where

$$\begin{aligned} M &= M_1 * M_2 \\ &= \delta_{16}[1,1,1,1,5,6,5,6,9,9,11,11,13,14,15,16,1,5,1,5,6,6,6,6,9,13,11,15,14,14,16,16,1,1, \\ &\quad 9,9,5,6,13,14,11,11,11,11,15,16,15,16,1,5,9,13,6,6,14,14,11,15,11,15,16,16,16,16]. \end{aligned}$$

Remark 2. Note that one advantage of networks over lattices is that they are composed of basic binary operators; thus, we may use STP to solve the structure matrix solely from its operator expression, which cannot be done for arbitrary multiple-valued networks. The ASSR makes the computation and analysis of system (6) much easier as it allows one to use existing results in multiple-valued logical networks to investigate the control properties of the system. For example, one may consider problems such as the controllability and observability of (9).

Example 2. Consider the controllability problem of the network (9). Denote the matrix M by $[M_1, \ldots, M_{16}]$, where $M_i \in \mathcal{L}_{16\times 16}$. Following [40], construct its controllability matrix as

$$C = \sum_{j=1}^{16} \left(\sum_{i=1}^{16} M_i \right)^{(j)}. \tag{10}$$

Taking the addition and multiplication in (10) to be of Boolean type (that is, $1+1=1$), we solve the controllability matrix as

$$C = \begin{pmatrix}
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&1&1&0&1&0&0&0&0&0&0&0&0&0\\
0&1&0&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&1&0&0&1&0&0&1&0&0&0\\
0&0&0&1&0&0&1&1&0&1&0&0&1&1&0&0\\
0&0&0&1&0&0&1&0&0&1&0&1&1&0&1&0\\
0&0&0&1&0&0&1&1&0&1&0&1&1&1&1&1
\end{pmatrix},$$

which gives full information on the controllability of each state in (9). That is, a state δ_{16}^i is controllable from δ_{16}^j if and only if the (i,j)th entry of C is nonzero (for details, one may refer to Zhao et al. [43]).

Another special property for networks over lattices is that the subnetworks defined over a sublattice of L are invariant subspaces of the network over L; that is, the dynamics starting from a point in the sublattice will remain in it.

Example 3. Recall system (9) and remove the controls in the network. Assume $S = \{P_1, P_2, P_4\} \subset L$, which is clearly a sublattice of L. Reassign the nodes as $P_1 \sim \delta_3^1$, $P_2 \sim \delta_3^2$, and $P_4 \sim \delta_3^3$. It is easy to figure out that, setting $z = x|_S$, we have

$$z(t+1) = M|_S z(t),$$

where $M|_S = \delta_9[1,1,1,4,5,5,7,8,9]$.

If we add the controls back and restrict them to the sublattice S, then it turns out that the system is a control-invariant subspace, which can be viewed as defined over the sublattice with the dynamics

$$z(t+1) = P|_S u(t) z(t),$$

where

$$P|_S = \delta_9[1,1,1,2,5,5,3,6,9,1,2,2,5,5,5,6,6,9,1,2,3,5,5,6,9,9,9].$$

3.2 Control networks over product lattices

Next, we consider the networks over a special class of finite lattices, which are the Cartesian product of some finite lattices. Suppose $L = L_1 \times \cdots \times L_p$, where L_i is a finite lattice of the cardinal k_i, $i = 1, \ldots, p$. Define the partial order relation $\leqslant$ on L by

$$(x_1, \ldots, x_p) \leqslant (y_1, \ldots, y_p) \Leftrightarrow x_1 \leqslant y_1, \ldots, x_p \leqslant y_p.$$

Then, a network over L with ASSR (8), denoted by Σ, can be decomposed into several factors $\Sigma_1, \ldots, \Sigma_p$, where the factor Σ_i is a network consisting of the corresponding ith factor of the state $(x_1(t), \ldots, x_p(t))$, with the binary operations in L_i, $i = 1, \ldots, p$.

Then, a decomposition theorem follows.

Theorem 3. The system Σ is controllable (observable, synchronizable, stabilizable) if and only if the factor systems Σ_i, $i = 1, \ldots, p$ are controllable (observable, synchronizable, stabilizable).

Proof. We only need to show that a network defined over a product lattice of L_1 and L_2 can be decomposed into two systems Σ_1 and Σ_2, which are defined over L_1 and L_2, respectively; that is, any trajectory $\{z(t)|t \geqslant 0\} \subset L$ starting from a point (x_0, y_0) can be decomposed into two trajectories $\{x(t, x_0)|t \geqslant 0\}$ and $\{y(t, y_0)|t \geqslant 0\}$ starting from points $x_0 \in L_1$ and $y_0 \in L_2$, respectively, such that $z(t) = (x(t, x_0), y(t, y_0))$, $\forall t > 0$. Because a factor of a product lattice can be viewed as a sublattice, factor lattices are invariant subspaces; such decomposition is obvious, and the conclusion follows.

Remark 3. For a network over the above product lattice with an ASSR (7), the ASSR of its subsystem over the factor lattice L_i, $i = 1, \dots, p$ is derived as

$$\tilde{x}^i(t+1) = H_i \times_{\mathcal{B}} M \times_{\mathcal{B}} H_i^{\mathrm{T}} \tilde{x}^i(t),$$

where $\tilde{x}^i(t) \in \Delta_{k_i^n}$ and $H_i \in \mathcal{L}_{k_i^n \times k^n}$ is the structure matrix of the projection map $L \to L_i$. This is because L_i is invariant with respect to the evolution of the subnetwork Σ_i. For details of the construction of the above structure matrix, one may refer to Ji et al. [36].

As is known, when we adopt the STP method, the complexity of the analysis of the logical network increases exponentially with respect to the number of nodes [22,23]. Applying Theorem 3, we may reduce the complexity of the analysis of the network over a product lattice $L = L_1 \times \cdots \times L_p$ by decomposing it into systems over subsystems over sublattices L_i, $i = 1, \dots, p$ and reducing its dimension. For example, suppose that the network has n nodes, $\mathrm{card}(L_i) = k_i$, and $k = k_1 \cdots k_p$. Then, the dimension of the ASSR of the system is k^n. If the complexity of executing an algorithm for the analysis of the network is $O(k^n)$, then, after the decomposition, the complexity is reduced to $O(\sum_{i=1}^p k_i^n)$, which greatly eases the computation load.

For an application of Theorem 3, we give an example of systems evolving over product lattices. Consider the following linear switched system with logical switching [44]:

$$x(t+1) = A_{\sigma(t)} x(t) + B_{\sigma(t)} u(t), \tag{11a}$$

$$\eta(t) = Q(x(t)), \tag{11b}$$

$$\begin{cases} \lambda(t+1) = \gamma(\lambda(t), \eta(t)), \\ \sigma(t) = \beta(\lambda(t), \eta(t)), \end{cases} \tag{11c}$$

where $Q : D \to \mathcal{D}_k$ is a quantizer, $D = [\eta_1, \xi_1) \times \cdots \times [\eta_n, \xi_n) \subset \mathbb{R}^n$, $\sigma(t) \in \mathcal{D}_N$ is the switching signal, Eq. (11c) is an ℓ-valued logical dynamic system with γ, β logical functions, $\lambda(t) \in \mathcal{D}_\ell$ is a logical variable, k, N, and ℓ are integers, A_i and B_i are matrices, and $\eta_i < \xi_i$ are real numbers, $i = 1, \dots, N$. $x(t) \in D$ is the continuous-state variable. Commonly, the sampling of the switching signal is based on coordinate partitions [45]; that is, the quantizer Q is defined as

$$Q(x_1, \dots, x_n) = \alpha_{i_1, i_2, \dots, i_n}, \text{ if } \beta^1_{i_1} \leqslant x_1 < \beta^1_{i_1+1},\ \beta^2_{i_2} \leqslant x_2 < \beta^2_{i_2+1}, \dots,\ \beta^n_{i_n} \leqslant x_n < \beta^n_{i_n+1},$$

where $\alpha_{i_1,\dots,i_n} \in \mathcal{D}_k$ is pairwise distinct over $i_1 = 1, \dots, N_1$, $i_2 = 1, \dots, N_2$, $\dots$, $i_n = 1, \dots, N_n$, with $N_1, \dots, N_n$ integers. In short, the quantizer assigns each state $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ an integer in $\mathcal{D}_k$, according to the interval $[\beta^j_{i_j}, \beta^j_{i_j+1})$ that each component x_j lies in, and the domain $D \subset \mathbb{R}^n$ is partitioned as

$$D = \left\{[\beta^1_1, \beta^1_2) \cup [\beta^1_2, \beta^1_3) \cup \cdots \cup [\beta^1_{N_1}, \beta^1_{N_1+1})\right\} \times \cdots \times \left\{[\beta^n_1, \beta^n_2) \cup [\beta^n_2, \beta^n_3) \cup \cdots \cup [\beta^n_{N_n}, \beta^n_{N_n+1})\right\}.$$

Apparently, such partition of $D \subset \mathbb{R}^n$ gives rise to a product lattice structure over $\mathcal{D}_k$, and $\mathcal{D}_k$ can hence be decomposed into components $D_{N_1} \times \cdots \times \mathcal{D}_{N_n}$. As the discretization of the linear switched system (11a) with respect to the quantizer (11b) is a finite transition system over $\mathcal{D}_k$, if the dynamics of the discretized transition system of $\eta(t)$ is expressed using the operators over this lattice, then the system after discretization becomes a dynamics evolving on a product lattice. The analysis of system (11) relies on the mergence of the discretized variable $\eta(t)$ and the logical variable $\lambda(t)$ [44]. Therefore, when $\eta(t)$ evolves over a product lattice, using the decomposition of the system over factor sublattices, the analysis for the merged system can be simplified.

However, the full potential of Theorem 3 is not yet revealed if the underlying lattice structure of the network is known a priori, because in this case, one can see from the beginning that the system is built up from subsystems over sublattices. In Section 4, we focus on (control) networks whose lattice structure is unknown, and Theorem 3 is useful for those systems found by our algorithms to be defined over product lattices.

4 Recovering lattice structures

After presenting the general expressions and control property analysis of networks over lattices, we consider the verification of such networks. As aforementioned, networks over lattices are special cases of multiple-valued logical networks. Thus, a natural question arises: how can one recognize them from the vast class of networks over finite sets? Further, how can one construct or recover the order relationships on underlying sets from the algebraic state space expressions of networks?

The aim of this section is twofold: first, to reconstruct the lattice structure on $\mathcal{D}_k$ from the structure matrix of a network defined over a lattice of k elements; second, to verify if there exists a partial order on $\mathcal{D}_k$ allowing a given network to be one defined over a lattice or at least be generated by some basic operators on the lattice.

4.1 Recovering order relations through lattice functions

Consider a system over $\mathcal{D}_k$ with ASSR (7). The following algorithm can decompose it into a component-wise expression, whose proof is straightforward.

Proposition 3. Consider a matrix $M \in \mathcal{L}_{k^n \times k^n}$. There exists a unique decomposition $M = M_1 * \cdots * M_n$, where $M_i = R_i M \in \mathcal{L}_{k \times k^n}$, and $R_i = \mathbf{1}_{k^{i-1}} \otimes I_k \otimes \mathbf{1}_{k^{n-i}}$, where $\mathbf{1}_n$ is the n-dimensional row vector with all entries equal 1.

In the following, we show that the information of comparability between the elements in a lattice can be encoded in the structure matrix of any lattice function over it. That is, if M is the structure matrix of an n-node network defined over a lattice, then any M_i constructed as in Proposition 3, which is the structure matrix of the dynamics of the i-th node, $i = 1, \ldots, n$, is enough to rebuild the lattice structure.

The case of 2-ary functions, which corresponds to a network of 2 nodes, is trivial because, by the so-called absorption property, a binary lattice function can only be $\vee$ or $\wedge$ (after some procedure of simplification). Therefore, it suffices to check the conditions in Proposition 2 on its structure matrix and recover the partial order by $x \vee y = y \Leftrightarrow x \leqslant y$.

The general n-ary case requires a different approach. Without loss of generality, assume that functions discussed in the following do not have "dumb" indices; that is,

$$\begin{aligned} &\forall i = 1, \ldots, n, \ \exists a_i \neq b_i \in \mathcal{D}_k, \ \text{s.t.} \\ &f(x_1, \ldots, x_{i-1}, a_i, x_{i+1}, \ldots, x_n) \neq f(x_1, \ldots, x_{i-1}, b_i, x_{i+1}, \ldots, x_n). \end{aligned} \tag{12}$$

Specifically, in network (3), if the index i is dumb in the function f_j, it means that the variable x_i does not influence the one-step transition of the variable x_j. In fact, one can verify if a variable in the expression of a multiple-valued logical function is dumb using the method provided by [23] and further remove them from the explicit expression of the function.

First, we point out the fact that given two comparable elements a and b in any lattice $(L, \vee, \wedge)$, the operators $\vee$ and $\wedge$ restricted to $\{a, b\}$ behave in the same way as the disjunction and conjunction operators over the classical Boolean algebra $\{0, 1\}$. Hence, we may try to use the "restricted" structure matrix of a k-valued logical function to find all comparable pairs on $\mathcal{D}_k$.

For an n-ary logical function f over $\mathcal{D}_k$, consider its restriction to a pair of elements $a, b \in \mathcal{D}_k$; that is, the value that $f(x_1, \ldots, x_n)$ takes when $x_1, \ldots, x_n$ take values in $\{a, b\}$ only. This transforms f to another logical function $f_{ab} : \mathcal{D}_2^n \to \mathcal{D}_k$.

Denote by $M \in \mathcal{L}_{k \times k^n}$ the structure matrix of f. $\forall a, b \in \mathcal{D}_k$. If their vector forms are $\delta_k^{i_a}$ and $\delta_k^{i_b}$, then identify them as δ_2^1 and δ_2^2, respectively. By making no distinction between a variable and its vector form ($a \sim \delta_k^{i_a}$, $b \sim \delta_k^{i_b}$, $1 \sim \delta_2^1$, $0 \sim \delta_2^2$), we may express a variable $x \in \{a, b\}$ through a Boolean one $\tilde{x} \in \{0, 1\}$ in vector form as

$$x = \delta_k[i_a, i_b]\tilde{x}. \tag{13}$$

Hence, by substitution, the restriction of f on $\{a, b\}^n$, viewed as a function $f_{ab} : \mathcal{D}_2^n \to \mathcal{D}_k$, has the following vector form expression:

$$f(x_1, \ldots, x_n)\big|_{\{a,b\}^n} = f_{ab}(\tilde{x}_1, \ldots, \tilde{x}_n) = M_{ab}\tilde{x}_1 \ldots \tilde{x}_n, \tag{14}$$

where

$$M_{ab} = M \prod_{j=0}^{n-1} (I_{2^j} \otimes \delta_k[i_a, i_b]). \tag{15}$$

The structure matrix (15) is obtained from (13) and the commutative properties in Proposition 1.

Theorem 4. Let $\mathcal{D}_k$ endowed with a partial order relation $\leqslant$ be a lattice and $f : \mathcal{D}_k^n \to \mathcal{D}_k$ be a lattice function satisfying (12). Given two elements $a, b \in \mathcal{D}_k$ with vector expressions $\delta_k^{i_a}$ and $\delta_k^{i_b}$, respectively, a and b are comparable ($a \leqslant b$ or $b \leqslant a$) if and only if

$$\mathrm{Col}(M_{ab}) \subset \{\delta_k^{i_a}, \delta_k^{i_b}\}; \tag{16a}$$

$$\mathrm{Col}(M_{ab}^i) \subset \{\delta_k^{i_a} - \delta_k^{i_b}, \mathbf{0}\},\ i = 1, \ldots, n, \tag{16b}$$

$$M_{ab}\delta_{2^n}^1 = \delta_k^{i_a},\ M_{ab}\delta_{2^n}^{2^n} = \delta_k^{i_b}, \tag{16c}$$

where $\mathbf{0} = (0, \ldots, 0) \in \mathbb{R}^k$, M_{ab} is defined as in (15), and

$$M_{ab}^i := M_{ab} W_{[2,2^{i-1}]} \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Proof. (Necessity) When a and b are comparable, since $\vee$ and $\wedge$ restricted to $\{a, b\}$ are identical to disjunction and conjunction operators on the classical Boolean algebra $\{0, 1\}$ (denoted by $\vee_2$ and $\wedge_2$, respectively), $f|_{\{a,b\}^n}$ as a lattice function can be viewed as belonging to the class generated by $\vee_2$ and $\wedge_2$. Hence, its image is in the set of $\{a, b\}$. By the property of these operators, $f|_{\{a,b\}^n}$ is monotonic and reproducing; that is (adopting the notations in (14)), $\forall \tilde{x}_1, \ldots, \tilde{x}_n \in \{0, 1\}$,

$$\begin{aligned}
&f_{ab}(\tilde{x}_1, \ldots, \tilde{x}_n) \in \{a, b\},\\
&f_{ab}(\tilde{x}_1, \ldots, \tilde{x}_{i-1}, 1, \tilde{x}_{i+1}, \ldots, \tilde{x}_n) - f_{ab}(\tilde{x}_1, \ldots, 0, \ldots, \tilde{x}_n) \in \{a - b, 0\}, \quad \forall i = 1, \ldots, n,\\
&f_{ab}(1, \ldots, 1) = a,\ f_{ab}(0, \ldots, 0) = b,
\end{aligned}$$

where, in the second equation, the variables only differ in the ith entry, $i = 1, \ldots, n$, and we can say that $f(x_1, \ldots, x_n) \leqslant f(y_1, \ldots, y_n)$ if $x_i \leqslant y_i$, $x_i, y_i \in \{a, b\}$, $i = 1, \ldots, n$. Translating the above equations into the vector expression yields

$$\begin{aligned}
&M_{ab}\tilde{x}_1 \ldots \tilde{x}_n \in \{\delta_k^{i_a}, \delta_k^{i_b}\},\\
&M_{ab}\tilde{x}_1 \ldots \delta_2^1 \ldots \tilde{x}_n - M_{ab}\tilde{x}_1 \ldots \delta_2^2 \ldots \tilde{x}_n = M_{ab}W_{[2,2^{i-1}]}(\delta_2^1 - \delta_2^2)\tilde{x}_1 \ldots \tilde{x}_{i-1}\tilde{x}_{i+1} \ldots \tilde{x}_n \in \{\delta_k^{i_a} - \delta_k^{i_b}, \mathbf{0}\},\\
&M_{ab}\delta_2^1 \ldots \delta_2^1 = \delta_k^{i_a},\ M_{ab}\delta_2^2 \ldots \delta_2^2 = \delta_k^{i_b},\\
&\forall \tilde{x}_1, \ldots, \tilde{x}_n \in \Delta_2,\ i = 1, \ldots, n,
\end{aligned}$$

which are (16a)–(16c), respectively.

(Sufficiency) Conversely, if $f|_{\{a,b\}^n}$ satisfies the conditions (16a)–(16c), it can be viewed as generated by $\vee_2$ and $\wedge_2$ over the lattice $\{a, b\}$ because, according to the classical result by Post [46], the class of reproducing and monotonic n-ary functions over $\mathcal{D}_2$ are exactly the one generated by $\vee_2$ and $\wedge_2$. Therefore, a and b have the order relation obtained from these operators.

Theorem 4 shows that given any lattice function f over a k-element lattice, we can find all the comparable pairs a, b in $\mathcal{D}_k$ by checking the conditions on corresponding M_{ab} and further obtain an undirected graph of k nodes, called the comparability graph, where there is an edge between two nodes if and only if they are comparable.

Then, algorithms can be applied to the comparability graph, assigning to it an orientation to recover the partial order on $\mathcal{D}_k$, which is, however, not unique [47]. Details of these algorithms are stated later in Subsection 4.2.

Remark 4. The significance of Theorem 4 is that, once the underlying lattice structure of a network over a lattice is recovered, one may use it to simplify the analysis of control problems. For example, if we find out that the network is defined over a product lattice, then the results in Subsection 3.2 can be applied to decompose a network.

4.2 Realization of the underlying lattice structure of a network

Next, we consider designing a lattice structure over $\mathcal{D}_k$ to make a k-valued network be generated by some basic operators of the lattice. The only data available are those of the structure matrix. The construction requires two steps: first, derive a relation from f according to Theorem 4; second, verify if it is a partial order making $\mathcal{D}_k$ a lattice.

According to the results in Lau [18] (Chapter 11, Subsection 11.4), monotonic functions over a lattice are generated by $\vee$ and $\wedge$ and piecewise constant functions $m_{a,b}$, defined as

$$m_{a,b}(x) = \begin{cases} b, & a \leqslant x, \\ \mathbf{0}, & \text{otherwise}. \end{cases} \tag{17}$$

We call $\vee$, $\wedge$ and $\{m_{a,b}\}_{a,b\in\mathcal{D}_k}$ basic operators on a lattice.

We present the following algorithm to construct a lattice structure on $\mathcal{D}_k$ to make the network generated by basic operators on this lattice.

Proposition 4. Consider a network over $\mathcal{D}_k$ with n nodes and its ASSR as $x(t+1) = Mx(t)$, where $x(t) = x(t) = \ltimes_{j=1}^{n} x_j(t)$, $M \in \mathcal{L}_{k^n\times k^n}$. Execute the following procedure to derive a partial order on $\mathcal{D}_k$.

- Step 1. Apply Proposition 3 to decompose M into componentwise form $M_1, \ldots, M_n$, where M_i is the structure matrix of the i-th node $x_i(t+1) = f_i(x_1(t), \ldots, x_n(t))$ with ASSR $x_i(t+1) = M_i x(t)$, $i = 1, \ldots, n$. Remove the dumb indices in each function M_i (see the algorithm provided by Cheng [23]).
- Step 2. For the i-th function f_i with structure matrix M_i, draw the corresponding graph G_i of n nodes, where there is an edge between nodes $a, b \in \mathcal{D}_k$ if and only if (a, b) is a pair on which f_i satisfies the conditions (16a) and (16b). Take the conjunction of the graphs G_i, $i = 1, \ldots, n$ and denote it by G.
- Step 3. Give an orientation to the resulting graph G and verify whether it is transitive. If it is, then the topological sorting of the oriented graph derives a partial order on $\mathcal{D}_k$.

If the partial order derived from the network over $\mathcal{D}_k$ following the above procedure makes a lattice, then the network is generated from the class of functions $\{\vee, \wedge, m_{a,b}\}$, where $\vee$ and $\wedge$ are the sup and inf operators of the lattice, respectively, and $m_{a,b}$ are piecewise constant functions defined as in (17), $a, b \in \mathcal{D}_k$.

Proof. The pairs on which the function with structure matrix M_i is monotonic (i.e., satisfies (16a) and (16b), by the proof of Theorem 4) are candidates of comparable pairs to allow this function to be generated from basic operators of a lattice (see Chapter 11 of Lau [18]). After checking the updating function of the i-th node on each pair $\{a, b\} \subset \mathcal{D}_k$, we obtain the comparability graph of the partial order for the dynamics of the i-th node. Since only when two elements are comparable in the order relation derived from each node can they be labeled as comparable in the partial order corresponding to the whole system, one needs to take conjunction of the n graphs (i.e., there is an edge between $\{a, b\}$ in the graph G if and only if such edge exists for each graph G_i, $i = 1, \ldots, n$) and check the conditions for being a lattice on the corresponding order relation after orienting the graph.

Proposition 4 gives a criterion for whether a network can be realized through basic operators of some lattice when only its structure matrix is known.

Remark 5. The procedure for constructing the graph G in Step 2 is depicted in pseudocode form as in Algorithm 1. Let (N, E) be an undirected graph of k nodes, that is, $N := \{1, \ldots, k\}$, and $E := \{e(a, b) : \mathcal{D}_k \times \mathcal{D}_k \to \{0, 1\}\}_{a,b\in 1,\ldots,k}$, where $e(a, b) = 1$ if and only if there is an edge between the vertices a and b. Then, Algorithm 1 returns the adjacency information of the vertices.

Remark 6. The algorithm in Step 3 for transitive orientation has been well established since 1999 [47], and we only give a sketch of it because of space limitations. Let G be a prime undirected graph with vertex set V; start with a partition $\{\{v\}, V\backslash\{v\}\}$, and define it as an ordered list. Then, choose a pivot vertex $x \in V$ and split each partition class Y into two parts: the vertices adjacent to x (denoted by Y_a) and those non-adjacent to x (denoted by Y_n). Then, place them in consecutive positions in the ordered list. Let Y_n occupy the earlier one in the two new positions if x proceeds Y and the latter if not. Then, move on to another pivot vertex. Refining the partition following the procedure for choosing pivots, one obtains an ordered partition where each class only contains a single element, which is the topological sorting of the orientation. One may refer to previous studies [48, 49] for proofs and details.

In the end, we claim that there exist canonical expressions for monotonic functions over a lattice.

Algorithm 1 Deriving the comparability graph from the ASSR.

Input: Structure matrices $M_i \in \mathcal{L}_{k\times k^n}$, $i = 1, \ldots, n$.
Output: Graph $G = (N, E)$.

Set $e(a, b) = 0$, $e_i(a, b) = 0$ for all $a, b \in \{1, \ldots, k\}$, $i = 1, \ldots, n$;
for $i = 1 : n$ **do**
  for $a = 1 : k$ **do**
    for $b = 1 : a$ **do**
      if $e_i(a, b) = 0$ **then**
        goto here;
      else
        $M_i^{ab} = M_i \prod_{j=0}^{n-1} (I_{2^j} \otimes \delta_k[i_a, i_b])$;
        if M_i^{ab} satisfies (16) **then**
          $e_i(a, b) = 1$;
        else
          $e_i(a, b) = 0$;
        end if
      end if
      here;
    end for
  end for
  $e(a, b) = e(a, b) \wedge e_i(a, b)$;
end for
return $N = \{1, \ldots, k\}$, $E = \{e(a, b)\}_{a,b=1,\ldots,k}$.

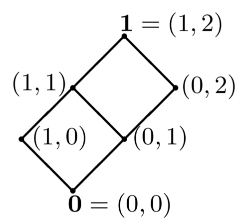

Figure 2 Hasse diagram of $\mathcal{D}_2 \times \mathcal{D}_3$.

Proposition 5 ([18], Subsection 11.4). If $f : \mathcal{D}_k^n \to \mathcal{D}_k$ is monotonic over a lattice $(\mathcal{D}_k, \leqslant)$, then $\forall \boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{D}_k^n$,

$$f(\boldsymbol{x}) = \bigvee_{\boldsymbol{a}=(a_1,\ldots,a_n)\in\mathcal{D}_k^n} \left(\bigwedge_{i=1}^{n} m_{a_i, f(\boldsymbol{a})}(x_i) \right), \tag{18}$$

where $m_{a_i,f(\boldsymbol{a})}$ is defined as in (17).

One can see that additional constraints are needed to ensure that a function is a lattice function (i.e., generated solely by $\vee$ and $\wedge$). However, finding such conditions would be difficult because of (18) as any lattice function is monotonic and hence will always allow an expression that contains m_{ab}.

5 Numerical examples

This section provides numerical examples to illustrate the results in Sections 3 and 4.

We first consider the observability problem of a control network defined over a product lattice and see how it can be solved by decomposing the system into subsystems over factor lattices.

Example 4. Consider a network over the lattice $\mathcal{D}_2 \times \mathcal{D}_3$, where $\mathcal{D}_2 = \{0, 1\}$, $\mathcal{D}_3 = \{0, 1, 2\}$ are ordered as chains canonically. The Hasse diagram of the product lattice is shown in Figure 2.

The dynamics of the network is

$$\begin{cases} x_1(t+1) = x_2(t) \wedge u(t), \\ x_2(t+1) = x_1(t) \vee x_2(t), \end{cases} \qquad y(t) = x_2(t). \tag{19}$$

Converting (19) to ASSR, we respectively obtain the structure matrices of the dynamics and the output as

$$L = \delta_{36}[1, 3, 3, 5, 5, \ldots, 32, 33, 34, 35, 36] \in \mathcal{L}_{36\times 216},$$

$$E = \delta_6[1, 2, 3, 4, 5, 6, \ldots, 1, 2, 3, 4, 5, 6] \in \mathcal{L}_{6\times 36}.$$

We consider the observability of the system (19), showing how existing results can be applied to the ASSR and how it can be solved using its subnetworks over sublattices. Suppose $x_1(t) = (x_1^1(t), x_2^1(t))$, $x_2(t) = (x_1^2(t), x_2^2(t))$, $y(t) = (y_1(t), y_2(t)) \in \Delta_2 \times \Delta_3$. Then, one can respectively calculate the ASSR of the corresponding subnetworks over the factor sublattices as

$$\begin{cases} z_1(t+1) = \tilde{L}_1 z_1(t), \\ y_1(t) = E_1 z_1(t); \end{cases} \quad \begin{cases} z_2(t+1) = \tilde{L}_2 z_2(t), \\ y_2(t) = E_2 z_2(t), \end{cases}$$

where $z_1(t) = x_1^1(t)x_2^1(t)$, $z_2(t) = x_2^1(t)x_2^2(t)$, and

$$\begin{aligned} &\tilde{L} = \delta_4[1, 2, 2, 2, 1, 2, 4, 4], \quad E_1 = \delta_2[1, 2, 1, 2], \\ &\tilde{L}_2 = \delta_9[1, 2, 3, 2, 2, 3, 3, 3, 3, 1, 2, 3, 5, 5, 6, 6, 6, 6, 1, 2, 3, 5, 5, 6, 9, 9, 9], \\ &E_2 = \delta_3[1, 2, 3, 1, 2, 3, 1, 2, 3]. \end{aligned}$$

First, consider the network over $\mathcal{D}_2$. Following Cheng et al. [50], construct an auxiliary network as

$$\begin{cases} z_1(t+1) = \tilde{L}_1 u(t) z_1(t), \\ z_1^*(t+1) = \tilde{L}_1 u(t) z_1^*(t). \end{cases}$$

Set $w(t) = z(t)z^*(t)$; then, the ASSR is

$$w(t+1) = Gu(t)w(t),$$

where $G = \tilde{L}_1(I_8 \otimes \tilde{L}_1)(I_2 \otimes W_{[2,4]})\mathrm{diag}(\delta_2^1, \delta_2^2)$.

Solving the output distinguishable pairs, we derive the set

$$W = \{\delta_{16}^2, \delta_{16}^4, \delta_{16}^5, \delta_{16}^7, \delta_{16}^{10}, \delta_{16}^{12}, \delta_{16}^{13}, \delta_{16}^{15}\}.$$

Then, the controllability matrix from Δ_{16} to W is

$$\begin{aligned} C_W^1 &= I_W \left[\sum_{i=1}^{16} (G\delta_2^1 + G\delta_2^2)^{(i)} \right] \\ &= [0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0], \end{aligned}$$

where I_W is the index matrix of W.

Therefore, the distinguishable pairs of subnetwork is $S_1 = \{(\delta_4^1, \delta_4^2), (\delta_4^1, \delta_4^3), (\delta_4^1, \delta_4^4)\}$, and the system is not observable.

A similar argument can be applied to the subsystem over $\mathcal{D}_3$, showing that

$$\begin{aligned} C_W^2 = [&0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, \\ &0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0], \end{aligned}$$

and the distinguishable pairs of the subnetwork is

$$\begin{aligned} S_2 = \{&(\delta_9^1, \delta_9^2), (\delta_9^1, \delta_9^3), (\delta_9^1, \delta_9^4), (\delta_9^1, \delta_9^5), (\delta_9^1, \delta_9^6), (\delta_9^1, \delta_9^7), (\delta_9^1, \delta_9^8), (\delta_9^1, \delta_9^9), (\delta_9^2, \delta_9^3), (\delta_9^2, \delta_9^6), (\delta_9^2, \delta_9^7), (\delta_9^2, \delta_9^8), \\ &(\delta_9^2, \delta_9^9), (\delta_9^3, \delta_9^4), (\delta_9^3, \delta_9^5), (\delta_9^4, \delta_9^6), (\delta_9^4, \delta_9^7), (\delta_9^4, \delta_9^8), (\delta_9^4, \delta_9^9), (\delta_9^5, \delta_9^6), (\delta_9^5, \delta_9^7), (\delta_9^5, \delta_9^8), (\delta_9^5, \delta_9^9)\}. \end{aligned}$$

Therefore, we conclude that the whole system is not observable. More precisely, $(\delta_2^{i_1}, \delta_2^{j_1}), (\delta_2^{i_2}, \delta_2^{j_2}) \in (\Delta_2 \times \Delta_3)^2$ is distinguishable if and only if $(\delta_2^{i_1}, \delta_2^{j_1}) \in S_1$, $(\delta_2^{i_2}, \delta_3^{j_2}) \in S_2$.

Next, we give an example of how the algorithm in Proposition 4 works for constructing a lattice structure on a finite set from the structure matrix of a network so that this network can be expressed through basic operators on the lattice.

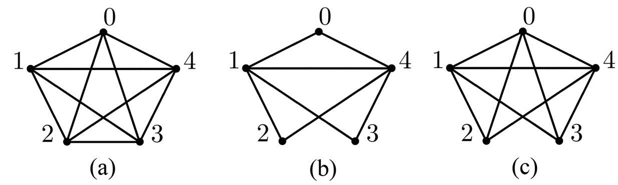

Figure 3 Comparability graphs corresponding to (a) M_1, (b) M_2, and (c) M_3.

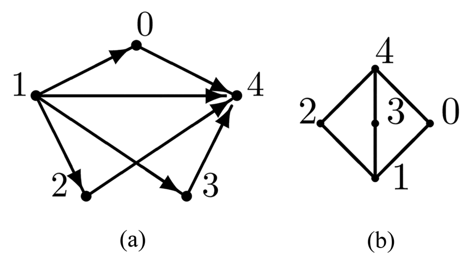

Figure 4 Transitive orientation of the conjunction of comparability graphs (a) and the Hasse diagram of the lattice (b).

Example 5. Consider the following network over $\mathcal{D}_5$ with 3 nodes and ASSR as

$$x(t+1) = Mx(t), \tag{20}$$

where $x(t) = x_1(t)x_2(t)x_3(t)$, and

$$\begin{aligned} M = \delta_{125}\ &[1,31,61,91,121,26,31,61,91,121,51,56,61,91,121,76,81,86,91,121,101,106,111,116,121,\\ &1,32,61,92,122,32,32,67,92,117,51,57,61,92,122,82,82,92,92,117,102,107,112,117,122,1,\\ &31,63,93,123,26,31,63,93,123,63,68,63,93,118,88,93,88,93,118,103,108,113,118,123,1,\\ &32,63,94,124,32,32,69,94,119,63,69,63,94,119,94,94,94,94,119,124,119,119,119,124,1,\\ &32,63,94,125,27,32,64,94,125,53,59,63,94,125,99,94,94,94,125,125,120,120,120,125]. \end{aligned}$$

We aim to check if there is an underlying partial order on $\mathcal{D}_5$ to make this system be defined over a lattice or to realize the transition law by some lattice operators.

Decomposing M according to Proposition 3, we derive the structure matrix of the dynamics of each node as

$$M_i = \mathbf{1}_{5^{i-1}} \otimes I_5 \otimes \mathbf{1}_{5^{3-i}} \in \mathcal{L}_{5\times 5^3},\ \ i = 1,2,3.$$

Next, we try to derive comparability graphs from them. Checking the conditions (16a) and (16b) on M_1, M_2, and M_3 for each pair of $a, b \in \mathcal{D}_5$, one can obtain their comparability graphs, as shown in Figures 3(a)–(c) respectively. Then, take their conjunction to get the comparability graph of the system (which is, naturally, the one in Figure 3(b)) and orient it according to the vertex-partitioning algorithm in McConnell and Spinrad [48]. First, choose 1 as a source, and the initial partition is $\{\{1\},\{2,3,4,0\}\}$. After a pivot on 2, as $\{3,0\}$ is not adjacent to 2, it goes before $\{2,4\}$. Then, after a pivot on 3, we obtain the topological sorting of the graph as

$$1 \mid 0 \mid 3 \mid 2 \mid 4, \tag{21}$$

where an element is less than the ones following it in the sequence. Applying (21) to Figure 4(a) gives the conjunction of comparability graphs a transitive orientation, as shown in Figure 4(b). Hence, we obtain a partial order on $\mathcal{D}_5$, denoted by R.

Finally, after checking the conditions on this orientation for being a lattice over $\mathcal{D}_5$, we may draw the Hasse diagram of this order relation, as shown in Figure 4.

Therefore, $(\mathcal{D}_5, R)$ is the lattice structure that makes the network (20) become one generated by basic operators on the lattice, and one can write it explicitly in these operators according to (18).

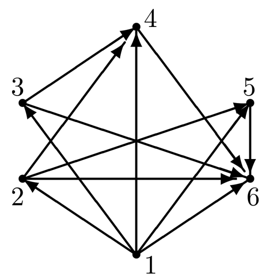

Figure 5 Oriented comparability graph over $\mathcal{D}_6$ in Example 4.

6 Conclusion

In this paper, control networks over finite lattices were first considered. Using STP, the ASSR of such networks was obtained, making control problems easily solvable. Algorithms were proposed to recover the partial order from a lattice function and construct lattice structures for an arbitrary network to allow it to be generated by basic operators over the lattice.

In various applications, the results in Subsection 3.2 and Section 4 can be combined: given a network over an unknown lattice, one may first recover the order relation using Theorem 4 and then decompose the system into subsystems defined over factor sublattices if the lattice is a product lattice. Theorem 3 helps to simplify the computation of the control properties. For example, if we apply the algorithm in Proposition 4 to the network dynamics in Example 4, we can reconstruct the oriented comparability graph as in Figure 5 and recover the lattice structure depicted in Figure 2. Then, decomposition can be applied, reducing the complexity of the system analysis from $O(6^2)$ to $O(2^2 + 3^2)$.

However, although (due to the canonical expression (18)), the difficulty of finding restrictions on the monotonic function class over a lattice to exclude the functions $m_{a,b}$ seems intrinsic, finding criteria for multiple-valued logical functions to be generated solely by the operators $\vee$ and $\wedge$ is a problem still worth investigating.

Meanwhile, using the framework proposed in this paper, problems concerning switched, delayed, and probabilistic networks over finite lattices can be further investigated.

Acknowledgements This work was supported partly by National Natural Science Foundation of China (Grants Nos. 62073315, 62350037).

References

1 Kauffman S A. Metabolic stability and epigenesis in randomly constructed genetic nets. J Theor Biol, 1969, 22: 437–467
2 Boole G. The Mathematical Analysis of Logic. New York: Philosophical Library, 1847
3 Post E L. Introduction to a general theory of elementary propositions. Am J Math, 1921, 43: 163–185
4 Rosenbloom P C. Post algebras. I. postulates and general theory. Am J Math, 1942, 64: 167–188
5 Epstein G. The lattice theory of post algebras. Trans Amer Math Soc, 1960, 95: 300–317
6 Traczyk T. Post algebras through P_0 and P_1 lattices, In: Proceedings of the Computer Science and Multiple-Valued Logic Theory and Applications, 1977
7 Shum K W, Sun Q T. Lattice network codes over optimal lattices in low dimensions, In: Proceedings of the 7th International Workshop on Signal Design and its Applications in Communications (IWSDA), Bengaluru, 2015. 113–117
8 Tunali N E, Huang Y C, Boutros J J, et al. Lattices over Eisenstein integers for compute-and-forward. IEEE Trans Inform Theor, 2015, 61: 5306–5321
9 Bouyioukos C, Kim J T. Gene regulatory network properties linked to gene expression dynamics in spatially extended systems. In: Proceedings of the 10th European Conference on Artificial Life, Budapest, 2009. 321–328
10 Bykovsky A Y. Heterogeneous network architecture for integration of AI and quantum optics by means of multiple-valued logic. Quantum Rep, 2020, 2: 126–165
11 Bykovsky A Y. Multiple-valued logic modelling for agents controlled via optical networks. Appl Sci, 2022, 12: 1263
12 Qiao S, Zhu P, Feng J. Fuzzy bisimulations for nondeterministic fuzzy transition systems. IEEE Trans Fuzzy Syst, 2023, 31: 2450–2463
13 Qiao S, Feng J, Zhu P. Distribution-based limited fuzzy bisimulations for nondeterministic fuzzy transition systems. J Franklin Inst, 2024, 361: 135–149
14 Richard A, Comet J P. Necessary conditions for multistationarity in discrete dynamical systems. Discrete Appl Math, 2007, 155: 2403–2413
15 Sun Q T, Yuan J H, Huang T, et al. Lattice network codes based on Eisenstein integers. IEEE Trans Commun, 2013, 61: 2713–2725
16 Russell S J, Norvig P. Artificial Intelligence: A Modern Approach. 3rd ed. Upper Saddle River: Prentice Hall, 2010
17 Adamatzky A. On dynamically non-trivial three-valued logics: oscillatory and bifurcatory species. Chaos Solitons Fractals, 2003, 18: 917–936
18 Lau D. Function Algebras on Finite Sets: Basic Course on Many-valued Logic and Clone Theory. Berlin: Springer, 2006
19 Rine D C. Computer Science and Multiple-valued Logic: Theory and Applications. New York: North-Holland Publishing Company, 2014
20 García-Martínez M, Campos-Cantón I, Campos-Cantón E, et al. Difference map and its electronic circuit realization. Nonlinear Dyn, 2013, 74: 819–830

21 Suneel M. Electronic circuit realization of the logistic map. Sadhana, 2006, 31: 69–78
22 Cheng D Z. Input-state approach to Boolean networks. IEEE Trans Neural Netw, 2009, 20: 512–521
23 Cheng D, Qi H. Controllability and observability of Boolean control networks. Automatica, 2009, 45: 1659–1667
24 Fornasini E, Valcher M E. Observability, reconstructibility and state observers of Boolean control networks. IEEE Trans Automat Contr, 2013, 58: 1390–1401
25 Zhang K. A survey on observability of Boolean control networks. Control Theor Technol, 2023, 21: 115–147
26 Li H, Wang Y. Further results on feedback stabilization control design of Boolean control networks. Automatica, 2017, 83: 303–308
27 Zhang X, Wang Y, Cheng D. Output tracking of Boolean control networks. IEEE Trans Automat Contr, 2020, 65: 2730–2735
28 Li Y, Zhu J. Necessary and sufficient vertex partition conditions for input-output decoupling of Boolean control networks. Automatica, 2022, 137: 110097
29 Lu J, Li H, Liu Y, et al. Survey on semi-tensor product method with its applications in logical networks and other finite-valued systems. IET Control Theor Appl, 2017, 11: 2040–2047
30 Cheng D, Ji Z. On networks over finite rings. J Franklin Institute, 2022, 359: 7562–7599
31 Zou S, Zhu J. On sub-networks over proper ideals of dynamical networks over finite rings. In: Proceedings of the 42nd Chinese Control Conference (CCC), Tianjin, 2023. 81–86
32 Lin L, Cao J, Zhu S, et al. Synchronization analysis for stochastic networks through finite fields. IEEE Trans Automat Contr, 2021, 67: 1016–1022
33 Lin L, Jiang Z, Lin H, et al. On quotients of stochastic networks over finite fields. IEEE Trans Control Netw Syst, 2024, 11: 878–889
34 Meng M, Li X, Xiao G. Synchronization of networks over finite fields. Automatica, 2020, 115: 108877
35 Yan Y, Cheng D, Feng J E, et al. Survey on applications of algebraic state space theory of logical systems to finite state machines. Sci China Inf Sci, 2023, 66: 111201
36 Ji Z, Zhang X, Cheng D. Aggregated (Bi-)simulation of finite-valued networks. 2023. ArXiv:2303.14390
37 Zhong J, Liu Y, Lu J, et al. Pinning control for stabilization of Boolean networks under knock-out perturbation. IEEE Trans Automat Contr, 2022, 67: 1550–1557
38 Zhao Y, Ghosh B K, Cheng D. Control of large-scale Boolean networks via network aggregation. IEEE Trans Neural Netw Learn Syst, 2015, 27: 1527–1536
39 Cheng D, Qi H, Zhao Y. An Inroduction to Semi-tensor Product of Matrices and Its Applications. Singapore: World Scientific, 2012
40 Cheng D, Qi H, Li Z. Analysis and Control of Boolean Networks, A Semi-tensor Product Approach. New York: Springer, 2011
41 Bergman G M. An Invitation to General Algebra and Universal Constructions. 2nd ed. Berlin: Springer, 2015
42 Stanley R P. Enumerative Combinatorics. 2nd ed. Cambridge: Cambridge University Press, 2012
43 Zhao Y, Qi H, Cheng D. Input-state incidence matrix of Boolean control networks and its applications. Syst Control Lett, 2010, 59: 767–774
44 Guo Y, Gong P, Wu Y, et al. Stabilization of discrete-time switched systems with constraints by dynamic logic-based switching feedback. Automatica, 2023, 156: 111190
45 Pola G, Di Benedetto M D. Control of cyber-physical-systems with logic specifications: a formal methods approach. Annu Rev Control, 2019, 47: 178–192
46 Post E L. The Two-valued Iterative Systems of Mathematical Logic. Princeton: Princeton University Press, 1941
47 Brandstädt A, Le V B, Spinrad J P. Graph Classes: A Survey. Philadelphia: Society for Industrial and Applied Mathematics, 1999
48 McConnell R M, Spinrad J P. Linear-time transitive orientation. In: Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, 1999. 19–25
49 McConnell R M, Spinrad J P. Modular decomposition and transitive orientation. Discrete Math, 1999, 201: 189–241
50 Cheng D, Li C, He F. Observability of Boolean networks via set controllability approach. Syst Control Lett, 2018, 115: 22–25