Structural Oscillatority Analysis of Boolean Networks

Shun-ichi Azuma, *Senior Member, IEEE*, Takahiro Yoshida, and Toshiharu Sugie, *Fellow, IEEE*

***Abstract*—Boolean networks are system models whose binary-state nodes are interconnected. Although a number of studies have been conducted so far, it is usually assumed that the *full* information on the target system is available for analysis and design. This paper addresses a *structural analysis* problem for Boolean networks, i.e., the case when the information on the network structure is *available* but that on the node dynamics is *unavailable*. In particular, we consider here oscillatority (instability). First, the notion of *structural oscillatority* is formulated based on an equivalence relation of network structures. We next present a necessary and sufficient condition for the so-called *cactus* Boolean networks to be structurally oscillatory. This condition captures structural oscillatority by a simple characterization in terms of the network structure (more concretely, the number of inhibiting edges in each simple cycles), which enables us to apply it to large-scale Boolean networks.**

***Index Terms*—Boolean network, network topology, structural oscillatority, structural stability.**

I. Introduction

BOOLEAN networks [1] are system models whose binary-state nodes are interconnected. Although the state of each node is restricted to be either 0 (inactive) or 1 (active), the system models are known as good approximations of various biological networks [2], [3].

So far, a number of results have been obtained in the fields of control and systems biology, such as stability (existence of attractors) [4]–[10], oscillatority [11], controllability and observability [12]–[14], and control synthesis [13], [15]–[20]. Other results can be found in, e.g., [21], [22]. It is assumed there (and in other existing results) that the *full information* on the target system is available, which may not fit in real situations. In fact, the perfect identification of a large-scale Boolean network is almost impossible because the required number of experimental data exponentially grows with the maximum indegree [23]. So it is often difficult to construct a model of the target system, which prevents us from using the existing results. Moreover, a Boolean network can be regarded as the combination of a *network structure* and a *node dynamics*, and the former has been considered to be more fundamental than the latter especially in biology. This follows from the fact that the former has been actually identified for many biological networks as an activation/inhibition relationship among the elements and the results are archived in databases such as KEGG [24], but the latter has been less investigated. Therefore, in biology, it is desirable to develop a method for analyzing dynamical properties only based on the information on the network structure, i.e., without using the information on the node dynamics.

Motivated by this circumstance, the authors have recently formulated a *structural stability* problem and derived a series of results on the *monostability*, i.e., the existence of a unique equilibrium [25], [26]. On the other hand, in addition to stability, *oscillatority* (that is, instability) is also important in analyzing dynamic functions, especially in living organisms [27], such as circadian rhythm, cell division, and heartbeat. However, the structural oscillatority problem for Boolean networks is still open.

This paper, thus, addresses the structural oscillatority of Boolean networks. In particular, we assume that the network structure represents the interaction among the nodes with the direction of correlation (which corresponds to the activation/inhibition in biology) and consider the problem of finding a class of Boolean networks that are structurally oscillatory. As a solution to the problem, we present a necessary and sufficient condition for Boolean networks with a *cactus* network structure, which is strongly connected and has no edge contained in two or more different simple cycles, to be structurally oscillatory. Structural oscillatority is captured by a simple characterization in terms of the network structure (more concretely, the number of inhibiting edges in each simple cycle), which enables us to apply it to large-scale Boolean networks. Note that this class of Boolean networks is specific but practically important because cactus network structures include typical ones such as cyclic networks, 8-shaped networks, and flower-shaped networks.

For the contribution, we would like to stress the following points. First, our result is useful not only for verifying the structural oscillatority of a given system but also for synthesizing an

Manuscript received September 14, 2016; revised April 21, 2017, July 14, 2017, and April 23, 2018; accepted May 10, 2018. Date of publication May 18, 2018; date of current version May 28, 2019. This work was supported in part by Grant-in-Aid for Scientific Research on Innovative Areas (*Molecular robotics*) #15H00814 and in part by Grant-in-Aid for Scientific Research (B) #17H03280 from the Ministry of Education, Culture, Sports, Science and Technology of Japan. Recommended by Associate Editor S. Hirche. *(Corresponding author: Shun-ichi Azuma.)*

S. Azuma is with the Graduate School of Engineering, Nagoya University, Nagoya 464-8603, Japan (e-mail: shunichi.azuma@mae.nagoya-u.ac.jp).

T. Yoshida is with the Toyota Motor Corporation, Toyota 471-8572, Japan (e-mail: takahiro_yoshida_ah@mail.toyota.co.jp).

T. Sugie is with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: sugie@i.kyoto-u.ac.jp).

Digital Object Identifier 10.1109/TCNS.2018.2838521

oscillatory system with unknown nodes. Second, our result is an important first step to establish a structure-based analysis and design framework of Boolean networks, because it is *not* trivial in the first place to determine whether there exists a structurally oscillatory Boolean network or not, while this paper finds a general class of such systems.

This paper is organized as follows. In Section II, the systems to be studied, i.e., Boolean networks, and the notion of structural oscillatority are introduced. Section III presents a necessary and sufficient condition for the structural oscillatority of Boolean networks with a cactus network structure, and Section IV gives the proof. Finally, Section V concludes this paper.

Note that this paper is based on our preliminary version [28], published in a conference proceedings. However, this journal version generalizes the preliminary result and contains the following new materials: 1) complete proofs omitted in the conference version, 2) a generalized condition for Boolean networks with a cactus network structure to be structurally oscillatory.

Notation: *(i) Sets and vectors*: The cardinality of the finite set $\mathbf{S}$ is denoted by $|\mathbf{S}|$. Given the numbers $x_1, x_2, \ldots, x_n$ and the set $\mathbf{I} := \{i_1, i_2, \ldots, i_m\} \subset \{1, 2, \ldots, n\}$, we use $(x_i)_{i\in\mathbf{I}}$ to represent the tuple $(x_{i_1}, x_{i_2}, \ldots, x_{i_m})$. For instance, if $\mathbf{I} = \{1, 3, 4\}$, $(x_i)_{i\in\mathbf{I}} = (x_1, x_3, x_4)$.

(ii) Boolean functions: For the Boolean variables $x \in \{0, 1\}$ and $y \in \{0, 1\}$, the logical OR and logical AND operations are denoted by $x \vee y$ and $x \wedge y$, respectively. We use $\bar{x}$ to represent the negation of x. For the vector v of Boolean variables, we use $\bar{v}$ to express the element-wise negation of v. The Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ of $x_1, x_2, \ldots, x_n$ is said to be *dependent on* x_i if there exists a tuple $(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, x_{i+2}, \ldots, x_n) \in \{0, 1\}^{n-1}$ satisfying $f(x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) \neq f(x_1, x_2, \ldots, x_{i-1}, \bar{x}_i, x_{i+1}, \ldots, x_n)$. For example, $x_1 \vee x_2$ is dependent on x_1 and x_2, while $(x_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_2) \vee x_3$ is not dependent on x_2 but is dependent on x_1 and x_3 because it is equal to $x_1 \vee x_3$. If f is dependent on all the arguments, it is said to be *minimally represented*.

(iii) Graphs: For the directed graph $G = (\mathbf{V}, \mathbf{E})$, the node sequence $(i_1, i_2, \ldots, i_m) \in \mathbf{V}^m$ is called a *directed path* from i_1 to i_m (or simply called a *path*) if $(i_k, i_{k+1}) \in \mathbf{E}$ holds for every $k \in \{1, 2, \ldots, m-1\}$, where $m \in \{2, 3, \ldots\}$. The path $(i_1, i_2, \ldots, i_m)$ is called a *simple cycle* if $i_1 = i_m$ and $i_1, i_2, \ldots, i_{m-1}$ are distinct from each other. If node $i \in \mathbf{V}$ is a part of the path $(i_1, i_2, \ldots, i_m)$ (or the simple cycle $(i_1, i_2, \ldots, i_m)$ with $i_1 = i_m$), i.e., $i = i_k$ for some $k \in \{1, 2, \ldots, m\}$, the node is said to be *contained in the path* (or *contained in the simple cycle*). The same terminology is applied to edges. The graph G is said to be *strongly connected* if there exists a directed path from i to j for each pair $(i, j) \in \mathbf{V} \times \mathbf{V}$. These notions are similarly defined for *edge-labeled directed graphs*, denoted by $G = (\mathbf{V}, \mathbf{E}, L)$ with a labeling function L for edges. Next, consider the directed graphs $G_1 = (\mathbf{V}_1, \mathbf{E}_1)$ and $G_2 = (\mathbf{V}_2, \mathbf{E}_2)$ where $\mathbf{V}_1$ and $\mathbf{V}_2$ are node sets and $\mathbf{E}_1$ and $\mathbf{E}_2$ are edge sets. If there exists a one-to-one mapping $h : \mathbf{V}_1 \to \mathbf{V}_2$ such that $(h(v), h(w)) \in \mathbf{E}_2$ for each $(v, w) \in \mathbf{E}_1$, the graphs are said to be *isomorphic*.

II. Problem Formulation

A. System Description

Consider the Boolean network with n nodes, given by

$$x_i(t+1) = f_i((x_j(t))_{j\in\mathbf{N}_i}, (\bar{x}_j(t))_{j\in\bar{\mathbf{N}}_i}) \quad (i = 1, 2, \ldots, n) \tag{1}$$

where $x_i(t) \in \{0, 1\}$ is the state of node i. The set $\mathbf{N}_i \subseteq \{1, 2, \ldots, n\}$ contains the indices of *activating neighbor nodes* whose state affects the update of the state of node i, while $\bar{\mathbf{N}}_i \subseteq \{1, 2, \ldots, n\}$ contains those of *inhibiting neighbor nodes* whose negated state affects the update of the state of node i. When (1) represents a gene regulatory network, the activating neighbor nodes and inhibiting ones correspond to activating genes and inhibiting genes, respectively. The function $f_i : \{0, 1\}^{|\mathbf{N}_i|} \times \{0, 1\}^{|\bar{\mathbf{N}}_i|} \to \{0, 1\}$ is a Boolean function assumed to

(a) be not dependent on $x_j(t)$ and its negation $\bar{x}_j(t)$ at the same time (i.e., $\mathbf{N}_i \cap \bar{\mathbf{N}}_i = \emptyset$);
(b) be minimally represented [under (a)];
(c) be composed of logical AND and OR operators; and
(d) be identical to 0 or 1 if $\mathbf{N}_i = \bar{\mathbf{N}}_i = \emptyset$.

Under (a), (c), and (d), f_i is a *sign-definite* function.[1] The Boolean networks composed of sign-definite functions are often called *regulatory Boolean networks* [7], which are rather restrictive but are known to be an important class of systems. Meanwhile, (b) implies that each Boolean function f_i does not contain any redundant argument.

For the Boolean network given by

$$\begin{cases} x_1(t+1) = x_2(t) \\ x_2(t+1) = x_4(t) \vee \bar{x}_3(t) \\ x_3(t+1) = x_4(t) \\ x_4(t+1) = \bar{x}_1(t) \end{cases} \tag{2}$$

we have $\mathbf{N}_1 = \{2\}, \bar{\mathbf{N}}_1 = \emptyset, \mathbf{N}_2 = \{4\}, \bar{\mathbf{N}}_2 = \{3\}, \mathbf{N}_3 = \{4\}, \bar{\mathbf{N}}_3 = \emptyset, \mathbf{N}_4 = \emptyset, \bar{\mathbf{N}}_4 = \{1\}$, $f_1(x_2) = x_2$, $f_2(x_4, \bar{x}_3) = x_4 \vee \bar{x}_3$, $f_3(x_4) = x_4$, and $f_4(\bar{x}_1) = \bar{x}_1$. It is clear that (a)–(d) hold for f_i $(i = 1, 2, 3, 4)$.

Note here that, for instance, $f_i((x_j(t))_{j\in\mathbf{N}_i}, (\bar{x}_j(t))_{j\in\bar{\mathbf{N}}_i}) := (x_1(t) \vee x_2(t)) \wedge (\bar{x}_1(t) \vee \bar{x}_2(t))$, which corresponds to the exclusive-OR operation for $x_1(t)$ and $x_2(t)$, is excluded in this paper because (b)–(d) hold but (a) does not hold.

The Boolean network in (1) can be regarded as the combination of a *network structure* and a *node dynamics*. So we denote the system by

$$\Sigma(G, F)$$

where G and F represent the *network structure* and *node dynamics*, respectively, which are defined based on (1) as follows.

[1] In general, the Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ is said to be *monotone* with respect to $i \in \{1, 2, \ldots, n\}$ if (a) $f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \leq f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$ for every $(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, x_{i+2}, \ldots, x_n) \in \{0, 1\}^{n-1}$ or (b) $f(x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n) \geq f(x_1, x_2, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$ for every $(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, x_{i+2}, \ldots, x_n) \in \{0, 1\}^{n-1}$. In particular, f is said to be *sign-definite* if it is monotone for every $i \in \{1, 2, \ldots, n\}$ [7].

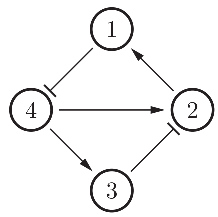

Fig. 1. Network structure G of the Boolean network in (2).

1) $G = (\mathbf{V}, \mathbf{E}, L)$ is the edge-labeled directed graph with the node set $\mathbf{V} := \{1, 2, \ldots, n\}$, the edge set $\mathbf{E} := \{(j, i) \in \{1, 2, \ldots, n\}^2 | j \in \mathbf{N}_i \cup \bar{\mathbf{N}}_i\}$, and the labeling function $L : \mathbf{E} \to \{-1, 1\}$ satisfying $L(j, i) = 1$ for $j \in \mathbf{N}_i$ and $L(j, i) = -1$ for $j \in \bar{\mathbf{N}}_i$.
2) $F : \{0,1\}^{|\bigcup_{i=1}^n \mathbf{N}_i|} \times \{0,1\}^{|\bigcup_{i=1}^n \bar{\mathbf{N}}_i|} \to \{0,1\}^n$ is the collection of the Boolean functions $f_1, f_2, \ldots, f_n$.

The network structure G is the information on whether a certain node affects the update of another one via its logical state or the opposite of it. The node dynamics F, on the other hand, is the information about the explicit logical expression of each function f_i. For example, for system (2), G is given by Fig. 1 and F is the tuple (f_1, f_2, f_3, f_4) for the functions f_i $(i = 1, 2, 3, 4)$ shown after (2). Note in Fig. 1 that the edges with label 1 are depicted by $\to$ and those with -1 are denoted by $\dashv$.

In the following part, the edges with label 1 are called *activating edges*, while those with -1 are called *inhibiting edges*. Moreover, if node j is an activating or inhibiting neighbor node of node i, i.e., $j \in \mathbf{N}_i \cup \bar{\mathbf{N}}_i$, node j is called a *neighbor node* of node i. And also, the node dynamics F is said to be *regular* if the Boolean functions f_i $(i = 1, 2, \ldots, n)$ satisfy the conditions (a)–(d).

B. Attractors and Structural Equivalence

In considering the behavior of $\Sigma(G, F)$ as a structural property, we need the notions of *attractors* and *structural equivalence*.

Since $\Sigma(G, F)$ has only 2^n possible state values, the trajectory reaches a set in the state space. Such a set is called an *attractor*. More precisely, the notion is defined as follows. Let $x(t) \in \{0,1\}^n$ be the state of $\Sigma(G, F)$, i.e., $x(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_n(t)]^\top$. For each initial state $x(0) \in \{0,1\}^n$, there exist a finite time $t \in \{0, 1, \ldots\}$ and a sequence $(a_1, a_2, \ldots, a_l) \in \{0,1\}^n \times \{0,1\}^n \times \cdots \times \{0,1\}^n$ of distinct Boolean vectors such that $x(t+1) = a_1, x(t+2) = a_2, \ldots, x(t+l) = a_l, x(t+l+1) = a_1, x(t+l+2) = a_2, \ldots$. The sequence $(a_1, a_2, \ldots, a_l)$ is called an *attractor of length* l. For instance, the system in (2) has the following three attractors of length 1, length 2, and length 4, respectively: $([1 \quad 1 \quad 0 \quad 0]^\top)$, $([0 \quad 1 \quad 1 \quad 0]^\top, [1 \quad 0 \quad 0 \quad 1]^\top)$, and $([1\ 1\ 1\ 0]^\top, [1\ 0\ 0\ 0]^\top, [0\ 1\ 0\ 0]^\top, [1\ 1\ 0\ 1]^\top)$. In particular, we call here an attractor of length 1 a *point attractor*.

Next, the notion of structural equivalence is given. Consider two Boolean networks $\Sigma(G_1, F_1)$ and $\Sigma(G_2, F_2)$ with regular node dynamics F_1 and F_2 (see the end of Section II-A for the definition of regular node dynamics). If the systems share the same network structure, i.e., $G_1 = G_2$, they are said to be *structurally equivalent*. For instance, the system in (2) is structurally equivalent to

$$\begin{cases} x_1(t+1) = x_2(t) \\ x_2(t+1) = x_4(t) \wedge \bar{x}_3(t) \\ x_3(t+1) = x_4(t) \\ x_4(t+1) = \bar{x}_1(t). \end{cases} \tag{3}$$

In fact, both (2) and (3) have the same network structure as shown in Fig. 1 and have regular node dynamics, although their dynamics of node 2 are different from each other.

Note here that a Boolean network is structurally equivalent to itself.

C. Notion of Structural Oscillatority

Now, the notions of *oscillatority* and *structural oscillatority* are introduced for the Boolean network $\Sigma(G, F)$.

Definition 1: (i) The Boolean network $\Sigma(G, F)$ is said to be *oscillatory* if there exists no point attractor in $\Sigma(G, F)$.

(ii) The Boolean network $\Sigma(G, F)$ is said to be *structurally oscillatory* if *all* the Boolean networks which are structurally equivalent to $\Sigma(G, F)$ are oscillatory.

Two remarks are given.

First, oscillatority and structurally oscillatory imply that the system oscillates for any initial state. These notions are useful for, e.g., the case when one needs an oscillator but the initial state cannot be configured or an unexpected state jump sometimes occurs due to disturbances.

Second, oscillatority is a system property depending on both the network structure G and the node dynamics F. In contrast, structural oscillatority depends only upon G. So structural oscillatority is a useful concept when the node dynamics F is unknown.

Oscillatority and structural oscillatority are illustrated in the following.

Example 1: Consider the following Boolean network $\Sigma(G, F)$ whose network structure is shown in Fig. 2:

$$\begin{cases} x_1(t+1) = x_3(t) \wedge \bar{x}_4(t) \\ x_2(t+1) = x_1(t) \\ x_3(t+1) = \bar{x}_2(t) \\ x_4(t+1) = x_1(t). \end{cases} \tag{4}$$

There exist only two systems which are structurally equivalent to $\Sigma(G, F)$: itself and the system obtained by replacing $\wedge$ with $\vee$ in

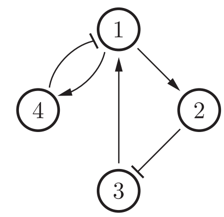

Fig. 2. Network structure G of the Boolean network in (4).

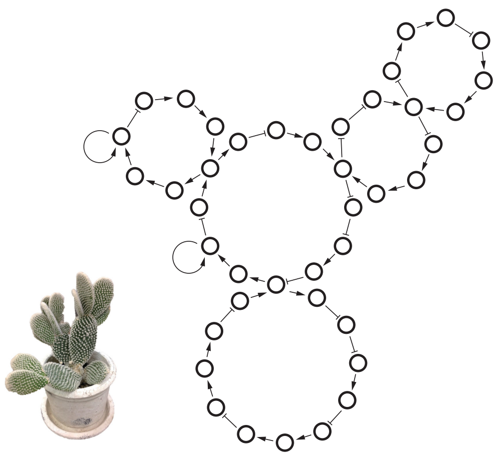

Fig. 3. Example of cactus network structures.

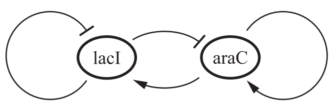

Fig. 4. Genetic network of Laci-Arac oscillator.

(4). By calculating the state transition for each $x(0) \in \{0,1\}^4$, it turns out that the former system has the attractor of length 5, that is, $([0\ \ 0\ \ 0\ \ 0]^\top, [0\ \ 0\ \ 1\ \ 0]^\top, [1\ \ 0\ \ 1\ \ 0]^\top, [1\ \ 1\ \ 1\ \ 1]^\top, [0\ \ 1\ \ 0\ \ 1]^\top)$, as a unique attractor, which implies that it is oscillatory. In a similar way, it follows that the latter has the attractor $([0\ \ 0\ \ 0\ \ 0]^\top, [1\ \ 0\ \ 1\ \ 0]^\top, [1\ \ 1\ \ 1\ \ 1]^\top, [1\ \ 1\ \ 0\ \ 1]^\top, [0\ \ 1\ \ 0\ \ 1]^\top)$ of length 5, as a unique attractor. So it is also oscillatory. Since both the structurally equivalent systems are oscillatory, $\Sigma(G,F)$ is structurally oscillatory.

Note that if $\Sigma(G,F)$ is not oscillatory, it is not structurally oscillatory, but not vice versa. Note also that if $\Sigma(G,F)$ is structurally oscillatory, any system which is structurally equivalent to $\Sigma(G,F)$ is structurally oscillatory, and vice versa.

III. Structural Oscillatority Condition for Cactus Boolean Networks

Consider the Boolean network $\Sigma(G,F)$ in (1). If the network structure G is strongly connected and has no edge contained in two or more different simple cycles as illustrated in Fig. 3, then G is said to be a *cactus* and $\Sigma(G,F)$ is called a *cactus Boolean network* (see the notation part in Section I for the definitions of strongly connected graphs and simple cycles). For example, the network structure G in Fig. 2 is a cactus, while G in Fig. 1 is not a cactus because edge $(4,1)$ [and $(1,2)$] belongs to both simple cycles $(1,2,4)$ and $(1,2,3,4)$.

This notion is a directed variant of cactus graphs studied in [29], and, in the directed case, it is well known that G is composed of simple cycles in which two simple cycles have at most one node in common [29], [30]. This class of network structures includes typical networks such as cyclic networks, 8-shaped networks, and flower-shaped networks [26]. For example, cactus network structures are found in biology, as illustrated in Fig. 4.

For cactus Boolean networks, a necessary and sufficient condition for structural oscillatority is obtained as follows.

Theorem 1: The cactus Boolean network $\Sigma(G,F)$ is structurally oscillatory if and only if all the simple cycles of G contain an odd number of inhibiting edges.

Proof: See Section IV. ■

Theorem 1 captures the structural oscillatority of the cactus Boolean networks by a simple characterization in terms of the network structure, i.e., the number of inhibiting edges. Since this condition can be applied to large-scale Boolean networks (i.e., with large n), it is useful not only for checking the structural oscillatority but also for designing an oscillatory Boolean network with unknown node dynamics.

Example 2: Consider the Boolean network $\Sigma(G,F)$ in (4). As mentioned above, the network structure G is a cactus and both simple cycles $(1,2,3)$ and $(1,4)$ contain a single inhibiting edge. By Theorem 1, $\Sigma(G,F)$ is structurally oscillatory.

Example 3: Consider a Boolean network $\Sigma(G,F)$ whose network structure is given in Fig. 3. It is not structurally oscillatory by Theorem 1, because there exists a simple cycle containing an even number of inhibiting edges (in fact, a simple cycle includes two inhibiting edges and two simple cycles include no inhibiting edge).

We next give an example of a biological network.

Example 4: Fig. 4 shows the genetic network of the Laci-Arac oscillator [31], which is a synthetic oscillator in the bacterium *Escherichia coli*. According to the dynamics described in [31], a Boolean network model is derived as follows:

$$\begin{cases} x_1(t+1) = \bar{x}_1(t) \vee x_2(t) \\ x_2(t+1) = \bar{x}_1(t) \wedge x_2(t) \end{cases} \tag{5}$$

where $x_1(t) \in \{0,1\}$ and $x_2(t) \in \{0,1\}$ are the expression states of lacI and araC genes, respectively. As shown in the figure, the network structure is a cactus and the self-loop on araC contains no inhibiting edge. This fact and Theorem 1 imply that the system is not structurally oscillatory. In fact, it is easy to see that (5) has no point attractor but the structurally equivalent system, which is obtained by replacing $\wedge$ with $\vee$ in the dynamics of node 2, has the point attractor $([1\ \ 1]^\top)$. This consequence suggests that it is important to design the node dynamics carefully in order to generate oscillation in the system. Meanwhile, another option is to remove the self-loop on araC in some way to make the system structurally oscillatory.

IV. Proof of Theorem 1

A. State-Transition Graph and Oscillatority

As a preliminary, we present the notions of *state-transition graphs* and their *isomorphism*, and show the relation with oscillatority.

Let us express the Boolean network $\Sigma(G,F)$ in (1) as

$$x(t+1) = f(x(t), \bar{x}(t))$$

where $f\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$, and $x(t) = [x_1(t)\ \ x_2(t)\ \ \cdots\ \ x_n(t)]^\top \in \{0,1\}^n$ and $\bar{x}(t)$ is the element-wise negation of $x(t)$ as defined before. The state transition of $\Sigma(G,F)$ can be represented by the directed graph with node set $\{0,1\}^n$ and edge set $\{(x,x_+) \in \{0,1\}^n \times \{0,1\}^n | x_+ = f(x,\bar{x})\}$. This graph is called the *state transition graph*.

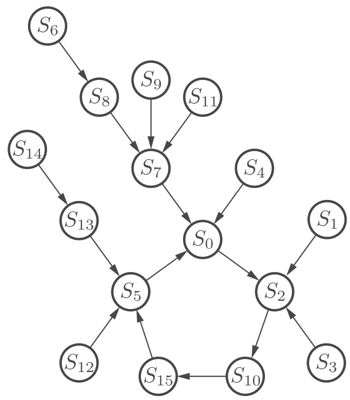

Fig. 5. State transition graph of the Boolean network in (4).

For instance, the state transition graph of (4) is given in Fig. 5 where $S_0, S_1, \ldots, S_{15}$ correspond to the state values $[0\ 0\ 0\ 0]^\top, [0\ 0\ 0\ 1]^\top, \ldots, [1\ 1\ 1\ 1]^\top$, respectively. Since $\Sigma(G, F)$ is a finite-state machine, the state transition graph has at least one cycle, and $\Sigma(G, F)$ is oscillatory if and only if the state transition graph has no cycle of length 1.

Based on the state transition graphs, we can introduce an equivalence relation for Boolean networks. Two Boolean networks $\Sigma(G_1, F_1)$ and $\Sigma(G_2, F_2)$ are said to be *state-transition-isomorphic* if their state transition graphs are isomorphic. The relation is denoted by $\Sigma(G_1, F_1) \sim \Sigma(G_2, F_2)$ or $\Sigma(G_2, F_2) \sim \Sigma(G_1, F_1)$.

The following result is a straightforward consequence of the definitions of oscillatority [Definition 1(i)] and state-transition isomorphism.

Lemma 1: Assume $\Sigma(G_1, F_1) \sim \Sigma(G_2, F_2)$. The system $\Sigma(G_1, F_1)$ is oscillatory if and only if $\Sigma(G_2, F_2)$ is oscillatory.

It is remarked that a similar statement may not hold for the *structural* oscillatority because the isomorphism is not on network structures but on state-transition graphs (more precisely, it is an open problem whether a similar statement holds or not for the structural oscillatority).

B. Main Part of Proof

We first prove the necessity of Theorem 1 by showing the contraposition: if G has a simple cycle including an even number of inhibiting edges, then $\Sigma(G, F)$ is not structurally oscillatory.

Consider the cactus Boolean network $\Sigma(G, F)$ where G has a simple cycle including an even number of inhibiting edges. For the system, we introduce three cactus Boolean networks $\Sigma(G^*, F_1)$, $\Sigma(G^*, F_2)$, and $\Sigma(G, F_3)$ (i.e., introduce a tuple (G^*, F_1, F_2, F_3)) such that they satisfy the following conditions:

(C1) $\Sigma(G, F) \sim \Sigma(G^*, F_1)$ and $\Sigma(G^*, F_2) \sim \Sigma(G, F_3)$;

(C2) G^* is a cactus satisfying the following two properties: (a) there exists no node such that the indegree is greater than 1 and the incoming edges are all inhibiting, (b) for each node of indegree 1, the incoming edge is activating;

(C3) F_2 contains only logical AND operators (i.e., contains no logical OR operator);

(C4) F_i $(i = 1, 2, 3)$ are regular.

$$\begin{tikzcd}
\Sigma(G,F) \sim \Sigma(G^*,F_1) \\
\Sigma(G,F_3) \sim \Sigma(G^*,F_2)
\end{tikzcd}$$

structurally-equivalent structurally-equivalent

Fig. 6. Relation among $\Sigma(G, F)$, $\Sigma(G^*, F_1)$, $\Sigma(G^*, F_2)$, and $\Sigma(G, F_3)$.

Note here that, $\Sigma(G, F)$ and $\Sigma(G, F_3)$ are structurally equivalent and also $\Sigma(G^*, F_1)$ and $\Sigma(G^*, F_2)$ are structurally equivalent under (C4). The relation among $\Sigma(G, F)$ and the other three systems is shown in Fig. 6. We show in Appendix I that there exist $\Sigma(G^*, F_1)$, $\Sigma(G^*, F_2)$, and $\Sigma(G, F_3)$ (i.e., G^*, F_1, F_2, and F_3) satisfying (C1)–(C4) if G is a cactus, which has a simple cycle including an even number of inhibiting edges.

Since (C3) implies that the dynamics of $\Sigma(G^*, F_2)$ is described only with the logical AND operators, it turns out from (C2) and (C3) that $([0\ 0\ \cdots\ 0]^\top)$ is an attractor of $\Sigma(G^*, F_2)$, which implies that $\Sigma(G^*, F_2)$ is not oscillatory. So it follows from the latter condition of (C1) (i.e., $\Sigma(G^*, F_2) \sim \Sigma(G, F_3)$) and Lemma 1 that $\Sigma(G, F_3)$ is not oscillatory. This fact, the structural equivalence between $\Sigma(G, F_3)$ and $\Sigma(G, F)$ (see Fig. 6), and the definition of structural oscillatority [Definition 1(ii)] prove that $\Sigma(G, F)$ is not structurally oscillatory.

On the other hand, the sufficiency is proven by the existing result in [7] ([7, Theorem 2]). The result has shown that the following fact holds for the regulatory Boolean networks: if $\Sigma(G, F)$ has a point attractor, then there exists a simple cycle containing an even number of inhibiting edges in G. Moreover, $\Sigma(G, F)$ is a regulatory Boolean network as stated in Section II-A. These prove the sufficiency of Theorem 1.

V. Conclusion

In this paper, structural oscillatority has been discussed for Boolean networks. A necessary and sufficient condition has been presented for the class of Boolean networks with a cactus network structure. It characterizes the structural oscillatority by a simple characteristic (the number of inhibiting edges) of the network structure. This result allows us not only to verify the structural oscillatority but also to synthesize an oscillatory Boolean network with unknown nodes.

In the future, we plan to generalize our framework to handle a more general class of Boolean networks. We have recently found that a general class of Boolean networks can be equivalently transformed into a Boolean network with a cactus network structure [32], and such an idea may be useful for the generalization. Meanwhile, it is also interesting to address design problems for Boolean networks with unknown nodes.

Appendix I
Existence of Boolean Networks $\Sigma(G^*, F_1)$, $\Sigma(G^*, F_2)$, and $\Sigma(G, F_3)$ in Section IV-B

As stated in Section IV-B, there exist cactus Boolean networks $\Sigma(G^*, F_1)$, $\Sigma(G^*, F_2)$, and $\Sigma(G, F_3)$ satisfying (C1)–(C4) if the cactus Boolean network $\Sigma(G, F)$ has a network structure G with a simple cycle including an even number of inhibiting

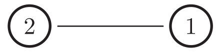

Fig. 7. Undirected graph $\hat{G}$ and spanning tree $\tilde{G}$ for G in Fig. 2.

edges. To prove this fact, we show here how to derive $\Sigma(G^{*}, F_1)$, $\Sigma(G^{*}, F_2)$, and $\Sigma(G, F_3)$.

Before moving to the main part, we describe a brief outline of this section. As a preliminary, we first present some notions for cactus network structures in Appendix I-A. They are useful for addressing the connection among simple cycles contained in a cactus network structure. In Appendix I-B, we introduce a transformation (Algorithm 1), denoted by T_i, for Boolean networks in the form of (1), from which a transformation, called the *cycle transformation* and denoted by $C_{\ell i}$, is defined for *cactus* Boolean networks. Several properties of the transformations T_i and $C_{\ell i}$ are given in Lemmas 3–9. The cycle transformation $C_{\ell i}$ is used for obtaining $\Sigma(G^{*}, F_1)$ (in Algorithm 2) and the transformation T_i is used for obtaining $\Sigma(G, F_3)$. Finally, Appendix I-C provides the systems $\Sigma(G^{*}, F_1)$, $\Sigma(G^{*}, F_2)$, and $\Sigma(G, F_3)$ satisfying (C1)–(C4) by using Lemmas 2– 4, 6, 8, and 9 (some of which are derived by Lemmas 5 and 7).

A. Parent, Child, and Depth of Simple Cycles in Cactus Network Structures

We first present several notions for *cactus* network structures.

Consider the cactus network structure $G = (\mathbf{V}, \mathbf{E}, L)$, where $\mathbf{V} := \{1, 2, \dots, n\}$ is the node set, $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ is the edge set, and $L : \mathbf{E} \to \{-1, 1\}$ is the labeling function. As stated in Section III, G is composed of simple cycles in which two simple cycles have at most one node in common. This implies that each block[2] of G is a simple cycle [29], [30]. So we call each simple cycle of G *cycle* ℓ ($\ell = 1, 2, \dots, s$) in this section (throughout Appendix I), where s is the number of simple cycles.

In the cactus network structure G, cycles ℓ_1 and ℓ_2 are said to be *adjacent* if they are distinct and have a node in common. Let $\hat{G} = (\hat{\mathbf{V}}, \hat{\mathbf{E}})$ be the undirected graph, which describes the adjacency relation of simple cycles in G, i.e., $\hat{\mathbf{V}} := \{1, 2, \dots, s\}$ and $\hat{\mathbf{E}}$ is defined as the set of unordered pairs $\{\ell_1, \ell_2\}$ such that cycles ℓ_1 and ℓ_2 are adjacent. Since $\hat{G}$ is a connected undirected graph by definition, $\hat{G}$ has a spanning tree. So we arbitrarily pick a spanning tree from the undirected graph $\hat{G}$ and denote it by $\tilde{G}$. For example, both $\hat{G}$ and $\tilde{G}$ are given by Fig. 7 for the cactus network structure G in Fig. 2, where cycles 1 and 2 indicate $(1, 2, 3, 1)$ and $(1, 4, 1)$, respectively.

Now, we introduce the notions of a *parent*, a *child*, and *depth* for simple cycles of G, in a similar way to the well-known notions for an undirected tree with a root. Assume that a cycle $\ell_0 \in \{1, 2, \dots, s\}$ is arbitrarily chosen and referred to as the *root cycle*. Cycle ℓ_1 is called a *parent* of cycle ℓ_2 if node ℓ_1 is a parent of node ℓ_2 in the undirected tree $\tilde{G}$ with the root ℓ_0. In a similar way, cycle ℓ_1 is called a *child* of cycle ℓ_2 if node ℓ_1 is a child of node ℓ_2 in the rooted tree $\tilde{G}$. The *depth of cycle* ℓ_1 in G is defined as the depth of node ℓ_1 in the rooted tree $\tilde{G}$. Note that these depend on the choice of the root cycle ℓ_0.

The following lemma is straightforwardly obtained from a well-known property of rooted trees [33].

Lemma 2: For the cactus network structure G, let cycle $\ell_0 \in \{1, 2, \dots, s\}$ be the root cycle. Each cycle $\ell \in \{1, 2, \dots, s\} \setminus \{\ell_0\}$ (except for the root cycle) has a unique parent.

B. Transformations of Boolean Networks Preserving State-Transition Isomorphism

1) Transformation T_i: For the Boolean network $\Sigma(G, F)$ in (1) (whose network structure is not necessarily a cactus), consider the following transformation with respect to node $i \in \mathbf{V}$.

Algorithm 1 (Transformation T_i):

(Step 1) Invert the labels (i.e., change 1 into -1 and conversely) of the edges connected to node i (the incoming and outgoing edges of node i).

(Step 2) In the dynamics of node i, i.e., in the function f_i, convert each logical AND operator, logical OR operator, binary number 0, and number 1 into a logical OR operator, logical AND operator, number 1, and number 0, respectively.

The resulting system is denoted by $T_i(\Sigma(G, F))$.

For instance, we obtain $T_1(\Sigma(G, F))$ as

$$\begin{cases} x_1(t+1) = x_4(t) \vee \bar{x}_3(t) \\ x_2(t+1) = \bar{x}_1(t) \\ x_3(t+1) = \bar{x}_2(t) \\ x_4(t+1) = \bar{x}_1(t) \end{cases}$$

for the system $\Sigma(G, F)$ in (4).

The composition $T_j(T_i(\Sigma(G, F)))$ is denoted by $T_j \circ T_i(\Sigma(G, F))$. Similarly, the composition of multiple transformations $T_{i_1}, T_{i_2}, \dots, T_{i_\nu}$ is expressed by $T_{i_\nu} \circ T_{i_{\nu-1}} \circ \dots \circ T_{i_1}(\Sigma(G, F))$.

The following lemmas show basic properties of the transformation T_i.

Lemma 3 ([26]): For the Boolean network $\Sigma(G, F)$, suppose that $i \in \mathbf{V}$ is given. Then, $\Sigma(G, F) \sim T_i(\Sigma(G, F))$.

Note here that this lemma and the definition of the state-transition isomorphism imply that

$$\Sigma(G, F) \sim T_{i_\nu} \circ T_{i_{\nu-1}} \circ \dots \circ T_{i_1}(\Sigma(G, F)) \tag{6}$$

for any $\nu \in \{1, 2, \dots\}$ and $(i_1, i_2, \dots, i_\nu) \in \mathbf{V}^{\nu}$.

Lemma 4: For the Boolean network $\Sigma(G, F)$ (with $G = (\mathbf{V}, \mathbf{E}, L)$), suppose that $i \in \mathbf{V}$ is given. Let $G^{\star} = (\mathbf{V}^{\star}, \mathbf{E}^{\star}, L^{\star})$ and $F^{\star}$ be the network structure and node dynamics of $T_i(\Sigma(G, F))$, respectively. Then, the following statements hold.

(i) $F^{\star}$ is regular.

(ii) $\mathbf{V}^{\star} = \mathbf{V}$ and $\mathbf{E}^{\star} = \mathbf{E}$.

Proof: Let $f_1, f_2, \dots, f_n$ and $f_1^{\star}, f_2^{\star}, \dots, f_n^{\star}$ denote the Boolean functions contained in F and $F^{\star}$, respectively.

(i) It is trivial from the two steps of Algorithm 1 that if the conditions (a), (c), and (d) in Section II-A hold for $f_1, f_2, \dots, f_n$, they also hold for $f_1^{\star}, f_2^{\star}, \dots, f_n^{\star}$.

Next, consider the condition (b) in Section II-A for $f_1^{\star}, f_2^{\star}, \dots, f_n^{\star}$. From Algorithm 1, node i of $T_i(\Sigma(G, F))$

[2] A graph $G = (\mathbf{V}, \mathbf{E})$ is said to be *biconnected* if, for each pair $(i, j) \in \mathbf{V} \times \mathbf{V}$, there exist two directed paths from i to j which have no node in common other than i and j. For a graph H, a maximal biconnected subgraph is called a *block*[33].

has the dynamics $x_i(t+1) = f_i^\star((\bar{x}_j(t))_{j\in\mathbf{N}_i}, (x_j(t))_{j\in\bar{\mathbf{N}}_i})$. It follows from Step 2 of Algorithm 1 and De Morgan's laws that $f_i^\star((\bar{x}_j(t))_{j\in\mathbf{N}_i}, (x_j(t))_{j\in\bar{\mathbf{N}}_i})$ is equal to the negation of $f_i((x_j(t))_{j\in\mathbf{N}_i}, (\bar{x}_j(t))_{j\in\bar{\mathbf{N}}_i})$ [26]. Thus, if the condition (b) holds for f_i, it also holds for $f_i^\star$. On the other hand, for node $j \in \mathbf{V}\backslash\{i\}$, f_j and $f_j^\star$ are the same from Algorithm 1, while their arguments are not necessarily the same in the following sense: if node i is a neighbor node of node j, one of f_j and $f_j^\star$ includes $x_i(t)$ and the other includes $\bar{x}_i(t)$. In this case, by considering the definition of minimal representation, it is clear that if (b) holds for f_j, it also holds for $f_j^\star$. So we can conclude that $F^\star$ is regular.

(ii) It is trivial that Step 1 of Algorithm 1 converts only the labeling function L to $L^\star$. On the other hand, Step 2 and the former statement, i.e., (i), imply that, for each $j \in \mathbf{V}$, $f_j^\star$ is minimally represented with the state or negated state of the same neighbor nodes as of the node j of $\Sigma(G,F)$ (see the end of Section II-A for the definition of *neighbor nodes*). So we obtain (ii). ■

This lemma implies that the transformation T_i converts a Boolean network into a Boolean network while preserving the regularity of node dynamics and the neighbor nodes for each node (where note that T_i may change how the neighbor nodes affect each node, i.e., activation or inhibition). So if $\Sigma(G,F)$ is a cactus Boolean network, then $T_i(\Sigma(G,F))$ is a cactus Boolean network. Moreover, it turns out from (ii) that if $\Sigma(G,F)$ has a simple cycle, then $T_i(\Sigma(G,F))$ has the same simple cycle. So, in the following part, when a simple cycle of $\Sigma(G,F)$ is called *cycle* ℓ, the corresponding cycle of $T_i(\Sigma(G,F))$ is also called *cycle* ℓ, and vice versa.

The next lemma presents useful properties of the transformation T_i for *cactus* Boolean networks.

Lemma 5: For the cactus Boolean network $\Sigma(G,F)$, suppose that $i \in \mathbf{V}$ is given. Then, the following statements hold.

(i) Assume that node i is contained in a simple cycle, called cycle ℓ. Cycle ℓ of $\Sigma(G,F)$ contains an even number of inhibiting edges if and only if cycle ℓ of $T_i(\Sigma(G,F))$ contains an even number of inhibiting edges.

(ii) Let $G^\star$ be the network structure of $T_i(\Sigma(G,F))$. Suppose that a regular node dynamics $\tilde{F}$ is given so that $\Sigma(G^\star,\tilde{F})$ is a Boolean network in the class of Section II-A. Then, the network structure of $T_i(\Sigma(G^\star,\tilde{F}))$ is equal to G.

Proof: (i) Consider $\Sigma(G,F)$. Since G is a cactus and node i is contained in cycle ℓ, node i has only one incoming edge and only one outgoing edge in cycle ℓ (if it has other incoming or outgoing edges, the edges are contained in other simple cycles). On the other hand, the transformation T_i inverts the labels of the incoming and outgoing edges of node i. These imply that the transformation T_i adds two inhibiting edges, subtracts two inhibiting edges, or does not change the number of inhibiting edges, *in cycle* ℓ. This proves (i).

(ii) Step 1 of Algorithm 1 is a mapping, which is a one-to-one correspondence between G and $G^\star$ and whose inverse mapping is itself. Moreover, it follows from Step 2 that each node of $T_i(\Sigma(G^\star,\tilde{F}))$ has the same neighbor nodes as of the corresponding node of $\Sigma(G^\star,\tilde{F})$, and then it turns out from Lemma 4(i) and the regularity of $\tilde{F}$ that all the Boolean functions of $T_i(\Sigma(G^\star,\tilde{F}))$ are minimally represented. Thus, (ii) is given. ■

Moreover, Lemma 5(ii) is extended as follows.

Lemma 6: For the cactus Boolean network $\Sigma(G,F)$, suppose that $\nu \in \{1,2,\ldots\}$ and $(i_1,i_2,\ldots,i_\nu) \in \mathbf{V}^\nu$ are given. Let $G^\star$ be the network structure of $T_{i_\nu} \circ T_{i_{\nu-1}} \circ \cdots \circ T_{i_1}(\Sigma(G,F))$. Suppose that a regular node dynamics $\tilde{F}$ is given so that $\Sigma(G^\star,\tilde{F})$ is a Boolean network in the class of Section II-A. Then, the network structure of $T_{i_1} \circ T_{i_2} \circ \cdots \circ T_{i_\nu}(\Sigma(G^\star,\tilde{F}))$ (which is in the inverse order of $T_{i_\nu} \circ T_{i_{\nu-1}} \circ \cdots \circ T_{i_1}$) is equal to G.

Proof: For $k \in \{1,2,\ldots,\nu\}$, let $G_k^\star$ be the network structure of $T_{i_k} \circ T_{i_{k-1}} \circ \cdots \circ T_{i_1}(\Sigma(G,F))$ and let $G_0^\star$ be the network structure of $\Sigma(G,F)$. Then, $G_\nu^\star = G^\star$ and $G_0^\star = G$. We prove the statement by mathematical induction.

First, we prove that $T_{i_\nu}(\Sigma(G^\star,\tilde{F}))$ has the network structure $G_{\nu-1}^\star$. By definition, $T_{i_\nu}(T_{i_{\nu-1}} \circ T_{i_{\nu-2}} \circ \cdots \circ T_{i_1}(\Sigma(G,F)))$ has the network structure $G_\nu^\star$ (i.e., $G^\star$), in which $T_{i_{\nu-1}} \circ T_{i_{\nu-2}} \circ \cdots \circ T_{i_1}(\Sigma(G,F))$ has the network structure $G_{\nu-1}^\star$. Hence, it turns out from Lemma 5(ii) and $G_\nu^\star = G^\star$ that $T_{i_\nu}(\Sigma(G^\star,\tilde{F}))$ has the network structure $G_{\nu-1}^\star$.

Next, we show that, for each $k = \nu-1,\nu-2,\ldots,1$, $T_{i_k} \circ T_{i_{k+1}} \circ \cdots \circ T_{i_\nu}(\Sigma(G^\star,\tilde{F}))$ has the network structure $G_{k-1}^\star$ under the assumption that $T_{i_{k+1}} \circ T_{i_{k+2}} \circ \cdots \circ T_{i_\nu}(\Sigma(G^\star,\tilde{F}))$ has the network structure $G_k^\star$. By the definition of $G_k^\star$, $T_{i_k}(T_{i_{k-1}} \circ T_{i_{k-2}} \circ \cdots \circ T_{i_1}(\Sigma(G,F)))$ has the network structure $G_k^\star$. By applying this fact, the above assumption, and Lemma 5(ii) to the systems $T_{i_k}(T_{i_{k+1}} \circ T_{i_{k+2}} \circ \cdots \circ T_{i_\nu}(\Sigma(G^\star,\tilde{F})))$ and $T_{i_k}(T_{i_{k-1}} \circ T_{i_{k-2}} \circ \cdots \circ T_{i_1}(\Sigma(G,F)))$, the network structure of the former system is equal to that of $T_{i_{k-1}} \circ T_{i_{k-2}} \circ \cdots \circ T_{i_1}(\Sigma(G,F))$, i.e., $G_{k-1}^\star$. ■

2) Cycle Transformation $C_{\ell i}$:

Next, we introduce a transformation along a simple cycle of the cactus network structure G, which is called the *cycle transformation*.

Consider the *cactus* Boolean network $\Sigma(G,F)$. For a cycle ℓ and a node i in the cycle, let $\bar{T}_{\ell i}$ be the transformation of $\Sigma(G,F)$ defined as follows: if node i has an inhibiting incoming edge in cycle ℓ, $\bar{T}_{\ell i}$ is equal to T_i; otherwise (if node i has an activating incoming edge in cycle ℓ), it is equal to the identity transformation, which transforms $\Sigma(G,F)$ into itself. Note here that node i has only one incoming edge in cycle ℓ because G is a cactus. The resulting system is denoted by $\bar{T}_{\ell i}(\Sigma(G,F))$ and the composition of multiple transformations is defined in a similar way to the case of T_i.

The *cycle transformation* $C_{\ell i}$ with respect to cycle ℓ and node i (in cycle ℓ) is defined as $\bar{T}_{\ell j_{m_\ell}} \circ \bar{T}_{\ell j_{m_\ell - 1}} \circ \cdots \circ \bar{T}_{\ell j_1}$, where $m_\ell \in \{1,2,\ldots,n\}$ is the positive integer and $j_1,j_2,\ldots,j_{m_\ell} \in \mathbf{V}$ are the nodes such that $(i,j_1,j_2,\ldots,j_{m_\ell},i)$ corresponds to cycle ℓ. The resulting system is denoted by $C_{\ell i}(\Sigma(G,F))$. Note that the cycle transformation $C_{\ell i}$ does *not* contain the transformation $\bar{T}_{\ell i}$, i.e., T_i, which is with respect to node i. Fig. 8 illustrates the network structure for the cycle transformation with respect to cycle 1, defined as $(2,3,\ldots,7,1,2)$, and node 2.

The following lemma shows that $C_{\ell i}$ corresponds to the composition of T_j for some nodes j.

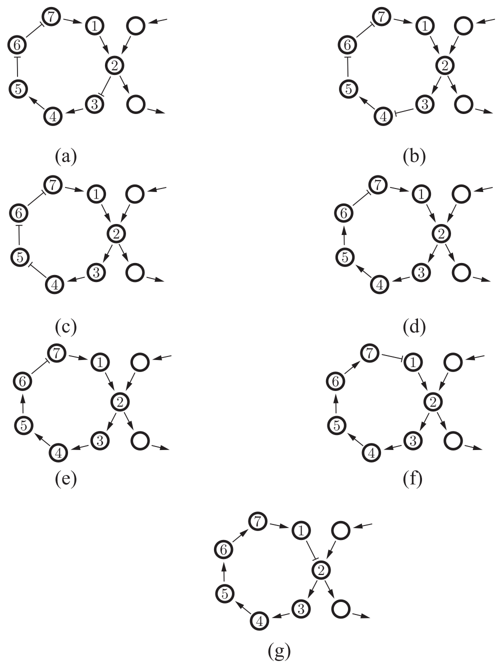

Fig. 8. Cycle transformation $C_{12}(\Sigma(G,F))$ where cycle 1 is $(2,3,\ldots,7,1,2)$. (a) $\Sigma(G,F)$. (b) $\bar{T}_{13}(\Sigma(G,F))$. (c) $\bar{T}_{14}\circ\bar{T}_{13}(\Sigma(G,F))$. (d) $\bar{T}_{15}\circ\bar{T}_{14}\circ\bar{T}_{13}(\Sigma(G,F))$. (e) $\bar{T}_{16}\circ\bar{T}_{15}\circ\cdots\circ\bar{T}_{13}(\Sigma(G,F))$. (f) $\bar{T}_{17}\circ\bar{T}_{16}\circ\cdots\circ\bar{T}_{13}(\Sigma(G,F))$. (g) $C_{12}(\Sigma(G,F))$ [i.e., $\bar{T}_{11}\circ\bar{T}_{17}\circ\bar{T}_{16}\circ\cdots\circ\bar{T}_{13}(\Sigma(G,F))$].

Lemma 7: For the cactus Boolean network $\Sigma(G,F)$, suppose that $\ell\in\{1,2,\ldots,s\}$ and $i\in\mathbf{V}$ are given so that node i is contained in cycle ℓ. Then, there exist a positive integer $\nu\in\{1,2,\ldots\}$ and node sequence $(i_1,i_2,\ldots,i_\nu)\in\mathbf{V}^\nu$ such that $C_{\ell i}(\Sigma(G,F))$ is equal to $T_{i_\nu}\circ T_{i_\nu-1}\circ\cdots\circ T_{i_1}(\Sigma(G,F))$.

Proof: Let $(i,j_1,j_2,\ldots,j_{m_\ell},i)$ denote cycle ℓ.

We first consider the case when there exists no node $j\in\{j_1,j_2,\ldots,j_{m_\ell}\}$, which has an inhibiting incoming edge in cycle ℓ. Then, $C_{\ell i}$ is equal to the identity transformation from the definitions of $C_{\ell i}$ and $\bar{T}_{\ell i}$. On the other hand, for any $i\in\mathbf{V}$, $T_i\circ T_i$ is equal to the identity transformation from the definition of T_i (Algorithm 1). These imply that $C_{\ell i}(\Sigma(G,F))$ is equal to $T_i\circ T_i(\Sigma(G,F))$ for any $i\in\mathbf{V}$.

Next, consider another case (when there exists a node $j\in\{j_1,j_2,\ldots,j_{m_\ell}\}$, which has an inhibiting incoming edge in cycle ℓ). Without loss of generality, assume that $j_1<j_2<\cdots<j_{m_\ell}$. From the node set $\{j_1,j_2,\ldots,j_{m_\ell}\}$, let us extract the nodes, which have an inhibiting incoming edge in cycle ℓ, and let $i_1,i_2,\ldots,i_\nu$ denote the extracted nodes so that $i_1<i_2<\cdots<i_\nu$, where ν is the number of the extracted nodes. Then, it follows from the definitions of $C_{\ell i}$ and $\bar{T}_{\ell i}$ that $C_{\ell i}(\Sigma(G,F))$ is equal to $T_{i_\nu}\circ T_{i_\nu-1}\circ\cdots\circ T_{i_1}(\Sigma(G,F))$. ■

From this result and Lemma 4, it turns out that the resulting system $C_{\ell i}(\Sigma(G,F))$ is a cactus Boolean network, which has the same simple cycles as of $\Sigma(G,F)$. In a similar way to the case of T_i, when a simple cycle of $\Sigma(G,F)$ is called *cycle* ℓ, the corresponding cycle of $C_{\ell i}(\Sigma(G,F))$ is also called *cycle* ℓ, and vice versa.

From (6) and Lemma 7, we obtain the following result on the state-transition isomorphism between $\Sigma(G,F)$ and $C_{\ell i}(\Sigma(G,F))$.

Lemma 8: For the cactus Boolean network $\Sigma(G,F)$, suppose that $\ell\in\{1,2,\ldots,s\}$ and $i\in\mathbf{V}$ are given so that node i is contained in cycle ℓ. Then, $\Sigma(G,F)\sim C_{\ell i}(\Sigma(G,F))$.

Moreover, the following lemma is concerned with the edge labels of the network structure of $C_{\ell i}(\Sigma(G,F))$.

Lemma 9: For the cactus Boolean network $\Sigma(G,F)$, suppose that $\ell\in\{1,2,\ldots,s\}$ and $i\in\mathbf{V}$ are given so that node i is contained in cycle ℓ. Let $\sigma_\ell\in\{0,1,\ldots\}$ be the number of inhibiting edges in cycle ℓ. Then, the following statements hold for edges (and their labels) of the network structure of $C_{\ell i}(\Sigma(G,F))$.

(i) If σ_ℓ is odd, node i of $C_{\ell i}(\Sigma(G,F))$ has an inhibiting incoming edge in cycle ℓ; otherwise, it has an activating incoming edge in cycle ℓ. The other nodes in cycle ℓ have an activating incoming edge in cycle ℓ.

(ii) For $\Sigma(G,F)$, assume that cycle ℓ has a parent and node i is shared by cycle ℓ and its parent (if a parent exists, it is unique from Lemma 2 and there exists a parent of cycle ℓ in $C_{\ell i}(\Sigma(G,F))$). In the parent of cycle ℓ of $C_{\ell i}(\Sigma(G,F))$, the label of any edge is equal to the label of the corresponding edge of $\Sigma(G,F)$.

Proof: (i) Let $(i,j_1,j_2,\ldots,j_{m_\ell},i)$ denote cycle ℓ. First, consider the system $\bar{T}_{\ell j_1}(\Sigma(G,F))$. By the definition of $\bar{T}_{\ell i}$, node j_1 (of $\bar{T}_{\ell j_1}(\Sigma(G,F))$) has an activating incoming edge in cycle ℓ. Next, consider $k\in\{1,2,\ldots,m_\ell-1\}$ and the system $\bar{T}_{\ell j_k}\circ\bar{T}_{\ell j_{k-1}}\circ\cdots\circ\bar{T}_{\ell j_1}(\Sigma(G,F))$. By the definition of $\bar{T}_{\ell i}$, the transformed system $\bar{T}_{\ell j_{k+1}}(\bar{T}_{\ell j_k}\circ\bar{T}_{\ell j_{k-1}}\circ\cdots\circ\bar{T}_{\ell j_1}(\Sigma(G,F)))$ has the following properties: (a) node j_{k+1} has an activating incoming edge in cycle ℓ; (b) each node $j\in\{j_1,j_2,\ldots,j_k\}$ has an incoming edge with the original label (of $\bar{T}_{\ell j_k}\circ\bar{T}_{\ell j_{k-1}}\circ\cdots\circ\bar{T}_{\ell j_1}(\Sigma(G,F))$) in cycle ℓ. These facts (including the facts for each $k\in\{1,2,\ldots,m_\ell-1\}$) and the definition of $C_{\ell i}(\Sigma(G,F))$ prove the latter statement. Moreover, the former statement is straightforwardly obtained from the latter statement, Lemma 5(i), and Lemma 7.

(ii) From Lemma 2, node i is the only common node of cycle ℓ and its parent in $\Sigma(G,F)$. Furthermore, the cycle transformation $C_{\ell i}$ corresponds to the composition of the transformations $\bar{T}_{\ell j}$ for some nodes j but except for i. Therefore, $C_{\ell i}$ does not affect any nodes in the parent of cycle ℓ of $C_{\ell i}(\Sigma(G,F))$. So we obtain (ii). ■

The results are illustrated in Fig. 8(a) and (g) for the case when $\ell=1$, $i=2$, and σ_1 is odd.

C. Derivation of $\Sigma(G^*,F_1)$, $\Sigma(G^*,F_2)$, and $\Sigma(G,F_3)$

Now, we consider the cactus Boolean network $\Sigma(G,F)$ such that G has a simple cycle including an even number of inhibiting edges. Boolean networks $\Sigma(G^*,F_1)$, $\Sigma(G^*,F_2)$, and $\Sigma(G,F_3)$ satisfying (C1)–(C4) are derived from $\Sigma(G,F)$ (i.e., G and F) in the following way.

1) If G itself satisfies the two properties in (C2), $\Sigma(G^*,F_1)$ is given by $G^*:=G$ and $F_1:=F$ (i.e., $\Sigma(G^*,F_1)$ is equal to $\Sigma(G,F)$); otherwise, $\Sigma(G^*,F_1)$ (i.e., the pair (G^*,F_1)) is given as $T_{i_\nu}\circ T_{i_\nu-1}\circ\cdots\circ T_{i_1}(\Sigma(G,F))$ for some positive integer $\nu\in\{1,2,\ldots\}$ and node sequence $(i_1,i_2,\ldots,i_\nu)\in\mathbf{V}^\nu$ (which is detailed later).

2) $\Sigma(G^*, F_2)$ is given by G^* in the previous step and F_2, which is given by modifying F_1 in the previous step so that all the logical OR operators are replaced with the logical AND operators in F_1.
3) If G itself satisfies the two properties in (C2), $\Sigma(G, F_3)$ is given by G of the original system $\Sigma(G, F)$ and $F_3 := F_2$ (i.e., $\Sigma(G, F_3)$ is equal to $\Sigma(G, F_2)$); otherwise, $\Sigma(G, F_3)$ (i.e., the pair (G, F_3)) is given as $T_{i_1} \circ T_{i_2} \circ \cdots \circ T_{i_\nu}(\Sigma(G^*, F_2))$ for ν and $(i_1, i_2, \ldots, i_\nu)$ in the first step, where note that the network structure of $T_{i_1} \circ T_{i_2} \circ \cdots \circ T_{i_\nu}(\Sigma(G^*, F_2))$ becomes equal to G of the original system $\Sigma(G, F)$, as explained later.

Note in the second step that the regularity of the node dynamics is preserved by the replacement of the logical OR operators with the logical AND operators because the resulting node dynamics (of the replacement) is a constant or a product of all the original arguments of F_1.

If G satisfies the two properties in (C2), it is easy to see that (C1)–(C4) hold for the resulting systems $\Sigma(G^*, F_1), \Sigma(G^*, F_2)$, and $\Sigma(G, F_3)$ with $G^* = G$, $F_1 = F$, and $F_3 = F_2$, and the regularity of F_2.

In the following part, we detail how to obtain $\Sigma(G^*, F_1)$ and $\Sigma(G, F_3)$ in the above procedure, in the case when G itself does not satisfy the two properties in (C2).

First, $\Sigma(G^*, F_1)$ is given as the resulting system, denoted by $\Sigma(s)$, of the following algorithm.

Algorithm 2:

(Step 1) In $\Sigma(G, F)$, pick a simple cycle, which contains an even number of inhibiting edges and designate it for the root cycle. Moreover, index the simple cycles of $\Sigma(G, F)$ as $1, 2, \ldots, s$ in an ascending order of the depth of the cycle (where the root cycle becomes cycle 1).

(Step 2) Let $\Sigma(0)$ denote the system $\Sigma(G, F)$.

(Step 3) The following procedure is executed for each $\ell = 1, 2, \ldots, s$ (in ascending order): If $\ell = 1$, let node i be arbitrarily chosen in cycle ℓ; otherwise, let node i be a common node of cycle ℓ and its parent. Let $\Sigma(\ell)$ denote the system $C_{\ell i}(\Sigma(\ell - 1))$.

This algorithm applies the cycle transformation $C_{\ell i}$ to each cycle ℓ. Step 1 is feasible subject to the previously given assumption (G has at least one simple cycle including an even number of inhibiting edges). Step 3 is also feasible because of Lemma 2 and the fact that cycle 1 is the root cycle. It is remarked that the algorithm recursively applies $C_{\ell i}$ to $\Sigma(G, F)$, which, together with Lemma 7, implies that $\Sigma(\ell)$ ($\ell = 1, 2, \ldots, s$) corresponds to $T_{i_\nu} \circ T_{i_\nu - 1} \circ \cdots \circ T_{i_1}(\Sigma(G, F))$ for some positive integer $\nu \in \{1, 2, \ldots\}$ and node sequence $(i_1, i_2, \ldots, i_\nu) \in \mathbf{V}^\nu$.

For the resulting system $\Sigma(s)$, we obtain the following result.

Lemma 10: For the cactus Boolean network $\Sigma(G, F)$, assume that G has a simple cycle including an even number of inhibiting edges and G does not satisfy the two properties in (C2). Let $\Sigma(s)$ be the resulting system of Algorithm 2. Then, the following statements hold.

(i) $\Sigma(s) \sim \Sigma(G, F)$.

(ii) Let G^* be the network structure of $\Sigma(s)$. Then, (C2) holds for G^*.

(iii) The node dynamics of $\Sigma(s)$ is regular.

Proof: As mentioned before, $\Sigma(s)$ corresponds to $T_{i_\nu} \circ T_{i_\nu - 1} \circ \cdots \circ T_{i_1}(\Sigma(G, F))$ for some $\nu \in \{1, 2, \ldots\}$ and $(i_1, i_2, \ldots, i_\nu) \in \mathbf{V}^\nu$. So (i), (iii), and the fact that G^* is a cactus are given by Lemma 8, Lemma 4(i), and Lemma 4(ii), respectively.

Next, we prove that the *cactus* network structure G^* satisfies the two properties in (C2). In the algorithm, cycle transformation is applied to each cycle (a) once, (b) in the order of their depth (i.e., cycle transformations are *not* applied in the order of a cycle, its parent), and (c) with respect to a common node of the cycle and its parent. This fact, the definition of cycles $1, 2, \ldots, s$, Lemma 2, and Lemma 9(ii) imply that Step 3 for cycle ℓ does not affect the nodes, edges, and labels in cycles $1, 2, \ldots, \ell - 1$. So Lemma 9(i) can be applied to all the resulting cycles. Then, the following facts are obtained for each node j in each cycle $\ell = 1, 2, \ldots, s$:

1) if node j is of indegree 1, it has an activating incoming edge in cycle ℓ;
2) if node j is of indegree greater than 1 and is not shared by its parent (i.e., is shared by a child), it has an activating incoming edge in cycle ℓ.

The former is trivial from Lemma 9(i) and the choice of node i in Step 3 (where $C_{\ell i}$ is for node i of indegree greater than 1). The latter is given by Lemma 9(i), the choice of node i in Step 3 (where node i is shared by cycle ℓ and its parent if $\ell \geq 2$), and the assumption that the root cycle has an even number of inhibiting edges. On the other hand, in the cactus network G^*, each node of indegree greater than 1 is shared by (at least) two cycles and one of which is a child of the other, because $\tilde{G}$, which is introduced in Appendix I-A, is a undirected tree. These imply that, in G^*, each node of indegree 1 has an activating incoming edge and each node of indegree greater than 1 has at least one activating incoming edge. In this way, it is proven that G^* satisfies the two properties in (C2). ■

From Lemma 10, it turns out that $\Sigma(G^*, F_1)$ is obtained as $\Sigma(s)$, i.e., G^* and F_1 are obtained as the network structure and node dynamics of $\Sigma(s)$.

Finally, we show how to derive $\Sigma(G, F_3)$ from $\Sigma(G^*, F_2)$, which is given by modifying the node dynamics of $\Sigma(G^*, F_1)$ as stated before. Since the resulting system $\Sigma(s)$ of Algorithm 2 corresponds to $T_{i_\nu} \circ T_{i_{\nu-1}} \circ \cdots \circ T_{i_1}(\Sigma(G^*, F_2))$ for some positive integer $\nu \in \{1, 2, \ldots\}$ and node sequence $(i_1, i_2, \ldots, i_\nu) \in \mathbf{V}^\nu$ (which are obtained by the execution process of Algorithm 2), we obtain the following result from Lemmas 3, 4, and 6.

Lemma 11: For the cactus Boolean network $\Sigma(G, F)$, assume that G has a simple cycle including an even number of inhibiting edges and G does not satisfy the two properties in (C2). For $\Sigma(G, F)$ and the resulting system $\Sigma(s)$ of Algorithm 2, let $\nu \in \{1, 2, \ldots\}$ and $(i_1, i_2, \ldots, i_\nu) \in \mathbf{V}^\nu$ be a positive integer and node sequence such that $T_{i_\nu} \circ T_{i_\nu - 1} \circ \cdots \circ T_{i_1}(\Sigma(G, F))$ is equal to $\Sigma(s)$. Let G^* be the network structure of $\Sigma(s)$ and suppose that a regular node dynamics F_2 is (arbitrarily) given so that $\Sigma(G^*, F_2)$ is a Boolean network in the class of Section II-A. Let Σ_3 be the system $T_{i_1} \circ T_{i_2} \circ \cdots \circ T_{i_\nu}(\Sigma(G^*, F_2))$. Then, the following statements hold.

(i) $\Sigma_3 \sim \Sigma(G^*, F_2)$.

(ii) The network structure of Σ_3 is equal to G.

(iii) The node dynamics of Σ_3 is regular.

Proof: Statements (i) and (iii) are trivial from the definition of Σ_3, Lemma 3 [i.e., (6)], and Lemma 4(i). Meanwhile,

(ii) is proven as follows. By assumption, G^* is the network structure of $T_{i_\nu} \circ T_{i_{\nu-1}} \circ \cdots \circ T_{i_1}(\Sigma(G, F))$. Thus, applying Lemma 6 to Σ_3, i.e., to $T_{i_1} \circ T_{i_2} \circ \cdots \circ T_{i_\nu}(\Sigma(G^*, F_2))$, we obtain (iii). ■

By regarding (i) and (iii) as the latter condition of (C1) and (C4) for $i = 3$ and considering (ii), we can conclude that $\Sigma(G, F_3)$ is obtained as Σ_3, i.e., $T_{i_1} \circ T_{i_2} \circ \cdots \circ T_{i_\nu}(\Sigma(G^*, F_2))$, for $\Sigma(G^*, F_2)$ specified in the beginning of this section.

References

[1] S. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, no. 5215, pp. 177–178, 1969.

[2] L. A. Amaral, A. Diaz-Guilera, A. A. Moreira, A. K. Goldberger, and L. A. Lipsitz, "Emergence of complex dynamics in a simple model of signaling networks," *Proc. Nat. Acad. Sci.*, vol. 101, no. 44, pp. 15551–15555, 2004.

[3] S. E. Harris, B. K. Sawhill, A. Wuensche, and S. Kauffman, "A model of transcriptional regulatory networks based on biases in the observed regulation rules," *Complexity*, vol. 7, no. 4, pp. 23–40, 2002.

[4] A. Mochizuki, "An analytical study of the number of steady states in gene regulatory networks," *J. Theor. Biol.*, vol. 236, no. 3, pp. 291–310, 2005.

[5] B. Drossel, T. Mihaljev, and F. Greil, "Number and length of attractors in a critical Kauffman model with connectivity one," *Phys. Rev. Lett.*, vol. 94, no. 8, 2005, Art. no. 088701.

[6] S. Zhang, M. Hayashida, T. Akutsu, W. Ching, and M. K. Ng, "Algorithms for finding small attractors in Boolean networks," *EURASIPJ. Bioinformat. Syst. Biol.*, vol. 2007, 2007, Art. no. 20180.

[7] J. Aracena, "Maximum number of fixed points in regulatory Boolean networks," *Bull. Math. Biol.*, vol. 70, no. 5, pp. 1398–1409, 2008.

[8] H. Li and Y. Wang, "Consistent stabilizability of switched Boolean networks," *Neural Netw.*, vol. 46, pp. 183–189, 2013.

[9] H. Li, Y. Wang, and Z. Liu, "Stability analysis for switched Boolean networks under arbitrary switching signals," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1978–1982, Jul. 2014.

[10] K. Kobayashi and K. Hiraishi, "Design of Boolean networks based on prescribed singleton attractors," in *Proc. 13th Eur. Control Conf.*, 2014, pp. 1504–1509.

[11] E. Fornasini and M. E. Valcher, "On the periodic trajectories of Boolean control networks," *Automatica*, vol. 49, no. 5, pp. 1506–1509, 2013.

[12] D. Cheng and H. Qi, "Controllability and observability of Boolean control networks," *Automatica*, vol. 45, no. 7, pp. 1659–1667, 2009.

[13] Y. Zhao, D. Cheng, and H. Qi, "Input-state incidence matrix of Boolean control networks and its applications," *Syst. Control Lett.*, vol. 59, no. 12, pp. 767–774, 2010.

[14] D. Laschov and M. Margaliot, "Controllability of Boolean control networks via the Perron-Frobenius theory," *Automatica*, vol. 48, no. 6, pp. 1218–1223, 2012.

[15] T. Akutsu, M. Hayashida, W. Ching, and M. K. Ng, "Control of Boolean networks: hardness results and algorithms for tree structured networks," *J. Theor. Biol.*, vol. 244, no. 4, pp. 670–679, 2007.

[16] D. Laschov and M. Margaliot, "A maximum principle for single-input Boolean control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 4, pp. 913–917, Apr. 2011.

[17] K. Kobayashi and K. Hiraishi, "Optimal control of gene regulatory networks with effectiveness of multiple drugs: A Boolean network approach," *BioMed Res. Int.*, vol. 2013, 2013, Art. no. 246761.

[18] K. Kobayashi and K. Hiraishi, "Structural control of probabilistic Boolean networks and its application to design of real-time pricing systems," in *Proc. 19th IFAC World Congr.*, vol. 47, pp. 2442–2447, 2014.

[19] K. Kobayashi and K. Hiraishi, "ILP/SMT-based method for design of Boolean networks based on singleton attractors," *IEEE/ACM Comput. Biol. Bioinformat.*, vol. 11, no. 6, pp. 1253–1259, Nov./Dec. 2014.

[20] N. Bof, E. Fornasini, and M. E. Valcher, "Output feedback stabilization of Boolean control networks," *Automatica*, vol. 57, pp. 21–28, 2015.

[21] T. Akutsu, M. Hayashida, and T. Tamura, "Algorithms for inference, analysis and control of Boolean networks," *Algebr. Biol.* (Lecture Notes in Computer Science, 5147). Berlin, Germany: Springer-Verlag, 2008.

[22] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-tensor Product Approach*. Berlin, Germany: Springer-Verlag, 2011.

[23] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," in *Proc. Pac. Symp. Biocomput.*, 1999, pp. 17–28.

[24] KEGG: Kyoto Encyclopedia of Genes and Genomes, 1995. [Online]. Available: www.genome.jp

[25] S. Azuma, T. Yoshida, and T. Sugie, "Stability analysis of Boolean networks with partial information," in *Proc. 21th Int. Symp. Math. Theory Netw. Syst.*, 2014, pp. 1395–1397.

[26] S. Azuma, T. Yoshida, and T. Sugie, "Structural monostability of activation-inhibition Boolean networks," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 2, pp. 179–190, Jun. 2017.

[27] K. Kruse and F. Jülicher, "Oscillations in biology," *Current Opinion Cell Biol.*, vol. 17, no. 1, pp. 20–26, 2005.

[28] S. Azuma, T. Yoshida, and T. Sugie, "Structural oscillatority analysis of Boolean networks," in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 2865–2868.

[29] M. Bona, M. Bousquet, G. Labelle, and P. Leroux, "Enumeration of m-Ary cacti," *Adv. Appl. Math.*, vol. 24, no. 1, pp. 22–56, 2000.

[30] M. Arcak, "Diagonal stability on cactus graphs and application to network stability analysis," *IEEE Trans. Autom. Control*, vol. 56, no. 12, pp. 2766–2777, Dec. 2011.

[31] J. Stricker, S. Cookson, M. R. Bennett, W. H. Mather, L. S. Tsimring, and J. Hasty, "A fast, robust and tunable synthetic gene oscillator," *Nature*, vol. 456, no. 7221, pp. 516–519, 2008.

[32] S. Azuma, T. Kure, and T. Sugie, "A sufficient condition for cactus-expandable graphs," in *Proc. 56th IEEE Conf. Decis. Control*, 2017, pp. 4394–4397.

[33] F. Harary, *Graph Theory*. Reading, MA, USA: Addison-Wesley, 1969.

Shun-ichi Azuma (S'03–M'05–SM'15) was born in Tokyo, Japan, in 1976. He received the B.Eng. degree in electrical engineering from Hiroshima University, Higashi Hiroshima, Japan, in 1999, and the M.Eng. and Ph.D. degrees in control engineering from Tokyo Institute of Technology, Tokyo, Japan, in 2001 and 2004, respectively. He was a Research Fellow with the Japan Society for the Promotion of Science from 2004 to 2005, an Assistant Professor with the Department of Systems Science, Graduate School of Informatics, Kyoto University, Uji, Japan, from 2005 to 2011, and an Associate Professor from 2011 to 2017. He is currently a Professor with Nagoya University, Nagoya, Japan.

His research interests include analysis and control of hybrid systems.

Dr. Azuma is an Associate Editors for the IEEE CSS Conference Editorial Board from 2011, the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS from 2013, and *IFAC Journal Automatica* from 2014.

Takahiro Yoshida received the B.E. degree in engineering and the M.E. degree in informatics from Kyoto University, Kyoto Japan, in 2013 and 2015, respectively.

He is currently with the Toyota Motor Corporation, Toyota, Japan.

Toshiharu Sugie (F'07) received the B.E., M.E., and Ph.D. degrees in engineering form Kyoto University, Kyoto, Japan, in 1976, 1978, and 1985, respectively.

From 1978 to 1980, he was a Research Member with the Musashino Electric Communication Laboratory, NTT, Musashino, Japan. From 1984 to 1988, he was a Research Associate with the Department of Mechanical Engineering, University of Osaka Prefecture, Osaka, Japan. In 1988, he joined Kyoto University, where he is currently a Professor with the Department of Systems Science. His research interests include robust control, identification for control, and control application to mechanical systems.

Dr. Sugie was an Editor for *Automatica*, and was also an Associate Editor for *Asian Journal of Control and International Journal of Systems Science*. He is a Fellow of the Society of Instrument and Control Engineers, Japan.