Efficient Verification of Observability and Reconstructibility for Large Boolean Control Networks With Special Structures

Kuize Zhang, *Senior Member, IEEE*, and Karl Henrik Johansson, *Fellow, IEEE*

***Abstract*—Verifying observability and reconstructibility of Boolean control networks (BCNs) is NP-hard in the number of nodes. A BCN is observable (reconstructible) if one can use an input sequence and the corresponding output sequence to determine the initial (current) state. In this article, we study when a node aggregation approach can be used to overcome the computational complexity in verifying these properties. We first define a class of node aggregations with subnetworks being BCNs. For acyclic node aggregations in this class, all corresponding subnetworks being observable (reconstructible) implies that the whole BCN is observable (reconstructible), although the converse is not true. In general, for cyclic node aggregations, the whole BCN being observable (reconstructible) does not imply that all subnetworks are observable (reconstructible), or vice versa. We design an algorithm to search for all acyclic node aggregations in this class, and show that finding acyclic node aggregations with small subnetworks can significantly reduce the computational complexity in verifying observability (reconstructibility). We also define a second class of node aggregations with subnetworks being finite-transition systems (more general than BCNs), which compensates for the drawback of the first class when the BCN has only a small number of output nodes. Finally, we use a BCN T-cell receptor kinetics model from the literature with 37 state nodes and 3 input nodes to illustrate the efficiency of the results derived from the two node aggregation methods. For this model, we derive the unique minimal set of 16 state nodes needed to be directly measured to make the overall BCN observable. We also compute 5 of the 16 state nodes needed to be directly measured to make the model reconstructible.**

***Index Terms*—Boolean control network, node aggregation, observability, reconstructibility, verification.**

I. Introduction

Boolean networks (BNs), introduced by Kauffman [13] to model genetic regulatory networks in 1969, are a class of discrete-time and discrete-space dynamical systems. In a BN, nodes can be in one of two discrete states "1" and "0," which represent the gene state "ON" (high concentration of a protein) and "OFF" (low concentration), respectively. Every node updates its state according to a Boolean function of the network node states. When external regulation or perturbation are considered, BNs are naturally extended to Boolean control networks (BCNs) [11]. Although BNs or BCNs are simplified models of biological systems, they can be used to characterize many important phenomena, e.g., cell cycles [8], cell apoptosis [24]. Hence the study of BNs and BCNs has received wide attention [14], [33].

The study of control-theoretic properties of BCNs dates back to 2007, when the problem of verifying controllability of a BCN was proved to be NP-hard in the number of nodes [1]. Since then, many basic properties of BCNs have been characterized, e.g., controllability [5], [34], observability [5], [9], [19], [29], [34], reconstructibility [9], [30], identifiability [7], invertibility [31], realizability [4], disturbance decoupling [3], Kalman decomposition [35], and related aspects [10], [17], [18], [20], [21], [25]. Among the above results, [3]–[5], [7], [9], [10], [17], [18], [20], [21], [25], [34], [35] are mainly based on the semitensor product framework originally proposed in [5] and [19] is based on an algebraic method; [29], [30] are mainly based on finite automata and graph theory; and [31] is mainly based on symbolic dynamics.

A BCN is observable (resp. reconstructible) if after a sufficiently long time period, the initial (resp. current) state can be determined by the input sequence and the output sequence. Observability is stronger than reconstructibility for BCNs (but this does not hold for more general control systems.). Hence, more observation information is needed to determine the initial state than to determine the current state. This property is of both theoretical and practical importance, and essential for state estimation, observer design, and controller synthesis. A quantitative description of a complex dynamical system is normally based on its state information, but the practical use is inherently limited by the ability to estimate the system's state. The problem of how to use a subset of nodes (as output nodes) to observe the whole network's state has important applications in systems biology and many other areas, e.g., in [22] it is argued that many biological networks are large and not all nodes can be directly measured.

Verifying observability and reconstructibility of a BCN is NP-hard in the number of nodes [16], [30], hence computationally

Manuscript received August 28, 2019; revised January 1, 2020; accepted January 15, 2020. Date of publication January 22, 2020; date of current version December 3, 2020. This work was supported in part by the Knut and Alice Wallenberg Foundation and in part by the Swedish Foundation for Strategic Research, and the Swedish Research Council. This paper was presented in part at the 37th Chinese Control Conference, Guangzhou, China, July 27–30, 2019 [26]. Recommended by Associate Editor Prof. S. B. Andersson. *(Corresponding author: Kuize Zhang.)*

The authors are with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 10044 Stockholm, Sweden (e-mail: kuzhan@kth.se; kallej@kth.se).

Digital Object Identifier 10.1109/TAC.2020.2968836

intractable for large networks. Existing verification algorithms [9], [29], and [30] run in exponential time in the number of nodes, so generally they cannot be used to deal with large networks (with more than approximately 30 nodes) in a reasonable amount of time. It seems unlikely that there exist fast algorithms for verifying these properties for general BCNs, but an interesting direction is to focus on special network structures. It is natural to develop node aggregation methods for which observability and reconstructibility for the overall network follow from the verification of these properties for the subnetworks of the aggregated network. An aggregated network consists of super nodes and directed edges, where each super node corresponds to a collection of nodes of the original BCN. Node aggregation methods have been used in pagerank algorithms [12] and social networks [23], as well as in controllability analysis of BCNs [32] and fixed-point computation of BNs [33]. The advantage of node aggregation methods has been illustrated through a BCN T-cell receptor kinetics model [15] in both [33] and [32]. This model has 37 state nodes and 3 input nodes,[1] i.e., it has 2^{37} states and 2^3 possible inputs. Due to the NP-hardness of verifying controllability of BCNs together with the NP-completeness of verifying existence of fixed points of BNs [2], generally it is impossible to use the general methods given in [2] and [6] to compute attractors or check controllability and stabilizability in a reasonable amount of time for large networks. However, applying some special node aggregation methods to the T-cell model, these two problems have been solved [32], [33]. In [33], an efficient way to search attractors of BNs is proposed based on such a method; particularly, for an acyclic node aggregation (i.e., when the aggregated network is acyclic), all attractors of a BN are obtained by composing attractors of the corresponding subnetworks. A similar idea has been used to deal with controllability and stabilizability of BCNs in [32], where it is proved that if a BCN is controllable (stabilizable) then all subnetworks are controllable (stabilizable) for a node aggregation of which each subnetwork has at least one state node. However, the converse is not true. It is partially because in order to verify controllability and stabilizability, external nodes (i.e., input nodes) of BCNs must be considered. In this article, we show that for observability and reconstructibility, the whole BCN being observable does not imply that all subnetworks are observable, and the latter does not imply the former either. It is because not only input nodes but also output nodes must be considered. Hence, the previous node aggregation methods cannot be used to deal with observability or reconstructibility.

There are five main contributions in this article as follows.

1) We first define a class of node aggregations for BCNs with all subnetworks being BCNs. It is proven that for acyclic node aggregations in this class, all subnetworks being observable (resp. reconstructible) implies that the whole BCN is observable (resp. reconstructible), but the converse is not true in general.
2) For cyclic node aggregations in this class, we prove that generally the whole BCN being observable (resp. reconstructible) does not imply that all corresponding subnetworks are observable (resp. reconstructible), or vice versa.
3) An efficient algorithm to compute all acyclic node aggregations in this class is developed. It is shown that finding such acyclic node aggregations with as small subnetworks as possible (sometimes significantly) reduces the computational complexity in verifying observability (resp. reconstructibility).
4) We also define a second class of node aggregations with subnetworks being finite-transition systems (FTSs), which compensates for some of the drawbacks of the first class when BCN has only a small number of output nodes. We prove similar observability and reconstructibility verification results for BCNs based on the second class of node aggregations.
5) Finally, for a BCN T-cell receptor kinetics model with 37 state nodes and 3 input nodes [15], by finding suitable acyclic node aggregations in the first class, we derive the unique minimal set of 16 state nodes that need to be directly measured to make the overall BCN observable. We then find 5 of the 16 state nodes to be directly measured for the BCN to be reconstructible. We also use the T-cell model to illustrate the efficiency of the results derived from the second class of node aggregations.

The remainder of the article is organized as follows. In Section II, basic concepts on BCNs, observability, reconstructibility, and node aggregations are introduced. In Section III, observability results based on the first class of node aggregations are proved. In addition, an algorithm for computing acyclic aggregated graphs is given and its complexity is discussed. In Section IV, results on reconstructibility based on the first class of node aggregations are derived. In Section V, related results based on the second class of node aggregations are obtained. In Section VI, the BCN T-cell receptor kinetics model is used to illustrate the efficiency of the main results given in Sections III–V. Section VII concludes the article.

Compared to its conference version [26], the current version gives a much more detailed description of the results, and contains substantial new contributions including Algorithm 1 for computing acyclic aggregations, the results on reconstructibility, the results based on the second class of node aggregations, and detailed observability and reconstructibility analysis of the T-cell model.

II. Preliminaries

In this section, we formally introduce BCNs and their observability and reconstructibility, together with the notion of node aggregation.

A. BCNs

Set $\mathcal{D} := \{0, 1\}$; $[i, j] := \{i, i+1, \dots, j\}$ with $i \le j$ being integer numbers; $C_i^j := \frac{i!}{j!(i-j)!}$ with $i \ge j$ being positive

[1] In [33] in order to compute attractors, the 3 input nodes are assumed to have constant values.

integers. 2^S stands for the power set of a set S. A BCN $\mathfrak{B}$ is described as

$$\begin{aligned}x_1(t+1) &= f_1(x_1(t),\dots,x_n(t),u_1(t),\dots,u_m(t))\\ x_2(t+1) &= f_2(x_1(t),\dots,x_n(t),u_1(t),\dots,u_m(t))\\ &\vdots\\ x_n(t+1) &= f_n(x_1(t),\dots,x_n(t),u_1(t),\dots,u_m(t))\\ y_1(t) &= h_1(x_1(t),\dots,x_n(t))\\ y_2(t) &= h_2(x_1(t),\dots,x_n(t))\\ &\vdots\\ y_q(t) &= h_q(x_1(t),\dots,x_n(t)) \qquad (1)\end{aligned}$$

where $t=0,1,\dots$ denote discrete time steps, $x_i(t)$, $u_j(t)$, $y_k(t)\in\mathcal{D}$ denote the values of the state node x_i, input node u_j, output node y_k at time step t, respectively, the maps $f_i:\mathcal{D}^{m+n}\to\mathcal{D}$ and $h_k:\mathcal{D}^n\to\mathcal{D}$ are Boolean functions, $i\in[1,n]$, $j\in[1,m]$, and $k\in[1,q]$. $\mathfrak{B}$ is represented in the compact stacked form

$$\begin{aligned}x(t+1) &= f(x(t),u(t))\\ y(t) &= h(x(t)) \qquad (2)\end{aligned}$$

with straightforward definitions of x,u,y,f, and h.

A BCN $\mathfrak{B}$ evolves over a unique graph $G=(\mathcal{N},E)$ associated with $\mathfrak{B}$. The vertex set $\mathcal{N}$ of G consists of input nodes, state nodes, and output nodes of $\mathfrak{B}$. The edges (i.e., elements of E) of G are defined as follows: For every two state nodes x_i and x_j and every input node u_k, there exists an edge from x_j (resp. u_k) to x_i, denoted by $(x_j,x_i)\in E$ (resp. $(u_k,x_i)\in E$), if and only if at each time step t, the value $x_i(t+1)$ of x_i depends on the value $x_j(t)$ of x_j (resp. the value $u_k(t)$ of u_k), and for every state node x_i and output node y_j, there exists an edge from x_i to y_j, denoted by $(x_i,y_j)\in E$, if and only if the value $y_j(t)$ of y_j depends on the value $x_i(t)$ of x_i. A subgraph G' of $G=(\mathcal{N},E)$ generated by a subset $\mathcal{N}'$ of $\mathcal{N}$ is the graph $(\mathcal{N},'E\cap(\mathcal{N}'\times\mathcal{N}'))$ consisting of $\mathcal{N}'$ and all edges of G between vertices of $\mathcal{N}'$. In graph G, an input node has zero indegree (i.e., the number of entering edges at the node is zero), an output node has zero outdegree (i.e., the number of leaving edges at the node is zero), and state nodes may have both positive indegree and positive outdegree. In graph G, for each edge (v_i,v_j), also denoted by $v_i\to v_j$, v_i (resp. v_j) is called a parent (resp. child) of v_j (resp. v_i). A path is a finite sequence $v_1\to v_2\to\cdots\to v_n$ of edges. In particular, a path $v_1\to v_2\to\cdots\to v_n$ is called a cycle if $v_1=v_n$. Two vertices v and v' are called strongly connected if they belong to some cycle. Graph G is strongly connected if every two vertices are strongly connected. A strongly connected component of G is a subgraph $G'=(\mathcal{N},'E')$ of G such that either $\mathcal{N}'$ contains only one vertex that does not belong to any cycle of G, or G' is strongly connected and for every $v'\in\mathcal{N}'$ and $v''\in\mathcal{N}\setminus\mathcal{N}'$, v' and v'' are not strongly connected. Let G'' be obtained from G by adding edge $v_1\to v_2$ if $(v_2,v_1)\in E$, but $(v_1,v_2)\notin E$ for any $v_1,v_2\in\mathcal{N}$. Hence G and G'' share the same set of vertices. Two vertices v and v' are called weakly connected in G if they are strongly connected in G''. A subgraph G' of graph G is called a weakly connected component of G if the subgraph of G'' generated by the vertices of G' is a strongly connected component of G''.

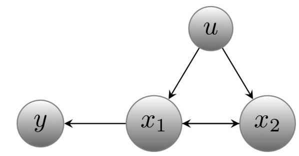


Fig. 1. Graph associated with BCNs (3) and (4).

A graph can be associated with different BCNs. For example, the graph shown in Fig. 1 is associated with the BCNs

$$\begin{aligned}x_1(t+1) &= x_2(t)\wedge u(t)\\ x_2(t+1) &= \neg x_1(t)\vee u(t)\\ y(t) &= x_1(t) \qquad (3)\end{aligned}$$

where $\wedge$, $\vee$, and $\neg$ denote AND, OR, and NOT, respectively, and

$$\begin{aligned}x_1(t+1) &= x_2(t)\bar{\vee}u(t)\\ x_2(t+1) &= \neg x_1(t)\wedge u(t)\\ y(t) &= x_1(t) \qquad (4)\end{aligned}$$

where $\bar{\vee}$ denotes XOR.

B. Observability

We are interested in the following notion of observability (first characterized in [9] and then studied in [29]).

Definition 2.1: A BCN $\mathfrak{B}$ is called observable if for all different initial states $x(0),x'(0)$, for each input sequence $\{u(0),u(1),\dots\}$, the corresponding output sequences $\{y(0),y(1),\dots\}$ and $\{y'(0),y'(1),\dots\}$ are different.

We use a graph-theoretic method proposed in [29] to verify observability. We simply call the "weighted pair graph" proposed in [29] the "observability graph".

Definition 2.2 ([29]): A graph $\mathcal{G}_o=(\mathcal{V},\mathcal{E},\mathcal{W})$ is called the observability graph of a BCN $\mathfrak{B}$ if its vertex set $\mathcal{V}$ is $\{\{x,x'\}\in\mathcal{D}^n\times\mathcal{D}^n|h(x)=h(x')\}$, its edge set $\mathcal{E}$ is $\{(\{x_1,x_1'\},\{x_2,x_2'\})\in\mathcal{V}\times\mathcal{V}|(\exists u\in\mathcal{D}^m)[(f(x_1,u)=x_2\wedge f(x_1',u)=x_2')\vee(f(x_1,u)=x_2'\wedge f(x_1',u)=x_2)]\}\subset\mathcal{V}\times\mathcal{V}$, and its weight function $\mathcal{W}:\mathcal{E}\to 2^{\mathcal{D}^m}$ assigns to each edge $(\{x_1,x_1'\},\{x_2,x_2'\})\in\mathcal{E}$ a set $\{u\in\mathcal{D}^m|(f(x_1,u)=x_2\wedge f(x_1',u)=x_2')\vee(f(x_1,u)=x_2'\wedge f(x_1',u)=x_2)\}$ of inputs, where we denote $\{x_1,x_1'\}\xrightarrow{u}\{x_2,x_2'\}$ if $u\in\mathcal{W}((\{x_1,x_1'\},\{x_2,x_2'\}))$.

A vertex $\{x,x'\}$ is called diagonal if $x=x'$, and nondiagonal otherwise. We call the subgraph of $\mathcal{G}_o$ generated by diagonal vertices, denoted by $\diamond$, the diagonal subgraph. Similarly we call the subgraph of $\mathcal{G}_o$ generated by nondiagonal vertices the nondiagonal subgraph.

Proposition 2.1 ([29]): A BCN $\mathfrak{B}$ is not observable if and only if its observability graph has a nondiagonal vertex v and a cycle C, such that there is a path from v to a vertex of C.

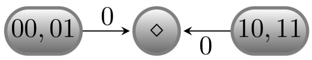


Fig. 2. Observability graph of BCN (3), where ◇ denotes its diagonal subgraph.

Fig. 3. Observability graph of BCN (4).

The following corollary follows from Proposition 2.1 since every diagonal subgraph has a cycle.

Corollary 2.2: If in the observability graph of a BCN $\mathfrak{B}$ there is a path from a nondiagonal vertex to a diagonal vertex, then $\mathfrak{B}$ is not observable.

The computational cost of constructing the observability graph of a BCN is at most $(2^n + 2^n(2^n-1)/2)2^m = 2^{n+m} + 2^{2n+m-1} - 2^{n+m-1}$. Hence the computational complexity of using Proposition 2.1 to check observability is $O(2^{2n+m-1})$. On the other hand, the size (i.e., the number of vertices and the number of edges) of the graph associated with a BCN is at most $n+m+q+mn+n(n+q)$, which is significantly smaller than the size of the observability graph. However, it is impossible to design an algorithm to check observability by using only the graph, because there exists a graph that is associated with an observable BCN and another unobservable BCN. Consider BCNs (3) and (4) and the graph in Fig. 1 that is associated with (3) and (4). It follows from the observability graph of (3) (shown in Fig. 2) together with Corollary 2.2 that (3) is not observable. Similarly, from the observability graph of (4) (Fig. 3) and Proposition 2.1, it follows that (4) is observable.

C. Reconstructibility

We consider the following notion of reconstructibility (cf. [9], [30]).

Definition 2.3: A BCN $\mathfrak{B}$ is called reconstructible if there exists a positive integer T, such that for all initial states $x(0), x'(0)$, for each input sequence $\{u(0), u(1), \dots, u(T)\}$, if $x(T+1) \neq x'(T+1)$ then the corresponding output sequence $\{y(0), y(1), \dots, y(T+1)\}$ and $\{y'(0), y'(1), \dots, y'(T+1)\}$ are different.

Observability is stronger than reconstructibility for BCNs. If a BCN is observable, then for any sufficiently long input sequence, one can use the input sequence and the corresponding output sequence to determine the initial state. Hence, all subsequent states can be determined. Consequently, the BCN is reconstructible. However, the converse does not hold in general. We use a graph-theoretic method proposed in [30] to verify reconstructibility.

Definition 2.4 ([30]): A graph $\mathcal{G}_r = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ is called the reconstructibility graph of a BCN $\mathfrak{B}$ if $\mathcal{G}_r$ is the nondiagonal subgraph of the observability graph $\mathcal{G}_o$ of $\mathfrak{B}$, i.e., $\mathcal{G}_r$ is generated by all nondiagonal vertices of $\mathcal{G}_o$.

Proposition 2.3 ([30]): A BCN $\mathfrak{B}$ is not reconstructible if and only if its reconstructibility graph has a cycle.

The computational cost of constructing the reconstructibility graph is at most $(2^n(2^n-1)/2)2^m = 2^{2n+m-1} - 2^{n+m-1}$. Hence, the computational complexity of using Proposition 2.3 to check reconstructibility is $O(2^{2n+m-1})$. Similarly to observability, one cannot only use the graph associated with a BCN to check its reconstructibility.

D. BCN Node Aggregations

Given a BCN, denote the set of state nodes by $\mathcal{X} = \{x_1, \dots, x_n\}$, the set of input nodes $\mathcal{U} = \{u_1, \dots, u_m\}$, the set of output nodes $\mathcal{Y} = \{y_1, \dots, y_q\}$, and the set of all nodes $\mathcal{N} = \mathcal{X} \cup \mathcal{U} \cup \mathcal{Y}$.

Definition 2.5: A *node aggregation* of the graph $G = (\mathcal{N}, E)$ associated with a BCN $\mathfrak{B}$ is a partition

$$\{\mathcal{N}_1, \dots, \mathcal{N}_s\} \tag{5}$$

of its set of nodes, where $s \in [1, n+m+q]$, i.e., $\mathcal{N} = \mathcal{N}_1 \cup \cdots \cup \mathcal{N}_s$, each $\mathcal{N}_i$ is a nonempty subset of $\mathcal{N}$, and $\mathcal{N}_i \cap \mathcal{N}_j = \emptyset$ for all $i \neq j, i, j \in [1, s]$. Such $\mathcal{N}_i$ with $i \in [1, s]$ denotes a super node and the graph $G_A = (\mathcal{N}_A, E_A)$ is called the aggregated graph, where $\mathcal{N}_A = \{\mathcal{N}_1, \dots, \mathcal{N}_s\}$, $(\mathcal{N}_j, \mathcal{N}_k) \in E_A$ if and only if $j \neq k$ and in G there exists an edge from some $v_j \in \mathcal{N}_j$ to some $v_k \in \mathcal{N}_k$. A node aggregation is called acyclic if its aggregated graph contains no cycle.

For each super node $\mathcal{N}_i$, its indegree (resp. outdegree) is the sum of edges entering (resp. leaving) $\mathcal{N}_i$ in the aggregated graph, $i \in [1, s]$. An aggregated graph contains no self-loop (i.e., an edge from a super node to itself). Note that a node aggregation uniquely defines an aggregated graph and vice versa, we will not distinguish them. In the sequel we will also write "a node aggregation G_A" sometimes.

Consider a state $x(t) = (x_1(t), \dots, x_n(t)) \in \mathcal{D}^n$ at time t of a BCN $\mathfrak{B}$ and its node aggregation G_A, the component of $x(t)$ in $\mathcal{N}_i$ with $i \in [1, s]$ is defined by the set $\{x_j(t) | j \in [1, n], x_j \in \mathcal{N}_i\}$. The set $\mathcal{U}$ of input nodes can be empty. In this case, the first equation of (2) becomes $x(t+1) = f(x(t))$.

The purpose of aggregating network nodes is to verify observability and reconstructibility by verifying these notions for subnetworks. A node aggregation can result in a significant reduction of computational complexity if all subnetworks are small. To this end, the observability and reconstructibility of each subnetwork must be well defined, so we make the following assumption.

Assumption 1: Consider a node aggregation G_A of the graph $G = (\mathcal{N}, E)$ associated with a BCN $\mathfrak{B}$. Let G_i be the subgraph of G generated by $\mathcal{N}_i$, $i \in [1, s]$. For each $i \in [1, s]$, $\mathcal{N}_i \cap \mathcal{Y} \neq \emptyset$; for each state node $x \in \mathcal{N}_i \cap \mathcal{X}$, there is a path from x to some output node $y \in \mathcal{N}_i \cap \mathcal{Y}$ in G_i, such that for all $x' \in \mathcal{X}$ if $(x', y) \in E$ then $x' \in \mathcal{N}_i$.

Definition 2.6: Consider a node aggregation G_A of the graph $G = (\mathcal{N}, E)$ associated with a BCN $\mathfrak{B}$, where G_A satisfies Assumption 1. For each super node $\mathcal{N}_i$, the BCN corresponding to $\mathcal{N}_i$ (or the subgraph G_i of G generated by $\mathcal{N}_i$) is

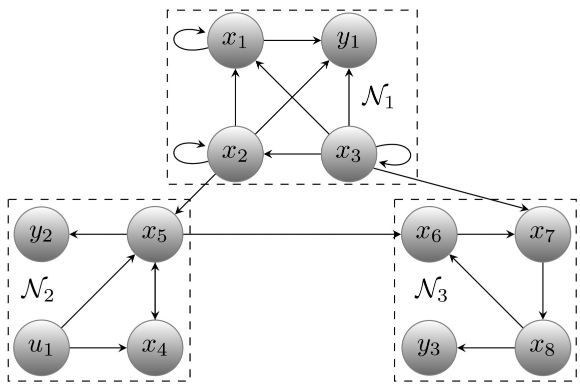


Fig. 4. Example of a node aggregation of BCN (6).

called the *BCN subnetwork* i, and is denoted $\mathfrak{B}_i$. In detail, the state node set of $\mathfrak{B}_i$ is $\mathcal{N}_i \cap \mathcal{X}$, the input node set is $\{u \in \mathcal{U} | (\exists x \in \mathcal{N}_i \cap \mathcal{X})[(u, x) \in E]\} \cup \{x \in \mathcal{X} \setminus \mathcal{N}_i | (\exists x' \in \mathcal{N}_i \cap \mathcal{X})[(x, x') \in E]\}$, and the output node set is $\{y \in \mathcal{N}_i \cap \mathcal{Y} | (\exists x \in \mathcal{N}_i \cap \mathcal{X})[(x, y) \in E]\}$. Such a node aggregation is also called a BCN node aggregation.

Under Assumption 1, each BCN subnetwork $\mathfrak{B}_i$ is of the form (1), but with $i \in [1, s]$. Hence Assumption 1 guarantees that the observability and reconstructibility of each $\mathfrak{B}_i$ are well defined. For each $\mathfrak{B}_i$, its input nodes may be input nodes of G or state nodes of G outside $\mathcal{N}_i$, but its state nodes and output nodes belong to $\mathcal{N}_i$.

Let us give an example to illustrate the above concepts.

Example 2.4: Consider the following BCN (with subnetworks $\mathfrak{B}_1, \mathfrak{B}_2, \mathfrak{B}_3$, corresponding to Fig. 4)

$$\mathfrak{B}_1 : \begin{cases} x_1(t+1) = x_1(t) \bar{\vee} (x_2(t) \wedge x_3(t)) \\ x_2(t+1) = x_2(t) \bar{\vee} x_3(t) \\ x_3(t+1) = \neg x_3(t) \\ y_1(t) = x_1(t) \wedge (x_2(t) \bar{\vee} x_3(t)) \end{cases}$$
$$\mathfrak{B}_2 : \begin{cases} x_4(t+1) = x_5(t) \wedge u_1(t) \\ x_5(t+1) = x_4(t) \bar{\vee} u_1(t) \bar{\vee} x_2(t) \\ y_2(t) = x_5(t) \end{cases} \tag{6}$$
$$\mathfrak{B}_3 : \begin{cases} x_6(t+1) = x_8(t) \bar{\vee} x_5(t) \\ x_7(t+1) = x_6(t) \bar{\vee} x_3(t) \\ x_8(t+1) = x_7(t) \\ y_3(t) = x_8(t). \end{cases}$$

In Fig. 4, all sets $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$ contain output nodes; $\mathcal{N}_1$ and $\mathcal{N}_3$ contain no input node; G_1 contains edges $x_1 \to y_1$, $x_2 \to y_1$, and $x_3 \to y_1$; G_2 contains path $x_4 \to x_5 \to y_2$; G_3 contains path $x_6 \to x_7 \to x_8 \to y_3$. Hence, this node aggregation satisfies Assumption 1. The corresponding aggregated graph is acyclic and shown in Fig. 5. Subnetworks $\mathfrak{B}_1, \mathfrak{B}_2$, and $\mathfrak{B}_3$ correspond to super nodes $\mathcal{N}_1, \mathcal{N}_2$, and $\mathcal{N}_3$, respectively. In addition, x_2 is an input node of $\mathfrak{B}_2$; x_3 and x_5 are input nodes of $\mathfrak{B}_3$.

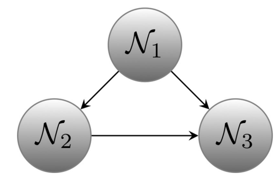


Fig. 5. Aggregated graph corresponding to Fig. 4.

III. Observability Analysis of BCNs Using BCN Node Aggregations

A. Observability Verification From BCN Node Aggregations

In this subsection, we investigate whether one can verify observability of a BCN $\mathfrak{B}$ by verifying observability of its BCN subnetworks $\mathfrak{B}_i$, $i \in [1, s]$. That is, we study whether $\mathfrak{B}$ being observable implies that all $\mathfrak{B}_i$ are also observable, or vice versa. After studying several different types of node aggregations, we find that for acyclic node aggregations satisfying Assumption 1, all $\mathfrak{B}_i$ being observable implies that $\mathfrak{B}$ is also observable, but not vice versa. The intuition here is that for acyclic aggregations, we can arrange all $\mathfrak{B}_i$ in a cascading order so that we can determine the initial states of all $\mathfrak{B}_i$ one by one. However, for cyclic aggregations, even satisfying Assumption 1, we will show that neither of the two directional implications holds by means of counterexamples.

Next we prove our first main result.

Theorem 3.1: Consider a BCN $\mathfrak{B}$ and one of its acyclic node aggregations G_A. If G_A satisfies Assumption 1 and all corresponding BCN subnetworks $\mathfrak{B}_i$ with $i \in [1, s]$ are observable then so is $\mathfrak{B}$.

Proof: Denote the graph associated with $\mathfrak{B}$ by $G = (\mathcal{N}, E)$. First we consider an acyclic node aggregation G_A of G, and prove there is a reordering (i.e., a bijection) $\tau : [1, s] \to [1, s]$, such that

$$\text{for each } i \in [1, s], \mathcal{N}^i := \bigcup_{j=1}^{i} \mathcal{N}_{\tau(j)} \text{ has zero indegree.} \tag{7}$$

Since G_A is acyclic, each subgraph of G_A has a super node with indegree 0. Suppose on the contrary that a subgraph G'_A of G_A has all super nodes with positive indegrees. Construct a new graph G''_A from G'_A by reversing the directions of all edges of G'_A. Then G''_A has a cycle, since G''_A has finitely many nodes and each node has a positive outdegree. Then G'_A and hence G_A have a cycle, which is a contradiction.

Make a copy $\overline{G}_A$ of G_A with nodes $\overline{\mathcal{N}}_1, \dots, \overline{\mathcal{N}}_s$. Choose $k_1 \in [1, s]$, such that $\overline{\mathcal{N}}_{k_1}$ has zero indegree in $\overline{G}_A$, remove $\overline{\mathcal{N}}_{k_1}$ and all edges leaving $\overline{\mathcal{N}}_{k_1}$ from $\overline{G}_A$, and set $\tau(1) = k_1$. Then in the remaining $\overline{G}_A$, there is $k_2 \in [1, s] \setminus \{k_1\}$, such that $\overline{\mathcal{N}}_{k_2}$ has zero indegree. Remove $\overline{\mathcal{N}}_{k_2}$ and all edges leaving $\overline{\mathcal{N}}_{k_2}$ from the remaining $\overline{G}_A$, and set $\tau(2) = k_2$. Repeat this procedure until $\overline{G}_A$ becomes empty, we obtain a bijection $\tau : [1, s] \to [1, s]$ that satisfies (7).

Second we assume that G_A satisfies Assumption 1 and all $\mathfrak{B}_i$ are observable, and prove that $\mathfrak{B}$ is also observable. We choose

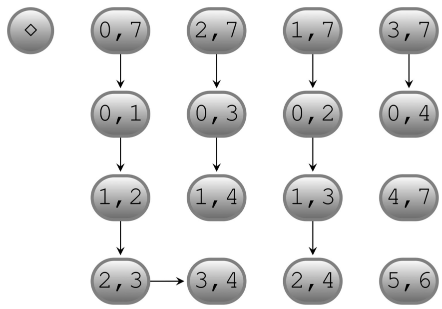


Fig. 6. Observability graph of subnetwork $\mathfrak{B}_1$ in (6), where $\diamond$ denotes the diagonal subgraph, numbers in circles are decimal representations for states of $\mathfrak{B}_1$, formally, $0\sim000$, $1\sim001$, $2\sim010$, $3\sim011$, $4\sim100$, $5\sim101$, $6\sim110$, $7\sim111$.

an arbitrary input sequence $\{u(0),u(1),\dots\}$ and two arbitrary initial states $x(0),x(0)'$, such that the components of $x(0),x(0)'$ in $\mathcal{N}_{\tau(k)}$ are not equal for some $k\in[1,s]$, but the components of $x(0),x(0)'$ in $\mathcal{N}_{\tau(i)}$ are equal for all $i\in[1,k-1]$. Then the corresponding output sequences are always the same in their $\mathcal{N}_{\tau(i)}$ components, where $i\in[1,k-1]$.

Note that

$$\begin{aligned}&\text{in } G\text{, for all } i,j\in[1,s],\\&\text{if there exist node } v\in\mathcal{N}_{\tau(i)} \text{ and node } v'\in\mathcal{N}_{\tau(j)}\\&\text{such that } (v,v')\in E \text{ then } i\le j.\end{aligned}\tag{8}$$

Then the dynamics of $\mathfrak{B}_{\tau(i)}$ does not affect that of $\mathfrak{B}_{\tau(k)}$ for any $i>k$. Hence the corresponding output sequences differ in their $\mathcal{N}_{\tau(k)}$ component since $\mathfrak{B}_{\tau(k)}$ is observable. That is, $\mathfrak{B}$ is observable. ∎

In [32], a node aggregation G_A satisfying (7) is called cascading; and it is pointed out that each cascading node aggregation is acyclic, which can also be seen by (8). In the proof of Theorem 3.1, the existence of the reordering τ satisfying (7) actually shows that each acyclic aggregation is cascading. Hence the following proposition follows.

Proposition 3.2: A node aggregation G_A is acyclic if and only if it is cascading.

Example 3.3: Recall Example 2.4. The node aggregation shown in Fig. 4 is acyclic and satisfies Assumption 1. Next we show that the BCN subnetworks $\mathfrak{B}_1,\mathfrak{B}_2,\mathfrak{B}_3$ in (6) are all observable. Then by Theorem 3.1, the whole BCN (6) is also observable. The observability graph of $\mathfrak{B}_1$ has eight diagonal vertices, and $1+C_6^2=16$ nondiagonal vertices. The observability graph of $\mathfrak{B}_1$ is shown in Fig. 6, where we note that there exists no path from a nondiagonal vertex to a diagonal vertex, and there exists no cycle in the subgraph generated by nondiagonal vertices. By Proposition 2.1, $\mathfrak{B}_1$ is observable.

For $\mathfrak{B}_2$, $x_5(0)=y_2(0)$, $x_4(0)=x_5(1)\bar{\vee}u_1(0)\bar{\vee}x_2(0)=y_2(1)\bar{\vee}u_1(0)\bar{\vee}x_2(0)$. $y_2(0)$ and $y_2(1)$ can be (directly) measured (since they are output nodes) and $u_1(0)$ and $x_2(0)$ are known (since they are input nodes), hence $\mathfrak{B}_2$ is observable.

For $\mathfrak{B}_3$, $x_8(0)=y_3(0)$, $x_7(0)=x_8(1)=y_3(1)$, $x_6(0)=x_7(1)\bar{\vee}x_3(0)=x_8(2)\bar{\vee}x_3(0)=y_3(2)\bar{\vee}x_3(0)$. $y_3(0),y_3(1)$, and $y_3(2)$ can be measured and $x_3(0)$ is known, hence $\mathfrak{B}_3$ is also observable.

The whole BCN (6) has $2^8=256$ states, 2 inputs, and $2^3=8$ outputs. Its observability graph has $(((1+C_6^2)\cdot 2+2^3)(2\cdot 2+2^2)((2C_4^2)\cdot 2+2^3)-2^8)/2=4992$ nondiagonal vertices, and $2^8=256$ diagonal vertices. Hence it is much more complex to directly use Proposition 2.1 to check observability of (6) than using Theorem 3.1 and Proposition 2.1 as above.

B. Limitation of BCN Node Aggregations

In this subsection we show some cases where the aggregation method cannot be used. We prove that for acyclic node aggregations satisfying Assumption 1, the whole BCN $\mathfrak{B}$ being observable does not imply that all BCN subnetworks $\mathfrak{B}_i$ are observable; for cyclic aggregations, even if satisfying Assumption 1, $\mathfrak{B}$ being observable does not imply that all $\mathfrak{B}_i$ are observable, or vice versa. These results also indicate that it is more difficult to find aggregations to verify observability than to find aggregations to verify controllability. The reason is as follows. In [32], controllability and stabilizability are considered for a BCN by removing all its output nodes. Under the assumption that in a node aggregation G_A, each subgraph G_i contains at least one state node, it is proved that the BCN is controllable only if each subnetwork is controllable, although the converse is not true. However, observability does not satisfy such a strong property (see Examples 3.4 and 3.5).

The first counterexample shows that for cyclic node aggregations satisfying Assumption 1, $\mathfrak{B}$ being observable does not in general imply that all $\mathfrak{B}_i$ are observable.

Example 3.4: Consider the following BCN (with three BCN subnetworks):

$$\begin{aligned}\mathfrak{B}_1&:\begin{cases}x_1(t+1)=u_1(t)\bar{\vee}x_2(t)\\ y_1(t)=x_1(t)\end{cases}\\ \mathfrak{B}_2&:\begin{cases}x_2(t+1)=u_2(t)\bar{\vee}x_1(t)\\ x_3(t+1)=u_2(t)\bar{\vee}x_4(t)\\ y_2(t)=x_2(t)\wedge x_3(t)\end{cases}\\ \mathfrak{B}_3&:\begin{cases}x_4(t+1)=u_3(t)\bar{\vee}x_3(t)\\ y_3(t)=x_4(t).\end{cases}\end{aligned}\tag{9}$$

It is not difficult to see that the node aggregation $\{\mathcal{N}_1=\{u_1,x_1,y_1\},\mathcal{N}_2=\{u_2,x_2,x_3,y_2\},\mathcal{N}_3=\{u_3,x_4,y_3\}\}$ satisfies Assumption 1 and contains cycles $\mathcal{N}_1\leftrightarrow\mathcal{N}_2$ and $\mathcal{N}_2\leftrightarrow\mathcal{N}_3$, BCN subnetworks $\mathfrak{B}_1,\mathfrak{B}_2,\mathfrak{B}_3$ in (9) correspond to super nodes $\mathcal{N}_1,\mathcal{N}_2,\mathcal{N}_3$, respectively. Subnetwork $\mathfrak{B}_1$ is observable, because $x_1(0)=y_1(0)$, and $y_1(0)$ can be measured. Symmetrically, $\mathfrak{B}_3$ is also observable. In the observability graph of $\mathfrak{B}_2$, there is an edge $\{00,01\}\xrightarrow{000}\{00,00\}$ from nondiagonal vertex $\{00,01\}$ to diagonal vertex $\{00,00\}$. Then by Corollary 2.2, $\mathfrak{B}_2$ is not observable. Now consider the whole BCN (9). We have $x_1(0)=y_1(0)$, $x_2(0)=x_1(1)\bar{\vee}u_1(0)=y_1(1)\bar{\vee}u_1(0)$, $x_3(0)=x_4(1)\bar{\vee}u_3(0)=y_3(1)\bar{\vee}u_3(0)$, $x_4(0)=y_3(0)$, $y_1(0),y_1(1),y_3(0),y_3(1)$ can be measured, $u_1(0)$ and $u_3(0)$ are known, hence (9) is observable.

The second counterexample shows that for acyclic aggregations satisfying Assumption 1, in general $\mathfrak{B}$ being observable does not imply that all $\mathfrak{B}_i$ are observable.

Example 3.5: Consider the following BCN:

$$\begin{aligned}\mathfrak{B}_1 &: \begin{cases} x_1(t+1) = x_2(t) \wedge u_1(t) \\ x_2(t+1) = x_1(t) \\ y_1(t) = x_1(t) \end{cases} \\ \mathfrak{B}_2 &: \begin{cases} x_3(t+1) = x_2(t) \\ x_4(t+1) = x_3(t) \\ y_2(t) = x_4(t). \end{cases}\end{aligned} \tag{10}$$

The node aggregation $\{\mathcal{N}_1 = \{u_1, x_1, x_2, y_1\}, \mathcal{N}_2 = \{x_3, x_4, y_2\}\}$ satisfies Assumption 1, and the corresponding aggregated graph $\mathcal{N}_1 \to \mathcal{N}_2$ contains no cycle. Subnetworks $\mathfrak{B}_1, \mathfrak{B}_2$ in (10) correspond to super nodes $\mathcal{N}_1, \mathcal{N}_2$, respectively. In the observability graph of $\mathfrak{B}_1$, there is an edge $\{10, 11\} \xrightarrow{0} \{01, 01\}$ from a nondiagonal vertex $\{10, 11\}$ to a diagonal vertex $\{01, 01\}$, then by Corollary 2.2, $\mathfrak{B}_1$ is not observable. Subnetwork $\mathfrak{B}_2$ is observable because $x_4(0) = y_2(0)$, $x_3(0) = x_4(1) = y_2(1)$, $y_2(0)$ and $y_2(1)$ can be measured. The whole BCN (10) is observable because $x_1(0) = y_1(0)$, $x_2(0) = x_3(1) = x_4(2) = y_2(2)$, $x_3(0) = x_4(1) = y_2(1)$, $x_4(0) = y_2(0)$, $y_1(0), y_2(0), y_2(1)$, and $y_2(2)$ can be measured.

The third counterexample shows that for cyclic aggregations satisfying Assumption 1, if all $\mathfrak{B}_i$ are observable, $\mathfrak{B}$ is not necessarily observable.

Example 3.6: Consider the following BCN:

$$\begin{aligned}\mathfrak{B}_1 &: \begin{cases} x_1(t+1) = u_1(t) \\ x_2(t+1) = x_1(t) \bar{\vee} x_4(t) \\ y_1(t) = x_2(t) \end{cases} \\ \mathfrak{B}_2 &: \begin{cases} x_3(t+1) = \neg(x_1(t) \bar{\vee} x_4(t)) \\ x_4(t+1) = u_2(t) \\ y_2(t) = x_3(t). \end{cases}\end{aligned} \tag{11}$$

The node aggregation $\{\mathcal{N}_1 = \{u_1, x_1, x_2, y_1\}, \mathcal{N}_2 = \{u_2, x_3, x_4, y_2\}\}$ satisfies Assumption 1, and its aggregated graph is a cycle $\mathcal{N}_1 \leftrightarrow \mathcal{N}_2$. For $\mathfrak{B}_1$, $x_2(0) = y_1(0)$, $x_1(0) = x_2(1) \bar{\vee} x_4(0) = y_1(1) \bar{\vee} x_4(0)$. Since $y_1(0)$ and $y_1(1)$ can be measured and $x_4(0)$ is known, $\mathfrak{B}_1$ is observable. Similarly $\mathfrak{B}_2$ is also observable. Consider a nondiagonal vertex $\{0110, 1111\}$ of the obse-rvability graph of (11), there is an edge $\{0110, 1111\} \xrightarrow{u_1 u_2} \{u_1 01 u_2, u_1 01 u_2\}$, where inputs $u_1, u_2 \in \mathcal{D}$, $\{u_1 01 u_2, u_1 01 u_2\}$ is a diagonal vertex of the observability graph. Hence by Corollary 2.2, (11) is not observable.

C. An Algorithm for Computing Acyclic BCN Node Aggregations

In this subsection we design Algorithm 1 for computing all acyclic BCN node aggregations satisfying Assumption 1 for the graph G associated with a BCN $\mathfrak{B}$. Each of such node aggregations can be easily found in polynomial time in the number of vertices and edges of G.

Algorithm 1

Input: A BCN $\mathfrak{B}$
Output: All acyclic aggregated graphs of the graph G associated with $\mathfrak{B}$ satisfying Assumption 1

1: Compute all strongly connected components of G, and regard each strongly connected component as a super node. Then an acyclic aggregated graph is obtained.
2: **if** the aggregated graph satisfies Assumption 1 **then**
3:   output the graph
4: **end if**
5: **while** some super node does not satisfy Assumption 1 **do**
6:   arbitrarily choose a super node $\mathcal{N}_i$ that does not satisfy Assumption 1
7:   **if** $\mathcal{N}_i$ contains only input nodes **then**
8:     arbitrarily find another super node $\mathcal{N}_j$, such that there exists an edge $\mathcal{N}_i \to \mathcal{N}_j$, and combine them to obtain a new super node
9:   **end if**
10:   **if** in $\mathcal{N}_i$, there exists a state node x_j that does not satisfy Assumption 1 **then**
11:     find an arbitrary path from x_j to an output node y_k in G, find all parents of y_k, and then combine all super nodes each of which contains either one node of the path or one parent of y_k
12:   **end if**
13: **end while**

Let us describe Algorithm 1. First, the algorithm computes an acyclic aggregated graph of graph G, where the vertices of the aggregated graph correspond to the strongly connected components of G. This aggregated graph is the finest acyclic aggregated graph of G. We say an aggregated graph G' of G is coarser than another aggregated graph G'' of G (or equivalently G'' is finer than G') if each vertex of G'' is a subset of some vertex of G'. There exist well-known algorithms to compute all strongly connected components with linear time complexity in the size of the graph, e.g., a variant of depth-first search. If the obtained aggregated graph satisfies Assumption 1, then the algorithm returns the aggregated graph. The rest of the algorithm (the "**while** ... **end while**" structure) combines several of the strongly connected components in order to obtain other acyclic aggregated graphs that satisfy Assumption 1. Given a node aggregation $\{\mathcal{N}_1, \dots, \mathcal{N}_s\}$, after we combine super nodes $\mathcal{N}_{i_1}, \dots, \mathcal{N}_{i_p}$, we obtain a new coarser node aggregation $\{\mathcal{N}_{i_1} \cup \cdots \cup \mathcal{N}_{i_p}, \mathcal{N}_j \text{ with } j \in [1, s] \setminus \{i_1, \dots, i_p\}\}$. Since the aggregated graph obtained in line 1 is acyclic, all other obtained aggregated graphs in the rest of the algorithm are also acyclic. The algorithm will return all acyclic aggregated graphs that satisfy Assumption 1 after running over all possible combinations of strongly connected components of G. There always exists an acyclic aggregated graph satisfying Assumption 1, e.g., the trival node aggregation consisting of only one super node $\mathcal{N}$.

After we have obtained an aggregated graph that satisfies Assumption 1, we can use Theorem 3.1 to check observability of $\mathfrak{B}$. If the obtained aggregated graph satisfies the condition in Theorem 3.1, then we know that $\mathfrak{B}$ is observable. Otherwise we compute another aggregated graph satisfying Assumption 1 by using Algorithm 1 to do the check. If $\mathfrak{B}$ is observable, then we will finally obtain (after repeating the above procedure several times) an acyclic aggregated graph that satisfies Assumption 1 and the condition in Theorem 3.1, because there always exists an acyclic aggregated graph satisfying Assumption 1 and the condition in Theorem 3.1, e.g., the trivial node aggregation containing only one super node $\mathcal{N}$. However, if $\mathfrak{B}$ is not observable, then after repeating the above procedure several times, we will obtain the trivial node aggregation that does not satisfy the condition in Theorem 3.1.

Next we analyze the computational complexity of using Theorem 3.1, Algorithm 1, and Proposition 2.1 to verify observability. We first find an acyclic node aggregation that satisfies Assumption 1 by using Algorithm 1, then check observability of all BCN subnetworks. If all BCN subnetworks are observable then the whole BCN is observable. Assume that we have obtained an acyclic node aggregation having k super nodes with almost the same size and satisfying Assumption 1. Then each super node approximately has n/k state nodes and m/k input nodes. The computational cost of verifying observability for all BCN subnetworks is approximately $k2^{(2n+m)/k-1}$ by Proposition 2.1. For large BCNs, $2n+m$ is large. When $k < l(2n+m)$ for some positive constant l, $k2^{(2n+m)/k-1}$ decreases as k increases. Hence roughly speaking, the more super nodes a node aggregation has and the closer the sizes of these super nodes are, the less computation it costs to verify observability for all BCN subnetworks. It is hard to find aggregations whose super nodes have approximately the same size, but we can find aggregations having sufficiently many super nodes instead (approximately equivalent to finding finer node aggregations). Also, since generally Algorithm 1 returns finer node aggregations earlier, we conclude that in most cases the method of using Theorem 3.1, Algorithm 1, and Proposition 2.1 to verify observability of large BCNs is quite efficient for a class of large BCNs.

IV. Reconstructibility Analysis of BCNs Using BCN Node Aggregations

In this section, we study whether the node aggregation method can be used to deal with reconstructibility of large BCNs. The focus is on the conceptual difference between observability and reconstructibility,

A. Reconstructibility Verification From BCN Node Aggregations

We next prove the second main result. Even if it is similar to Theorem 3.1, the proof is quite different. See discussion after the proof for some insight.

Theorem 4.1: Consider a BCN $\mathfrak{B}$ and one of its acyclic BCN node aggregations G_A. If G_A satisfies Assumption 1 and all BCN subnetworks $\mathfrak{B}_i$ with $i \in [1, s]$ are reconstructible then so is $\mathfrak{B}$.

Proof: By the proof of Theorem 3.1, there is a bijection $\tau : [1, s] \to [1, s]$, such that (7) holds. Then without loss of generality, we assume in G_A that for all $i, j \in [1, s]$, if there is an edge $\mathcal{N}_i \to \mathcal{N}_j$ then $i < j$. Next we assume that G_A satisfies Assumption 1 and all BCN subnetworks $\mathfrak{B}_i$ with $i \in [1, s]$ are reconstructible, and prove $\mathfrak{B}$ is also reconstructible.

Assume by contradiction that each $\mathfrak{B}_i$ is reconstructible but $\mathfrak{B}$ is not. Since $\mathfrak{B}$ is not reconstructible, by Proposition 2.3, there exist an input sequence $\{u(0), u(1), \dots\}$ and two different initial states $x(0), x(0)'$, such that the corresponding output sequences $\{y(0), y(1), \dots\}$ and $\{y(0)', y(1)', \dots\}$ are the same, and the corresponding states $x(t)$ and $x(t)'$ are different at any time step $t = 0, 1, \dots$. Hence, at any time step t, there is $i_t \in [1, s]$, such that the components of $x(t)$ and $x(t)'$ in $\mathcal{N}_{i_t}$ are different. Since $\mathfrak{B}_1$ is reconstructible and $\mathcal{N}_1$ has zero indegree in G_A, there is a positive integer j_1, such that the components of $x(k)$ and $x(k)'$ in $\mathcal{N}_1$ are the same for any time $k \geq j_1$. Since $\mathfrak{B}_2$ is reconstructible, and $\mathcal{N}_1$ is the only potential parent of $\mathcal{N}_2$ in G_A, there is a positive integer $j_2 > j_1$, such that the components of $x(k)$ and $x(k)'$ in $\mathcal{N}_2$ are the same for any time $k \geq j_2$. Similarly we have for any $l \in [1, s-1]$, there is a positive integer $j_l > j_{l-1}$, such that the components of $x(k)$ and $x(k)'$ in $\mathcal{N}_l$ are the same for any time $k \geq j_l$. Hence the components of $x(k)$ and $x(k)'$ in $\mathcal{N}_s$ are different for any time $k > j_{s-1}$. Due to the finite cardinality of the state space $\mathcal{D}^n$, there is a cycle in the reconstructibility graph of $\mathfrak{B}_s$. By Proposition 2.3, $\mathfrak{B}_s$ is not reconstructible, which is a contradiction, and completes the proof. ■

In order to verify reconstructibility of a large BCN, similarly to verifying observability, one can first use Algorithm 1 to find an acyclic node aggregation satisfying Assumption 1, and then check whether all BCN subnetworks are reconstructible according to Theorem 4.1.

The proof of Theorem 4.1 is more complex than that of Theorem 3.1. To see why, consider an acyclic node aggregation G_A satisfying Assumption 1. As in the proofs of Theorems 3.1 and 4.1, we can assume that the super nodes (arranged as $\mathcal{N}_1, \dots, \mathcal{N}_s$) of G_A satisfy that there exists no edge from $\mathcal{N}_i$ to $\mathcal{N}_j$ for any $i > j$ without loss of generality. To prove that all $\mathfrak{B}_i$ being observable implies that $\mathfrak{B}$ is also observable, we just need to assume two different initial states, and check the subnetwork $\mathfrak{B}_j$ with the smallest index j, such that the components of the initial states of $\mathfrak{B}$ in $\mathfrak{B}_j$ differ. However, to prove that all $\mathfrak{B}_i$ being reconstructible implies that $\mathfrak{B}$ is reconstructible, we have to check all $\mathfrak{B}_i$ in the above order one by one.

B. Limitation of BCN Node Aggregations

Similarly to observability, one cannot always use the aggregation method to verify reconstructibility of BCNs. Next we give counterexamples to show some cases where the aggregation method cannot be used in general.

The fourth counterexample shows that for acyclic aggregations satisfying Assumption 1, if $\mathfrak{B}$ is reconstructible, then in general the BCN subnetworks $\mathfrak{B}_i$ with $i \in [1, s]$ are not necessarily all reconstructible.

Example 4.2: Consider the following BCN (with three BCN subnetworks):

$$\begin{aligned} &\mathfrak{B}_1 : \begin{cases} x_1(t+1) = u_1(t) \bar{\vee} x_2(t) \\ y_1(t) = x_1(t) \end{cases} \\ &\mathfrak{B}_2 : \begin{cases} x_2(t+1) = x_3(t) \\ x_3(t+1) = x_2(t) \\ y_2(t) = x_2(t) \wedge x_3(t) \end{cases} \\ &\mathfrak{B}_3 : \begin{cases} x_4(t+1) = u_2(t) \bar{\vee} x_3(t) \\ y_3(t) = x_4(t). \end{cases} \end{aligned} \tag{12}$$

The acyclic node aggregation $\{\mathcal{N}_1 = \{u_1, x_1, y_1\}, \mathcal{N}_2 = \{x_2, x_3, y_2\}, \mathcal{N}_3 = \{u_2, x_4, y_3\}\}$ satisfies Assumption 1. In the proof of Proposition 3.4, we have shown that $\mathfrak{B}_1$ and $\mathfrak{B}_3$ are both observable, hence they are both reconstructible. For $\mathfrak{B}_2$, in its reconstructibility graph, there is a self-loop on nondiagonal vertex $\{10, 01\}$, then by Proposition 2.3, $\mathfrak{B}_2$ is not reconstructible. Now consider the whole BCN (12). We have $x_1(0) = y_1(0)$, $x_2(0) = x_1(1) \bar{\vee} u_1(0) = y_1(1) \bar{\vee} u_1(0)$, $x_3(0) = x_4(1) \bar{\vee} u_2(0) = y_3(1) \bar{\vee} u_2(0)$, $x_4(0) = y_3(0)$, $y_1(0), y_1(1), y_3(0), y_3(1)$ can be measured, $u_1(0)$ and $u_2(0)$ are known, hence (12) is observable, and then reconstructible.

The fifth counterexample shows that for cyclic aggregations satisfying Assumption 1, if $\mathfrak{B}$ is reconstructible, then in general not all $\mathfrak{B}_i$ are necessarily reconstructible.

Example 4.3: Consider the following BCN:

$$\begin{aligned} &\mathfrak{B}_1 : \begin{cases} x_1(t+1) = u_1(t) \bar{\vee} x_2(t) \\ y_1(t) = x_1(t) \end{cases} \\ &\mathfrak{B}_2 : \begin{cases} x_2(t+1) = x_1(t) \wedge x_3(t) \\ x_3(t+1) = x_4(t) \wedge x_2(t) \\ y_2(t) = x_2(t) \wedge x_3(t) \end{cases} \\ &\mathfrak{B}_3 : \begin{cases} x_4(t+1) = u_2(t) \bar{\vee} x_3(t) \\ y_3(t) = x_4(t). \end{cases} \end{aligned} \tag{13}$$

The cyclic node aggregation $\{\mathcal{N}_1 = \{u_1, x_1, y_1\}, \mathcal{N}_2 = \{x_2, x_3, y_2\}, \mathcal{N}_3 = \{u_2, x_4, y_3\}\}$ satisfies Assumption 1. In the proof of Proposition 3.4, we have shown that $\mathfrak{B}_1$ and $\mathfrak{B}_3$ are observable, hence they are reconstructible. For $\mathfrak{B}_2$, in its reconstructibility graph, there is a self-loop $\{10, 01\} \xrightarrow{11} \{10, 01\}$, then by Proposition 2.3, $\mathfrak{B}_2$ is not reconstructible. Now consider the whole BCN (13). Similarly to (12), the whole BCN is observable, and hence reconstructible.

The sixth counterexample shows that for cyclic aggregations satisfying Assumption 1, if all $\mathfrak{B}_i$ are reconstructible, then in general $\mathfrak{B}$ is not necessarily reconstructible.

Example 4.4: Consider the following BCN:

$$\begin{aligned} &\mathfrak{B}_1 : \begin{cases} x_1(t+1) = u_1(t) \vee x_1(t) \\ x_2(t+1) = x_1(t) \bar{\vee} x_4(t) \\ y_1(t) = x_2(t) \end{cases} \\ &\mathfrak{B}_2 : \begin{cases} x_3(t+1) = x_1(t) \bar{\vee} x_4(t) \\ x_4(t+1) = u_2(t) \vee x_4(t) \\ y_2(t) = x_3(t). \end{cases} \end{aligned} \tag{14}$$

The cyclic node aggregation $\{\mathcal{N}_1 = \{u_1, x_1, x_2, y_1\}, \mathcal{N}_2 = \{u_2, x_3, x_4, y_2\}\}$ satisfies Assumption 1. One directly sees that both $\mathfrak{B}_1$ and $\mathfrak{B}_2$ are observable, hence they are also reconstructible. In the reconstructibility graph of BCN (14), there is a self-loop $\{1001, 0000\} \xrightarrow{00} \{1001, 0000\}$, then by Proposition 2.3, (14) is not reconstructible.

V. Observability and Reconstructibility Analysis Using FTS Node Aggregations

Previously, we gave a class of node aggregations and studied when one can efficiently verify observability and reconstructibility for large BCNs by verifying these two notions for their subnetworks. Note that in this approach, every BCN subnetwork must have an output node, which makes the verification not very efficient when the whole BCN has only a small number of output nodes. In order to compensate for this drawback, in this section, we give a new class of node aggregations and a new notion of subnetwork, such that in a new subnetwork, there may exist no output node of the whole network. Note that a newly defined subnetwork is not necessarily an BCN but an FTS (cf. [27], [28]) that is more general than a BCN. We still adopt the notion of node aggregation as in Definition 2.5.

An FTS[2] is described as

$$\begin{aligned} x(t+1) &\in f(x(t), u(t)) \\ y(t) &= h(x(t)) \end{aligned} \tag{15}$$

where $t = 0, 1, \dots$, $x(t) \in \Delta_N$, $u(t) \in \Delta_M$, $y(t) \in \Delta_Q$, $f : \Delta_N \times \Delta_M \to 2^{\Delta_N} \setminus \emptyset$, $h : \Delta_N \to \Delta_Q$, finite sets $\Delta_N, \Delta_M, \Delta_Q$ have positive integer cardinality N, M, Q.

Due to the nondeterminism of (15), given an initial state and an input sequence, more than one output sequence may be produced. The notions of observability and reconstructibility for FTSs are as follows (called observability in the arbitrary-experiment case in [27] and detectability in [28], respectively). When an FTS reduces to a BCN, Definition 5.1 coincides with Definition 2.1, Definition 5.2 coincides with Definition 2.3.

Definition 5.1 ([27]): An FTS $\mathfrak{F}$ is called observable if for all different initial states $x(0), x'(0)$, for each input sequence $\{u(0), u(1), \dots\}$, any output sequence $\{y(0), y(1), \dots\}$ generated by $x(0)$ and $\{u(0), u(1), \dots\}$ is different from any output sequence $\{y'(0), y'(1), \dots\}$ generated by $x'(0)$ and $\{u'(0), u'(1), \dots\}$.

[2]In this article, we only need to consider total FTSs for which all states can be initial, see [28] for details. An FTS is called a finite labeled transition system in [27], and in [28] it is called a nondeterministic FTS.

Definition 5.2 ([28]): An FTS $\mathfrak{F}$ is called reconstructible if there exists a positive integer T, such that for every input sequence $\{u(0), u(1), \dots\}$, and every output sequence $\{y(0), y(1), \dots\}$, the state at time k can be uniquely determined by $\{u(0), \dots, u(k-1)\}$ and $\{y(0), \dots, y(k)\}$ for every time $k > T$.

By definition, if an FTS is observable, then for an arbitrary input sequence $\{u(0), u(1), \dots\}$ and the corresponding output sequence $\{y(0), y(1), \dots\}$, its state at any time can be uniquely determined. Note that here observability is not necessarily stronger than reconstructibility (see Example 5.2 for a counterexample). An algorithm[3] for verifying observability of $\mathfrak{F}$ that runs in polynomial time in the numbers of states, inputs, and outputs has been given in [27], and an algorithm[4] for verifying reconstructibility of $\mathfrak{F}$ that also runs in polynomial time in the numbers of states, inputs, and outputs has been given in [28]. Note that the algorithms in Propositions 2.1 and 2.3 for verifying observability and reconstructibility of BCNs also run in polynomial time in the numbers of states, inputs, and outputs.

The new assumption is as follows.

Assumption 2: Consider a node aggregation G_A of the graph $G = (\mathcal{N}, E)$ associated with a BCN $\mathfrak{B}$. For every $i \in [1, s]$ and every state node $x \in \mathcal{N}_i \cap \mathcal{X}$, either

1) in subgraph G_i generated by $\mathcal{N}_i$, there is a path from x to some output node $y \in \mathcal{N}_i \cap \mathcal{Y}$; or
2) there exists $i \neq j \in [1, s]$ and $x' \in \mathcal{N}_j \cap \mathcal{X}$, such that $(x, x') \in E$ and for all $x'' \in \mathcal{N}$ if $(x'', x') \in E$ then $x'' \in \mathcal{N}_i \cap \mathcal{X}$.

One can see that Assumption 2 does not imply Assumption 1, or vice versa.

With these preliminaries, we give the new notion of subnetwork.

Definition 5.3: Consider a node aggregation G_A of the graph $G = (\mathcal{N}, E)$ associated with a BCN $\mathfrak{B}$ that satisfies Assumption 2. For each super node $\mathcal{N}_i$, where $i \in [1, s]$, the FTS corresponding to $\mathcal{N}_i$ (or the subgraph G_i of G generated by $\mathcal{N}_i$) is called the *FTS subnetwork* i, and denoted by $\mathfrak{F}_i$. In detail, the state node set of $\mathfrak{F}_i$ is $\mathcal{N}_i \cap \mathcal{X}$, the input node set of $\mathfrak{F}_i$ is $\{u \in \mathcal{U} | (\exists x \in \mathcal{N}_i \cap \mathcal{X})[(u, x) \in E]\} =: \mathfrak{F}_i^u$,and the output node set of $\mathfrak{F}_i$ is a subset of $\{y \in \mathcal{N}_i \cap \mathcal{Y} | (\exists x \in \mathcal{N}_i \cap \mathcal{X})[(x, y) \in E]\} \cup \bigcup_{j \in [1,s], j \neq i} \{x \in \mathcal{N}_j | ((\exists x' \in \mathcal{N}_i \cap \mathcal{X})[(x', x) \in E]) \wedge ((\forall x'' \in \mathcal{N})[(x'', x) \in E \Rightarrow x'' \in \mathcal{N}_i \cap \mathcal{X}])\} =: \mathfrak{F}_i^y$. Such a node aggregation is called an *FTS node aggregation*.

In Definition 5.3, for FTS subnetwork $\mathfrak{F}_i$, the set of input nodes is $\mathfrak{F}_i^u$, but the set of output nodes is only a subset of $\mathfrak{F}_i^y$. Actually, if a proper subset of $\mathfrak{F}_i^y$ is enough for the corresponding $\mathfrak{F}_i$ to be observable (resp. reconstructible), then we do not need to impose that all nodes of $\mathfrak{F}_i^y$ are output nodes. For subnetwork $\mathfrak{F}_j$, since there may exist nodes outside $\mathfrak{F}_j$ that can affect dynamics of $\mathfrak{F}_j$ but are not input nodes of $\mathfrak{F}_j$ (e.g., in Fig. 7, node x_4 is not an input node of subnetwork $\mathfrak{F}_2$ but affects dynamics of $\mathfrak{F}_2$), one has the updating rule of $\mathfrak{F}_j$ is actually a set-valued function as described in (15). Hence $\mathfrak{F}_j$ is itself an FTS.

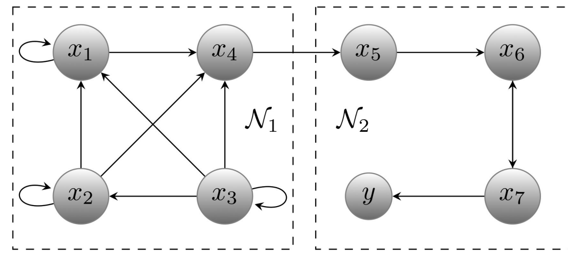


Fig. 7. Example of a node aggregation for Example 5.2.

We next prove the third main result. Differently from Theorems 3.1 and 4.1, we do not need to impose that the corresponding aggregated graph is acyclic. However, for an acyclic node aggregation that satisfies Assumption 2, the reordering τ in Theorem 5.1 always exists.

Theorem 5.1: Consider a BCN $\mathfrak{B}$ and one of its node aggregations G_A satisfying Assumption 2. Assume that there exists a reordering $\tau : [1, s] \to [1, s]$, such that for each $i \in [1, s]$, no output node of FTS subnetwork $\mathfrak{F}_{\tau(i)}$ belongs to $\mathcal{N}_{\tau(j)}$ for any $j < i$. If all $\mathfrak{F}_i$ with $i \in [1, s]$ are observable (resp. reconstructible), then so is $\mathfrak{B}$.

Proof: Arbitrarily choose an initial state $x(0)$ given but unknown, an input sequence $\{u(0), u(1), \dots\}$, and an arbitrary produced output sequence $\{y(0), y(1), \dots\}$. First, since subnetwork $\mathfrak{F}_{\tau(s)}$ is observable (resp. reconstructible), and all of its output nodes belong to $\mathcal{N}_{\tau(s)}$, we can determine its initial state (resp. current state at some time and all subsequent states) by using the above input sequence and output sequence; second, for $\mathfrak{F}_{\tau(s-1)}$, by using the obtained states of $\mathfrak{F}_{\tau(s)}$ (they may be values of some output nodes of $\mathfrak{F}_{\tau(s-1)}$) and the above input and output sequences, we can determine the initial state (resp. current state at some time and all subsequent states) of $\mathfrak{F}_{\tau(s-1)}$; ...; finally, we can determine the initial state (resp. current state at some time and all subsequent states) of $\mathfrak{F}_{\tau(1)}$. Hence $\mathfrak{B}$ is observable (resp. reconstructible). ■

Example 5.2: Consider the following BCN:

$$
\begin{aligned}
x_1(t+1) &= x_1(t) \bar{\vee} (x_2(t) \wedge x_3(t)) \\
x_2(t+1) &= x_2(t) \bar{\vee} x_3(t) \\
x_3(t+1) &= \neg x_3(t) \\
x_4(t+1) &= x_1(t) \wedge (x_2(t) \bar{\vee} x_3(t)) \\
x_5(t+1) &= x_4(t) \\
x_6(t+1) &= x_5(t) \bar{\vee} x_7(t) \\
x_7(t+1) &= x_6(t) \\
y(t) &= x_7(t).
\end{aligned} \tag{16}
$$

Consider the node aggregation shown in Fig. 7. This node aggregation satisfies Assumption 2 but does not satisfy Assumption 1. The unique node aggregation of (16) satisfying Assumption 1 only contains one super node $\{x_1, \dots, x_7, y\}$, since every BCN subnetwork must contain an output node and the whole network only contains one output node. Consider the following two FTS

[3]This algorithm reduces to the one shown in Proposition 2.1 when the FTS reduces to a BCN.

[4]This algorithm does not reduce to the one shown in Proposition 2.3 when the FTS reduces to a BCN.

TABLE I
UPDATING RULES FOR THE NODES OF THE T-CELL RECEPTOR KINETICS MODEL [15]

Nodes	Boolean rule	Nodes	Boolean rule	Nodes	Boolean rule
CD8	Input	Gads	LAT	PKCth	DAG
CD45	Input	Grb2Sos	LAT	PLCg(act)	(Itk∧PLCg(bind)∧SLP76∧ZAP70) ∨(PLCg(bind)∧Rlk∧SLP76∧ZAP70)
TCRlig	Input	IKKbeta	PKCth	PAGCsk	Fyn∨(¬TCRbind)
AP1	Fos∧Jun	IP3	PLCg(act)	PLCg(bind)	LAT
Ca	IP3	Itk	SLP76∧ZAP70	Raf	Ras
Calcin	Ca	IkB	¬IKKbeta	Ras	Grb2Sos∨ RasGRP1
cCbl	ZAP70	JNK	SEK	RasGRP1	DAG∧PKCth
CRE	CREB	Jun	JNK	Rlk	Lck
CREB	Rsk	LAT	ZAP70	Rsk	ERK
DAG	PLCg(act)	Lck	(¬PAGCsk)∧CD8∧CD45	SEK	PKCth
ERK	MEK	MEK	Raf	SLP76	Gads
Fos	ERK	NFAT	Calcin	TCRbind	(¬cCbl)∧TCRlig
Fyn	(Lck∧CD45)∨(TCRbind∧CD45)	NFkB	¬IkB	TCRphos	Fyn∨(Lck∧TCRbind)
				ZAP70	(¬cCbl)∧Lck∧TCRphos

subnetworks $\mathfrak{F}_1$ (a BCN) and $\mathfrak{F}_2$:

$$\mathfrak{F}_1 : \begin{cases} x_1(t+1) = x_1(t)\bar{\vee}(x_2(t) \wedge x_3(t)) \\ x_2(t+1) = x_2(t)\bar{\vee}x_3(t) \\ x_3(t+1) = \neg x_3(t) \\ x_4(t+1) = x_1(t) \wedge (x_2(t)\bar{\vee}x_3(t)) \\ y_1(t) = x_5(t+1) = x_4(t) \end{cases} \tag{17}$$

$$\mathfrak{F}_2 : \begin{cases} x_5(t+1) \in \{0,1\} \\ x_6(t+1) = x_5(t)\bar{\vee}x_7(t) \\ x_7(t+1) = x_6(t) \\ y(t) = x_7(t) \end{cases}$$

where $y_1(t) = x_5(t+1)$ means that x_5 is regarded as an output node of $\mathfrak{F}_1$; y is an output node of $\mathfrak{F}_2$; x_4 is not an input node of $\mathfrak{F}_2$, but affects the update of node x_5 (see $x_5(t+1) \in \{0,1\}$).

Observe that $x_7(0) = y(0)$, $x_6(0) = x_7(1)$, and $x_5(0) = x_6(1)\bar{\vee}x_7(0)$, so $\mathfrak{F}_2$ is observable. However, $\mathfrak{F}_2$ is not reconstructible since starting at initial state 010 (in order x_5, x_6, x_7), for output sequence $(01)^\omega$ (i.e., concatenation of infinitely many copies of 01), 001 and 101 are possible states at time $2k+1$, 000, 010, 100, 110 are possible states at time $2k+2$ for all natural numbers k. In Example 3.3, we have proved that BCN subnetwork $\mathfrak{B}_1$ of (6) is observable, hence $\mathfrak{F}_1$ (actually the same as $\mathfrak{B}_1$) of (16) is also observable. Then by Theorem 5.1, (16) is observable, and hence reconstructible.

Similarly to Algorithm 1, one can also design an algorithm for returning all node aggregations of a BCN satisfying Assumption 2. One can do similar analysis on the limitation of FTS node aggregations as on BCN node aggregations in the previous section.

VI. A BIOLOGICAL APPLICATION

In this section, we apply our results in Sections III–V to study observability and reconstructibility of the BCN T-cell receptor kinetics model [15], and then compare the obtained results.

A. The Model

T-cells are a type of white blood cells known as lymphocytes. These white blood cells play a central role in adaptive immunity and enable the immune system to mount specific immune

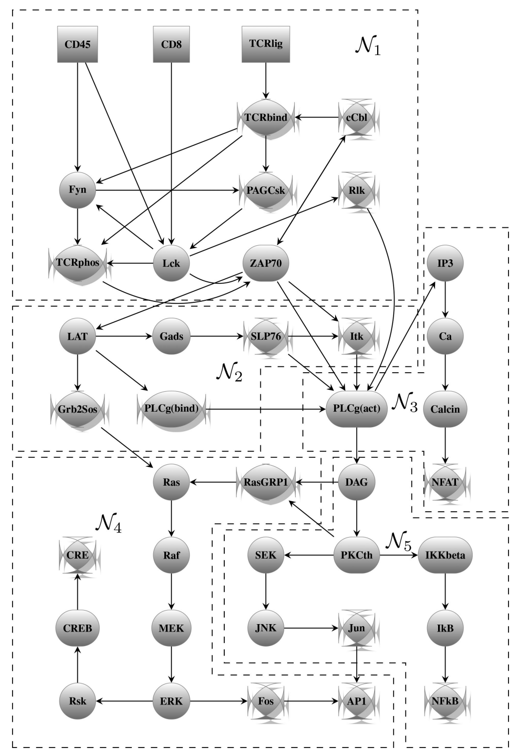


Fig. 8. Network of the T-cell receptor kinetics model (cf. [15]), where rectangles denote input nodes, the other nodes denote state nodes, particularly the nodes with shadows are chosen to be measured. The node aggregation shown in this figure is acyclic and satisfies Assumption 1.

responses. T-cells have the ability to recognize potentially dangerous agents and subsequently initiate an immune reaction against them. They do so by using T-cell receptors to detect foreign antigens bound to major histocompatibility complex molecules, and then activate, through a signaling cascade, several transcription factors. These transcription factors, in turn, influence the cell's fate such as proliferation. For the details, we refer the reader to [15]. The BCN T-cell receptor kinetics model given in [15] is shown in Table I, its network is shown in Fig. 8. In Fig. 8, there exist three input nodes and 37 state nodes. Hence, the model has 2^3 inputs and 2^{37} states. In order to do a quantitative analysis for the T-cell model, it would be better to obtain the state information of the model first. Next we study how to choose as few state nodes as possible to measure to make the model observable or reconstructible. We choose as few state nodes as possible because usually not all nodes can be directly measured [22]. It is useful in the biological application to find a minimal set of state nodes that need to be measured to make the whole BCN observable. It is almost impossible to use a PC to deal with such a large BCN directly. We next use the node aggregation approach instead.

B. Observability Analysis Based on a Particular Acyclic BCN Node Aggregation

In order to obtain the initial state of the BCN, one must choose some state nodes to measure, since there exists no output node in the network. In this sense, one should first choose some state nodes, and then assign each of these chosen state nodes one new output node as its child, such that each of these new added output nodes has only one parent. Then one can obtain the values of these chosen state nodes by measuring their corresponding output nodes. The chosen state nodes and their corresponding output nodes are represented as the nodes with shadows and their shadows, respectively, in Fig. 8. The main result obtained in this subsection is that we find the minimal set

$$\begin{aligned}&\{TCRbind, cCbl, PAGCsk, Rlk, TCRphos, SLP76,\\&Itk, Grb2Sos, PLCg(bind), CRE, AP1, NFkB,\\&NFAT, Fos, Jun, RasGRP1\}\end{aligned} \tag{18}$$

of state nodes that need to be measured to make the whole BCN observable. That is to say, if all nodes in (18) can be measured then the BCN is observable, and if any one of them cannot be measured then the BCN is not observable. In what follows, we prove this conclusion, and illustrate the process of looking for these state nodes.

We assume that it is not known which nodes in Fig. 8 are chosen to be measured, and next illustrate the process of looking for them. First of all, CRE, $AP1$, $NFkB$, and $NFAT$ must be chosen to be measured, because they have no children, and if any one of them cannot be measured, the whole BCN is not observable. By measuring these nodes, the initial values of $CREB$, Rsk, ERK, MEK, Raf, Ras, IkB, $IKKbeta$, $PKCth$, DAG, $Calcin$, Ca, $IP3$, and $PLCg(act)$ can be obtained. Taking $NFAT$ for example, by Table I, we have $Calcin(0) = NFAT(1)$, $Ca(0) = Calcin(1) = NFAT(2)$, $IP3(0) = NFAT(3)$, and $PLCg(act)(0) = NFAT(4)$, then one can obtain $NFAT(0)$, $Calin(0)$, $Ca(0)$, $IP3(0)$, and $PLCg(act)(0)$ by measuring $NFAT(0)$, $NFAT(1)$, $NFAT(2)$, $NFAT(3)$, and $NFAT(4)$, respectively. Second, since $AP1(t+1) = Fos(t) \wedge Jun(t)$ (cf. Table I) and both Fos and Jun only affect $AP1$ (cf. Fig. 8), Fos and Jun must be chosen to be measured. This is because if any one of Fos and Jun cannot be measured, say, Fos, then $Jun(0) = 0$ implies that $AP1(1) = 0$ no matter what $Fos(0)$ is and $Fos(0)$ cannot be obtained, i.e., the whole BCN is not observable. By measuring Jun, the initial values of JNK and SEK can be obtained. Third, since $Ras(t+1) = Grb2Sos(t) \vee RasGRP1(t)$ (also cf. Table I) and Ras is the unique child of both $Grb2Sos$ and $RasGRP1$ (also cf. Fig. 8), similarly to the case for Fos and Jun, we have $Grb2Sos$ and $RasGRP1$ must be chosen to be measured, or the whole BCN is not observable. By measuring $Grb2Sos$, the initial values of LAT and $ZAP70$ can be obtained. Fourth, we give an acyclic node aggregation for the BCN as shown in Fig. 8 (also cf. Fig. 9), and set that for each state node chosen to be measured, its corresponding new added output node belongs to the same part as the state node. By the above analysis, the BCN subnetworks $\mathfrak{B}_3, \mathfrak{B}_4$, and $\mathfrak{B}_5$ corresponding to subgraphs $\mathcal{N}_3, \mathcal{N}_4$, and $\mathcal{N}_5$ are observable. Next, we look for the minimal number of state nodes to be measured in subgraphs $\mathcal{N}_1$ and $\mathcal{N}_2$ to make the corresponding subnetworks observable by using Proposition 2.1. Since subgraph $\mathcal{N}_1$ has three input nodes and eight state nodes, and $\mathcal{N}_2$ has fewer input nodes and state nodes, we can use Proposition 2.1 to deal with them. For $\mathcal{N}_2$, if we choose all state nodes to be measured, then obviously the BCN subnetwork is observable. Since we want to know which node is necessary for the BCN subnetwork to be observable, we choose any five of the six state nodes in $\mathcal{N}_2$ to be measured, and verify observability of the BCN subnetworks by using Proposition 2.1. After verifying observability of these six subnetworks one by one, we find that $SLP76$, Itk, $Grb2Sos$, and $PLCg(bind)$ are necessary for $\mathfrak{B}_2$ to be observable, and the other two nodes are not. Also by Proposition 2.1, we obtain that if we choose only $SLP76$, Itk, $Grb2Sos$, and $PLCg(bind)$ to be measured, then the BCN subnetwork, denoted by $\mathfrak{B}_2$, is observable. Hence the set of these four nodes is the unique minimal set of nodes making the BCN subnetwork observable. Based on this, we choose $SLP76$, Itk, $Grb2Sos$, and $PLCg(bind)$ to be measured in $\mathcal{N}_2$. For $\mathcal{N}_1$, using the same method as for dealing with $\mathcal{N}_2$, by Proposition 2.1 we find that $TCRbind$, $cCbl$, $PAGCsk$, Rlk, and $TCRphos$ are necessary for the BCN $\mathfrak{B}_1$ to be observable, and the other 3 nodes are not. We also obtain that if we choose only $TCRbind$, $cCbl$, $PAGCsk$, Rlk, and $TCRphos$ to be measured, then the BCN subnetwork, denoted by $\mathfrak{B}_1$, is observable. Hence, the set of these five nodes is the unique minimal set of nodes making the BCN subnetwork observable.

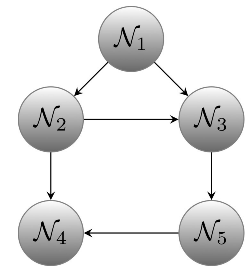


Fig. 9. Aggregated graph corresponding to Fig. 8.

Until now we have found all the 16 state nodes shown in (18). Next we prove that if we choose only these 16 nodes to be measured, then the whole BCN is observable. Note in that sense, the acyclic node aggregation shown in Fig. 8 satisfies Assumption 1, and we have proved that all subnetworks $\mathfrak{B}_i$ are observable, $i \in [1, 5]$, then by Theorem 3.1, the whole BCN is observable.

To conclude this part, we show that if any one of these 16 nodes in (18) cannot be measured, then the whole BCN is not observable. Previously we have shown that CRE, $AP1$, Jun, Fos, $NFkB$, $NFAT$, $RasGRP1$, and $Grb2Sos$ are necessary for the whole BCN to be observable, so if any one of these nodes cannot be measured, the whole BCN is not observable. Now consider Rlk, Itk, and $PLCg(bind)$. By Table I and Fig. 8, we have $PLCg(act)(t+1) = PLCg(bind)(t) \wedge SLP76(t) \wedge ZAP70(t) \wedge (Itk(t) \vee Rlk(t))$, all of Rlk, Itk and $PLCg(bind)$ affect only $PLCg(act)$. If Rlk (Itk, $PLCg(bind)$) cannot be measured and $SLP76(0) = 0$, then $PLCg(act)(1) = 0$ no matter what $Rlk(0)$ ($Itk(0)$, $PLCg(bind)(0)$) is, i.e., $Rlk(0)$ ($Itk(0)$, $PLCg(bind)(0)$) cannot be obtained, and hence the whole BCN is not observable. Consider $SLP76$. We have $SLP76$ affects only Itk and $PLCg(act)$, $PLCg(act)(t+1) = PLCg(bind)(t) \wedge SLP76(t) \wedge ZAP70(t) \wedge (Itk(t) \vee Rlk(t))$, and $Itk(t+1) = SLP76(t) \wedge ZAP70(t)$. If $SLP76$ cannot be measured, and $ZAP70(0) = 0$, then $PLCg(act)(1) = Itk(1) = 0$ no matter what $SLP76(0)$ is, i.e., $SLP76(0)$ cannot be obtained, and the whole BCN is not observable either. For $TCRphos$, we have $TCRphos$ affects only $ZAP70$, and $ZAP70(t+1) = (\neg cCbl(t)) \wedge Lck(t) \wedge TCRphos(t)$. For $PAGCsk$, we have $PAGCsk$ affects only Lck, and $Lck(t+1) = (\neg PAGCsk(t)) \wedge CD8(t) \wedge CD45(t)$. Similarly we have, if either $TCRphos$ or $PAGCsk$ cannot be measured, then the whole BCN is not observable. For $cCbl$, we have $cCbl$ affects only $TCRbind$ and $ZAP70$, $TCRbind(t+1) = (\neg cCbl(t)) \wedge TCRlig(t)$, and $ZAP70(t+1) = (\neg cCbl(t)) \wedge Lck(t) \wedge TCRphos(t)$. If $cCbl$ cannot be measured, and $TCRlig(0) = TCRphos(0) = 0$, then $TCRbind(1) = ZAP70(1) = 0$ no matter what $cCbl(0)$ is, i.e., $cCbl(0)$ cannot be obtained, and the whole BCN is not observable either. Finally, consider $TCRbind$. $TCRbind$ affects only Fyn, $TCRphos$, and $PAGCsk$. We have $Fyn(t+1) = CD45(t) \wedge (Lck(t) \vee TCRbind(t))$, $TCRphos(t+1) = Fyn(t) \vee (Lck(t) \wedge TCRbind(t))$, and $PAGCsk(t+1) = Fyn(t) \vee (\neg TCRbind(t))$. If $TCRbind$ cannot be measured, $CD45(0) = 0$, and $Fyn(0) = 1$, then $Fyn(1) = 0$, $TCRphos(1) = PAGCsk(1) = 1$ no matter what $TCRbind(0)$ is, i.e., $TCRbind(0)$ cannot be obtained forever, and then the whole BCN is not observable. This part has been finished. In addition, note that if Fyn cannot be measured, we cannot obtain that the whole BCN is not observable by using similar procedure. This is because Fyn affects only $PAGCsk$ and $TCRphos$, $PAGCsk(t+1) = Fyn(t) \vee (\neg TCRbind(t))$, $TCRphos(t+1) = Fyn(t) \vee (Lck(t) \wedge TCRbind(t))$; if $TCRbind(0) = 0$, then no matter what $Lck(0)$ is, $TCRphos(1) = Fyn(0)$; if $TCRbind(0) = 1$, then no matter what $Lck(0)$ is, $PAGCsk(1) = Fyn(0)$. That is, in both cases, $Fyn(0)$ can be measured. This procedure is not sufficient to prove that the whole BCN is observable either, so the node aggregation method and Proposition 2.1 are necessary.

On the other hand, a weaker type of observability (i.e., [29, Def. 5], not equivalent to the one studied in this article) of BCNs is characterized in [19] by using an algebraic method, and it is proved that for the BCN T-cell receptor kinetics model, the unique minimal set of nodes that need to be measured to make the whole BCN observable is

$$\begin{aligned}&\{TCRbind, Rlk, TCRphos, SLP76,\\ &Itk, Grb2Sos, PLCg(bind), CRE, AP1, NFkB,\\ &NFAT, Fos, Jun, RasGRP1\},\end{aligned} \tag{19}$$

which is a proper subset of (18).

C. Reconstructibility Analysis Based on a Particular Acyclic BCN Node Aggregation

In this subsection, we study reconstructibility of the BCN T-cell receptor kinetics model. Obviously, if we choose the 16 state nodes shown in (18) to be measured, then the whole BCN is reconstructible. However, to make the whole BCN reconstructible, we do not need to measure so many state nodes. In order to use the node aggregation method to characterize reconstructibility, we give a new node aggregation for its network consisting of two super nodes, where $\mathcal{N}_1$ is the same as the $\mathcal{N}_1$ in Fig. 8, while $\mathcal{N}_2$ is the set of all other nodes. The corresponding aggregated graph $\mathcal{N}_1 \to \mathcal{N}_2$ is also acyclic. For $\mathcal{N}_1$, by Proposition 2.3 we have that for each state node x in $\mathcal{N}_1$, if x cannot be measured and all other state nodes in $\mathcal{N}_1$ can be measured, then the BCN subnetwork $\mathfrak{B}_1$ is reconstructible. That is, none of state nodes of $\mathcal{N}_1$ is necessary for $\mathfrak{B}_1$ to be reconstructible.

We next assume that any state node in the set

$$\{TCRbind, cCbl, PAGCsk, Rlk, TCRphos\} \tag{20}$$

can be measured. In order to make this node aggregation satisfy Assumption 1, we add an output node to $\mathcal{N}_2$ satisfying $y(t) \equiv 1$, which actually means that none of state nodes in $\mathcal{N}_2$ can be measured.

We have proved that BCN subnetwork $\mathfrak{B}_1$ is observable, hence reconstructible. Next we prove that $\mathfrak{B}_2$ is also reconstructible. Then by Theorem 4.1, the whole BCN is reconstructible. Arbitrarily choose an initial state, an input sequence, and the corresponding output sequence. Since $\mathfrak{B}_1$ is observable, we actually know the values of every node in $\mathcal{N}_1$ at any time. For $\mathfrak{B}_2$, we have $LAT(t+1) = ZAP70(t)$, hence we know the value of LAT since time 1. Furthermore, we successively know the values of $Gads, Grb2Sos, PLCg(bind)$ since time 2; the value of $SLP76$ since time 3; the value of Itk since time 4; the value of $PLCg(act)$ since time 5; the values of $IP3, DAG$ since time 6; the values of $Ca, PKCth$ since time 7; the values of $Calcin, RasGRP1, SEK, IKKbeta$ since time 8; the values of $NFAT, Ras, JNK, IkB$ since time 9; the values of $Raf, Jun, NFkB$ since time 10; the value of MEK since time 11; the value of ERK since time 12; the values of Rsk, Fos since time 13; the values of $CREB, AP1$ since time 14; and the value of CRE since time 15. That is, $\mathfrak{B}_2$ is reconstructible.

Note that compared to measuring at least 16 state nodes to make the whole BCN observable, we only need to measure five state nodes to make the whole BCN reconstructible.

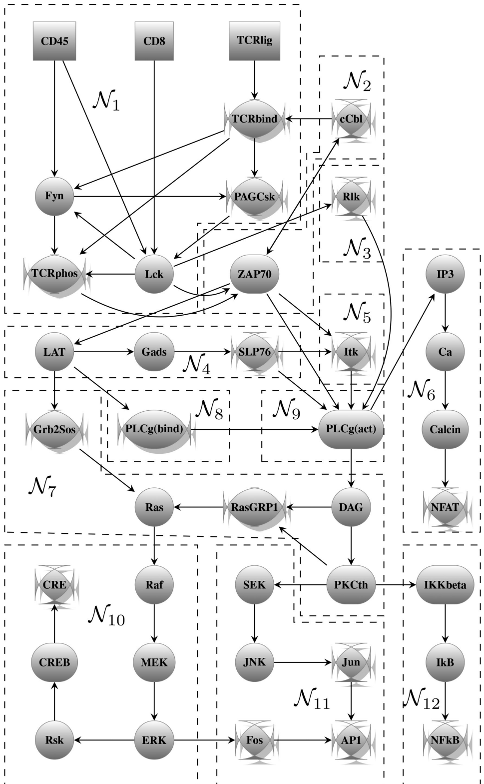


Fig. 10. Cyclic node aggregation of the network of the T-cell receptor kinetics model (cf. [15]) that satisfies Assumption 2.

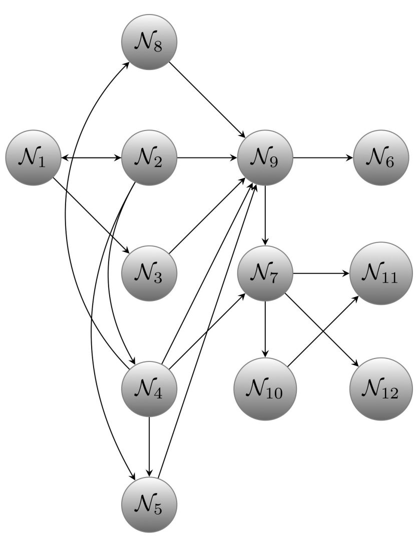


Fig. 11. Aggregated graph corresponding to Fig. 10.

D. Observability and Reconstructibility Analysis Based on a Particular FTS Node Aggregation

We use an FTS node aggregation proposed in Section V to verify observability for the T-cell model based on the selected output nodes as in (18). We choose a cyclic node aggregation as in Fig. 10. Its aggregated graph is shown in Fig. 11. This node aggregation contains 12 FTS subnetworks, and only one cycle $\mathcal{N}_1 \leftrightarrow \mathcal{N}_2$. The sets of output nodes for the corresponding FTS subnetworks $\mathfrak{F}_1, \dots, \mathfrak{F}_{12}$ are chosen as $\{TCRbind, PAGCsk, TCRphos, Rlk\}$, $\{cCbl\}$, $\{Rlk\}$, $\{SLP76\}$, $\{Itk\}$, $\{NFAT\}$, $\{Grb2Sos, RasGRP1, Raf, SEK\}$, $\{PLCg(bind)\}$, $\{IP3\}$, $\{CRE\}$, $\{Jun, AP1, Fos\}$, and $\{NFkB\}$, respectively. None of these 12 subnetworks is a BCN. One sees that the reordering $\mathfrak{F}_1, \mathfrak{F}_2, \mathfrak{F}_3, \mathfrak{F}_4, \mathfrak{F}_5, \mathfrak{F}_8, \mathfrak{F}_9, \mathfrak{F}_7, \mathfrak{F}_{10}, \mathfrak{F}_6, \mathfrak{F}_{11}, \mathfrak{F}_{12}$ satisfies the assumption in Theorem 5.1. One can also verify that $\mathfrak{F}_2, \dots, \mathfrak{F}_{12}$ are observable. For $\mathfrak{F}_1$, $Lck(0) = Rlk(1)$, $Fyn(0) = TCRphos(1)$ if $TCRbind(0) = 0$, $Fyn(0) = PAGCsk(1)$ if $TCRbind(0) = 1$, hence $\mathfrak{F}_1$ is also observable. Then by Theorem 5.1, the T-cell model is observable, and hence reconstructible.

Compared to the node aggregation shown in Fig. 8, the node aggregation in Fig. 10 contains more (smaller) subnetworks. Hence, the verification based on the latter node aggregation is more efficient for the T-cell model.

VII. Conclusion

We used a node aggregation method to reduce computational complexity of verifying observability and reconstructibility for large BCNs with special structures. We defined a first class of node aggregations whose subnetworks are BCNs, and then showed that even for this special class of node aggregations, the subnetworks being observable (reconstructible) does not imply the whole BCN being observable (reconstructible), and vice versa. However, for acyclic node aggregations in this class, we proved that the subnetworks being observable (reconstructible) implies that the whole BCN is observable (reconstructible). We proved that acyclic node aggregations are equivalent to cascading node aggregations frequently used in the literature. We showed that finding such node aggregations consisting of as many subnetworks as possible can reduce the computational complexity in verifying observability and reconstructibility. We also designed an efficient algorithm for searching all acyclic aggregations in this class.

In order to compensate for the drawback of the first class of node aggregations when the BCN has only a small number of output nodes, we also defined a second class of node aggregations with subnetworks being FTSs. We obtained several results on efficient observability and reconstructibility verification for large BCNs.

The first class of node aggregations characterized in this article can be used to deal with observability and reconstructibility also for discrete-time linear (special classes of nonlinear) control systems over Euclidean spaces with special network structures. Taking discrete-time linear time-invariant control systems, for example, if an acyclic node aggregation satisfying Assumption 1 has been found, then one can verify observability of the whole system by verifying observability of each subsystem by using the well-known observability rank criterion. Since given the dimensions of input space, state space, and output space, the set of observable linear time-invariant control systems is dense in the set of linear time-invariant control systems, the node aggregation method is feasible. Further discussion is left for future study. Our results in this article motivate also the study of observability and reconstructibility of large BCNs based on different types of node aggregations.

References

[1] T. Akutsu, M. Hayashida, W. K. 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.

[2] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions," *Genome Inform. Workshop Genome Informat.*, vol. 9, pp. 151–160, 1998.

[3] D. Cheng, "Disturbance decoupling of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 2–10, Jan. 2011.

[4] D. Cheng, Z. Li, and H. Qi, "Realization of Boolean control networks," *Automatica*, vol. 46, no. 1, pp. 62–69, 2010.

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

[6] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-tensor Product Approach*. London, U.K.: Springer-Verlag, 2011.

[7] D. Cheng and Y. Zhao, "Identification of Boolean control networks," *Automatica*, vol. 47, no. 4, pp. 702–710, 2011.

[8] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry, "Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle," *Bioinformatics*, vol. 22, no. 14, pp. e124–e131, 2006.

[9] E. Fornasini and M. E. Valcher, "Observability, reconstructibility and state observers of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 58, no. 6, pp. 1390–1401, Jun. 2013.

[10] Y. Guo, R. Zhou, Y. Wu, W. Gui, and C. Yang, "Stability and set stability in distribution of probabilistic Boolean networks," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 736–742, Feb. 2019.

[11] T. Ideker, T. Galitski, and L. Hood, "A new approach to decoding life: systems biology," *Annu. Rev. Genomics Human Genetics*, vol. 2, pp. 343–372, 2001.

[12] H. Ishii, R. Tempo, and E. W. Bai, "A web aggregation approach for distributed randomized pagerank algorithms," *IEEE Trans. Autom. Control*, vol. 57, no. 11, pp. 2703–2717, Nov. 2012.

[13] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, 1969.

[14] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, pp. 1662–1664, 2002.

[15] S. Klamt, J. Saez-Rodriguez, J. A. Lindquist, L. Simeoni, and E. D. Gilles, "A methodology for the structural and functional analysis of signaling and regulatory networks," *BMC Bioinf.*, vol. 7, no. 1, 2006, Art. no. 56.

[16] D. Laschov, M. Margaliot, and G. Even, "Observability of Boolean networks: A graph-theoretic approach," *Automatica*, vol. 49, no. 8, pp. 2351–2362, 2013.

[17] F. Li, "Pinning control design for the synchronization of two coupled Boolean networks," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 63, no. 3, pp. 309–313, Mar. 2016.

[18] H. Li and Y. Wang, "Logical matrix factorization with application to topological structure analysis of Boolean network," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1380–1385, May 2015.

[19] R. Li, M. Yang, and T. Chu, "Controllability and observability of Boolean networks arising from biology," *Chaos: An Interdisciplinary J. Nonlinear Sci.*, vol. 25, no. 2, 2015, Art. no. 023104.

[20] Z. Li and J. Song, "Controllability of Boolean control networks avoiding states set," *Sci. China Inf. Sci.*, vol. 57, no. 3, pp. 1–13, 2014.

[21] Y. Liu, H. Chen, J. Lu, and B. Wu, "Controllability of probabilistic Boolean control networks based on transition probability matrices," *Automatica*, vol. 52, pp. 340–345, 2015.

[22] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Observability of complex systems," *Proc. Nat. Acad. Sci.*, vol. 110, no. 7, pp. 2460–2465, 2013.

[23] A. Louati, M.-A. Aufaure, and Y. Lechevallier, "Graph Aggregation: Application to Social Networks." in *Advances in Theory and Applications of High Dimensional and Symbolic Data Analysis*, volume RNTI-E-25, Rong Guan, Yves Lechevallier, Gilbert Saporta, and Huiwen Wang, Eds. Paris, France: Hermann, 2013, pp. 157–177.

[24] S. Sridharan, R. Layek, A. Datta, and J. Venkatraj, "Boolean modeling and fault diagnosis in oxidative stress response," *BMC Genomics*, vol. 13, no. 6, 2012, Art. no. S4.

[25] Y. Wu and T. Shen, "A finite convergence criterion for the discounted optimal control of stochastic logical networks," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 262–268, Jan. 2018.

[26] K. Zhang and K. H. Johansson, "Efficient observability verification for large-scale Boolean control networks," in *Proc. 37th Chin. Control Conf.*, Jul. 2018, pp. 560–567.

[27] K. Zhang, T. Liu, and D. Cheng, "Observability of finite labeled transition systems," *IEEE Trans. Autom. Control*, vol. 63, no. 6, pp. 1591–1602, Jun. 2018.

[28] K. Zhang and M. Zamani, "Detectability of nondeterministic finite transition systems," in *Proc. IFAC PapersOnline*, vol. 50, no. 1, pp. 9272–9277, 2017.

[29] K. Zhang and L. Zhang, "Observability of Boolean control networks: A unified approach based on finite automata," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2733–2738, Sep. 2016.

[30] K. Zhang, L. Zhang, and R. Su, "A weighted pair graph representation for reconstructibility of Boolean control networks," *SIAM J. Control Optim.*, vol. 54, no. 6, pp. 3040–3060, 2016.

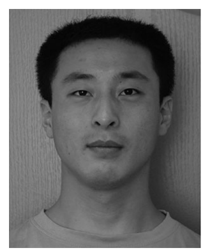
[31] K. Zhang, L. Zhang, and L. Xie, "Invertibility and nonsingularity of Boolean control networks," *Automatica*, vol. 60, pp. 155–164, 2015.

[32] Y. Zhao, B. K. Ghosh, and D. Cheng, "Control of large-scale Boolean networks via network aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1527–1536, Jul. 2016.

[33] Y. Zhao, J. Kim, and M. Filippone, "Aggregation algorithm towards large-scale Boolean network analysis," *IEEE Trans. Autom. Control*, vol. 58, no. 8, pp. 1976–1985, Aug. 2013.

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

[35] Y. Zou and J. Zhu, "Kalman decomposition for Boolean control networks," *Automatica*, vol. 54, pp. 65–71, 2015.



Kuize Zhang (Senior Member, IEEE) received the B.S. and Ph.D. degrees in mathematics and control science and engineering from Harbin Engineering University, Harbin, China, in 2009 and 2014, respectively.

He held visiting positions with the University of Turku, Turku, Finland, from September 2012 to September 2013, and the Chinese Academy of Sciences, Beijing, China, from January to October 2015. He was a Project Officer with Nanyang Technological University, Singapore, from October 2013 to October 2014, Postdoc with Chinese Academy of Sciences from November 2015 to October 2017, and Postdoc with Technical University of Munich, Munich, Germany, from September 2016 to August 2017. He is currently a Postdoc with the KTH Royal Institute of Technology, Stockholm, Sweden. He has authored or coauthored a book *Discrete-Time and Discrete-Space Dynamical Systems*. His current research interests include formal methods, Boolean networks, discrete-event systems, etc.

Dr. Zhang was the recipient of the Humboldt Research Fellowship, in 2019, and the 2016 Chinese Association of Automation Outstanding Doctoral Thesis Nomination Award.

Karl Henrik Johansson (Fellow, IEEE) received the M.Sc. and Ph.D. degrees from Lund University, Lund, Sweden. He is Director of the Stockholm Strategic Research Area ICT the next generation and Professor with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. He has held visiting positions with UC Berkeley, California Institute of Technology, Nanyang Technological University, Hong Kong University of Science and Technology Institute of Advanced Studies, and Norwegian University of Science and Technology. His research interests are in networked control systems, cyber–physical systems, and applications in transportation, energy, and automation.

Dr. Johansson is a member of the IEEE Control Systems Society Board of Governors, the International Federation of Automatic Control (IFAC) Executive Board, and the European Control Association Council. He is Fellow of the Royal Swedish Academy of Engineering Sciences. He is an IEEE Distinguished Lecturer. He is a Distinguished Professor with the Swedish Research Council and a Wallenberg Scholar. He was the recipient of several best paper awards and other distinctions. He was also the recipient of the Future Research Leader Award from the Swedish Foundation for Strategic Research and the triennial Young Author Prize from IFAC.