Online Observability of Boolean Control Networks

Guisen Wu, Liyun Dai, Zhiming Liu, Taolue Chen, Jun Pang, Hongyang Qu

Abstract—Four types of observability of Boolean control networks (BCNs) have been proposed to study the initial state of BCNs. However, all of them are offline observability meaning that once the input sequence is determined, it remains unchanged during the process of determining the BCN’s initial state. It makes us not find the input sequence to determine the initial state of some biological systems. In order to solve this problem and make some optimization, we propose the notion of online observability. In online observability, we deriving and deciding input sequence based on the output we observed in every time step when we determine the initial state in real time. Moreover, we also propose determination algorithms and applications of the online observability to further illustrate its advantages.

I. INTRODUCTION

Before introducing the Boolean control networks, we introduce the Boolean networks at first. In 1960s, Nobel Prize winners Jacob and Monod found that “Any cell contains a number of ‘regulatory’ genes that act as switches and can turn one another on and off. If genes can turn one another on and off, then you can have genetic circuits.” [1]. Inspired by these Boolean-type actions in genetic circuits, the Boolean networks (BNs) is firstly proposed by Kauffman [2] for modeling nonlinear and complex biological systems. It is a type of discrete systems which based on a directed graph. In a Boolean network, each node has only two states “0” and “1”, and each node can only be in one of these two values at a time step. The value of each node \( n_i \) at the next time step is affected by the value of another node \( n_j \) if there is a directed edge from \( n_j \) to \( n_i \), where \( n_j \) and \( n_i \) can be the same node. We regard these nodes as the neighboring nodes of \( n_i \) and the values of these neighboring nodes is the input. And then a new value of the node \( n_i \) is obtained through a series of logical operations. The logical operators used in the operation include: AND, OR, NO, XOR, and so on. Some general descriptions of the BNs and their applications to biological systems can be found in [2]. Since then research interests in BNs have been motivated by the large number of natural and artificial systems [3], [4], [5], [6], [7]. That these systems’ describing variables display only two distinct configurations, then these describing variables take only two values, i.e., \( \{0, 1\} \).

Then BNs are naturally extended to Boolean control networks (BCNs) when external regulation or perturbation is considered [9]. Different from BNs, there are three kinds of nodes in BCNs, they are input-nodes, state-nodes and output-nodes. In BCNs, we can only control the value of the input-nodes and observe the value of the output-nodes. However, the value of each state-node \( s \) can be reflected by the value of an output-node \( o \) if there exists a directed edge from \( s \) to \( o \). And the value of each state node \( s_i \) at the next time step is affected by the value of a state-node \( s_j \) (or an input-node \( i \)) if there is a directed edge from \( s_j \) (or \( i \)) to \( s_i \), where \( s_j \) and \( s_i \) can be the same node. Therefore, there are also a series of logical operations (updating rules) to obtain the new values of the state nodes and output nodes of BCNs. BCNs can be used to solve various real problems, for instance, structural and functional analysis of signaling and regulatory networks [10], [11], abduction based drug target discovery [12], and pursuing evasion problems in polygonal environments [13].

As the wide application of BCNs, there are a lot of research work about the control-theoretic problems of BCNs. The work in [14] proves that the problem of determining the controllability of BCNs is NP-hard in the number of nodes. In addition, it points out that “One of the major goals of systems biology is to develop a control theory for complex biological systems.” Since then, the study on control-theoretic problems in the areas of BNs and BCNs has drawn great attention [15], [16], [17], [18], [8]. What is more, the controllability and observability are the basic control-theoretic problems of BCNs.

However, in this paper we research the observability of the BCNs. The concept of observability was proposed firstly in [15]. To date, there are four types of observability have been proposed. And they are mainly about how to get some information of the initial value of the state-nodes of the BCNs by the value of their input-nodes and output-nodes. Because the value of state-nodes can be reflected by the value of output-nodes. Moreover, the value of state-nodes at the next time step is affected by the value of state-nodes and input-nodes. Such that we can control and predict the new value of the state-nodes by the value of input-nodes. Therefore, with the updating rules of BCNs we get some information about the initial value of state-nodes of BCNs by the value of input-nodes and output-nodes. Then based on actual application needs, different types of observability are defined to get different kinds of information about the initial value of state-nodes.

For convenience, we use the vectors input \( i \), state \( s \) and output \( o \) to represent the value of the input-nodes, state-nodes and output-nodes of a BCN respectively. Then an input sequence \( i_0, i_1, \ldots, i_{p-1} \) consists of several inputs in sequential time steps, a state sequence \( s_0, s_1, \ldots, s_{p-1} \) consists of several states in sequential time steps, and an output sequence \( o_0, o_1, \ldots, o_{p-1} \) consists of several outputs in sequential time steps. Such that, for the initial state \( s_0 \) of a BCN and its

Guisen Wu, Liyun Dai, and Zhiming Liu are with the School of Computer and Information Science, Southwest University Chongqing, 400715 China. E-mail: \{wgs233, dailiyun, zhimingliu88\}@swu.edu.cn.

Taolue Chen is with the Department of Computer Science and Information Systems, Birkbeck, University of London. E-mail: taolue@cs.bbk.ac.uk.

Jun Pang is with the Faculty of Science, Technology, Engineering, University of Luxembourg e-mail: jun.pang@uni.lu.

Hongyang Qu was with the Department of Automatic Control and Systems Engineering, University of Sheffield. E-mail: h.qu@sheffield.ac.uk.
input sequence $i_0, i_1, \ldots, i_{p-1}$, we have a corresponding state sequence $s_0, s_1, \ldots, s_p$ and a corresponding output sequence $o_0, o_1, \ldots, o_p$ for this BCN.

As we mentioned before, there are four types of observability have been proposed. The four existing observability of BCNs are as follows.

1) The first type of observability proposed in 2009 [15], and it means that every initial state $s_0$ can be determined by an input sequence $i_0, i_1, \ldots, i_{p-1}$. Such that, as we inputing $i_0, i_1, \ldots, i_{p-1}$, the corresponding output sequence $(o_0, o_1, \ldots, o_p)$ of $s_0$ is different from the corresponding output sequence of any other types of initial state. Therefore, we can distinguish the $s_0$ from other types of initial state by this input sequence, and then we can determine whether the initial state is $s_0$ by this input sequence.

2) The second observability proposed in 2010 [16], and it is determined in [19]. It stands that for every two distinct initial states $s_0$ and $s_0'$, there exists an input sequence $i_0, i_1, \ldots, i_{p-1}$ which can distinguish them. Such that, as we inputing $i_0, i_1, \ldots, i_{p-1}$, the corresponding output sequences $(o_0, o_1, \ldots, o_p, o_0', o_1', \ldots, o_p')$ of them are different from each other. Therefore, we can distinguish between these two types of initial state $s_0$ and $s_0'$ by the corresponding input sequence.

3) The third observability proposed in 2011 [17], and it states that there is an input sequence $i_0, i_1, \ldots, i_{p-1}$ that determines the initial state $s_0$. Such that, the corresponding output sequence of all types of initial state are different. Therefore, we can determine the initial state of the BCN by this input sequence for every initial state.

4) The fourth observability proposed in 2013 [18] is essentially the observability of linear control systems, i.e., every sufficient long input sequence $i_0, i_1, \ldots, i_{p-1}$ can determine the initial state $s_0$. Such that, the corresponding output sequence of any types of initial state are different. Therefore, we can determine the initial state of the BCN by every sufficient long input sequence for every initial state.

In Section [17] we will present the formal definition of four observability completely and introduce their implication relationship

In the four existing observability, we can not determine the initial state of BCNs in real time by the first and second observability. Although we can determine the initial state of BCNs in real time by the third and fourth observability, the requirements for BCNs to determine the initial state are very harsh. Thus, we consider that whether we can determine the initial state of some BCNs which can not be determined by the third and fourth observability.

In the process determining the initial state, we infer the set of possible initial states $S_k$ by observing the output of $BCN$ at every time step $k$. The input $i_k$ we chose should make every two distinct states $(S_k^1, S_k^2)$ in the set of possible initial states will not turn into be the same state after affected by $i_k$. And the new set of possible initial states $S_{k+1}$ is derived by the input $i_k$ we chose and the new output $o_{k+1}$ we observe at the time step $k + 1$, and we have $|S_{k+1}| \leq |S_k|$. If $|S_{k+1}| = 1$, we can determine the initial state $s_0$ of the $BCN$.

What’s more, in the third and fourth observability, if a $BCN$ satisfies the observability there has to exist an input sequence that determines its initial state $s_0$. But we can also determine the set of possible initial states $S_0$ by initial output $o_1$ we observe, and then we use different input sequences to determine initial state ($s_0 \in S_0$) for different sets of possible initial states. In this case, the requirements for BCNs to determine the initial state would be less harsh because we utilize the set of possible initial states $S_0$ to find the input sequence. Then we propose the online observability in this paper which means that we find input sequence by the $S_k$, where $S_k$ is the set of initial states derived at every time step $k$. With the online observability, we can determine the initial state of some BCNs which can not be determined before.

In the online observability, we make full use of the input and output of BCNs at every time step to determine their initial state. Therefore, a BCN is online observable iff we can determine its initial state $s_0$ in real time for every initial state $s_0$. Comparing with the existing first and second observability, the online observability has the real-time property because we can determine the initial state in real time. However, comparing with the existing third and fourth observability, it has interactivity that we would make full use of the input and output of a BCN to determine its initial state. With the real-time property and interactivity, we call this type of observability online observability.

In order to study online observability better, we formulate the formal definition for it. Firstly, we define the derivation function to describe the derivation process of the state of BCNs by their output and input at every time step. Secondly, we propose the definition of the $K$-step determinability to present that we can use a set of possible current states to determine the current state of BCNs in real time in finite time steps. The derivation function and $K$-step determinability are all preparations for defining the the online observability. Thirdly, we define the online observability that for every set of possible states which derived at the time step 0, it satisfy $K$-step determinability. By the definition, we prove that online observability is the necessary and sufficient condition of determine the initial state of BCNs in real time for every initial state. Finally, we compare the online observability with existing four observability. That the online observability implies the first and second observability but not imply the third and fourth observability.

After we defined the online observability and compared it with with existing four observability, we propose two algorithms to determine it. The first one is the supertree-based algorithm, the supertree of a $BCN$ intuitively depicts how to derive the state of the $BCN$ by alternately observing the output and then deriving and deciding the input until we can determine the state of $BCN$. But there are some shortcomings in this algorithm, so we propose the algorithm based on directed graph. In the algorithm based on directed graph, we check whether the $K$-step determinability of the sets with less possible states and then check the sets with more possible states. So that, it can help us find all paths to determine the
initial state of BCNs.

Finally, in order to further illustrate the advantages of the online observability, we present how to use it to do some optimization in the process of determining the initial state. The first one is to find the shortest path and the second one is to avoid entering critical states. It because that the interactivity let the online observability has better performance than existing observability in the dynamic analysis of BCNs.

In conclusion, in this paper we make the following contributions.

Contributions: Firstly, we propose and formally define the concept of online observability of BCNs. Comparing with existing observability, the online observability can help to determine the initial state of some biological systems. Secondly, in addition to theoretical research, we also provide two algorithms to determine the online observability for BCNs. Finally, we present some optimization brought by the online observability of BCNs. Including methods to find shortest path and approaches to avoid entering critical states in the process of determining the initial state of BCNs. These optimization further explain the advantages of online observability of BCNs.

Then the remainder of this paper is organized as follows. Structure: Section II introduces necessary preliminaries about BCNs, algebraic forms of BCNs and the four existing types of observability of BCNs. Section III presents the definition of derivation function, k steps determinability and online observability of BCNs. Section IV presents how to determine the online observability of BCNs by super tree and directed graph. Section V talks about some applications of the online observability of BCNs. We also compare the online observability with offline observability in this section. Section VI ends up with the introduction of our future work.

II. PRELIMINARIES

In this section we introduce the definition of BCNs and their algebraic forms, as well as the four existing types of observability.

\( \mathbb{B} : \) the set \{0, 1\}; \( t = 0, 1, \ldots \) represents the discrete time.

A. Boolean Control Networks

A Boolean control network can be defined as a directed graph together with logical equations to describe the updating rules of the nodes of the directed graph. The formal definition of BCN is given as follows.

Definition 1 (Boolean Control Networks, [9]): A BCN consists of the topology and the associated updating rules. The topology is captured by a directed graph which consists of input-nodes, state-nodes, output-nodes, and directed edges which connect nodes.

- Every node in a BCN can take a logic value from \{0, 1\} at a discrete time \( t = 0, 1, 2, \ldots \).
- Every directed edge from a state-node \( s_1 \) (or an input-node \( i_1 \)) to a state-node \( s_2 \) means that the logic value of \( s_2 \) at time step \( t+1 \) is affected by the logic value of \( s_1 \) (or \( i_1 \)) at time step \( t \).
- Every directed edge from a state-node \( s_1 \) to an output-node \( o_1 \) means that the logic value of \( o_1 \) at time step \( t \) is affected by the logic value of \( s_1 \) at time step \( t \).

Assuming that the BCN has \( n \) state-nodes, \( m \) input-nodes and \( q \) output-nodes. Then the updating rules of the BCN can be described as following formulas:

\[
\begin{align*}
    s(t+1) &= f(i(t), s(t)) \\
    o(t) &= h(s(t))
\end{align*}
\]

where

- \( s(t) \in \mathbb{B}^n \) is a state which represent the logic value of all state-nodes at time step \( t \);
- \( i(t) \in \mathbb{B}^m \) is an input which represent the logic value of all input-nodes at time step \( t \);
- \( o(t) \in \mathbb{B}^q \) is a output which represent the logic value of all output-nodes at time step \( t \);
- \( f : \mathbb{B}^{n+m} \rightarrow \mathbb{B}^n \) and \( h : \mathbb{B}^n \rightarrow \mathbb{B}^q \) are logical functions that represent the updating rules of the BCN.

In order to better illustrate the definition of BCN, we give an example as follows.

![Fig. 1. A Boolean control network with two input-nodes A and B, four state-nodes C, D, E and F, and two output-nodes G, W. We use blue, black and orange, to distinguish three types of nodes and three types of edges.](image-url)

Example 1: In Fig.1 we have a BCN with two input-nodes A and B, four state-nodes C, D, E and F, and two output-nodes G, W. We use blue, black and orange, to distinguish three types of nodes and three types of edges.
for a BCN to be converted into its algebraic form. What’s more, for convenience, we will use this example to explain various concepts throughout this paper.

Fig. 2. The truth table which describe the updating rules of the BCN shown in Fig.

B. The algebraic forms of BCNs

As mentioned in the Section 2, STP is one of useful tools to deal with both BNs and BCNs related problems [15]. The STP algebraic form of BCNs helps us to introduce four existing observability of BCN and define the online observability of BCN. The definition of STP is as follows.

Definition 2 (STP): [18] Let $X \in \mathbb{R}_{m \times n}$, $Y \in \mathbb{R}_{p \times q}$ and $\alpha = \text{lcm}(n, p)$ be the least common multiple of $n$ and $p$. The STP of $X$ and $Y$ is defined as

$$X \ltimes Y = (X \otimes I_{\alpha/n})(Y \otimes I_{\alpha/p}),$$

where $\otimes$ denotes the Kronecker product.

After introducing the definition of STP of matrices, we introduce some related notations at first [20]:

- $\delta_n$: the $i$-th column of the identity matrix $I_n$;
- $\Delta_n$: the set $\{\delta_n, \ldots, \delta_n\}$;
- $\delta_n[i_1, \ldots, i_s]: \{\delta_n[1], \ldots, \delta_n[s] \mid \{i_1, \ldots, i_s\} \subseteq \{1, 2, \ldots, n\}\}$
- the logical matrix;
- $L_{n \times s}$: the set of $n \times s$ logical matrices.

Using STP of matrices, the updating rules of the BCN [1] can be equivalently represented in the following algebraic form:

Definition 3:

$$s(t + 1) = L \ltimes i(t) \ltimes s(t)$$

$$o(t) = H \ltimes s(t)$$

where $s(t) \in \Delta_N$, $i(t) \in \Delta_M$, and $o(t) \in \Delta_Q$ denote the states, inputs and outputs respectively the same as in formula [1], but $s(t)$, $i(t)$ and $o(t)$ in formula [2] written with the special vector forms; $L \in L_{N \times (N \times M)}$ and $H \in L_{Q \times N}$ denote the relation matrices, where $N = 2^n$, $M = 2^m$, and $Q = 2^q$.

Since STP keeps most properties of the conventional product [15], the associative law, the distributive law, etc., we usually omit the symbol “$\ltimes$” hereinafter. For instance, the formula

$$s(t + 1) = L \ltimes i(t) \ltimes s(t)$$

will be written as

$$s(t + 1) = Li(t)s(t).$$

After the introduction of the algebraic forms of BCNs, we introduce the process of constructing a BCN algebraic form. In order to construct the algebraic form of BCN [2] we give a mapping

$$\tau : \{0, 1\} \mapsto \{\delta_2^1, \delta_2^2\},$$

where $\tau(0) = \delta_2^1$, $\tau(1) = \delta_2^2$. Therefore, the logical variable $A(t)$ takes value from these two vectors, i.e., $A(t) \in \{\delta_2^1, \delta_2^2\}$. Using the STP of matrices, we have

$$i(t) = i_1(t) \ldots i_m(t);$$

$$s(t) = s_1(t) \ldots s_n(t);$$

$$o(t) = o_1(t) \ldots o_p(t).$$

And according to [21], for the logical function of each state-node $f_p$ which can be found in the updating rules [1] that

$$f_p(i_1(t), \ldots, i_m(t), s_1(t), \ldots, s_n(t)),$$

there exists a logical matrix $L_p \in L_{2 \times NM}$ such that

$$\tau(f_p(i_1(t), \ldots, i_m(t), s_1(t), \ldots, s_n(t)))) = L_p(i(t)s(t)) \quad (3)$$

Therefore for state-nodes $s_1, \ldots, s_n$, we have $n$ logical matrices $L_1, \ldots, L_n$ for them, respectively. If for each state-node $s_p$ the logical matrix has its form

$$L_p = [\delta_{2}^1, \ldots, \delta_{2}^{PNM}],$$

then we have that

$$L = [\delta_{N}^{R_1}, \ldots, \delta_{N}^{R_{PNM}}]$$

where

$$\delta_{N}^{R_1} = \delta_{2}^{11} \ldots \delta_{2}^{n_1};$$

$$\delta_{N}^{R_{PNM}} = \delta_{2}^{1m} \ldots \delta_{2}^{n_{PNM}}.$$
C. Four existing observability of BCNs

After introducing the algebraic forms of BCNs, we introduce four existing types of observability of BCNs in this subsection. In order to introduce four existing types of observability of BCNs, we define the mappings [20]:

\[
\begin{align*}
L_p^0 & : (\Delta_M)^p \mapsto (\Delta_N)^p, i_0 \ldots i_{p-1} \mapsto s_1 \ldots s_p \\
L_\infty^p & : (\Delta_M)^\infty \mapsto (\Delta_N)^\infty, i_{01} \ldots i_{s_1s_2} \ldots \\
(HL)_p^0 & : (\Delta_M)^p \mapsto (\Delta_N)^p, i_0 \ldots i_{p-1} \mapsto s_1 \ldots o_p \\
(HL)^\infty & : (\Delta_M)^\infty \mapsto (\Delta_N)^\infty, i_{01} \ldots \mapsto o_1o_2 \ldots
\end{align*}
\]

(5)

Where \( \Delta_N, \Delta_M \) and \( \Delta_Q \) are three alphabets, \( s_0 \in \Delta_N, p \in \mathbb{Z}_+ \) and \( \infty \) is the infinite natural numbers. For all \( p \in \mathbb{Z}_+ \),

\[ I = i_0 \ldots i_{p-1} \in (\Delta_M)^p \]
is an input sequence,

\[ L_p^0(I) = s_1 \ldots s_p \in (\Delta_N)^p \]
is a state sequence and

\[ (HL)_p^0 = a_0 \ldots o_p \in (\Delta_N)^p \]
is an output sequence. For the \( \infty \),

\[ I = i_0 \ldots \in (\Delta_M)^\infty \]
is an infinitely long input sequence,

\[ L_\infty^0(I) = s_1 \ldots \in (\Delta_N)^\infty \]
is an infinitely long state sequence and

\[ (HL)^\infty(I) = a_0 \ldots \in (\Delta_N)^\infty \]
is an infinitely long output sequence. From the algebraic forms of BCNs, in the formula [5], for any \( 1 \leq k \leq |I| \) we have

\[ s_k = L_i(k-1)s_k-1. \]

In the formula [5] and formula [6], for any \( 1 \leq k \leq |I| \) we have

\[ o_k = Hs_k = HL_i(k-1)s_k-1. \]

Then four existing types of observability of BCNs can be defined as follows.

Definition 4: The first type of observability is that, a BCN is called observable, if for every initial state \( s_0 \in \Delta_N \), there exists an input sequence \( I \in (\Delta_M)^p \) for some \( p \in \mathbb{Z}_+ \) such that for every two states \( s_0 \neq s_0' \in \Delta_N \), \( Hs_0 = Hs_0' \) implies \( (HL)^p_{s_0}(I) \neq (HL)^p_{s_0'}(I) \) [16].

The first observability means that a BCN is called observable if there exists an input sequence which can distinguish any two states in the same BCN.

Definition 5: The second type of observability is that, a BCN is called observable if for any distinct states \( s_0, s_{0}' \in \Delta_N \), there exists an input sequence \( I \in (\Delta_M)^p \) for some \( p \in \mathbb{Z}_+ \), such that \( Hs_0 = Hs_{0}' \) implies \( (HL)^p_{s_0}(I) \neq (HL)^p_{s_{0}'}(I) \) [16].

The second observability means that a BCN is called observable if there exists an input sequence which can distinguish them.

Example 5: For example, for the BCN mentioned in Example 7, we have for every two distinct initial states of the BCN, there exists an input sequence which can distinguish them for instance,

- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;
- the states \( \delta_1^3 \) and \( \delta_2^3 \) can be distinguished by any input sequence which with the prefix \( \delta_3^3 \), \( \delta_4^3 \) or \( \delta_5^3 \) etc;

Therefore we have this BCN satisfies the existing first observability.

Definition 6: The third type of observability is that, a BCN is called observable if for any distinct states \( s_0, s_{0}' \in \Delta_N \), there exists an input sequence \( I \in (\Delta_M)^p \) for some \( p \in \mathbb{Z}_+ \), such that for any distinct states \( s_0, s_{0}' \in \Delta_N \), \( Hs_0 = Hs_{0}' \) implies \( (HL)^p_{s_0}(I) \neq (HL)^p_{s_{0}'}(I) \) [17].

The third observability means that a BCN is called observable if there exists an infinitely long input sequence which can determine the initial state \( s_0 \) of the BCN for every \( s_0 \in \Delta_N \).

Example 5: For example, for the BCN mentioned in Example [7] we have for every distinct initial states of the BCN, there exists an input sequence which can determine the initial state \( s_0 \) of the BCN for every \( s_0 \in \Delta_N \).

Then four existing types of observability of BCNs can be defined as follows.

Definition 7: The fourth type of observability is that, a BCN is called observable, if for any distinct states \( s_0, s_{0}' \in \Delta_N \), for any input sequence \( I \in (\Delta_M)^\infty \), \( Hs_0 = Hs_{0}' \) implies \( (HL)^\infty_{s_0}(I) \neq (HL)^\infty_{s_0'}(I) \) [8].

The fourth observability means that a BCN is called observable if every sufficient long input sequence can determine the initial state \( s_0 \) of the BCN for every \( s_0 \in \Delta_N \).

Example 6: For example, for the BCN mentioned in Example [7] we have there exists at least one sufficient long input sequence that can not determine the initial state of this BCN.

What is more, we know the implication relationships of four existing types of observability from the definitions of them [20].
Where the implication relationship is that “The first observability implies the second observability,” means “If a BCN satisfies the first observability then it satisfies the second observability.” For the details of the proving process of this proposition, we refer readers to [20].

After introducing the definition of four existing observability and their implication relationship, we discuss how to determine the initial state of some BCNs in real time by four existing types of observability.

- The first and second observability can not help us to determine the initial state of some BCNs in real time. Because in the first observability we need to assume that the initial state \( s_0 \) of a BCN is \( s_i \), and then check it by corresponding input sequence \( I_i \) of this state. If the initial state we assume is correct, then we can determine the initial state \( s_0 = s_i \). But if the assumption is not correct, we can not determine the initial state of the BCN. Therefore we need to check several test cases (with the same initial state \( s_0 \) of this BCN) untill we can determine the initial state of them. So do the existing second observability of BCNs.

- We can use the third existing observability and fourth existing observability to determine the initial state of BCNs in real time, because we do not need to presuppose the initial state. But the requirements for BCNs are very harsh when we use the third observability and fourth observability. Because we do not make full use of the output and input in the process of determining the initial state. The output of BCNs we observed at every time step can help us further determine the range of the initial state. Then we can use different input sequence to determine the initial state based on the range of the initial state. However, in the third and fourth existing observability, we have to use the same input sequence to determine the initial state.

In some biological systems (depicted by BCNs), the initial state of them can be checked at most once, i.e., we can only check the initial state of some biological systems in real time. Therefore, we can not use the first observability and second observability to determine the initial states of them in real time. Moreover, in some biological systems, it would takes many costs to check these biological systems. Hence we will spend a lot of overhead to determine the initial states of them by the first observability and second observability. Furthermore, we also can not use the third observability and fourth observability to determine the initial states of some biological systems if they do not satisfy the requirements of the third and fourth observability. With these disadvantages of four existing observability, we propose the online observability of BCNs to solve this problem.

Problem: Finding the necessary and sufficient condition of determine the initial state \( s_0 \) of BCNs in real time for every \( s_0 \in \Delta_N \).

III. THE ONLINE OBSERVABILITY OF BCNs

In this section we propose the online observability to solve the problem mentioned before, and we will introduce its related information in detail.

In the existing third and fourth observability, there has to exist an input sequence that determines its initial state \( s_0 \) if a BCN satisfies the observability. But we can derive the set of possible initial states \( S_0 \) by initial output \( o_1 \) we observe at time step 0, and then we find input sequence to determine initial state \( (s_0 \in S_0) \) for different sets of possible initial states. These input sequences can be different, and we can also determine the initial state of the BCN. In this case, the requirements for the BCN to determine the initail state would be less harsh since we utilize the set of possible initial states \( S_0 \) to find the input sequence. Furthermore, if we utilize the sets of possible initial states \( S_0 \) and \( S_1 \) derived in time step 0 and 1 to find the input sequence to determine the initial state, the requirements for BCNs would further less harsh. According to this law, in the online observability we find input sequence by the \( S_t \) we derive at every time step, such that the requirements would be least harsh. Thus, a BCN satisfies the online observability if its initial state \( s_0 \) can be determined in real time for every \( s_0 \in \Delta_N \).

The reason why we called this type of observability online observability is as follows.

- Firstly, in the online observability, we determine the initial state of BCNs in real time. In other words, we only need one test case to determine the initial state of BCNs. We call this property real-time property.

- Secondly, in the online observability, we use the outputs we observe to derive and decide the input sequence at every time step in the process of determining the initial state of BCNs. By this way, we can make full use of the outputs and inputs to determine the initial state of BCNs.

We call this property interactivity.

With the real-time and interactivity, we called this type of observability online observability.

After introducing the properties of the online observability, we briefly present how to determine the initial state of BCNs in real time. At every time setp \( t \), we observe the output \( o_t \) of BCNs, then we can infer the set of possible states \( S_t \) by the output \( o_t \) at first. Secondly, with the set of possible states \( S_t \) we can derive the set of possible inputs \( I_t \), such that for every \( i_t \in I_t \) we have that \( s_t, s_j \) will not turn into the same state.

\[ s_i \in \Delta_N \]

Fig. 3. The implication relationships graph between existing observability 1, 2, 3, 4, where “\( \rightarrow \)” means “implies”. 
state after being affected by the input \( i_k \) i.e., \( L s_i i_k \neq L s_j i_k \) for any distinct \( s_i, s_j \in S_t \). This principle will be shown in the equation \( (11) \). And then, we choose an input from \( I_t \).

Thirdly, the set of possible initial states \( S_{t+1} \) of next time step is derived by the input \( i_k \) we chose and the new output \( o_{t+1} \) we observe. The cardinal number of possible states set \( |S_{t+1}| \) does not change or decrease in this process shown in the equations \( (9) \) and \( (10) \). If the cardinal number of new possible states set \( |S_{t+1}| \) turn into be 1 then we can determine the state and the initial state of \( BCN \).

Therefore, in the definition of online observability, firstly we need to describe how to derive the set of possible states of the \( BCN \) by the output and input in every time step. So we define the derivation function to solve this problem. Secondly, we need to describe whether we can use a set of possible states determine the initial state in real time for every \( s_0 \in \Delta N \) by the above mentioned process, then we define the \( K \)-step determinability.

And if the current state of \( BCN \) is shown in \( (11) \), we can only deduce the possible set is the empty set \( \emptyset \). It means that if we don’t know anything about the state of a \( BCN \), then we can not deduce anything no matter what we do.

\[
\text{Df} (\emptyset, i, o) = \emptyset
\]  (7)

In equation \( (8) \), we have that for any possible states set \( S \), we neither input anything nor observe the output. In this case we can only deduce that the possible states set is \( S \). It means that we can not know more information about a \( BCN \) than what I knew before observing the output and deciding the input of this \( BCN \).

\[
\text{Df} (S, \varepsilon, \varepsilon) = S
\]  (8)

In equation \( (9) \), when the possible states set \( S = \Delta N \), and we observe that the outputs of \( BCN \) is \( \delta^1 \) before we decide input. In this case we can derive that the possible states would be \( \delta_1^1, \delta_2^1 \) or \( \delta_3^1 \). This equation shows how to derive the set of possible states by observing the output at every time step. So we have that the cardinal number of the possible states set may decrease after we observing the output of \( BCN \). As the cardinal number of the possible states decreases, we can further determine the range of values for the initial state. And it would help us derive the set possible inputs. Therefore, the derivation function would make the online observability satisfy the interactivity.

\[
\text{Df} (\{\delta^1_{16}, \delta^2_{16}, \delta^3_{16}\}, \delta^1_{16}, \varepsilon) = \{\delta_{10}^1, \delta_{16}^4, \delta_{16}^3\}
\]  (10)

And then, if the possible states set \( S = \{\delta^4_{16}, \delta^5_{16}, \delta^6_{16}\} \) and we input \( \delta^1_{16} \). Before we observe the output of the \( BCN \) we can only derive that the possible states would be \( \delta_{16}^4, \delta_{16}^5, \delta_{16}^6 \) shown in equation \( (10) \). In other words, the cardinal number of the possible states set does not decrease before observing the output of this \( BCN \). Therefore the input \( \delta^1_{16} \) is a suitable input, this equation shows how to derive the set of possible inputs.

\[
\text{Df} (\{\delta^4_{16}, \delta^5_{16}, \delta^6_{16}\}, \delta^1_{4}, \varepsilon) = \{\delta^9_{16}, \delta^4_{16}, \delta^3_{16}\}
\]  (11)

Finally if the set of possible states is \( \{\delta^9_{16}, \delta^4_{16}, \delta^6_{16}\} \) and the inputs is \( \delta^1_{4} \). Before we observe the output of \( BCN \) we can derive that the possible states should be \( \delta^9_{16} \) or \( \delta^3_{16} \) shown in equation \( (11) \). Because both \( \delta^9_{16} \) and \( \delta^3_{16} \) are turn into be the same state \( \delta^3_{16} \) after affected by \( \delta^1_{4} \). And if the current state of the \( BCN \) is one of them then we can not determine the state any more. Therefore, the derivation function helps us define the online observability to satisfy the real-time property.

We give a lemma to to better illustrate the real-time property.

**Lemma 1:** At time step \( k \), \( S \) is the set of current possible states we derived and \( i(k) \) is the input we chose. If we can...
determine the current state \( s(k) \) of this \( BCN \) in real time for every \( s(k) \in S \), then we have \( |\Delta f (S, i(k), \varepsilon) | = |S| \).

**Proof 1:** At time step \( k \), \( S \) is the set of current possible states we derived and \( i(k) \) is the input we chose. And we can determine the current state \( s(k) \) of this \( BCN \) in real time for every \( s(k) \in S \).

Firstly, we assume that \( |\Delta f (S, i(k), \varepsilon) | < |S| \). Then we have that there are two possible states \( s_i(k), s_j(k) \in S \), \( s_i(k) \neq s_j(k) \) such that

\[
s_i(k + 1) = L \times i(k) \times s_i(k) = L \times i(k) \times s_j(t) = s_j(k + 1).
\]

And then for any \( p \in \mathbb{N}^* \) we have

\[
s_i(k + p) = s_j(k + p);
\]

\[
o_i(k + p) = H \times s_i(k + p) = H \times s_j(k + p) = o_j(k + p).
\]

Therefore we cannot distinguish between \( s_i(k) \) and \( s_j(k) \) any more. If the current state \( s(k) \) is one of the two possible states \( s_i(k) \) and \( s_j(k) \), then we can not determine the \( s(k) \) in real time.

But we have that we can determine the current state \( s(k) \) of this \( BCN \) in real time for every \( s(k) \in S \), then the presumption \( |\Delta f (S, i(k), \varepsilon) | < |S| \) is wrong, thus \( |\Delta f (S, i(k), \varepsilon) | = |S| \).

At time step \( k \) and \( S \) is the set of possible states we derived. As the derivation function can only describe the derivation process the set of possible states \( S \) and the set of possible inputs of the \( BCN \) at a time step. But we need to know how to choose the input at every time step to determine the initial state of the \( BCN \) in real time. Thus we propose the \( k \)-step determinability for the set of possible states \( S \) to depict whether we can determine the initial state in real time.

**B. \( k \)-step determinability**

After we defined the derivation function, we present the definition of \( k \)-step determinability for the set of states \( S \), where the \( k \in \mathbb{N} \).

**Definition 9 (\( k \)-Step Determinability):**

When \( k = 0 \), a set of states \( S \) the \( S \) is 0-step deterministic iff the cardinal number of this states \( |S| = 1 \).

When \( k > 0 \), a set of states \( S \) is \( k \)-step deterministic iff the cardinal number of this states set \( |S| > 1 \) and for this set of states \( S \) there exists \( i_p \in \Delta M \) such that

- \( |\Delta f (S, i_p, \varepsilon) | = |S| \), and
- for each \( o_j \) in \( \Delta Q \) such that \( |\Delta f (S, i_p, o_j) | \neq 0 \), there exists a \( k' < k \) such that \( |\Delta f (S, i_p, o_j) | \) is \( k' \)-step deterministic.

In the time step \( t \) and \( S \) is the set of current possible states we derived. If \( k = 0 \), from the definition of \( k \)-step determinability we know that the cardinality number of possible states set \( |S| = 1 \), then we can determine the current state of the \( BCN \). Therefore we can determine the state without deriving and deciding any input and outputting output in “\( k = 0 \)” step. If \( k > 0 \), we have \( |S| > 1 \). Guided by the **Lemma 4** we use the formula \( |\Delta f (S, i_p, \varepsilon) | = |S| \) to ensure that no different states will turn into be the same state after being affected by the input \( i_p \). Without this formula, we can not guarantee that the current state \( s(t) \) can be determined in real time for every \( s(t) \in S \). Furthermore, for each \( o_j \) in \( \Delta Q \) such that \( |\Delta f (S, i_p, o_j) | \neq 0 \) there exists a \( k' < k \) that \( |\Delta f (S, i_p, o_j) | \) is \( k' \)-step deterministic, it make sure that we can can determine the initial state in real time. The \( k \)-step deterministic \( (k > 0) \) is defined recursively, and it requires the definition of 0-step deterministic.

In order to better illustrate this definition, we give the following example.

**Example 7:** In the \( BCN \) mentioned in **Example 7** For the set of state \( \{\delta_1^1, \delta_2^1, \delta_3^1\} \), the cardinality number of this set \( |\{\delta_1^1, \delta_2^1, \delta_3^1\}| = 3 > 1 \), and for this set there exists \( \delta_1^1 \) such that

- \( |\Delta f (\{\delta_1^1, \delta_2^1, \delta_3^1\}, \delta_4^1, \varepsilon) | = |\{\delta_1^1, \delta_2^1, \delta_3^1\}| \);
- \( |\Delta f (\{\delta_1^1, \delta_2^1, \delta_3^1\}, \delta_4^1, \delta_1^1) | = |\{\delta_1^1\}| = 1 \), and \( \{\delta_1^1\} \) is 0-step deterministic;
- \( |\Delta f (\{\delta_1^1, \delta_2^1, \delta_3^1\}, \delta_4^1, \delta_2^1) | = |\{\delta_1^1\}| = 1 \), and \( \{\delta_1^1\} \) is 0-step deterministic;
- \( |\Delta f (\{\delta_1^1, \delta_2^1, \delta_3^1\}, \delta_4^1, \delta_3^1) | = |\{\delta_1^1\}| = 1 \), and \( \{\delta_1^1\} \) is 0-step deterministic.

Therefore the set of states \( \{\delta_1^1, \delta_2^1, \delta_3^1\} \) is 1-step deterministic.

In addition, we propose some lemma for the \( k \)-step determinability.

**Lemma 2:** In the time step \( t \) and \( S \) is the set of current possible states we derived. If \( S \) is \( k \)-step deterministic, then we can determine the current state \( s(t) \) of this \( BCN \) for every \( s(t) \in S \) in real time at time step \( (t + k) \).

**Proof 2:** If \( k = 0 \), then we have \( |S| = 1 \). Therefore, we can determine the current state \( s(t) \) of this \( BCN \) in real time at time step \( t \), and then the **Lemma 2** is right.

If for \( k = 0, \ldots, k = p \) the **Lemma 2** is right. When \( k = p + 1 \) and the \( S \) is \( k \)-step deterministic, then we have for this set of states \( S \) there exists \( i_p \in \Delta M \) such that

- \( |\Delta f (S, i_p, \varepsilon) | = |S| \), and
- for each \( o_j \) in \( \Delta Q \) such that \( |\Delta f (S, i_p, o_j) | \neq 0 \) there exists a \( k' < (p + 1) \) that \( |\Delta f (S, i_p, o_j) | \) is \( k' \)-step deterministic.

However, for \( k = 0, \ldots, k = p \) the **Lemma 2** is right, thus, we can determine the state \( s(t + 1) \) of this \( BCN \) for every \( s(t + 1) \in \Delta f (S, i_p, \varepsilon) \) in real time at time step \( (t + 1 + k') \). What is more, we have \( |\Delta f (S, i_p, \varepsilon) | = |S| \), then for every \( s(t + 1) \in \Delta f (S, i_p, \varepsilon) \) we can find the corresponding one \( s(t) \) for it. So we can determine the state \( s(t) \) of this \( BCN \) for every \( s(t) \in S \) in time step \( (t + p + 1) \). So we have if for \( k = 0, \ldots, k = p \) the **Lemma 2** is right then the **Lemma 2** is right too when \( k = p + 1 \).

The **Lemma 2** is right when \( k = 0 \), and if for \( k = 0, \ldots, k = p \) the **Lemma 2** is right then the **Lemma 2** is right too when \( k = p + 1 \). Thus we have the **Lemma 2** is right for any \( k \in \mathbb{N} \).

The **Lemma 2** indicates that for the set of current possible states \( S \) we derived, \( S \) is \( k \)-step deterministic is the sufficient condition of determine the current state \( s(t) \) of this \( BCN \) for every \( s(t) \in S \) in real time at time step \( (t + k) \). Therefore, the \( k \)-step determinability helps us define the online observability to satisfy the real-time property.

**Lemma 3:** In the time step \( t \) and \( S \) is the set of current possible states we derived. If we can determine the current
Therefore the set of states $S(t)\in\mathcal{S}$ in real time in time step $t+k$, then $S$ is $k$-step deterministic.

**Proof 3:** If $k = 0$ and we can determine the current state $s(t)$ of this $BCN$ for every $s(t)\in S$ in time step $t$. So we have $|S| = 1$, and then $S$ is 0-step deterministic. Therefore, we have the Lemma 3 is right when $k = 0$.

If for $k = 0,\ldots,k = p$ the Lemma 3 is right. When $k = p + 1$ and we can determine the current state $s(t)$ of this $BCN$ for every $s(t)\in S$ in real time in time step $t+k$, then from the Lemma 7 we have for this set of states $S$ there exists $i_p \in \Delta_M$ such that $|\mathcal{D}(S,i_p,\varepsilon)| = |S|$. And $s(t+1)$ can be determined in time step $(t+1+p)$ for every $s(t+1)\in \mathcal{D}(S,i_p,\varepsilon)$. Then we have that for each $o_j$ in $\Delta_Q$ such that $|\mathcal{D}(S,i_p,o_j)| \neq 0$ there exists a $k' < (p+1)$ that $\mathcal{D}(S,i_p,o_j)$ is $k'$-step deterministic. Therefore, we have $S$ is $(p+1)$-step deterministic. And then have if for $k = 0,\ldots,k = p$ the Lemma 3 is right then the Lemma 3 is right too when $k = p+1$.

The Lemma 3 is right when $k = 0$, and if for $k = 0,\ldots,k = p$ the Lemma 3 is right then the Lemma 3 is right too when $k = p+1$. Thus we have the Lemma 3 is right for any $k \in \mathbb{N}$.

The Lemma 3 indicates that for the set of current possible states $S$ we derived, $S$ is $k$-step deterministic is the necessary condition of determine the current state $s(t)$ of this $BCN$ for every $s(t)\in S$ in real time at time step $(t+k)$. Therefore, the $k$-step determinability helps us define the online observability to satisfy the interactivity.

**Lemma 4:** If a set of states $S$ is $k$-step deterministic and $k_1 < k_2$, then $S$ is $k_2$-step deterministic. But if the set states $S$ is $k_1$-step deterministic and $k_1 > k_2$, we can not deduce that $S$ is $k_2$-step deterministic.

**Proof 4:** A set of states $S$ is $k_1$-step deterministic and the $k_1 < k_2$.

If $k_1 = 0$ then $|S| = 1$. Therefore for this set of states $S$ there exists $i_p \in \Delta_M$ such that
- $|\mathcal{D}(S,i_p,\varepsilon)| = |S| = 1$, and
- for each $o_j$ in $\Delta_Q$ such that $|\mathcal{D}(S,i_p,o_j)| \neq 0$ and $\mathcal{D}(S,i_p,o_j)$ is $k'$-step deterministic and $k' = 0 < k_1 < k_2$.

Therefore the set of states $S$ is $k_2$-step deterministic.

If $k_1 > 0$, then $|S| > 1$. Therefore for this set of states $S$ there exists $i_p \in \Delta_M$ such that
- $|\mathcal{D}(S,i_p,\varepsilon)| = |S|$, and
- for each $o_j$ in $\Delta_Q$ such that $|\mathcal{D}(S,i_p,o_j)| \neq 0$ and $\mathcal{D}(S,i_p,o_j)$ is $k'$-step deterministic and $k' < k_1 < k_2$.

Therefore the set of states $S$ is $k_2$-step deterministic.

But if the states set $S$ is $k_1$-step deterministic and $k_1 > k_2$. What is more, for this set of states $S$ there is no such $i_p$ in $\Delta_M$ such that
- $|\mathcal{D}(S,i_p,\varepsilon)| = |S|$, and
- for each $o_j$ in $\Delta_Q$ such that $|\mathcal{D}(S,i_p,o_j)| \neq 0$ and $\mathcal{D}(S,i_p,o_j)$ is $k'$-step deterministic with $k' < k_2 < k_1$.

Therefore the set of states $S$ is not $k_2$-step deterministic.

So we have the Lemma 4 is right.

From the Lemma 4, we know that the exists the infimum of $k$ for the $k$-step determinability of a set of states $S$, which indicate the shortest path to definitely determine the state of a $BCN$ by this possible states set $S$ in real time.

**Definition 10** ($k_{inf}(S)$): If for a set of states $S$, there exists a $k_i \in \mathbb{N}$ such that $S$ is $k_i$-step deterministic but $S$ is not $(k_i-1)$-step deterministic, then $k_i$ is the $k_{inf}(S)$ of $S$.

In Section V we will represent the details on how to find the shortest path when we use the online observability to determine the initial state of a $BCN$. And some other some other advantages of the online observability.

**Lemma 5:** If a set of states $S_j$ is $k$-step deterministic and $S_i \subset S_j$, then $S_i$ is $k$-step deterministic, where $|S_j| > 1$.

**Proof 5:** A set of states $S_j$ is $k$-step deterministic and $S_i \subset S_j$, where $|S_j| > 1$. Therefore we have $k > 0$ from this precondition.

If $k = 1$, we have for this set of states $S_j$ there exists $i_p$ in $\Delta_M$ such that
- $|\mathcal{D}(S_j,i_p,\varepsilon)| = |S_j|$, and
- for each $o_j$ in $\Delta_Q$ such that $|\mathcal{D}(S_j,i_p,o_j)| = 1$ and $\mathcal{D}(S_j,i_p,o_j)$ is 0-step deterministic.

Then we have for the set of states $S_i$ there exists $i_p$ in $\Delta_M$ such that
- $|\mathcal{D}(S_i,i_p,\varepsilon)| = |S_i|$, and
- for each $o_j$ in $\Delta_Q$ such that $|\mathcal{D}(S_i,i_p,o_j)| = 1$, the $\mathcal{D}(S_i,i_p,o_j)$ is 0-step deterministic.

And then $S_i$ is 1-step deterministic, thus we have the Lemma 5 is right when $k = 1$. If for $k = 1,\ldots,k = p$ the Lemma 5 is right. When $k = p+1$, we have for this set of states $S_j$ there exists $i_p$ in $\Delta_M$ such that
- $|\mathcal{D}(S_j,i_p,\varepsilon)| = |S_j|$, and
- for each $o_j$ in $\Delta_Q$ such that $|\mathcal{D}(S_j,i_p,o_j)| \neq 0$ and $\mathcal{D}(S_j,i_p,o_j)$ is $k'$-step deterministic with $k' < (p+1)$.

Then we have for the set of states $S_i$ there exists $i_p$ in $\Delta_M$ such that
- $|\mathcal{D}(S_i,i_p,\varepsilon)| = |S_i|$, and
- for each $o_j$ in $\Delta_Q$ such that $0 \neq |\mathcal{D}(S_i,i_p,o_j)| \leq |\mathcal{D}(S_j,i_p,o_j)|$ and $\mathcal{D}(S_i,i_p,o_j)$ is $k'$-step deterministic with $k' < (p+1)$.

Then we have the $S_i$ is $(p+1)$-step deterministic. Therefore we have if for $k = 1,\ldots,k = p$ the Lemma 5 is right, then the Lemma 5 is right too when $k = p+1$.

The Lemma 5 is right when $k = 1$, and if for $k = 1,\ldots,k = p$ the Lemma 5 is right then the Lemma 5 is right too when $k = p+1$. Thus we have the Lemma 5 is right for any $k \in \mathbb{N}$.

The Lemma 4 would help us design the algorithm to determine the online observability of a $BCN$, and we will represent the details in the Section VII.

C. Online observability

After the previous preparation, we present the formal definition of the online observability. The formal definition of the online observability of $BCNs$ is as follows.

**Definition 11** (Online Observability of $BCNs$): A $BCN$ is online observable, iff for every $o_j \in \Delta_Q$ such that...
$|Df (\Delta_N, \varepsilon, o_j)| > 0$, the $Df (\Delta_N, \varepsilon, o_j)$ is $K$-step deterministic.

However, since we don not need to find the the specific value of $k$ when we define the online observability, we define the concept of $K$-Step Determinability for convenience.

**Definition 12 (K-Step Determinability):** In the time step $t$ and $S$ is the set of current possible states we derived. If there exists a $k_i \in \mathbb{N}$ such that $S$ is $k_i$-step deterministic, then the set of states $S$ is $K$-step deterministic.

In order to better illustrate the definition of online observability, we give the following example.

**Example 8:** In the BCN mentioned in Example 7. We have that:

- $|Df (\Delta_N, \varepsilon, o_1)| = \{\delta_1^{16}, \delta_2^{16}, \delta_3^{16}\} \neq 0$, and this set of states is 1-step deterministic (also $K$-step deterministic);
- $|Df (\Delta_N, \varepsilon, o_2)| = \{\delta_7^{16}, \delta_3^{16}, \delta_4^{16}, \delta_5^{16}\} \neq 0$, and this set of states is 1-step deterministic (also $K$-step deterministic);
- $|Df (\Delta_N, \varepsilon, o_3)| = \{\delta_8^{16}, \delta_9^{16}, \delta_5^{16}, \delta_1^{16}\} \neq 0$, and this set of states is 2-step deterministic (also $K$-step deterministic);
- $|Df (\Delta_N, \varepsilon, o_4)| = \{\delta_8^{16}, \delta_9^{16}, \delta_3^{16}, \delta_2^{16}\} \neq 0$, and this set of states is 2-step deterministic (also $K$-step deterministic).

Therefore we have this BCN satisfies the online observability.

From the informal definition and formal definition of online observability, we propose a theorem for the online observability to solve the problem we proposed in the Section 7.

**Theorem 1:** The necessary and sufficient condition of determine the initial state $s_0$ of a BCN for every $s_0 \in \Delta_N$ in real time is the online observability of this BCN.

**Proof 6:** A BCN is online observable, then for every $o_j$ in $\Delta_Q$ such that $|Df (\Delta_N, \varepsilon, o_j)| > 0$, the $Df (\Delta_N, \varepsilon, o_j)$ is $K$-step deterministic. From the Lemma 2, we have if the $Df (\Delta_N, \varepsilon, o_j)$ is $K$-step deterministic, then we have there exists a $k_i$ such that we can determine the $s_o$ in real time at time step $k_i$ for every $s_0 \in Df (\Delta_N, \varepsilon, o_j)$. What is more for every $o_j$ in $\Delta_Q$ such that $|Df (\Delta_N, \varepsilon, o_j)| > 0$, the $Df (\Delta_N, \varepsilon, o_j)$ is $K$-step deterministic. Then we have there exists a $k_i$ such that we can determine the $s_0$ in time step $k_i$ for every $s_0 \in \Delta_N$. Therefore we can determine the initial state $s_0$ of BCNs in real time for every $s_0 \in \Delta_N$.

If we can determine the initial state $s_0$ of BCNs for every $s_0 \in \Delta_N$ in real time, and $Df (\Delta_N, \varepsilon, o_j)$ is the set of possible states we derived at time step 0. From the Lemma 3, we have for any set of initial possible states $Df (\Delta_N, \varepsilon, o_j)$ we derived, there exists a $k_i$ such that we can determine the $s_0$ in real time at time step $k_i$ for every $s_0 \in Df (\Delta_N, \varepsilon, o_j)$. Thus we have $Df (\Delta_N, \varepsilon, o_j)$ is $k_i$-step deterministic and then $K$-step deterministic for every $o_j$ in $\Delta_Q$ such that $|Df (\Delta_N, \varepsilon, o_j)| > 0$. Therefore this BCN is online observable.

So we have the necessary and sufficient condition of determine the initial state $s_0$ of a BCN for every $s_0 \in \Delta_N$ in real time is the online observability of this BCN.

What is more, we propose some lemma for the $K$-step determinability as well.

**Lemma 6:** If a set of states $S_i$ is $K$-step deterministic and $S_i \subseteq S_j$, then $S_i$ is $K$-step deterministic, where $|S_j| > 1$.

**Proof 7:** A set of states $S_j$ is $K$-step deterministic and $S_i \subseteq S_j$. Then there exists a $k_i \in \mathbb{N}$ such that $S_j$ is $k_i$-step deterministic. By the Lemma 5, we have $S_i$ is $k_i$-step deterministic too, and then the $S_i$ is $K$-step deterministic as well.

With the Lemma 6, we would have its equivalent proposition is right as well.

**Lemma 7:** If a set of states $S_i$ is not $K$-step deterministic and $S_i \subseteq S_j$, then $S_j$ is not $K$-step deterministic.

**Proof 8:** A set of states $S_i$ is not $K$-step deterministic and $S_i \subseteq S_j$. We assume that the set of states $S_j$ is $K$-step deterministic first, then we have $S_i$ is $K$-step deterministic too. But $S_i$ is not $K$-step deterministic, therefore we have the assumption is wrong, and then the $S_j$ is not $K$-step deterministic.

The Lemma 6 and Lemma 7 help us to determine whether a BCN is online observable in the Section 7.

**D. Online observability and existing observability**

After defining online observability of BCNs, we discuss the implication relationships between four existing observability and online observability. The implication relationship is about the conditions that need to meet these observability. For instance, in the Theorem 2, “The online observability implies the existing first observability,” means that “If a BCN satisfies the online observability then it satisfies the existing first observability.” However, “The existing first observability does not imply the online observability,” means that “A BCN that satisfies the existing first observability does not have to satisfy the online observability.”

**Theorem 2:** The online observability implies the existing first observability, but the existing first observability does not imply the online observability.

**Proof 9:** A BCN satisfies the online observability. From the Theorem 7, we have that we can determine the initial state $s_0$ of this BCN in real time for every $s_0 \in \Delta_N$. Therefore for every initial state $s_0 \in \Delta_N$, there exists an input sequence $I \in (\Delta_k)^p$ for some $p \in \mathbb{Z}_+$ such that the $s_0$ can be determined by the input sequence $I$, and then this BCN satisfies the existing first observability.

However, if a BCN satisfies the existing first observability. What is more there exist two state $s, s' \in Df (\Delta_N, \varepsilon, o_j)$ where $|Df (\Delta_N, \varepsilon, o_j)| > 0$, and the $u_0, u_1, \ldots, u_{p-1}$ is the corresponding one unique input sequences to determine $s$, and the $\bar{u}_0, \bar{u}_1, \ldots, \bar{u}_{p-1}$ is the corresponding one unique input sequences to determine $s'$. If the first input of the input sequences $I$ and $I'$ are different, then we can not determine the initial state $s_0$ of this BCN in real time for every $s_0 \in \{s, s'\}$. Therefore this BCN does not satisfy the online observability.

In the existing first observability, we presuppose the initial state of a BCN at first, and then we use its corresponding input sequence to distinguish it from other types of initial state. In other words, we do not consider the real-time property in the existing first observability which would make the condition more harsh. So we have the online observability implies the
existing first observability, but the existing first observability does not imply the online observability. Moreover, from the implication relationships between the existing first and second observability which mentioned in the Section [7] we have the implication relationships between the online observability and the existing second observability

**Theorem 3:** The online observability implies the existing second observability, but the existing second observability does not imply the online observability.

**Theorem 4:** The online observability does not imply the existing third observability, but the existing third observability implies the online observability.

**Proof 10:** A BCN is online observable. What is more there are \( o, o' \in \Delta_Q \), such that there does not exist \( i_p \in \Delta_M \) satisfy that \( |\text{def} (\Delta_N, i_p, o, o')| = |\text{def} (\Delta_N, i_p, o')| \) and \( |\text{def} (\Delta_N, i_p, o')| = |\text{def} (\Delta_N, i_p, o)| \). Therefore there does not exist an input sequence \( I \in (\Delta_M)^p \) for some \( p \in \mathbb{Z}_+ \), such that for any distinct states \( s_0, s'_0 \in \Delta_N \), \( Hs_0 = Hs'_0 \) implies \( (HL)_{s_0}^p (I) \neq (HL)_{s'_0}^p (I) \). So we have this BCN does not satisfy the existing third observability.

If a BCN satisfies the existing third observability, then there exists an input sequence \( I \in (\Delta_M)^p \) for some \( p \in \mathbb{Z}_+ \), such that for any distinct states \( s_0, s'_0 \in \Delta_N \), \( Hs_0 = Hs'_0 \) implies \( (HL)_{s_0}^p (I) \neq (HL)_{s'_0}^p (I) \). And then we can use the input sequence \( I \) to determine the initial state \( s_0 \) of this BCN in real time for every \( s_0 \in \Delta_N \). So we have this BCN satisfies the online observability.

In the existing third type of observability, there has to exist an input sequence that can distinguish any distinct states of the BCN. In other words, we do not consider the interactivity in the existing third observability which would make the condition less harsh. So we have the online observability does not imply the existing third observability, but the existing third observability implies the online observability. Moreover, from the implication relationships between the existing third and fourth observability which mentioned in the Section [7] we have the implication relationships between the online observability and the fourth observability.

**Theorem 5:** The online observability does not imply the existing fourth observability, but the existing fourth observability implies the online observability.

Therefore, the implication relationships graph between four existing observability and online observability is shown in Fig.4.

From the implication relationships between online observability and existing observability we know that the online observability can help us solve some problem which can not be solved by existing observability.

- Firstly, if the systems described by BCNs are online observable but not satisfy the existing third and fourth observability, then the online observability can help us determine their initial state in real time.
- Secondly, it takes least observation costs for us to determine the initial state of some systems described in real time by BCNs. There are some biological systems depicted by BCNs, such as the immune systems which can be depicted as the BCN T-cell receptor kinetics model [11]. And there exist input-nodes and state-nodes in this model, for the purpose of obtain the initial state of this BCN, we must select some state-nodes to be observe at first. However, if we use the online observability of BCNs to determine the initial state of the BCN T-cell receptor kinetics model in real time., then we need least observation costs.

What is more, with the online observability, we can make some optimizations in the process of determining the initial state. We will represent them in the Section [V].

**IV. DETERMINING THE ONLINE OBSERVABILITY OF BCNs**

After defining the online observability and comparing it with the existing four observability, we propose two algorithms to determine the online observability of BCNs. The first one is the supertree-based algorithm, and the second one is the algorithm based on directed graph. Based on the definition of online observability, we propose the supertree to describe the process of determining the initial state of a BCN. And then, we propose the algorithm to determine the online observability of BCNs based on the supertree. But the supertree-based algorithm can not help us find all paths to determine the initial state of a BCN. In order to improve the shortcomings of the supertree-based algorithm, we propose the algorithm based on directed graph. The algorithm based on directed graph may take longer time for us to determine its online observability. But if we want do some optimization in the process of determining the initial state of a BCN, this algorithm would be better. What is more, we also analyze the complexity of the algorithm based on directed graph in this section. Finally, we represent how to determine the initial state of a BCN by the directed graph.

**A. Supertree-based algorithm**

According to the definition of online observability, we alternately observe the output and then derive and decide the input in the process of determining the initial state of a BCN. When the cardinal number of the set of possible states comes into be 1, we can determine the current state of this BCN and then its initial state. We define the supertree for BCNs to describe this process, and then propose the supertree-based algorithm to determine the online observability for BCNs.
convenience, we use the set of state $S_i$ inside a node to represent this node, and the input $i_p$ or output $o_j$ in an edge to represent the edge.

Definition 13 (Supertree): For a BCN, every node $S_i$ in the supertree is $K$-step deterministic. The root node of its supertree is $\Delta_N$, while the leaf nodes of the supertree are the nodes with cardinal number $1 (|S_i| = 1)$. In addition to the leaf nodes, if a node $S_i$ in the $2k + 1$ layer of the supertree and

$$|\text{df} (S_i, \varepsilon, o_j)| > 0,$$

then $\text{df} (S_i, \varepsilon, o_j)$ is one of its son nodes, and $o_j$ is the edge from $S_i$ to $\text{df} (S_i, \varepsilon, o_j)$ for each $o_j \in \Delta_Q$. If a node $S_i$ in the $2k + 2$ layer of the supertree and

$$|\text{df} (S_i, i_p, \varepsilon)| = |S_i|,$$

then $\text{df} (S_i, i_p, \varepsilon)$ is the son node of $S_i$ and $i_p$ is the edge from $S_i$ to $\text{df} (S_i, i_p, \varepsilon)$ for each $i_p \in \Delta_M$.

![Fig. 5. Branch of the super tree which represents $\delta_{BCN}$](image)

In the Definition 13, for a BCN, we can only infer that the possible states set is $\Delta_N$ at the beginning, thus the root node of the super tree is $\Delta_N$. And then, we can determine the state of the BCN when the cardinal number of the possible states set turns into 1. Therefore, the leaf nodes of the supertree are the nodes with cardinal number 1. In the process of determining the initial state of a BCN, we observe the output of the BCN to derive the set of possible states at first. After that, we derive and decide the input and then derive the new possible states set of the BCN. We alternately observe the output and then derive and decide the input until we can determine the state of BCN. Therefore we use $\text{df} (S_i, \varepsilon, o_j)$ to find child nodes for every $S_i$ in $2k + 1$ layer, and using $\text{df} (S_i, i_p, \varepsilon)$ to find child nodes for every $S_i$ in $2k + 2$ layer. The formula

$$|\text{df} (S_i, \varepsilon, o_j)| > 0$$

ensures the node $\text{df} (S_i, \varepsilon, o_j)$ is not empty. The formula

$$|\text{df} (S_i, i_p, \varepsilon)| = |S_i|$$

guarantee we can determine the state of $\Delta_N$ in the end (Section III Equation 7). Therefore, the paths to determine the initial state of a BCN are described in the supertree, and then we can use it to determine the online observability for a BCN.

Based on the definition of the supertree, we propose the supertree-based algorithm (Algorithm 1) to determine the online observability for BCNs.

**Algorithm 1 Supertree-based algorithm**

**Input:** The algebraic form of BCN

**Output:** The super tree of BCN

```
1: Ob = false
2: NodesArray = $\Delta_N$
3: while Ob == false do
4:   Build child nodes for NodesArray by $\text{df} (S_i, \varepsilon, o_j)$
5:   NodesArray = Child nodes of $\text{NodeArray}$ except leaf nodes
6:   Check this BCN by the super tree
7:   if $BCN$ is online observable then
8:      Ob = true
9:   else
10:      Build child nodes for NodesArray by $\text{df} (S_i, i_p, \varepsilon)$
11:     NodesArray = Child nodes of $\text{NodeArray}$
12: end if
13: end while
14: Delete uncertain branches
15: NodesArray = $\Delta_N$
16: return NodesArray
```

In the supertree-based algorithm we build trees by breadth first. We check the online observability of the BCN by the supertree after $2k + 2$ layer of the supertree was built for every $k \in \mathbb{N}$. If the BCN is online observable, then we stop building the supertree and delete the uncertain branches. Finally, we return the NodesArray which is the root node of the supertree, and then we can determine the initial state of the BCN by the supertree.

In order to better illustrate how to use the super tree to determine the online observability, we give the following example.

Example 9: In the BCN mentioned in Example 7 From the definition of online observability we need to determine whether the $\text{df} (\Delta_N, \varepsilon, o_j)$ is $K$-step deterministic for every $o_j \in \Delta_Q$ such that $|\text{df} (\Delta_N, \varepsilon, o_j)| > 0$. Therefore, we build child nodes for $\Delta_N$ by the $\text{df} (S_i, \varepsilon, o_j)$. For instance,

$$\text{df} (\Delta_N, \varepsilon, \delta_{4}) = \{\delta^1_{16}, \delta^2_{16}, \delta^3_{16}\}$$

and we cannot determine whether $\{\delta^1_{16}, \delta^2_{16}, \delta^3_{16}\}$ is $K$-step deterministic or not, and then we cannot determine the online observability of this BCN by the supertree now. Therefore we build child nodes for it, and then build child nodes for its child nodes as shown in Fig 5. Then we check the second and third layer of this branch, we have the nodes $\{\delta^1_{16}\}$, $\{\delta^2_{16}\}$ and $\{\delta^3_{16}\}$ are $K$-step deterministic, and then we have the node $\{\delta^1_{16}, \delta^2_{16}, \delta^3_{16}\}$ is $K$-step deterministic. We use the same method to check other nodes, and then determine the online observability for the BCN. Finally, we delete uncertain branches except the branches which can help us to determine online observability, such as the first branch ($\{\delta^1_{16}, \delta^2_{16}, \delta^3_{16}\}$).
And then, the supertree can help us to determine the initial state of the BCN.

However, if we want use the supertree-based algorithm to find all paths to determine the initial state of a BCN. In this case, we need to check the nodes that appear multiple times in the supertree, and this nodes would take many additional time and space overhead. For instance, in the Fig.5 there are two nodes take \{δ_{16}^{1}, δ_{16}^{2}\} in the fourth layer. Moreover, the same nodes in a path will form a loop, the loops in the supertree will prevent us from building a complete tree. For example, there are the \{δ_{16}^{1}, δ_{16}^{3}\} in fourth layer and the \{δ_{16}^{2}, δ_{16}^{3}\} in fifth layer, and they would form a loop. With the shortcomings of the supertree, we propose the algorithm based on directed graph to help us find all paths to determine the initial state of a BCN.

B. Algorithm based on directed graph

In order to improve the shortcomings of the supertree-based algorithm, we proposed the algorithm based on directed graph. The biggest difference of these two algorithms is the way how the supertree and directed graph constructed. That supertree is built from the root node (\(\Delta N_j\)) to leaf nodes (contain 1 state), while the directed graph is built from smallest nodes (contain 1 state) to largest node (contain largest number of states). In addition, there is not any repeated node in the directed graph because every node appears only once in the directed graph. What is more, even there are some loops in the directed graph, the loops would not prevent us from building the directed graph completely.

Therefore, we have the definition of directed graph for BCNs.

Definition 14 (Directed Graph): Firstly, every node \(S_i\) in the directed graph is \(K\)-step deterministic, and there are no duplicate nodes in the graph.

Secondly, for every node \(S_i\) and \(|S_i| > 1\), we have that for every distinct two \(s_a, s_b \subseteq S_i, H_{s_a} = H_{s_b}\).

Finally, for the edges of the directed graph.

- If \(|S_i| = 1\), then there are not edge from it to other nodes.
- If \(|S_i| > 1\), and there are exist one edge \(i_p\) from it to one nodes, then there exist \(z \geq 1\) such that there are \(z\) edges contain \(i_p\) from it to nodes \(S_1, \ldots, S_z\) that

\[|S_i| = |S_1| + \ldots, |S_z|\]

and

\[\Delta f (S_i, i_p, \varepsilon) = S_1 \lor \ldots, \lor S_z.\]

From the Lemma\[3\] in the Section\[3\] we have that if the set of states \(S_i\) is not \(K\)-step deterministic and \(S_i \subset S_j\), then \(S_j\) is not \(K\)-step deterministic. Therefore, we check whether the nodes with fewer states are \(K\)-step deterministic at first, and then we check whether the nodes with more states are \(K\)-step deterministic in the process of building the directed graph for a BCN. Once we can find a node \(S_i\) is not \(K\)-step deterministic, then we make sure that there exists \(o_j \in \Delta Q\) such that \(\Delta f (\Delta N_j, \varepsilon, o_j) > 0\) and \(S_i \subset \Delta f (\Delta N_j, \varepsilon, o_j)\), then \(\Delta f (\Delta N_j, \varepsilon, o_j)\) is not \(K\)-step deterministic, and then this BCN is not online observable.

With the definition of directed graph and the way to construct the directed graph. We propose the algorithm based on directed graph presented in the Algorithm\[2\]. And the Algorithm\[3\] present the algorithm to build nodes which is used in the Algorithm\[2\].

Algorithm 2 Algorithm based on directed graph

Input: The algebraic forms of BCN

Output: The directed graph of BCN

1. \(k = 1\)
2. \(Ob = true\)
3. buildnode\((k)\)
4. \(k = k + 1\)
5. \(NodesArray = \)buildnode\((k)\)
6. while \(NodesArray[] = \)null do
7. for each \(S_i \in NodesArray\) do
8. if \(k == 2\) then
9. \(S_i = \Delta M\)
10. else
11. Find \(S_i\) by other nodes
12. end if
13. for each \(i_p \in S_i\) do
14. Check \(S_i\) by \(i_p\)
15. Build edges for \(S_i\)
16. end for
17. if \(S_i\) has not any edge. then
18. \(Ob = false\)
19. return Null
20. end if
21. end for
22. \(k = k + 1\)
23. \(NodesArray = \)buildnode\((k-1)\)
24. end while
25. \(NodesArray = \)buildnode\((k-1)\)
26. return \(NodesArray\)

Algorithm 3 buildnode\((\text{int} k)\)

Input: The number of states \(k\)

Output: The nodes with \(k\) states which with the same corresponding outputs

1. Build all nodes with \(p\) states
2. if Failed to build then
3. return Null
4. else
5. Classify these nodes
6. Sort the states in these nodes
7. Sort these nodes
8. return nodes
9. end if

There are some details in Algorithm\[2\] and Algorithm\[3\] are as follows:

- Build all nodes with \(k\) states: Firstly, we classify all states by their corresponding outputs \((\Delta f (\Delta N, \varepsilon, o_j))\), then we have all of the states sets. The states set contains all states that with the same corresponding outputs. Secondly, we compare \(k\) with the cardinal number \(|\Delta f (\Delta N, \varepsilon, o_j)|\).

- In order to improve the shortcomings of the supertree-based algorithm, we proposed the algorithm based on directed graph. The biggest difference of these two algorithms is the way how the supertree and directed graph constructed. That supertree is built from the root node \((\Delta N_j)\) to leaf nodes (contain 1 state), while the directed graph is built from smallest nodes (contain 1 state) to largest node (contain largest number of states). In addition, there is not any repeated node in the directed graph because every node appears only once in the directed graph. What is more, even there are some loops in the directed graph, the loops would not prevent us from building the directed graph completely.

- Therefore, we have the definition of directed graph for BCNs.

Definition 14 (Directed Graph): Firstly, every node \(S_i\) in the directed graph is \(K\)-step deterministic, and there are no duplicate nodes in the graph.

Secondly, for every node \(S_i\) and \(|S_i| > 1\), we have that for every distinct two \(s_a, s_b \subseteq S_i, H_{s_a} = H_{s_b}\).

Finally, for the edges of the directed graph.

- If \(|S_i| = 1\), then there are not edge from it to other nodes.
- If \(|S_i| > 1\), and there are exist one edge \(i_p\) from it to one nodes, then there exist \(z \geq 1\) such that there are \(z\) edges contain \(i_p\) from it to nodes \(S_1, \ldots, S_z\) that

\[|S_i| = |S_1| + \ldots, |S_z|\]

and

\[\Delta f (S_i, i_p, \varepsilon) = S_1 \lor \ldots, \lor S_z.\]

From the Lemma\[3\] in the Section\[3\] we have that if the set of states \(S_i\) is not \(K\)-step deterministic and \(S_i \subset S_j\), then \(S_j\) is not \(K\)-step deterministic. Therefore, we check whether the nodes with fewer states are \(K\)-step deterministic at first, and then we check whether the nodes with more states are \(K\)-step deterministic in the process of building the directed graph for a BCN. Once we can find a node \(S_i\) is not \(K\)-step deterministic, then we make sure that there exists \(o_j \in \Delta Q\) such that \(|\Delta f (\Delta N_j, \varepsilon, o_j)| > 0\) and \(S_i \subset \Delta f (\Delta N_j, \varepsilon, o_j)\), then \(\Delta f (\Delta N_j, \varepsilon, o_j)\) is not \(K\)-step deterministic, and then this BCN is not online observable.
of each states set we built before. If $k$ greater than $\lceil \text{df}(\Delta_N, \varepsilon, o_j) \rceil$, then we could not get $k$ states from this states set. Else we can get $C_k^{\text{df}(\Delta_N, \varepsilon, o_j)}$, sets with $k$ states from this states set. Finally, we use all of the sets of states found in second step to build nodes we need.

- Sort the states in these nodes and sort these nodes: For example, the nodes $\{\delta^{16}_{16}, \delta^{16}_{16}\}$, $\{\delta^{16}_{16}, \delta^{16}_{16}\}$ and $\{\delta^{16}_{16}, \delta^{16}_{16}\}$ shown in Fig 6. We sort the states inside the nodes at first, and then sort the nodes by the states of them.

- Find $Sis$ by other nodes: From the Lemma 3 and Lemma 4 in the Section III we have if $S_i \subset S_j$ then for any input $i$ which can not make $S_i$ K-step deterministic, it can not make $S_j$ K-step deterministic either. Therefore, for the node $S_i$ with $k$ sorted states inside it, we can use the node with the first $k - 1$ states of $S_i$ and the node with the last $k - 1$ states of $S_i$ to find the suitable inputs set $Sis$ for $S_i$. For example, we can search correct inputs sets which make $\{\delta^{16}_{16}, \delta^{16}_{16}, \delta^{16}_{16}\}$ and $\{\delta^{16}_{16}, \delta^{16}_{16}, \delta^{16}_{16}\}$ K-step deterministic at first. After that, take the intersection of these sets to be the suitable inputs set of $\{\delta^{16}_{16}, \delta^{16}_{16}, \delta^{16}_{16}, \delta^{16}_{16}\}$.

- Check $S_i$ by $i_p$: According to the order determined in previous steps, we check every node in order. If for one input $i_p \in Sis$ implies $|\text{df}(S_i, i_p, \varepsilon)| < |S_i|$, we can make sure the $i_p$ is a wrong input. Else if for each $O_j \in \Delta_Q$, $|\text{df}(S_i, i_p, o_j)| > 0$ and $\text{df}(S_i, i_p, o_j)$ is K-step deterministic then $i_p$ is a correct input. Therefore, we can connect the node $S_i$ to each node $\text{df}(S_i, i_p, o_j)$ with directed edge. Else if there exist $o_j \in \Delta_Q$ and we can not make sure whether $\text{df}(S_i, i_p, o_j)$ is K-step deterministic, then we check it in the next round.

![Fig. 6. Part of the directed graph which represents $\{\delta^{16}_{16}, \delta^{16}_{16}\}$ and $\{\delta^{16}_{16}, \delta^{16}_{16}\}$. The green, black, orange, blue edges show the inputs $\delta^{16}_{16}, \delta^{16}_{16}, \delta^{16}_{16}$ respectively.](image-url)

What is more, based on the definitions of existing four types of observability, we can also use the directed graph to determine the existing second and fourth type of observability for BCNs.

- Checking the existing second observability: When we try to build bottom layer and penultimate layer of the directed graph, and there are exist some nodes in penultimate layer has no edges from it to other nodes. Therefore, there are distinct states $s_0, s'_0 \in \Delta_N$, and there does not exist any input sequence $I \in (\Delta_N)^p$ for any $p \in \mathbb{Z}_+$, such that $H(s_0) = H s_0'$ implies $(HL)^p_{s_0}(I) \neq (HL)^p_{s'_0}(I)$. And then, this BCN does not satisfy existing second observability.

- Checking the existing fourth observability: When we try to build edges for every layer, and if there exist one node whose right inputs set is not $\Delta_M$, then there exists an input sequence $I \in (\Delta_M)^\infty$ does not satisfy that for any distinct states $s_0, s'_0 \in \Delta_N$, $H(s_0) = H s_0'$ implies $(HL)^\infty_{s_0}(I) \neq (HL)^\infty_{s'_0}(I)$. And then this BCN does not satisfy existing fourth observability.

### C. Complexity analysis

As the algorithm by the directed graph is better than by supertree when we want to find all paths to determine the initial state of BCNs. We analyze the complexity of this algorithm briefly in this paper.

- Firstly, we need to calculate the number of layers in the directed graph i.e. the upper bound of the number $(k)$ of the states of the nodes in the directed graph. We have that

$$k_{upb} = \max(\text{df}(\Delta_N, \varepsilon, \delta^1_M), \ldots, \text{df}(\Delta_N, \varepsilon, \delta^M_M))$$

(12)

Because the states of the same nodes in the directed graph should have the same corresponding output. Therefore, the $k_{upb}$ indicates the number of layers in the directed graph, and it depends on the relationship between states and outputs of the BCNs.

- Secondly, we need to calculate the number of nodes which with $k$ states, we have that

$$\text{Non}(k) = C^k_{|S_i|} + \ldots + C^k_{|S_p|},$$

(13)

where

$$S_1, \ldots, S_p \in \{\text{df}(\Delta_N, \varepsilon, \delta^1_M), \ldots, \text{df}(\Delta_N, \varepsilon, \delta^M_M)\}$$

and $|S_1|, \ldots, |S_p| \geq k$. The $\text{Non}(k)$ indicates the number of nodes which built by the $\text{buildnode}(k)$ function, and it also depends on the relationship between states and outputs of the BCNs.

- Thirdly, we need to calculate the cardinal number of states inputs set of each node $|\text{sis}(S_i)|$. If $|S_i| = 2$ then $\text{sis}(S_i) = \Delta_M$. If $|S_i| > 2$ then $\text{sis}(S_i)$ is derived by other nodes, therefore it depends on the updating rules of the BCNs.

- Finally, we need to calculate the time used to check whether a input which in the suitable inputs set of a node is a right input for this node $T(S_i, i_p)$, and it depends on the updating rules of the BCNs as well.

After completing the previous analysis, we calculate the complexity by layer by layer, then we have the time we need to determine the online observability.

$$T = \sum_{k=1}^{k_{upb}} \sum_{i=1}^{\text{Non}(k)} \sum_{p=1}^{\text{sis}(S_i)} T(S_i, i_p)$$
From the definition, we know that the $k_{uph}$ and the $Non(k)$ are depend on the relationship between states and outputs of the BCNs, and the $|Sts(S_i)|$ and $T(S_i, i_p)$ are depend on the updating rules of the BCNs. Therefore, it is hard to give an accurate complexity of the algorithm by the number of the nodes of the BCNs without the complete information of their updating rules. We just give a brief introduction of complexity analysis in this paper, and we would do more research about this problem in the future.

D. Determining initial state

After introducing the algorithms to determine the online observability of the BCNs, we present the way to determine the initial state of a BCN by the directed graph. If a system described by BCN is online observable, and the directed graph of it has been built, then we can determine the initial state of this system (or BCN) in real time. In order to illustrate the process of determining the initial state of a BCN, we give one example as follows.

Example 10: In the BCN mentioned in Example 7. The process of determining its initial state is shown in the Fig.

- Firstly, we observe the output of the BCN. If the output we observe is $\delta^1$ then we can derive that the set of possible initial states should be $\{\delta^1_{16}, \delta^2_{16}, \delta^3_{16}\}$, and we record them as the initial states and current states of the BCN in the table.
- Secondly, we derive and decide the input ($\delta^1$) and observe the output ($\delta^3$), then we can derive that the set possible current states $\{\delta^2_{16}, \delta^3_{16}\}$, and then we record them as current states set in their corresponding positions.
- Repeat the second step until the cardinal number of the possible states set turns into 1. In that time we can determine the current state ($\delta^0_{16}$) and the corresponding initial state ($\delta^1_{10}$) of the BCN.

Although the way of determine the initial state of a BCN by the directed graph is very brief, but it would help us present how to do some optimization in the process of determining the initial state of the BCNs.

V. Optimization

In the Section III we present that the online observability is the the necessary and sufficient condition of determine the initial state $s_0$ of the BCNs in real time. Therefore, it can help us determine the initial state of some BCNs in real time which can not be determined by the existing third and fourth observability. In addition, we can also use the online observability to some optimization, including finding the shortest path and avoid entering critical states in the process of determining the initial state of the BCNs.

A. Finding shortest path

When we need to determine the initial state of a BCN, an important aspect that we will consider is to find the shortest path to determine the initial state. In general, we can not find the shortest path definitely. Fortunately, we can use the directed graph to make the best decision. For the path to determine the initial state of the BCNs, we introduce two functions $Pe(S, i_p)$ and $Pv(S, i_p)$ to describe its expected value and variance, respectively. The definition of them are as follows.

Some necessary statements before defining the functions $Pe(S, i_p)$ and $Pv(S, i_p)$:

- $S$: the set of states.
- $\{i_1, i_2, \ldots, i_z\}$: the right inputs set of $S$;
- $\{S_p^1, S_p^2, \ldots, S_p^k\}$: the set of state sets, and its elements correspond to the possible outputs $\{o_1, o_2, \ldots, o_k\}$.

As we choose the input $i_p$ to determine initial state the BCN by $S$, for each $i_p$ in $\{i_1, i_2, \ldots, i_z\}$.

Definition 15 ($Spe(S)$):

$$Spe(S) = \min\{Pe(S, i_1), Pe(S, i_2), \ldots, Pe(S, i_z)\}.$$  

Definition 16 ($Pe(S, i_p)$): When the $|S| = 1$, we have that $Pe(S, i_p) = 0$ for every $i_p$ in $\{i_1, i_2, \ldots, i_z\}$. According to Definition 15 $Spe(S) = 0$ if $|S| = 1$. When the $|S| > 1$, we have that

$$Pe(S, i_p) = 1 + \sum_{j=1}^{k} \frac{Spe(S^j_p)|S^j_p|}{|S|}.$$  

In the definition of $Spe(S)$, the function shortest path expected value $Spe(S)$ is to find the $i_p$ from $\{i_1, i_2, \ldots, i_z\}$ to calculate least $Pe(S, i_p)$ for $S$. From the definition of $Pe(S, i_p)$, we have that if $|S| = 1$ then we can make sure the state of BCNs. Thus we need not choose the input anymore to determine the state of BCNs. Therefore, for any input the path expected value $Pe(S, i_p)$ would be 0 and the shortest path expected value $Spe(S)$ also would be 0. But if $|S| > 1$ we still need to choose input and observe the output. Only by this way we can determine the state of of BCNs. And we recursively define the $Pe(S, i_p)$ and $Spe(S)$ for each input $i_p$ in the right inputs set.

If we want to find the shortest path to determine the initial state of a BCN, we can choose an input $i_p$ with least $Pe(S, i_p)$ by the function $Spe(S)$. This input $i_p$ may help us find the shortest path to determine the initial state. But the output of BCNs we observe is uncertain after we choose the input $i_p$, hence selecting the $i_p$ which with least $Pe(S, i_p)$ may leads...
to a very long path to determine the initial state of BCNs. For better performance, we define the $P_{V}(S, i_{p})$ to avoid this risk.

**Definition 18 ($P_{V}(S, i_{p})$):** When the $|S| = 1$, we have that $P_{V}(S, i_{p}) = 0$ for every $i_{p}$ in $\{i_{1}, i_{2}, \ldots, i_{z}\}$. But when the $|S| > 1$, we have that

$$P_{V}(S, i_{p}) = \frac{\sum_{j=1}^{k}(S_{p}^{j} - P_{e}(S, i_{p}) + 1)^{2}|S_{p}^{j}|}{|S|}$$

From the definition of the $P_{V}(S, i_{p})$, we have that if the $P_{V}(S, i_{p})$ of input $i_{p}$ is not very large, the risk of choosing $i_{p}$ would be not great either.

**B. Avoiding entering critical states**

In biological systems which are depicted by the BCNs, some of the genes’ states may correspond to unfavorable or even dangerous situations [22]. So another important aspect that we consider is to avoid entering critical states in the process of determining the BCN’s initial state. Therefore, we also construct two functions $C_{e}(S, i_{p})$ and $C_{V}(S, i_{p})$ to describe expected value and variance of the times of entering critical states in the process determine the initial state of the BCNs. The definition of $C_{e}(S, i_{p})$ is as follows.

**Definition 19 ($L_{c}(S)$):**

$$L_{c}(S) = \min(C_{e}(S, i_{1}), C_{e}(S, i_{2}), \ldots, C_{e}(S, i_{z}))$$

According to Definition 18

$$C_{e}(S, i_{p}) = |S \cap S_{cr}|$$

But when the $|S| > 1$ we have that

$$C_{e}(S, i_{p}) = |S \cap S_{cr}| + \sum_{j=1}^{z}L_{c}(S_{p}^{j})|S|$$

Where $S_{cr}$ is the critical states set of the BCN we research. The definition of $C_{e}(S, i_{p})$ has some difference with $P_{e}(S, i_{p})$, because of the critical states set $S_{cr}$. So that we can analyze the possibility of entering the critical states after we derived the possible states set of BCNs, and we can get the definitions of $C_{V}(S, i_{p})$ in the similar way.

**Definition 20 ($C_{V}(S, i_{p})$):** When the $|S| = 1$, we have that $C_{V}(S, i_{p}) = 0$ for every $i_{p}$ in $\{i_{1}, i_{2}, \ldots, i_{z}\}$. But when the $|S| > 1$, we have that

$$C_{V}(S, i_{p}) = \frac{\sum_{j=1}^{z}[L_{c}(S_{p}^{j}) - C_{e}(S, i_{p}) + |S \cap S_{cr}|]^{2}|S_{p}^{j}|}{|S|}$$

The use of $C_{e}(S, i_{p})$ and $C_{V}(S, i_{p})$ are similar to $P_{e}(S, i_{p})$ and $P_{V}(S, i_{p})$ respectively. They help us avoid entering critical states of BCNs in the process of determining the initial state of BCNs. With these four functions $P_{e}(S, i_{p})$, $P_{V}(S, i_{p})$, $C_{e}(S, i_{p})$, and $C_{V}(S, i_{p})$, we can make the best decision we like.

In the four existing types of observability, they have not property interactivity. This leads to we can not analyze the state of the BCNs dynamically, hence it would be hard to do some optimization in the process of determining the initial state of the BCNs. However, this problem can be solved by the online observability of the BCNs better.

**VI. Conclusions**

In this paper, firstly we proposed the online observability of BCNs and define its mathematical form. Secondly we propose two algorithms based on the super tree and directed graph to determine the online observability. After introducing determination algorithm we present some optimization brought by the online observability and then talk about some advantages of it.

But even we use the super tree and directed graph, it is still hard to determine the the online observability of a BCN which with a large number of nodes. Therefore, in the future we will try to separate the BCN into the subnets. We determine their online observability respectively, and then we determine the online observability of original BCN. Furthermore, we also want to try to use some knowledge about formal methods to earn scalability for BCNs. In addition to the theoretical aspect, the realistic application is also very important. Hence we will also try to find some realistic example which can be modeled by BCNs. So that we can research these realistic examples well and determine the online observability their models for better performance.

**References**

[1] F. Jacob and J. Monod, “Genetic regulatory mechanisms in the synthesis of proteins,” *Journal of Molecular Biology*, vol. 3, no. 3, pp. 318–356, 1961.

[2] S. A. Kauffman, “Metabolic stability and epigenesis in randomly constructed genetic nets,” *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1968.

[3] T. Akutsu, S. Miyano, and S. Kuhara, “Inferring qualitative relations in genetic networks and metabolic pathways,” *Bioinformatics*, vol. 16, no. 8, pp. 727–734, 2000.

[4] I. Shmulevich, E. R. Dougherty, and W. Zhang, “From boolean to probabilistic boolean networks as models of genetic regulatory networks,” *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[5] A. Faur, 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.

[6] D. G. Green, T. G. Leishman, and S. Sadedin, “The emergence of social consensus in boolean networks,” in *Artificial Life ’07. IEEE Symposium on*, 2007, pp. 402–408.

[7] Y. Lou and Y. Hong, “Multi-agent decision in boolean networks with private information and switching interconnection,” in *Chinese Control Conference*, 2010, pp. 4530–4535.

[8] E. Fornasini and M. E. Valcher, “Observability, reconstructibility and state observers of boolean control networks,” *IEEE Transactions on Automatic Control*, vol. 58, no. 6, pp. 1390–1401, 2013.

[9] T. Ideker, T. Galitski, and L. Hood, “A new approach to decoding life: systems biology,” *Annu Rev Genomics Hum Genet*, vol. 2, no. 1, pp. 343–372, 2001.

[10] K. Kaufman, M., Andris, F., and Leo, O., “A logical analysis of t cell activation and anergy,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 7, pp. 3894–3899, 1999.

[11] 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,” *Bioinformatics*, vol. 7, no. 1, p. 56, 2006.

[12] D. Biane and F. Delaplace, “Abduction based drug target discovery using boolean control network,” *Bioinformatics*, vol. 27, no. 7, pp. 873–880, 2011.

[13] J. Thanburg, P. Ogren, and X. Hu, “A boolean control network approach to pursue evasion problems in polygonal environments,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 4506–4511.
[14] T. Akutsu, M. Hayashida, W. K. Ching, and M. K. Ng, “Control of boolean networks: hardness results and algorithms for tree structured networks,” *Journal of Theoretical Biology*, vol. 244, no. 4, p. 670, 2007.

[15] D. Cheng and H. Qi, “Controllability and observability of boolean control networks,” *Automatica*, vol. 45, no. 7, pp. 1659–1667, 2009.

[16] Y. Zhao, H. Qi, and D. Cheng, “Input-state incidence matrix of boolean control networks and its applications,” *Systems & Control Letters*, vol. 59, no. 12, pp. 767–774, 2010.

[17] D. Cheng, H. Qi, and Z. Li, “Identification of boolean control networks,” *Automatica*, vol. 47, no. 4, pp. 702–710, 2011.

[18] ——, *Analysis and Control of Boolean Networks*. Springer London, 2011.

[19] R. Li, M. Yang, and T. Chu, “Controllability and observability of boolean networks arising from biology,” *Chaos An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 2, pp. 1778–1792, 2015.

[20] K. Zhang and L. Zhang, “Observability of boolean control networks: A unified approach based on finite automata,” *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2733–2738, 2016.

[21] D. Cheng, “Semi-tensor product of matrices and its applications - a survey,” *Methods & Applications of Analysis*, vol. 321, no. 2, pp. 61–79, 2003.

[22] Z. Q. Li and J. L. Song, “Controllability of boolean control networks avoiding states set,” *Science China Information Sciences*, vol. 57, no. 3, pp. 1–13, 2014.