A-002

# A Fixed-Parameter Algorithm for Detecting a Singleton Attractor in an AND/OR Boolean Network with Bounded Treewidth

Tatsuya Akutsu† Takeyuki Tamura†

## 1 Introduction

The *Boolean network* (BN) is known as a discrete mathematical model of gene regulatory networks [3]. In a BN, each vertex corresponds to a gene and takes one of two values 0 and 1, where 0 (resp., 1) means that the corresponding gene is inactive (resp., active). The value of a vertex at a given time step is determined according to a regulation rule, which is a Boolean function of the values of the predecessors of the vertex at the previous time instant. The values of vertices are updated synchronously, and the (global) *state* of a network at a given time step is the vector of its vertex values. Beginning from any initial state, the system eventually falls into an *attractor*, which is classified into two types: a *singleton attractor* corresponding to a stable state, and a *periodic attractor* corresponding to a sequence of states that repeats periodically.

It is known that the problem of finding an attractor of the shortest period is NP-hard even for BNs with maximum in-degree 2 consisting of AND/OR of literals [5]. Due to this hardness and the fact that there exist $2^n$ global states for a BN with $n$ vertices, previous theoretical studies focused on the development of $o(2^n)$ time algorithms.

Since it is quite hard to develop such algorithms for general BNs, some restrictions were assumed on types of Boolean functions in all studies. For example, an $O(1.587^n)$ time algorithm and an $O(1.985^n)$ time algorithm were developed for detection of a singleton attractor [4] and an attractor of period 2 [1], respectively, both for *AND/OR BNs* which are BNs consisting of Boolean functions restricted to conjunctions and disjunctions of literals.

An $O(n^{2p(w+1)}poly(n))$ time algorithm was also developed for finding an attractor of period $p$ of a BN having bounded treewidth $w$ and consisting of nested canalyzing functions, where $p$ and $w$ are constants, and nested canalyzing functions are a super class of AND/OR functions [1]. They also presented a fixed-parameter algorithm (precisely, an algorithm working in $O(g(p,w,d)poly(n))$ time where $g(p,w,d)$ depends only on $p,w,d$) for a general BN with bounded degree and bounded treewidth [1]. However, it is unknown whether there exists a fixed-parameter algorithm even for an AND/OR BN with bounded treewidth but without degree constraint. In this short report, we present a fixed-parameter algorithm for detection of a singleton attractor in an AND/OR BN with bounded treewidth.

†Institute for Chemical Research, Kyoto University

## 2 Preliminaries

A BN $N(V,F)$ consists of a set $V$ of $n$ vertices and a corresponding set $F=\{f_v:\ v\in V\}$ of $n$ Boolean functions. Let $v(t)\in\{0,1\}$ represent the value of a vertex $v$ at time $t$, and denote by $\mathbf{v}(t)=\langle v(t):v\in V\rangle$ the *state of the network* at time $t$. The values of all vertices are updated simultaneously according to the corresponding Boolean functions, $v(t+1)=f_v(\mathbf{v}(t))$. A directed graph $G(V,E)$ can be associated with the network, with a directed edge $(u,v)\in E$ if and only if $f_v$ depends on $u$, where edges may be self-loops. The initial assignment of values $\mathbf{v}(1)=\langle v(1):v\in V\rangle$ uniquely determines the state of the network at all $t>1$. An initial state is called a *periodic attractor with period* $p$ if $\mathbf{v}(1)=\mathbf{v}(p+1)$ and $\mathbf{v}(1)\neq\mathbf{v}(q)$ holds for all $1<q<p+1$. An attractor with period $p=1$ is called a *singleton attractor*.

A *tree decomposition* of a graph $G(V,E)$ is a pair $\langle T(V_T,E_T),(B_t)_{t\in V_T}\rangle$, where $T(V_T,E_T)$ is a rooted tree and $(B_t)_{t\in V_T}$ is a family of subsets of $V$ such that

- for every $v_i\in V$, $B^{-1}(v_i)=\{t\in V_T|v_i\in B_t\}$ is nonempty and connected in $T$, and
- for every edge $\{v_i,v_j\}\in E$, there exists $t\in V_T$ such that $v_i,v_j\in B_t$.

The *width* of the decomposition is defined as $\max_{t\in V_T}(|B_t|-1)$ and the *treewidth of* $G$ is the minimum of the widths among all the tree decompositions of $G$. Graphs with treewidth at most $k$ are also known as *partial* $k$*-trees* [2].

## 3 Algorithm

Let $\langle T(V_T,E_T),(B_t)_{t\in V_T}\rangle$ be a tree decomposition of $G(V,E)$ associated to a given BN $N(V,F)$. Let $des(t)$ denote the set of descendant of $t\in V_T$ with including $t$. Let $V_t=\bigcup_{t'\in des(t)}B_{t'}$. For each $t\in V_T$, $p(t)$ denotes the parent node of $t$ in $V_T$.‡

For each $t\in V_T$, $\phi_t$ denotes a function from $B_t$ to $\{0,1,\omega\}$, where we call such $\phi_t$ an *assignment*. We define $\hat{\phi}_t$ by

$$\hat{\phi}_t(v)=\begin{cases}\phi_t(v), & \text{if } \phi_t(v)\in\{0,1\},\\ 0, & \text{if } \phi_t(v)=\omega \text{ and } f_v \text{ is AND},\\ 1, & \text{if } \phi_t(v)=\omega \text{ and } f_v \text{ is OR},\end{cases}$$

It is to be noted that if $\hat{\phi}_t(v)=1$ and $v$ is an AND vertex (resp., $\hat{\phi}_t(v)=0$ and $v$ is an OR vertex), Boolean

‡We use nodes and vertices for $\langle T(V_T,E_T),(B_t)_{t\in V_T}\rangle$ and $N(V,F)$, respectively.

values of its input vertices are uniquely determined (in a singleton attractor). Therefore, $\phi_t(v) = \omega$ holds only if $v$ is an AND vertex and $\hat{\phi}_t(v) = 0$, or $v$ is an OR vertex and $\hat{\phi}_t(v) = 1$.

For a 0-1 assignment $\alpha$ to $V' \subseteq V$, $\phi_t(v) \in \{0, 1\}$ is called *validated* (by $\alpha$ on $V'$) if the Boolean value $v$ is uniquely determined as $\alpha(v)$ from $\alpha$ regardless of an assignment to $V - V'$. We say that $\alpha$ *violates* a Boolean function $f_v$ (assigned to vertex $v$) if the value ($b_v$) of $f_v$ is uniquely determined by $\alpha$ but $b_v \neq \alpha(v)$. It is to be noted that we need not validate if $\hat{\phi}_t(v) = 1$ and $v$ is an AND vertex, or, $\hat{\phi}_t(v) = 0$ and $v$ is an OR vertex since it is enough for such a vertex to examine whether $\alpha$ does not violate $f_v$. $\phi_t$ is called *consistent* if the following is satisfied

(1) $\phi_t(v) \in \{0, 1\}$ holds for all $v \in B_t - B_{p(t)}$, where we let $B_{p(t)} = \emptyset$ if $t$ is the root,

(2) there exists a 0-1 assignment $\alpha$ to $V_t$ such that $\hat{\phi}_t(v) = \alpha(v)$ holds for all $v \in B_t$, $\alpha$ does not violate any Boolean function assigned to $V_t$, and $\phi_t(v)$ is validated for all $v$ such that $\phi_t(v) \in \{0, 1\}$ by $\alpha$ on $V_t$.

Let $A_t$ be the set of consistent assignments to $B_t$. We describe below how to compute $A_t$ by dynamic programming from leaves to the root in $V_T$. For each leaf $t$, $A_t$ is determined by $A_t = \{\phi_t | \ \phi_t$ satisfies conditions (1) and (2) for $\alpha = \hat{\phi}_t.\}$.

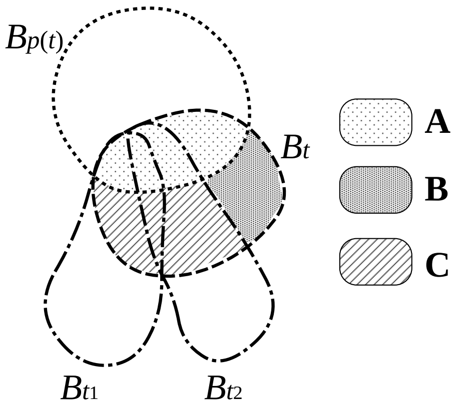


Figure 1: Part A ($B_t \cap B_{p(t)}$) can be validated later, Part B ($B_t - B_{p(t)} - \cup_j B_{t_j}$) must be validated by $\phi_t$, and Part C ($(B_t \cap (\cup_j B_{t_j})) - B_{p(t)}$) must be validated by $\phi_t$, $\phi_{t_1}$, $\phi_{t_2}$.

Let $t_1, \ldots, t_d$ be the children of $t$. For all possible $\phi_t$, we examine whether it is consistent as follows. We call $\phi_t$ and $\phi_{t_i}$ are *compatible* if, for each $v \in B_t \cap B_{t_i}$, $\phi_t(v) = \phi_{t_i}(v)$ holds, or $\phi_t(v) \in \{0, 1\}$, $\phi_{t_i}(v) = \omega$ and $\hat{\phi}_t(v) = \hat{\phi}_{t_i}(v)$ hold. We maintain a 0-1 table $X(x_1, \ldots, x_h)$ where $\{v_{j_1}, \ldots, v_{j_h}\} = B_t$ (i.e., $X$ is a 0-1 table having $2^{|B_t|}$ entries). We say that $\phi_t$ is a *candidate* assignment if $\phi_t(v) \in \{0, 1\}$ holds for any vertex in $B_t - B_{t(p)}$, $\phi_t$ does not violate any function assigned to $B_t$, and $\phi_t(v)$ is validated by $\phi_t$ for any vertex $B_t - B_{t(p)} - \cup_{j=1,\ldots,d} B_{t_j}$. The following is a pseudocode of the proposed algorithm (see also Fig. 1).

Procedure *FpFindAtt*<br>
**for all** leaves in $V_T$ **do** compute $A_t$;<br>
**for all** internal nodes $t$ in $V_T$ **do**<br>
  $A_t \leftarrow \emptyset$;<br>
  **for all** candidate assignments $\phi_t$ **do**<br>
    **for all** $(x_1, \ldots, x_h) \in \{0, 1\}^h$ **do**<br>
      $X(x_1, \ldots, x_h) \leftarrow 0$;<br>
    $X(x_1, \ldots, x_h) \leftarrow 1$ where $x_g = 1$ iff $\phi_t(v_{j_g})$<br>
    has already been validated, or $\phi_t(v_{j_g}) = \omega$;<br>
    **for** $i = 1$ **to** $d$ **do**<br>
      $Y \leftarrow X$;<br>
      **for all** $\phi_{t_i} \in A_{t_i}$ compatible with $\phi_t$ **do**<br>
        let $z_g = 1$ if $\phi_{t_i}(v_{j_g}) \in \{0, 1\}$,<br>
        otherwise $z_g = 0$;<br>
        **for all** $(x_1, \ldots, x_h)$ with $X(x_1, \ldots, x_h) = 1$ **do**<br>
          $Y(\max(x_1, z_1), \ldots, \max(x_h, z_h)) \leftarrow 1$;<br>
      $X \leftarrow Y$;<br>
    **if** $X(1, 1, \ldots, 1) = 1$ **then** $A_t \leftarrow A_t \cup \{\phi_t\}$

There exists a singleton attractor iff $A_r \neq \emptyset$ for the root $r$ of $T$. Furthermore, such a singleton attractor can be retrieved by using the standard traceback technique. Since the number of possible $\phi_t$ is bounded by $3^{k+1}$ per $t$ and the size of table $X$ is bounded by $2^{k+1}$ for partial $k$-trees, we have:

**Theorem 1** *The singleton attractor detection problem for an AND/OR BN with bounded treewidth $k$ can be solved in $O(f(k)poly(n))$ time, where $f(k)$ is a function depending only on $k$.*

# References


[1] T. Akutsu, S. Kosub, A. A. Melkman, and T. Tamura, Finding a periodic attractor of a Boolean Network, *IEEE/ACM Trans. Computational Biology and Bioinformatics* 9:151–160, 2012.

[2] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer, 2006.

[3] S. A. Kauffman, *The Origins of Order: Self-organization and Selection in Evolution*, Oxford Univ. Press, 1993.

[4] A. A. Melkman, T. Tamura, and T. Akutsu, Determining a singleton attractor of an AND/OR Boolean network in $O(1.587^n)$ time, *Information Processing Letters*, 110:565–569, 2010.

[5] T. Tamura and T. Akutsu, Detecting a singleton attractor in a Boolean network utilizing SAT algorithms, *IEICE Trans. Fundamentals*, E92-A:493-501, 2009.