Published in final edited form as:
SIAM J Appl Dyn Syst. 2013 ; 12(4): 1997–2011. doi:10.1137/13090537X.

A REDUCTION METHOD FOR BOOLEAN NETWORK MODELS PROVEN TO CONSERVE ATTRACTORS

ASSIEH SAADATPOUR[*], **RÉKA ALBERT**[†], **TIMOTHY C. RELUGA**[‡]

[†]Department of Physics, The Pennsylvania State University, University Park, PA 16802, USA

[‡]Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA

Abstract

Boolean models, wherein each component is characterized with a binary (ON or OFF) variable, have been widely employed for dynamic modeling of biological regulatory networks. However, the exponential dependencse of the size of the state space of these models on the number of nodes in the network can be a daunting prospect for attractor analysis of large-scale systems. We have previously proposed a network reduction technique for Boolean models and demonstrated its applicability on two biological systems, namely, the abscisic acid signal transduction network as well as the T-LGL leukemia survival signaling network. In this paper, we provide a rigorous mathematical proof that this method not only conserves the fixed points of a Boolean network, but also conserves the complex attractors of general asynchronous Boolean models wherein at each time step a randomly selected node is updated. This method thus allows one to infer the long-term dynamic properties of a large-scale system from those of the corresponding reduced model.

Keywords

Boolean models; Network reduction; Asynchronous methods; Attractors; Biological regulatory networks

AMS subject classifications.

92C42; 37G35

1. Introduction.

The ever-accelerating pace of experimental data generation has laid the foundation for developing network models of biological systems wherein the components of a system are represented by nodes and the interactions among them by edges. Analyzing these network models and studying their dynamics can unravel unknown facets of the underlying biological systems. Among different dynamic modeling approaches, discrete models, in which each component is assumed to have a finite number of qualitative states, have been increasingly employed in modeling biological regulatory networks [10, 18, 19, 20, 23]. The

[*]Corresponding author. Department of Physics, The Pennsylvania State University, University Park, PA 16802, USA (assieh@psu.edu).

simplest discrete dynamic models are the so-called Boolean models that assume only two states (ON or OFF) for each component [8, 21].

Since Boolean models are parameter free, they serve as a suitable starting point for modeling biological systems for which a detailed kinetic characterization of the interactions is not available. In particular, attractor analysis of these models is of immense biological importance as it can provide valuable insights into the long-term behaviors, i.e. observed phenotypes, of these systems in response to environmental stimuli and internal perturbations [1, 5, 6, 9, 15, 16]. For example, it allows one to predict the long-term activity levels of components or to determine key components influencing different cellular traits. However, the exponential dependence of the size of the state space of Boolean models on the number of nodes in the network makes the identification of all attractors of even relatively small systems computationally intractable. In particular, it has been proven that determination of the existence of fixed points in Boolean networks is a strong NP-complete problem [24]. There have been several efforts to reduce the state space of Boolean models by simplifying the underlying networks. In [2, 11, 14] a network reduction method based on the removal of stable variables (i.e, variables that stabilize in an attracting state after a transient period, irrespective of updating strategy or initial conditions) and leaf nodes (i.e., nodes with out-degree = 0) was proposed. In another study, Naldi et al. [12] proposed a reduction method for simplifying finite-state logical models by iteratively removing nodes without a self loop from the network. In this method, the logical rules for the reduced models were constructed using reduced ordered multivalued decision diagrams. This method was proven to preserve the fixed points of a system, but it may introduce spurious oscillations into the reduced model [12]. Subsequently, Veliz-Cuba [22] adapted this method for the reduction of Boolean networks by iteratively removing nodes without a self loop from the network and simplifying the redundant Boolean functions.

Boolean models of biological regulatory networks, such as signal transduction networks, which involve one or more sustained signals (source nodes), often contain stabilized nodes (stable variables). In a previous work [15], we proposed a two-step network reduction method that (*i*) identifies and eliminates the stabilized nodes; and (*ii*) iteratively merges simple mediator nodes, i.e., nodes having in-degree and out-degree of one. We note that the second step automatically excludes the removal of nodes with a self-loop because a node with in- and out-degree of one and with a self-loop would need to be isolated. We previously employed this reduction method to identify attractors of the abscisic acid signal transduction network in plants as well as the T-LGL leukemia survival signaling network in humans [15, 16], and using numerical simulations showed that it is effective in reducing the size of a Boolean network model without affecting its long-term dynamic properties. For both systems, the proposed reduction method enabled us to make predictions about the effect of node perturbations on the long-term behaviors of the systems. For example, we identified several potential therapeutic targets for T-LGL leukemia, some of which were supported by existing experimental evidence and the rest can guide future wet-bench experiments [16]. In this paper, we provide a rigorous mathematical proof that our method not only conserves the fixed points of a system, but also conserves the complex attractors of general asynchronous Boolean models wherein at each time step a randomly selected node is updated. We illustrate this reduction method on two toy networks and highlight the results of our previous effort on the attractor analysis of the abscisic acid signaling network as well.

2. Network reduction method.

A biological regulatory network can be represented by a directed graph $\mathscr{G} = (\mathscr{A}, \mathscr{E})$ where the set of vertices (nodes) $\mathscr{A}$ describes different components of the system, and the set of edges $\mathscr{E}$ denotes the regulatory interactions among the nodes. The orientation of each edge in the network is determined based on the direction of mass transfer or information propagation from the upstream to the downstream component. In addition, each edge has a positive or negative sign signifying activation or inhibition, respectively. The source nodes (i.e., nodes with in-degree = 0) of this graph, if they exist, represent external inputs (signals) to the network.

Boolean models assume each node of the network has only two states – ON (1) and OFF (0). The state of each node v is determined based on a Boolean function (rule) $B_v : \{0, 1\}^{m_v} \to \{0, 1\}$, where m_v is the number of regulators of v. In general B_v is expressed via the Boolean operators AND, OR, and NOT, but other implementations are also possible. In Boolean models, time is an implicit variable and can be implemented using synchronous or asynchronous update algorithms. Synchronous models assume similar timescales for all the processes involved in a system, which is often unrealistic for modeling biological regulatory networks [13]. Asynchronous models, on the other hand, allow updating the nodes' states individually based on their own timescales [21]. Several asynchronous algorithms have been proposed so far, including the random order asynchronous [3, 7], deterministic asynchronous [4], and general asynchronous [7] algorithms. In a previous work [15], we carried out a comparative study of these asynchronous methods applied to the same biological system. That study suggested that the general asynchronous method, wherein at each time step a randomly selected node is updated, is the most efficient and informative asynchronous updating strategy.

By updating the nodes' states according to the synchronous or asynchronous algorithms, one can obtain the state of the whole system at each time step, which is expressed by a vector whose v'th element represents the state of node v at that time step. We note that the Boolean model of a network with n nodes $(n = |\mathscr{A}|)$ has a total of 2^n states and at most n possible transitions for each state. These states and the allowed transitions among them form the state transition graph of the system. Starting from an initial state in the state transition graph and iteratively updating the nodes' states, the state of the system evolves over time and eventually converges to an attractor. Attractors, which describe the long-time behavior of a system, fall into two categories: fixed points (steady states), wherein the state of the system does not change, and complex attractors, wherein the system oscillates among a set of states. As fixed points are time-independent, they are the same in both synchronous and asynchronous methods. In contrast, complex attractors are highly dependent on the method of update. For example, it was observed that oscillations present in the synchronous method may be absent from the corresponding asynchronous methods [5, 15].

In the following we describe a reduction method that facilitates the identification of attractors in large networks, and provide a mathematical proof that it conserves attractors of a given Boolean network under the general asynchronous update method. This method consists of two steps: (*i*) identifying and eliminating the nodes whose states stabilize due to their regulation and irrespective of timing or initial condition; and (*ii*) iteratively merging simple mediator nodes, i.e., nodes having in-degree and out-degree of one. The first step is especially suited for biological regulatory networks with one or more source nodes whose states can be fixed at an ON or OFF value, e.g. based on the existing experimental evidence. The two steps are outlined in Algorithms 1 and 2.

2.1. Reduction algorithms.

Algorithm 1:

Identifying and eliminating the stabilized nodes.

```
Input: Boolean functions $B_1, B_2, \ldots, B_n$ corresponding to each node,
(I) for each node $i$
  if $B_i$ is a constant function
    for each node $j$ where $B_j$ depends on $i$
      - insert the value of $B_i$ into $B_j$
      - simplify $B_j$ using Boolean algebra
    remove node $i$ and function $B_i$
(II) Repeat step (I) until no additional constant Boolean function is obtained.
Output: $\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_p$. Note that $n - p$ variables have been stabilized and thus removed from the network.
```

Algorithm 2:

Merging simple mediator nodes.

```
Input: Boolean functions $\widetilde{B}_1, \widetilde{B}_2, \ldots, \widetilde{B}_p$ from Algorithm 1.
(I) for each node triple $(u, v, w)$
  if $\widetilde{B}_v$ depends only on $u$
    if $\widetilde{B}_w$ depends only on $v$
      if $\widetilde{B}_u$ does not depend on $w$, and $\widetilde{B}_w$ does not depend on $u$
        replace $\widetilde{B}_w(v)$ with $\widetilde{B}_w\left(\widetilde{B}_v(u)\right)$
        remove node $v$ and function $\widetilde{B}_v$
(II) Repeat step (I) until no node with in-/out-degree of one remains.
Output: $B_1^R, B_2^R, \ldots, B_q^R$, the Boolean rules for the reduced network $\mathscr{G}^{\mathscr{R}} = \left(\mathscr{V}^{\mathscr{R}}, \mathscr{E}^{\mathscr{R}}\right)$.
```

These two algorithms are illustrated in Figure 2.1. For the network given in part (a), node u is a source node. Let us assume that it is always ON. Then node v, which is inhibited by u, is stabilized in an OFF state after a time delay. Based on Algorithm 1, u and v can thus be removed from the network. In addition, the Boolean rule for node w, which depends on the stabilized node v, is simplified accordingly. For the network given in part (b), node v has an in-degree and out-degree of one and can be removed according to Algorithm 2 (node u will be connected to w in the reduced network).

2.2. Conservation of attractors.

In the following we prove that there is a one-to-one correspondence between attractors of the original Boolean model and those of the reduced model under Algorithms 1 and 2. As mentioned before, fixed points of a Boolean network model are update-independent. As such, the proof of the conservation of fixed points under the reduction method is independent of the method of update. Complex attractors, on the other hand, depend on the update method. In order to prove the conservation of complex attractors, we consider the general asynchronous approach in which at each time step a randomly selected node is updated.

Let us denote the state transition graphs of the original and reduced models by $\mathscr{M} = (\mathscr{S}, \mathscr{T})$ and $\mathscr{M}^R = \left(\mathscr{S}^R, \mathscr{T}^R\right)$, respectively, wherein $\mathscr{S}$ and $\mathscr{S}^R$ denote the set of states in the respective models, and the directed edges in $\mathscr{T}$ and $\mathscr{T}^R$ represent the allowed transitions among the states. In the following we consider the case of removing a single node v. The general case of removing a sequence of nodes follows by induction.

Definition 2.1. An attractor of a Boolean model, in the context of graph theory, is an absorbing set of states $C \subset \mathscr{S}$ that forms a strongly connected component of $\mathscr{M}$. Here, a strongly connected component is a set of states for which there is a path between every pair of states, and an absorbing set is a subgraph of the state transition graph that no path can leave it (i.e., with no out-component).

This definition automatically implies that an attractor is always a maximal strongly connected component.

Definition 2.2. Let the removed node v be the kth node in the list of nodes of the original Boolean model. The projection map $\pi : \mathscr{S} \to \mathscr{S}^R$ following the removal of v is given by

$$s = (s(1), \ldots, s(k-1), s(k), s(k+1), \ldots, s(n)) \mapsto \pi(s) = (s(1), \ldots, s(k-1), s(k+1), \ldots, s(n)),$$

where s(i) denotes the ith coordinate of s. For any set of states $C \subseteq \mathscr{S}$, we define $\pi(C) = \{\pi(s) : s \in C\}$, and the pre-image of any state set $C^R \subseteq \mathscr{S}^R$ as $\pi^{-1}\left(C^R\right) = \left\{s : s \in \mathscr{S}, \pi(s) \in C^R\right\}$.

Definition 2.3. Attractors of a Boolean network model are conserved under a reduction method if for any attractor C in the original model, $\pi(C)$ is an attractor of the reduced model, and the pre-image of any attractor in the reduced model contains exactly one attractor of the original model.

Theorem 2.4. The reduction method outlined in Algorithms 1 and 2 conserves the fixed points of a Boolean network model.

Proof. Since our reduction method does not remove nodes with a self-loop, the proof of Theorem 2.4 of [22] holds true in our case. □

Definition 2.5. [12] A transition $(s, s') \in \mathcal{T}$ is preserved under the reduction method if and only if $\pi(s) = \pi(s')$ or $(\pi(s), \pi(s')) \in \mathcal{T}^R$.

In the following theorem we show that when removing the stabilized nodes not only complex attractors but also the state transitions are preserved in the sense of Definition 2.5.

Theorem 2.6. The reduction rule outlined in Algorithm 1 conserves complex attractors of a Boolean network model under the general asynchronous update method.

Proof. Let us consider removing the stabilized node v with the fixed state c_v. We first show that when removing v any transition in $\mathcal{T}$ is preserved in the sense of Definition 2.5, and conversely any transition in $\mathcal{T}^R$ is a projection of at least one transition in $\mathcal{T}$. Let $(s, s') \in \mathcal{T}$. If $s = s'$, i.e., when the transition is a self loop, then $\pi(s) = \pi(s')$ and by Definition 2.5 this transition is preserved. Otherwise, as we consider the general asynchronous approach, s and s' differ at only one position. If s and s' differ at the position of node v, i.e., when the transition (s, s') involves a transient state of v before it stabilizes, then again $\pi(s) = \pi(s')$ and by Definition 2.5 the transition is preserved. If v has already stabilized, then it has the same state in both s and s', i.e., $s(k) = s'(k)$ considering that v is the kth node in the list of nodes of the original Boolean model. Thus there is $i \neq k$ such that $s(i) \neq s'(i)$ and $B_i(s) = s'(i)$. Since v is a stabilized node, we also have $B_i(s) = B_i^R(\pi(s))$. In addition, since $i \neq k$, $(\pi(s'))(i) = s'(i)$. Therefore $B_i^R(\pi(s)) = B_i(s) = s'(i) = (\pi(s'))(i)$. Thus $(\pi(s), \pi(s')) \in \mathcal{T}^R$, implying that transition (s, s') is preserved. Conversely, let $(z, z') \in \mathcal{T}^R$ with $z = (z(1),\ldots, z(k-1), z(k+1),\ldots, z(n))$ and $z' = (z'(1),\ldots, z'(k-1), z'(k+1),\ldots, z'(n))$. Thus there is $i \neq k$ such that $z(i) \neq z'(i)$ and $B_i^R(z) = z'(i)$. Note that $z(j) = z'(j)$ for any $j \neq i, k$. Without loss of generality assume that $i > k$. Let $s = (z(1),\ldots, z(k-1), c_v, z(k+1),\ldots, z(i-1), z(i), z(i+1),\ldots, z(n))$ and $s' = (z(1),\ldots, z(k-1), c_v, z(k+1),\ldots, z(i-1), z'(i), z(i+1),\ldots, z(n))$. Then $\pi(s) = z$ and $\pi(s') = z'$, and $B_i(s) = B_i^R(z) = z'(i) = s'(i)$. Thus $(s, s') \in \mathcal{T}$.

Now let C be an attractor of $\mathcal{M}$. We claim that $\pi(C)$ is an attractor of $\mathcal{M}^R$, and in addition C is the only attractor in $\pi^{-1}(\pi(C))$. Since any transition in C is preserved in the reduced model, $\pi(C)$ is strongly connected. We show that $\pi(C)$ is absorbing as well. If not, there must exist a transition $(z, z') \in \mathcal{T}^R$ such that $z \in \pi(C)$ and $z' \notin \pi(C)$. Then the argument above implies that there exists a transition $(s, s') \in \mathcal{T}$ such that $\pi(s) = z$ and $\pi(s') = z'$. $z' \notin \pi(C)$ implies $s' \notin C$. Then $s \in C$ and $s' \notin C$ implying that C is not absorbing, a contradiction. Thus $\pi(C)$ is an attractor of $\mathcal{M}^R$. In order to show the second part of the claim, take any $z \in \pi(C)$ and let $\pi^{-1}(z) = \{s_a, s_b\}$. Since v is a stabilized node, it must have the fixed state c_v in at least one of the states in $\pi^{-1}(z)$. Without loss of generality, let s_a be such a state. Then we have $s_a \in C$. Now s_b can have a transient state of v, and thus any subset of $\pi^{-1}(\pi(C))\backslash C$ containing s_b cannot be absorbing. This implies that C is the only attractor in $\pi^{-1}(\pi(C))$.

Suppose now that C^R is an attractor of $\mathscr{M}^R$. We claim that there exists a unique attractor C of $\mathscr{M}$ such that $C \subseteq \pi^{-1}(C^R)$. Our argument on preservation of transitions implies that $\pi^{-1}(C^R)$ contains a strongly connected component C of $\mathscr{M}$. If C is not absorbing, then there must exist a transition $(s, s') \in \mathscr{T}$ such that $s \in C$ and $s' \notin C$. This implies that $(\pi(s), \pi(s')) \in \mathscr{T}^R$ such that $\pi(s) \in C^R$ and $\pi(s') \notin C^R$, a contradiction with C^R being absorbing. Thus C is an attractor of $\mathscr{M}$. If such a C is not unique, i.e., there are two attractors C_1 and C_2 in $\pi^{-1}(C^R)$, then $\pi(C_1)$ and $\pi(C_2)$ are attractors of $\mathscr{M}^R$. Since C^R is an attractor containing $\pi(C_1)$ and $\pi(C_2)$, we have $\pi(C_1) = \pi(C_2) = C^R$. Since C_1 is the only attractor in $\pi^{-1}(\pi(C_1))$, we conclude that $C_1 = C_2$. □

Theorem 2.7. The reduction rule outlined in Algorithm 2 conserves complex attractors of a Boolean network model under the general asynchronous update method.

In order to prove Theorem 2.7 we make use of a symbolic-dynamics approach. To this end, we first introduce the following notations and definitions. Note that the set of nodes $\mathscr{A}$ can be thought of as an alphabet wherein each node represents a letter. In our model there is exactly one Boolean function that updates each node, so we use the letters in alphabet $\mathscr{A}$ as synonyms for the corresponding Boolean functions as well. Thus each letter can also represent a Boolean function and the words formed by these letters are transformations between states. In other words, each word represents an order of update in the general asynchronous method. Let $\mathfrak{W}(\mathscr{A})$ be the set of words composed from alphabet $\mathscr{A}$, including the empty word.

Consider node v satisfying the conditions of Algorithm 2, that is a node whose in-degree and out-degree is one with an edge coming from node u and an edge going to node w. Assume that there is no edge between u and w. Let $\mathscr{V} = \{v\} \subseteq \mathscr{A}$, $\mathscr{U} = \{v, w\} \subseteq \mathscr{A}$, and $\mathscr{U}^c = \mathscr{A} \backslash \mathscr{U}$. We also define a reduced alphabet $\mathscr{A}^R$ with node set $\mathscr{A} \backslash \mathscr{V}$ wherein the node w has a new Boolean rule $\widetilde{w}$ that copies (or negates) the state of u. Thus in terms of Boolean functions $\mathscr{A}^R = (\mathscr{A} \backslash \{v, w\}) \cup \{\widetilde{w}\}$. We note that $\widetilde{w}$ is equivalent to vw in the original model. Let $\mathscr{W} = \{\widetilde{w}\}$ and $\mathscr{W}^c = \mathscr{A}^R \backslash \mathscr{W}$.

Definition 2.8. Given a subset $N \subseteq \mathscr{A}$ of the nodes in $\mathscr{G}$, two states s_i and s_j are N-equivalent, denoted as $s_i =_N s_j$, if and only if for any node in N, s_i and s_j have the same value at that node.

Clearly, for any $M \subseteq N$, $s_i =_N s_j$ implies $s_i =_M s_j$.

Definition 2.9. State s_j is reachable from s_i through $\mathscr{A}$ under N, denoted as $s_i \succ_{\mathscr{A}, N} s_j$, if and only if there is a word $x \in \mathfrak{W}(\mathscr{A})$ such that $s_i x =_N s_j$. Here, $s_i x$ denotes the updated state after applying the letters in x to s_i.

We note that $s_i \succ_{\mathscr{A}, N} s_j$ is a transitive operation, so if $s_1 \succ_{\mathscr{A}, N} s_2$ and $s_2 \succ_{\mathscr{A}, N} s_3$ then $s_1 \succ_{\mathscr{A}, N} s_3$.

Theorem 2.10. $s_1 \succ \mathscr{A}, \mathscr{A}\, s_n$ if and only if there exist intermediate states $s_2, s_3, \ldots, s_{n-1}$ such that

$$s_1 \succ \mathscr{U}, \mathscr{A}\, s_2 \succ \mathscr{U}^c, \mathscr{A}\, s_3 \succ \mathscr{U}, \mathscr{A}\, s_4 \succ \ldots \succ \mathscr{U}^c, \mathscr{A}\, s_{n-1} \succ \mathscr{U}, \mathscr{A}\, s_n . \tag{2.1}$$

Proof. If $s_1 \succ \mathscr{A}, \mathscr{A}\, s_n$, then from Definition 2.9, there is a word $x \in \mathfrak{W}(\mathscr{A})$ such that $s_1 x = \mathscr{A}\, s_n$. As $\mathscr{A} = \mathscr{U} \cup \mathscr{U}^c$, the word x can be decomposed into (possibly empty) words with letters in $\mathscr{U}$ and $\mathscr{U}^c$. Thus there exist intermediate states $s_2, s_3, \ldots, s_{n-1}$ satisfying 2.1. Conversely, if 2.1 holds, then there exist words $x_1 \in \mathfrak{W}(\mathscr{U})$, $x_2 \in \mathfrak{W}(\mathscr{U}^c)$, …, $x_{n-1} \in \mathfrak{W}(\mathscr{U})$ such that $s_1 x_1 = \mathscr{A}\, s_2$, $s_2 x_2 = \mathscr{A}\, s_3, \ldots$, and $x_{n-1} = \mathscr{A}\, s_n$. Then $x := x_1 x_2 \ldots x_{n-1}$ is a word in $\mathfrak{W}(\mathscr{A})$ and $s_1 x = \mathscr{A}\, s_n$. Thus $s_1 \succ \mathscr{A}, \mathscr{A}\, s_n$. □

Lemma 2.11. If $s_i \succ \mathscr{U}, \mathscr{A}\, s_j$, then $s_i \succ \mathscr{U}, \mathscr{A}^R\, s_j$.

Proof. Since $s_i \succ \mathscr{U}, \mathscr{A}\, s_j$, there exists a word $x \in \mathfrak{W}(\mathscr{U})$ such that $s_i x = \mathscr{A}\, s_j$. As the node set of the reduced alphabet $\mathscr{A}^R$ is a subset of the node set of $\mathscr{A}\, s_j$. Thus $s_i \succ \mathscr{U}, \mathscr{A}^R\, s_j$. □

Lemma 2.12. If $s_i \succ \mathscr{U}, \mathscr{A}^R\, s_j$, then $s_i \succ \mathscr{W}, \mathscr{A}^R\, s_j$ or $s_i w \succ \mathscr{W}, \mathscr{A}^R\, s_j$.

Proof. Since $s_i \succ \mathscr{U}, \mathscr{A}^R\, s_j$, there exists a word $x \in \mathfrak{W}(\mathscr{U})$ such that $s_i x = \mathscr{A}^R\, s_j$. We need to show the existence of a word $y \in \mathfrak{W}(\mathscr{W})$ with $s_i y = \mathscr{A}^R\, s_j$, or a word $y' \in \mathfrak{W}(\mathscr{W})$ with $s_i w y' \succ \mathscr{A}^R\, s_j$. Since x can only alter the states of v or w, and v cannot be observed, it suffices to show that $s_i y$ (or $s_i w y'$) and $s_i x$ have the same value at the position of node w. Indeed, all possible choices for $x \in \mathfrak{W}(\mathscr{U})$ are equivalent to one of the following cases:

- If $x = v$, then y can be the empty word;
- If $x = vw$, then we can take $y = \widetilde{w}$;
- If $x \in \{w, wv\}$, then y' can be the empty word.

It is straightforward to see that for the above y’s (or y'’s), $s_i y$ (or $s_i w y'$) and $s_i x$ have the same value at the position of node w. □

Lemma 2.13. If $s_i \succ \mathscr{U}^c, \mathscr{A}\, s_j$, then $s_i \succ \mathscr{W}^c, \mathscr{A}\, s_j$.

Proof. It follows directly from the fact that $\mathscr{W}^c = \mathscr{A}^R \backslash \{\widetilde{w}\} = \mathscr{A} \backslash \{v, w\} = \mathscr{U}^c$. □

Lemma 2.14. If $s_i \succ \mathscr{W}^c, \mathscr{A}\, s_j$, then $s_i \succ \mathscr{W}^c, \mathscr{A}^R\, s_j$.

Proof. The proof is the same as the proof of Lemma 2.11. □

Proposition 2.15. If $s_1 \succ \mathscr{A}, \mathscr{A}\, s_n$, then either $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\, s_n$ or there exists a word $x \in \mathfrak{W}(\mathscr{W}^c)$ such that $s_1 x w \succ \mathscr{A}^R, \mathscr{A}^R\, s_n$.

Proof. If $s_1 \succ \mathscr{A}, \mathscr{A}\ s_n$, then Theorem 2.10 implies that there exist intermediate states s_2, s_3, …, s_{n-1} such that

$$s_1 \succ \mathscr{U}, \mathscr{A}\ s_2 \succ \mathscr{U}^c, \mathscr{A}\ s_3 \succ \mathscr{U}, \mathscr{A}\ s_4 \succ \ldots \succ \mathscr{U}^c, \mathscr{A}\ s_{n-1} \succ \mathscr{U}, \mathscr{A}\ s_n.$$

We prove the assertion by induction on the length of this sequence starting from the right end. If the length of the sequence is one, there are two possible cases:

- $s_1 \succ \mathscr{U}, \mathscr{A}\ s_n$: In this case, by Lemma 2.11 we have $s_1 \succ \mathscr{U}, \mathscr{A}^R\ s_n$. Then based on Lemma 2.12, $s_1 \succ \mathscr{W}, \mathscr{A}^R\ s_n$. or $s_1 w \succ \mathscr{W}, \mathscr{A}^R\ s_n$. Since $\mathscr{W} \subseteq \mathscr{A}^R$, $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or $s_1 w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. By taking x to be the empty word in $\mathfrak{W}(\mathscr{W}^c)$, we have $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or $s_1 x w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$.

- $s_1 \succ \mathscr{U}^c, \mathscr{A}\ s_n$: In this case, based on Lemma 2.13, $s_1 \succ \mathscr{W}^c, \mathscr{A}\ s_n$, and then by Lemma 2.14, $s_1 \succ \mathscr{W}^c, \mathscr{A}^R\ s_n$. Since $\mathscr{W}^c \subseteq \mathscr{A}^R$, $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$.

Assume $s_l \succ \mathscr{A}, \mathscr{A}\ s_n$ implies $s_l \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or there exists $x' \in \mathfrak{W}(\mathscr{W}^c)$ such that $s_l x' w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. Now let $s_{l-1} \succ \mathscr{A}, \mathscr{A}\ s_n$.

If $s_{l-1} \succ \mathscr{A}, \mathscr{A}\ s_n$ and $s_l \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$, then either $s_{l-1} \succ \mathscr{U}, \mathscr{A}\ s_l \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or $s_{l-1} \succ \mathscr{U}^c, \mathscr{A}\ s_l \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. In the first case, $s_{l-1} \succ \mathscr{U}, \mathscr{A}\ s_l$ implies $s_{l-1} \succ \mathscr{U}, \mathscr{A}^R\ s_l$ (Lemma 2.11). So based on Lemma 2.12 either $s_{l-1} \succ \mathscr{W}, \mathscr{A}^R\ s_l$ or $s_{l-1} w \succ \mathscr{W}, \mathscr{A}^R\ s_l$. Then $s_{l-1} \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. or $s_{l-1} w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. With x being the empty word, we have $s_{l-1} \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or $s_{l-1} x w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. In the second case, $s_{l-1} \succ \mathscr{U}^c, \mathscr{A}\ s_l$ implies $s_{l-1} \succ \mathscr{W}^c, \mathscr{A}\ s_l$ (Lemma 2.13), and as a result $s_{l-1} \succ \mathscr{W}^c, \mathscr{A}^R\ s_l$ (Lemma 2.14). Since $\mathscr{W}^c \subseteq \mathscr{A}^R$, $s_{l-1} \succ \mathscr{A}^R, \mathscr{A}^R\ s_l$. Therefore $s_{l-1} \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$.

If $s_{l-1} \succ \mathscr{A}, \mathscr{A}\ s_n$ and $s_l x' w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ for some $x' \in \mathfrak{W}(\mathscr{W}^c)$, then based on the former condition we have two cases:

- $s_{l-1} \succ \mathscr{U}, \mathscr{A}\ s_l$ and $s_l x' w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$: In this case, according to the first condition there exists a word $y \in \mathfrak{W}(\mathscr{U})$ such that $s_{l-1} y = \mathscr{A}\ s_l$. Then $s_{l-1} y x' w = \mathscr{A}\ s_l x' w$. Since $x' \in \mathfrak{W}(\mathscr{W}^c)$ and it does not change the state of v, $s_{l-1} y x' w = \mathscr{A}\ s_{l-1} y w x'$. Thus $s_{l-1} y w x' = \mathscr{A}\ s_l x' w$. Since $x' \in \mathfrak{W}(\mathscr{W}^c)$ and $\mathscr{W}^c \subseteq \mathscr{A}$, we then have $s_{l-1} y w \succ \mathscr{A}, \mathscr{A}\ s_l x' w$. Since $yw \in \mathfrak{W}(\mathscr{U})$, we get $s_{l-1} \succ \mathscr{U}, \mathscr{A}\ s_l x' w$. Now Lemmas 2.11 and 2.12 imply that $s_{l-1} \succ \mathscr{W}, \mathscr{A}^R\ s_l x' w$ or $s_{l-1} w \succ \mathscr{W}, \mathscr{A}^R\ s_l x' w$. Since $\mathscr{W} \subseteq \mathscr{A}^R$ and $s_l x' w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$, we then have $s_{l-1} \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or $s_{l-1} w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. By taking x to be the empty word, we get $s_{l-1} \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or $s_{l-1} x w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$.

- $s_{l-1} \succ \mathscr{U}^c, \mathscr{A}\ s_l$ and $s_l x' w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$: In this case, the first condition and Lemma 2.13 imply that $s_{l-1} \succ \mathscr{W}^c, \mathscr{A}\ s_l$. Thus there is a word $x'' \in \mathfrak{W}(\mathscr{W}^c)$ such that $s_{l-1}x'' = \mathscr{A}\ s_l$, implying $s_{l-1}x''x'w = \mathscr{A}\ s_l x' w \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. By taking $x := x''x' \in \mathfrak{W}(\mathscr{W}^c)$, we have $s_{l-1}xw \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$.

So, $s_{l-1} \succ \mathscr{A}, \mathscr{A}\ s_n$ implies $s_{l-1} \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or $s_{l-1}xw \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ for some $x \in \mathfrak{W}(\mathscr{W}^c)$. Thus, by induction, $s_1 \succ \mathscr{A}, \mathscr{A}\ s_n$ implies $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or there is $x \in \mathfrak{W}(\mathscr{W}^c)$ such that $s_1 xw \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. □

PROPOSITION 2.16. *If* s_1 and s_n are two states in an attractor of $\mathscr{A}$, then $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$.

Proof. Suppose s_1 and s_n belong to the attractor C of $\mathscr{A}$. Then from Definition 2.1, $s_1 vw \in C$, and that $s_1 vw \succ \mathscr{A}, \mathscr{A}\ s_n$. Now by Proposition 2.15, $s_1 vw \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ or $s_1 vwxw \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$ for some $x \in \mathfrak{W}(\mathscr{W}^c)$. In the former case, since vw is equivalent to $\widetilde{w}$ and thus is in $\mathfrak{W}(\mathscr{A}^R)$, we have $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. In the latter case, our simplification rules imply that $s_1 vwxw \succ \mathscr{A}^R\ s_1 vwx$ (because x on the left hand side does not alter the state of v, so the second w has no effect). Since $vwx \in \mathfrak{W}(\mathscr{A}^R)$, we then have $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ s_n$. □

Now using the preceding results we prove that if two states are reachable from each other in the original model, their projections under π have the same property.

Proposition 2.17. If s_a and s_b are two states in an attractor of $\mathscr{A}$, then $\pi(s_a) \succ \mathscr{A}^R, \mathscr{A}^R\ \pi(s_b)$ and $\pi(s_b) \succ \mathscr{A}^R, \mathscr{A}^R\ \pi(s_a)$.

Proof. From Definition 2.1, if s_a and s_b are in the same attractor, $s_a \succ \mathscr{A}, \mathscr{A}\ s_b$ and $s_b \succ \mathscr{A}, \mathscr{A}\ s_a$. By Proposition 2.16, $s_a \succ \mathscr{A}^R, \mathscr{A}^R\ s_b$ and $s_b \succ \mathscr{A}^R, \mathscr{A}^R\ s_a$. Then we have $\pi(s_a) \succ \mathscr{A}^R, \mathscr{A}^R\ \pi(s_b)$ and $\pi(s_b) \succ \mathscr{A}^R, \mathscr{A}^R\ \pi(s_a)$. □

Finally, we prove that there is a one-to-one correspondence between complex attractors of the original and reduced models.

LEMMA 2.18. For every complex attractor C in the original model $\mathscr{M}$, $\pi(C)$ is a complex attractor in the reduced model $\mathscr{M}^R$.

Proof. Let C be a complex attractor in $\mathscr{M}$. For any $z_a, z_b \in \pi(C)$ there exist $s_a, s_b \in C$ such that $\pi(s_a) = z_a$ and $\pi(s_b) = z_b$. Since C is a complex attractor, $s_a \succ \mathscr{A}, \mathscr{A}\ s_b$ and $s_b \succ \mathscr{A}, \mathscr{A}\ s_a$. Then Proposition 2.17 implies that $z_a \succ \mathscr{A}^R, \mathscr{A}^R\ z_b$ and $z_b \succ \mathscr{A}^R, \mathscr{A}^R\ z_a$. Thus $\pi(C)$ is strongly connected.

On the other hand, suppose $z_a \in \pi(C)$ and $z_b \notin \pi(C)$. Choose $s_a \in C \cap \pi^{-1}(z_a)$ and $s_b \in \pi^{-1}(z_b)$. Then $s_a v \in C$ and $s_b \notin C$. Since C is an attractor, $s_a v \nsucc \mathscr{A}, \mathscr{A}\ s_b$. Since the words in

$\mathscr{A}^R$ are a subset of the words in $\mathscr{A}$, $s_a v \nsucc \mathscr{A}, \mathscr{A}\ s_b$ implies $\pi(s_a) \nsucc \mathscr{A}^R, \mathscr{A}^R\ \pi(s_b)$. Thus we have $z_a \nsucc \mathscr{A}^R, \mathscr{A}^R\ z_b$. This means that no path leaves $\pi(C)$ and thus $\pi(C)$ is an absorbing set.

Since $\pi(C)$ is strongly connected and absorbing, by Definition 2.1 it is an attractor in $\mathscr{M}^R$. □

LEMMA 2.19. For every complex attractor C^R of $\mathscr{M}^R$, there exists an attractor C of $\mathscr{M}$ such that $C \subseteq \pi^{-1}(C^R)$.

Proof. Take any attractor C^R of $\mathscr{M}^R$. Let $z \in C^R$ and define $s_1 = \mathscr{A}\ \pi^{-1}(z)v$. We note that such a state is uniquely defined, since updating of node v erases any ambiguity in the preimage and that $s_1 v = \mathscr{A}\ s_1$.

Let G be the set of all states in $\mathscr{M}$ reachable from s_1, including itself. By definition, G is an absorbing set, and must contain an attractor. We claim that $G \subseteq \pi^{-1}(C^R)$, and thus the preimage $\pi^{-1}(C^R)$ must contain an attractor.

In order to prove $G \subseteq \pi^{-1}(C^R)$, we first note that

$$\pi(s_1) \succ \mathscr{A}^R\ \pi(\pi^{-1}(z)v) = \mathscr{A}^R z \in C^R,$$

so $s_1 \in \pi^{-1}(C^R)$. Now, take $t \in G\backslash\{s_1\}$. By the definition of G, $s_1 \succ \mathscr{A}, \mathscr{A}\ t$. Then by Proposition 2.15, either $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ t$ or there exists $x \in \mathfrak{W}(\mathscr{W}^c)$ such that $s_1 x w \succ \mathscr{A}^R, \mathscr{A}^R\ t$. If $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ t$, $\pi(s_1) \succ \mathscr{A}^R, \mathscr{A}^R\ \pi(t)$. Otherwise, $s_1 x w \succ \mathscr{A}^R, \mathscr{A}^R\ t$ for some $x \in \mathfrak{W}(\mathscr{W}^c)$. Since $s_1 = \mathscr{A}\ s_1 v$, $s_1 v x w \succ \mathscr{A}^R, \mathscr{A}^R\ t$. Since w commutes over $\mathfrak{W}(\mathscr{W}^c)$, $s_1 v x w \succ \mathscr{A}^R, \mathscr{A}^R\ t$. Now since vw is equivalent to $\widetilde{w} \in \mathscr{W} \subset \mathscr{A}^R$ and $\mathscr{W}^c \subset \mathscr{A}^R$, we have $s_1 \succ \mathscr{A}^R, \mathscr{A}^R\ t$ and $\pi(s_1) \succ \mathscr{A}^R, \mathscr{A}^R\ \pi(t)$. So, in both cases, $\pi(s_1) \succ \mathscr{A}^R, \mathscr{A}^R\ \pi(t)$. Since $\pi(s_1) \in C^R$ and C^R is an attractor, $\pi(t) \in C^R$. Thus $t \in \pi^{-1}(C^R)$ and the claim is proven. □

LEMMA 2.20. For any complex attractor C of $\mathscr{M}$, $C \subseteq D = \pi^{-1}(\pi(C))$ and C is the only attractor inside D.

Proof. $C \subseteq \pi^{-1}(\pi(C))$ follows immediately from Definition 2.2. Let $z \in \pi(C)$ and $\{s_a, s_b\} = \pi^{-1}(z)$, and without loss of generality, take $s_a \in C$. Under our model's restrictions, v either copies or negates the state of u. In either case, $s_a v = \mathscr{A}\ s_b$ or $s_b v = \mathscr{A}\ s_a$. If $s_a v = \mathscr{A}\ s_b$, then since $s_a \in C$ and C is an attractor, $s_b \in C$. If $s_b v = \mathscr{A}\ s_a$, then any subset of $D\backslash C$ containing s_b cannot be absorbing. Since this holds for all $z \in \pi(C)$, no subset of $D\backslash C$ can be an attractor. □

LEMMA 2.21. For any complex attractor C^R of $\mathscr{M}^R$, π^{-1} (C^R) contains a unique attractor of $\mathscr{M}$.

Proof. Suppose C^R is an attractor of $\mathscr{M}^R$. By Lemma 2.19, $\pi^{-1}(C^R)$ must contain an attractor. Suppose then C_1 and C_2 are attractors in $\pi^{-1}(C^R)$. By Lemma 2.18, $\pi(C_1)$ and $\pi(C_2)$ are attractors of $\mathscr{M}^R$. Since C^R is an attractor and contains $\pi(C_1)$ and $\pi(C_2)$, we must have $\pi(C_1) = \pi(C_2) = C^R$. By Lemma 2.20, C_1 is the unique attractor in $\pi^{-1}(\pi(C_1))$, so $C_1 = C_2$. We thus conclude that attractors in pre-images must be unique. □

Proof of Theorem 2.7. It follows directly from Lemmas 2.18–2.21.

In summary, our reduction method projects attractors to attractors, does not merge attractors, and does not create spurious attractors.

3. Examples and Applications.

Now we provide some examples of attractor conservation under our reduction method. The first example, given in Figure 3.1, shows that if we remove node v with in- and out-degree of one, then fixed points are conserved; both the original and reduced models have two fixed points. Note that the fixed points of the reduced model are projection under π of the ones in the original model.

The second example, given in Figure 3.2, shows that upon removal of node v with in- and out-degree of one complex attractors are conserved; both the original and reduced models have exactly one complex attractor in this case.

We previously employed our reduction method to simplify the abscisic acid (ABA) signaling network with 43 nodes [15] as well as the T-LGL leukemia survival signaling network with 60 nodes [16]. For the former network, which has five source nodes, we totally eliminated 93% of the nodes, and for the latter network with six source nodes, we removed 87% of the nodes. For example, for the ABA network after removing the stabilized nodes we obtained the sub-network with 13 nodes illustrated in Figure 3.3a (for the description of Boolean rules one can refer to [15]). Then we eliminated the nodes with in/out-degree of one (such as PLC, GC and cGMP) and also the leaf nodes (KEV and KAP). It was recently proven that the removal of leaf nodes conserves attractors of a Boolean model [11]. We note that in this example we also applied a more general reduction method of removing a node with in/out-degree of one even though its up-and down-stream nodes have a common edge, such as in the case of InsP3 following the removal of PLC. After reduction, we obtained a simplified network with only three nodes as depicted in Figure 3.3b. The state transition graph of this simplified network is given in Figure 3.3c. As we can see the system has only one fixed point in which all the three nodes stabilize in the OFF state. In [15] we verified using numerical simulations that the 13-node network has a single attractor as well.

For both systems, the reduction method enabled us to correctly identify attractors of the original systems and make predictions about the effect of node perturbations on the final outcomes. Some of our predictions were corroborated by the existing experimental data and the rest can direct followup wet-bench experiments. For example, for the ABA system we found that in the case of knocking out a particular node of the network, both the original and reduced models lead to oscillation, and for the T-LGL signaling network we identified 19 potential therapeutic targets for T-LGL leukemia, 67% of which were supported by experimental data.

4. Discussion.

In this paper, we presented a two-step reduction method for Boolean network models and rigorously proved that it conserves the attractors of the original system under the general asynchronous Boolean model. It eliminates stable variables that have the same value on every attractor and removes simple mediator nodes. A great advantage of our method over the other existing methods [12, 22] is that it does not create spurious attractors in the reduced model. This method is especially suited for simplifying large-scale biological regulatory networks that involve sustained signals. Its application to two relatively large signaling networks with more than 10^{12} states in their state transition graphs demonstrated its ability to identify all attractors of the underlying systems and to make experimentally testable predictions about the long-term behaviors of the systems. Integration of our reduction method with the removal of leaf nodes (nodes with out-degree=0) as proposed in [2, 11, 14] can be very effective in simplifying biological regulatory networks.

It is worth mentioning that when removing a sequence of nodes that satisfy the conditions of Algorithm 1 or 2, our reduction method is independent of the order in which the nodes are chosen for elimination. Indeed, as our method does not allow the removal of auto-regulated nodes, the proof of Proposition 2.3 in [22] holds in our case as well. We note, however, that in the process of node removals, some of the nodes that cannot be eliminated at early steps (due to, for example, having in- and out-degree of more than one) may be eligible for removal after elimination of other nodes in the network. In addition, although we presented Algorithm 1 before Algorithm 2 for simplicity of notation, these two steps are actually independent of each other and can be applied on a system in any order.

Taking a step further, the proof of attractor conservation in the case of removing simple mediator nodes in Boolean models can be straightforwardly extended to iteratively removing nodes with in-degree or out-degree of one with no dependency between the downstream or upstream nodes, respectively. This extended reduction rule has been previously employed in the context of network inference of a mammalian signal transduction network [17].

It should be noted that although the proposed reduction method conserves the attractors of a given system, it can change the state transition graph and thus may have an impact on the relative size of the basins of attraction (defined as the set of states that can reach an attractor). However, this change is not expected to be drastic as, for example, we found in a previous study that the basins of attraction of the attractors for two reduced sub-networks of the T-LGL signaling network were approximately of the same relative size [16]. Further work is needed to determine any possible relation between the sizes of basins of the attractors of the original and reduced models.

In exploring the mathematics of an exact model reduction, there are intersections with a number of areas of pure mathematics and theoretical computer science. For example, the asynchronous Boolean networks that we focused on in this paper are a special case of nondeterministic finite state machines and of finite automata. From an algebraic perspective, our analysis can be also related to the theory of trace monoid actions and automatic semigroups. As such, some of these formalisms could be potentially used to further simplify the results presented in this study or to develop new ones.

Overall, our reduction method can greatly facilitate studying the long-term behavior of Boolean dynamic models of large-scale biological regulatory networks. A future extension of this method includes relaxing the assumption of independency between upstream and downstream nodes of the simple mediator nodes. It would be also interesting to study the applicability of this reduction method in identifying attractors of a broader class of dynamic models of biological regulatory networks.

Acknowledgments.

This work was supported by the NSF grants CCF-0643529 and PHY-1205840 (to RA) as well as the NSF grant DMS-0920822, NIH grant PAR-08–224, and Bill and Melinda Gates Foundation grant 49276 (to TCR).

REFERENCES

[1]. Albert R and Othmer HG (2003), The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster, J Theor Biol 223 (1):1–18. [PubMed: 12782112]

[2]. Bilke S and Sjunnesson F (2002), Stability of the Kauffman model, Phys Rev E Stat Nonlin Soft Matter Phys 65:016129. [PubMed: 11800758]

[3]. Chaves M, Albert R, and Sontag ED (2005), Robustness and fragility of Boolean models for genetic regulatory networks, J Theor Biol 235 (3):431–449. [PubMed: 15882705]

[4]. Chaves M, Sontag ED, and Albert R (2006), Methods of robustness analysis for Boolean models of gene control networks, Syst Biol (Stevenage) 153 (4):154–167. [PubMed: 16986617]

[5]. Faure A, Naldi A, Chaouiya C, and Thieffry D (2006), Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle, Bioinformatics 22 (14):E124–E131. [PubMed: 16873462]

[6]. Garg A, Di Cara A, Xenarios I, Mendoza L, and De Micheli G (2008), Synchronous versus asynchronous modeling of gene regulatory networks, Bioinformatics 24 (17):1917–1925. [PubMed: 18614585]

[7]. Harvey I and Bossomaier T, Time out of joint: Attractors in asynchronous random Boolean networks, In: Husbands P and Harvey I, eds., Proceedings of the fourth European conference on artificial life, MIT Press., Cambridge, 1997, pp 67–75.

[8]. Kauffman SA (1993), Origins of Order: Self-Organization and Selection in Evolution, Oxford Univ. Press, New York.

[9]. Li FT, Long T, Lu Y, Ouyang Q, and Tang C (2004), The yeast cell-cycle network is robustly designed, P Natl Acad Sci USA 101 (14):4781–4786.

[10]. Li S, Assmann SM, and Albert R (2006), Predicting essential components of signal transduction networks: a dynamic model of guard cell abscisic acid signaling, PLoS Biol 4 (10):e312. [PubMed: 16968132]

[11]. Naldi A, Monteiro PT, and Chaouiya C, Efficient handling of large signalling-regulatory networks by focusing on their core control, In: Gilbert D and Heiner M, eds., 10th International Conference on Computational Methods in Systems Biology (2012), pp 288–306.

[12]. Naldi A, Remy E, Thieffry D, and Chaouiya C (2011), Dynamically consistent reduction of logical regulatory graphs, Theoret. Comput. Sci 412 (21):2207–2218.

[13]. Papin JA, Hunter T, Palsson B∅, and Subramaniam S (2005), Reconstruction of cellular signalling networks and analysis of their properties, Nat Rev Mol Cell Biol 6 (2):99–111. [PubMed: 15654321]

[14]. Richardson KA (2005), Simplifying Boolean networks, Adv. Complex Syst. 8 (4):365–381.

[15]. Saadatpour A, Albert I, and Albert R (2010), Attractor analysis of asynchronous Boolean models of signal transduction networks, J Theor Biol 266 (4):641–656. [PubMed: 20659480]

[16]. Saadatpour A, Wang RS, Liao A, Liu X, Loughran TP, Albert I, and Albert R (2011), Dynamical and structural analysis of a T cell survival network identifies novel candidate therapeutic targets for large granular lymphocyte leukemia, PLoS Comput Biol 7 (11):e1002267. [PubMed: 22102804]

[17]. Saez-Rodriguez J, Alexopoulos LG, Epperlein J, Samaga R, Lauffenburger DA, Klamt S, and Sorger PK (2009), Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction, Mol Syst Biol 5:331. [PubMed: 19953085]

[18]. Saez-Rodriguez J, Simeoni L, Lindquist JA, Hemenway R, Bommhardt U, Arndt B, Haus UU, Weismantel R, Gilles ED, Klamt S, and Schraven B (2007), A logical model provides insights into T cell receptor signaling, PLoS Comput Biol 3 (8):e163. [PubMed: 17722974]

[19]. Sanchez L and Thieffry D (2001), A logical analysis of the Drosophila gap-gene system, J Theor Biol 211 (2):115–141. [PubMed: 11419955]

[20]. Sanchez L, van Helden J, and Thieffry D (1997), Establishement of the dorso-ventral pattern during embryonic development of Drosophila melanogasater: a logical analysis, J Theor Biol 189 (4):377–389. [PubMed: 9446747]

[21]. Thomas R and D’Ari R (1990), Biological Feedback, CRC Press, Boca Raton

[22]. Veliz-Cuba A (2011), Reduction of Boolean network models, J Theor Biol 289C:167–172.

[23]. Zhang R, Shah MV, Yang J, Nyland SB, Liu X, Yun JK, Albert R, and Loughran TP Jr. (2008), Network model of survival signaling in large granular lymphocyte leukemia, Proc Natl Acad Sci U S A 105 (42):16308–16313. [PubMed: 18852469]

[24]. Zhao Q (2005), A remark on“Scalar equations for synchronous Boolean networks with biological applications” by C. Farrow, J. Heidel, J. Maloney, and J. Rogers, IEEE Trans Neural Netw 16 (6):1715–1716. [PubMed: 16342513]

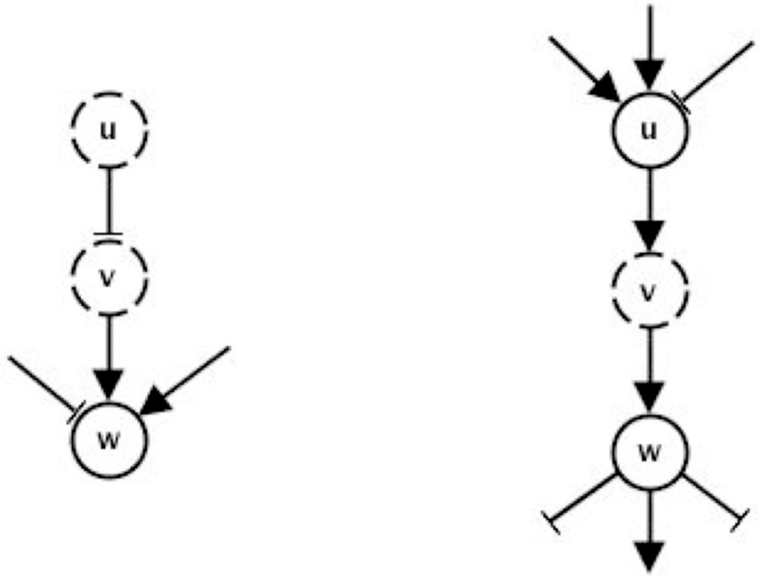


Fig. 2.1.

Illustration of Algorithms 1 and 2. Dashed nodes are removed by the algorithms. Arrowheads represent activation and blunt edges indicate inhibition. (a) A simple network illustrating the removal of stabilized nodes as described in Algorithm 1. Node *u* is a source node with a constant Boolean function. Both *u* and *v* are stabilized and can be removed from the network. (b) A simple network illustrating the removal of simple mediator nodes as described in Algorithm 2. Node *v* with in-degree and out-degree of one is removed.

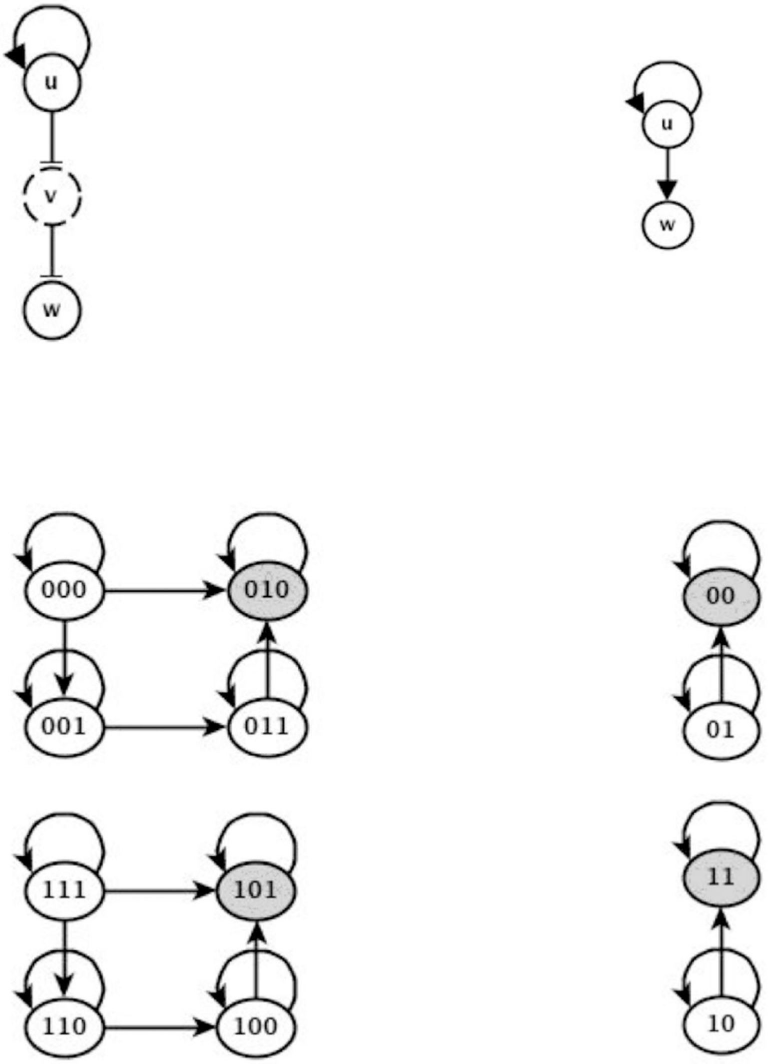


Fig. 3.1.

An example model-reduction illustrating conservation of fixed points. (a) Original network and respective Boolean rules. Arrowheads represent activation and blunt edges represent inhibition. The symbol * denotes the future state of the target node. (b) Reduced network and Boolean rules upon removal of node v from the network given in (a). (c) State transition graph of the original model. The binary digits from left to right represent the state of the nodes *u*, *v*, and *w*, respectively.(d) State transition graph of the reduced model. The binary digits from left to right represent the state of the nodes *u* and *w*, respectively. In (c) and (d), directed edges signify allowed transitions among states, and self-loops appear when a node is updated but its state does not change. The states with a gray background are fixed points of the systems.

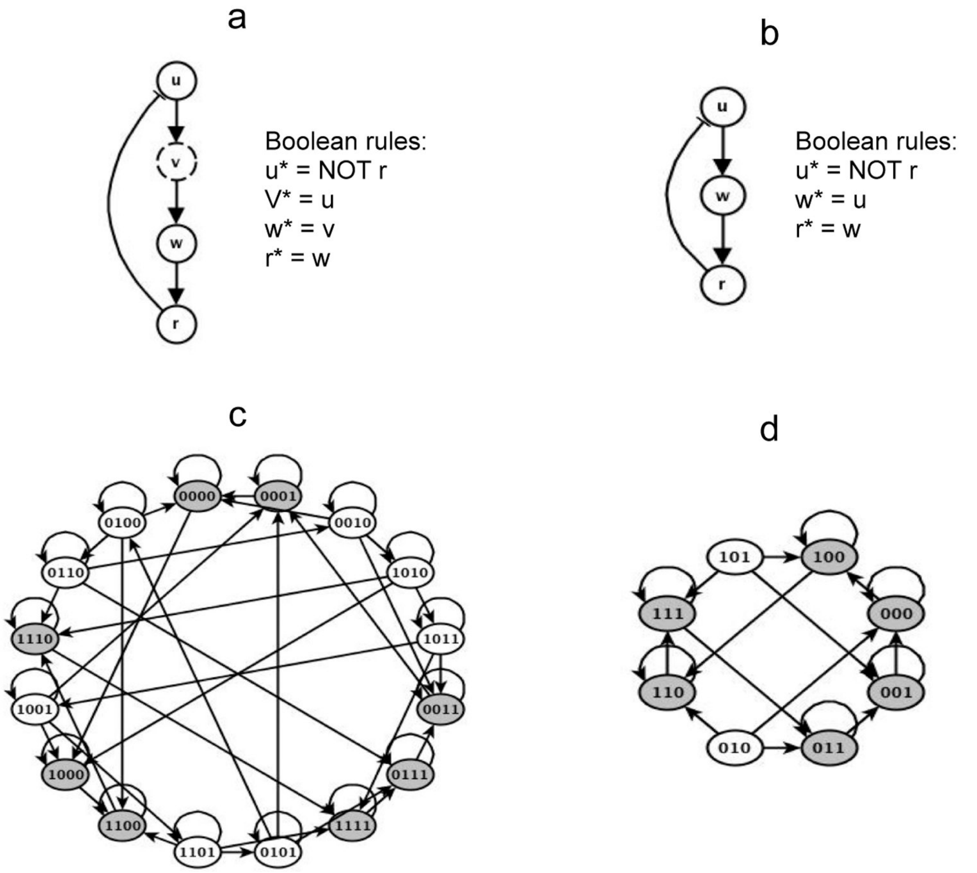
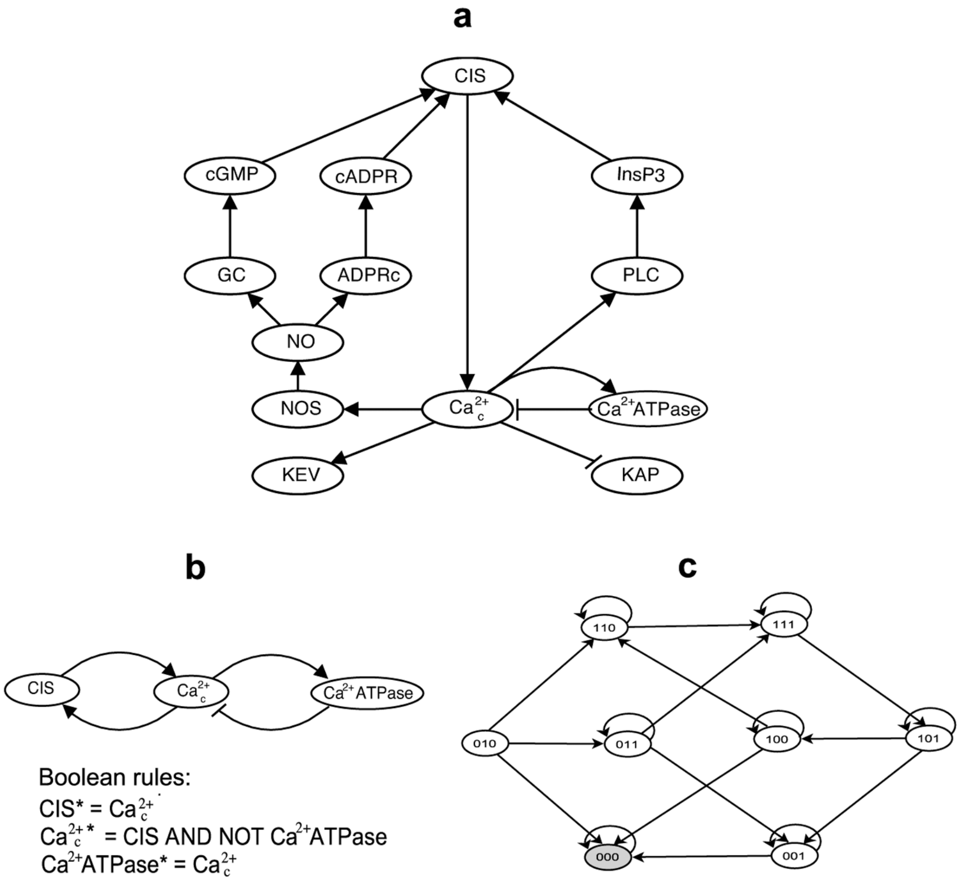


Fig. 3.2.

An example model-reduction illustrating conservation of complex attractors. (a) Original network and respective Boolean rules. Arrowheads represent activation and blunt edges represent inhibition. The symbol * denotes the future state of the target node. (b) Reduced network and Boolean rules upon removal of node *v* from the network given in (a). (c) State transition graph of the original model. The binary digits from left to right represent the state of the nodes *u*, *v*, *w* and *r*, respectively. (*d*) State transition graph of the reduced model. The binary digits from left to right represent the state of the nodes *u*, *w* and *r*, respectively. In (c) and (d), the directed edges signify allowed transitions among states, and self-loops appear when a node is updated but its state does not change. The states with a gray background form complex attractors of the systems.



Boolean rules:
$CIS^* = Ca_c^{2+}$
Ca_c^{2+*} = CIS AND NOT Ca^{2+}ATPase
Ca^{2+}ATPase* = Ca_c^{2+}

Fig. 3.3.

Reduced models of the ABA signaling network. (a) The sub-network obtained by removing the stabilized nodes. (b) The simplified sub-network and its corresponding Boolean rules obtained after shortening of the linear pathways. (c) The state transition graph of the sub-network given in (b). The binary digits from left to right represent the state of the nodes CIS, Ca_c^{2+}, and Ca^{2+}ATPase, respectively. The directed edges signify the allowed transitions among states. Self-loops appear when a node is updated but its state does not change. The state with a gray background is the only fixed point of the system. This figure has been adapted from [15].