The steady state system problem is NP-hard even for monotone quadratic Boolean dynamical systems

Winfried Just*

June 3, 2006

Abstract

In [2], the authors give a polynomial-time algorithm for deciding for a Boolean dynamical system in which each regulatory function is a monomial whether every limit cycle is a steady state. We show that the corresponding problem is NP-hard if the class of permissible regulatory functions contains the quadratic monotone functions $x_i \vee x_j$ and $x_i \wedge x_j$. We also show that the problem is NP-hard if the set of permissible regulatory functions includes all functions of the type $x_i x_j$ and $x_j + 1$.

1 Introduction

A *Boolean dynamical system* is a pair $< \mathbb{F}_2^n, f >$. The function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$ can be written as a vector of components $f = [f_1, \dots, f_n]$.

Boolean dynamical systems have recently found important applications as models of biochemical networks such as gene regulatory networks [8]. In these applications, x_i represents the concentration level of the i-th chemical in the network that is treated either as low (Boolean value 0) or high (Boolean value 1). The function f models the change of concentration levels over successive discrete time steps, and the i-th component f_i of f is the regulatory function for x_i.

Given a Boolean dynamical system, one would like to be able to deduce the dynamical properties of the system from the regulatory functions; ideally by means an efficient algorithm. Starting from any initial state vector $\bar{x} \in \mathbb{F}_2^n$, the system must eventually enter a limit cycle. Limit cycles of length one are called *steady states.* Boolean dynamical systems with the property that all limit cycles are steady states are called *steady state systems,* and we will refer to the problem of determining whether a given Boolean dynamical system is a steady state system as the *steady state system problem.*

It follows from the results of [5] which are based on earlier results of [3] that the steady state system problem has an efficient solution for linear Boolean dynamical systems. An

*Mathematical Biosciences Institute, 250 Mathematics Building, 231 W 18th Ave, Columbus, OH 43210 and Department of Mathematics, Ohio University, Athens, OH 45701

efficient algorithm for this has been developed in [6]. The structure of limit cycles has also been determined for affine Boolean systems in [9]. A Boolean dynamical system in which all regulatory functions are monomials, that is, are of the form $f_i = x_{j_1} x_{j_2} \dots x_{j_k}$ for some indices $j_1, j_2, \dots, j_k$, will be called a *monomial system.* It was shown in [2] that there exists a polynomial-time algorithm for deciding whether a given monomial system is a steady state system. The question naturally arises whether this result can be generalized to Boolean dynamical systems with regulatory functions restricted to other classes of functions, such as functions of the form $m(\bar{x}) + c$, where $m(\bar{x})$ is a monomial and $c \in \{0, 1\}$, or to the class of *monotone Boolean functions,* that is, combinations of the functions $x \vee y$ and $x \wedge y$.

In this note we show that the steady state system problem becomes NP-hard when the class of regulatory functions is allowed to contain either all quadratic monotone functions $x_i \vee x_j$, $x_i \wedge x_j$ or all functions of the kind $x_i x_j$ and $x_j + 1$. This shows that in at least one respect the results of [2] are nearly the best possible ones: While meaningful classification theorems for Boolean dynamical systems with more general than monomial regulatory functions may exist, one should not expect to be able to extract a polynomial-time algorithm for the steady state system problem from them.

When interpreting NP-hardness results, one must pay careful attention to the description length of an instance of the problem. Since Boolean functions in n variables can either be described by truth tables for 2^n input vectors or as a sum of at most 2^n monomials, in general the description length of a Boolean dynamical system in n variables is of the order 2^n; not a polynomial in n. However, all systems considered in this note will have regulatory functions that are either monomials plus a binary constant, are of the form $x_i \vee x_j$, $x_i \wedge x_j$ for some variables, or are combinations of at most m functions of the latter type. In the first two cases, the description length of the system will be bounded a polynomial in n; in the third case it will be bounded by a polynomial in n and m. When we claim that a decision problem for a class of Boolean dynamical systems is NP-hard, then we mean that unless $P = NP$, no deterministic algorithm for solving this problem can have an execution time that is bounded by a polynomial in k for instances of the problem of description length k.

It is know that, in general, determining whether a given Boolean dynamical system has *any* steady state is an NP-complete problem [1]. This is a different problem from the steady state system problem which asks whether *all* limit cycles are steady states. Note that the dual of the steady state system problem asks for the existence of a limit cycle of any length greater than one, and since the class P is closed under dual problems, all we need to show is that the dual problem is NP-hard. Note that neither the steady state problem nor its dual is obviously in the class NP, since limit cycles in general may be of exponential length in the number of variables. However, our proofs will show that for Boolean dynamical systems with regulatory functions restricted to the classes mentioned above, even the problem of deciding whether there exists a limit cycle of length two is NP-hard. Since verifying that two points form a limit cycle can easily be done in polynomial time, our results show that the problem of existence of limit cycles of length two becomes NP-complete, not merely NP-hard.

Throughout this paper we will use the notation $[n]$ for the set $\{1, \dots, n\}$.

2 Generalized monomial systems

We will say that $h : \mathbb{F}_2^n \to \mathbb{F}_2$ *depends on* x_j if there are values $x_1^*, \dots, x_{j-1}^*, x_{j+1}^*, \dots x_n^*$ such that $h(x_1^*, \dots, x_{j-1}^*, 0, x_{j+1}^*, \dots x_n^*) \neq h(x_1^*, \dots, x_{j-1}^*, 1, x_{j+1}^*, \dots x_n^*)$. The *support of* h, denoted by $supp(h)$, is the set of indices of variables that h depends on. We say that h is a *generalized monomial function* if there exist $v, u_i \in \mathbb{F}_2$ for $i \in supp(h)$ such that

$$h(x_1, \dots, x_n) = \prod_{i \in supp(h)} (x_i - u_i) + v.$$

A particular case of generalized monomial functions are functions h of the form $h(\bar{x}) = m(\bar{x}) + c$, where $m(\bar{x})$ is a monomial function and c is a Boolean constant. If each regulatory function is either of the form $x_i x_j$ for some (possibly equal) $i, j \in [n]$ or of the form $x_j + 1$, we call the system a *semiquadratic monomial + c system.*

We say that $< \mathbb{F}_2^n, f >$ with $f = [f_1, \dots, f_n]$ is a *consistently generalized monomial system* if there are $v_i \in \mathbb{F}_2$ for $i \in [n]$ such that for for all $i \in [n]$ we have:

$$f_i(x_1, \dots, x_n) = \prod_{j \in supp(f_i)} (x_j - v_j) + v_i.$$

Proposition 1 *Suppose $< \mathbb{F}_2^n, f >$ is a consistently generalized monomial system. Then $< \mathbb{F}_2^n, f >$ is isomorphic to a Boolean monomial system and the results of [2] apply.*

Proof: Let f be as in the assumption, and let v_i for $i \in \{1, \dots, n\}$ witness the fact that f is a consistently generalized monomial system. Define $\Phi : \mathbb{F}_2^n \to \mathbb{F}_2^n$ by $\Phi(x_1, \dots, x_n) = [x_1 - v_1, \dots, x_n - v_n]$. Clearly Φ is a bijection. Let $g = [g_1, \dots, g_n]$, where $g_i = \prod_{j \in supp(f_i)} x_j$. It is not hard to see that $f = \Phi^{-1} \circ g \circ \Phi$. Since each component of g is a monomial, the theorem follows. $\square$

Note that the notion of consistently generalized monomial system is stronger than the notion of a Boolean dynamical system in which every regulatory function is a generalized monomial function. We will show below (Theorem 2) that if we only assume that each regulatory function individually is a generalized monomial function, then the steady state system problem becomes NP-hard, even for semiquadratic monomial + c systems.

One can show that a Boolean function h is a generalized monomial function iff it is *canalizing* in every variable in its support, where a Boolean function $h : \mathbb{F}_2^n \to \mathbb{F}_2$ is *canalizing in* x_i if there exist $u, v \in \mathbb{F}_2$ such that $f(x_1, \dots, x_n) = v$ whenever $x_i = u$. It has been reported that most experimentally characterized gene regulatory functions are canalizing in several variables [4]. Thus Boolean dynamical systems in which every regulatory function is a generalized monomial function may be highly relevant in modeling of biochemical networks.

Theorem 2 *(i) The steady state system problem is NP-hard for the class of semiquadratic monomial + c Boolean systems.*
(ii) The problem of deciding whether a given semiquadratic monomial + c Boolean system has a limit cycle of length two is NP-complete.

Proof: Suppose $\{x_1, \ldots x_n\}$ is a set of Boolean variables, suppose $\psi = \psi_1 \wedge \cdots \wedge \psi_m$, where $\psi_k = y_{k,1} \vee y_{k,2} \vee y_{k,3}$ for all $k \in [m]$ and each $y_{k,j}$ is either $x_{i_{k,j}}$ or the negation $\neg x_{i_{k,j}}$ for some variable $x_{i_{k,j}}$. We will construct a semiquadratic monomial + c Boolean system $< \mathbb{F}_2^{2n+4m+2}, f >$ that has a limit cycle of length two iff it has a limit cycle of length > 1 iff ψ is satisfiable. Since the description length of this system is polynomial in n and m and since our construction can be carried out by a polynomial-time algorithm, this will show that 3-SAT is polynomial-time reducible to the steady state problem for semiquadratic monomial + c Boolean systems, and part (i) of the theorem will follow. Since it can easily be verified in polynomial time whether a given state is part of a cycle of length two, it will follow that the problem of deciding whether there exists such a cycle is NP-complete.

In the following description of the system $< \mathbb{F}_2^{2n+4m+2}, f >$, the letter i stands for a number in $[n]$ and the letter k stands for a number in $[m]$.

- $f_i = x_i$.
- $f_{n+i} = x_i + 1$.
- If $y_{k,1} = \neg x_{i_{k,1}}$ and $y_{k,2} = \neg x_{i_{k,2}}$, then $f_{2n+k} = x_{i_{k,1}} x_{i_{k,2}}$.
- If $y_{k,1} = x_{i_{k,1}}$ and $y_{k,2} = \neg x_{i_{k,2}}$, then $f_{2n+k} = x_{n+i_{k,1}} x_{i_{k,2}}$.
- If $y_{k,1} = \neg x_{i_{k,1}}$ and $y_{k,2} = x_{i_{k,2}}$, then $f_{2n+k} = x_{i_{k,1}} x_{n+i_{k,2}}$.
- If $y_{k,1} = x_{i_{k,1}}$ and $y_{k,2} = x_{i_{k,2}}$, then $f_{2n+k} = x_{n+i_{k,1}} x_{n+i_{k,2}}$.
- If $y_{k,3} = \neg x_{i_{k,3}}$, then $f_{2n+m+k} = x_{2n+k} x_{i_{k,3}}$.
- If $y_{k,3} = x_{i_{k,3}}$, then $f_{2n+m+k} = x_{2n+k} x_{n+i_{k,3}}$.
- $f_{2n+2m+k} = x_{2n+m+k} + 1$.
- $f_{2n+3m+1} = x_{2n+2m+1}$.
- $f_{2n+3m+k+1} = x_{2n+3m+k} x_{2n+2m+k+1}$ for $k \in [m-1]$.
- $f_{2n+4m+1} = x_{2n+4m+2} + 1$.
- $f_{2n+4m+2} = x_{2n+4m} x_{2n+4m+1}$.

Clearly, the above system is a semiquadratic monomial + c system.

Starting from an arbitrary Boolean vector $[x_1, \ldots x_{2n+4m+2}]$, after running the system for $m + 4$ steps or longer, the system will be in state $[z_1, \ldots, z_{2n+4m+2}]$, where the values of the first $2n + 4m$ variables will have settled as follows:

- $z_i = x_i$ for $i \in [n]$.
- $z_{n+i} = \neg x_i$ for $i \in [n]$.

- $z_{2n+k} = \neg(y_{k,1} \vee y_{k,2})$ for $k \in [m]$.
- $z_{2n+m+k} = \neg(y_{k,1} \vee y_{k,2} \vee y_{k,3})$ for $k \in [m]$.
- $z_{2n+2m+k} = \psi_k(x_1, \dots, x_n)$ for $k \in [m]$.
- $z_{2n+3m+k} = \psi_1(x_1, \dots, x_n) \wedge \dots \wedge \psi_k(x_1, \dots, x_n)$ for $k \in [m]$.
- In particular, $z_{2n+4m} = \psi(x_1, \dots, x_n)$.

If z_{2n+4m} settles to value 0, then $z_{2n+4m+2}$ will also settle to 0 and $z_{2n+3m+1}$ will settle to 1, so we reach a steady state. If z_{2n+4m} settles to value 1, then $z_{2n+4m+2}$ and $z_{2n+4m+1}$ will alternate between 0 and 1, and we reach a limit cycle of length two. The latter can (and will sometimes) happen iff ψ is satisfiable. □

3 Monotone Boolean dynamical systems

A *monotone* Boolean dynamical system is one in which every regulatory function is a combination of functions of the form $x_i \wedge x_j$ and $x_i \vee x_j$. Monotone Boolean functions have been widely studied and have found applications in a variety of fields (see [7] and references therein). Important examples of monotone Boolean functions are *threshold functions,* for example, the function that evaluates to 1 iff at least two of the variables x_1, x_2, x_3 are equal to 1 can be written as $(x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_1 \wedge x_3)$. We will call a monotone Boolean dynamical system *quadratic* if every regulatory function is of the form $x_i \vee x_j$ or $x_i \wedge x_j$. Note that this includes the case where $f_i(\bar{x}) = x_j$ for some i, j, since $x_j = x_j \vee x_j$.

Our proof of Theorem 2 heavily relies on the fact that negation of a Boolean variable can be expressed by a monomial plus a constant. Thus one might expect that the steady state system problem becomes computationally tractable for the class of monotone Boolean dynamical systems. Unfortunately, the problem remains NP-hard for this class of functions (Theorem 4).

Any nonconstant monotone Boolean function takes the value 0 on the input vector that consists only of 0's, and takes the value 1 on the input vector that consists only of 1's. Thus every monotone Boolean function dynamical system with nonconstant regulatory functions has at least two steady states, namely the vectors $[0, \dots, 0]$ and $[1, \dots, 1]$. Therefore the result of [1] does not directly apply to such systems. However, we can prove the following:

Theorem 3 *The problem of deciding whether a monotone Boolean dynamical system has at least three steady states is NP-complete.*

Proof: Suppose $\{x_1, \dots x_n\}$ is a set of Boolean variables, suppose $\psi = \psi_1 \wedge \dots \wedge \psi_m$, where $\psi_k = y_{k,1} \vee y_{k,2} \vee y_{k,3}$ for all $k \in [m]$ and each $y_{k,j}$ is either $x_{i_{k,j}}$ or the negation $\neg x_{i_{k,j}}$ for some variable $x_{i_{k,j}}$. We will construct a monotone Boolean system whose description length is polynomial in n and m that has a third steady state iff ψ is satisfiable.

We construct $< \mathbb{F}_2^{2n+3}, f >$ as follows:

- $f_i = (x_i \wedge x_{2n+1} \wedge x_{2n+3}) \vee x_{2n+2}$ for $i \in [2n]$.
- $f_{2n+1} = (x_1 \vee x_{n+1}) \wedge \cdots \wedge (x_n \vee x_{n+n})$.
- $f_{2n+2} = (x_1 \wedge x_{n+1}) \vee \cdots \vee (x_n \wedge x_{n+n})$.
- $f_{2n+3} = \psi^*(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n})$, where ψ^* is obtained from ψ by replacing every occurrence of $\neg x_i$ by x_{n+i}.

Clearly, each regulatory function of the above system is monotone and has polynomial description complexity in n and m.

We make the following observations about this system:

- We will call any state with $x_{2n+2} = 1$ or $x_i = x_{n+i} = 1$ for some $i \in [n]$ a *one-state.* If the system starts from a one-state, then the system will reach the steady state $[1, ..., 1]$.
- The system will never reach a one-state from any state that is not a one-state. Thus the basin of attraction of $[1, \ldots, 1]$ is the set of all one-states.
- We will call any state that is not a one-state and where $x_{2n+1} = 0$ or $x_{2n+3} = 0$ or $x_i = x_{n+i} = 0$ for some $i \in [n]$ a *zero-state.* If the system starts from a zero-state, then the system will reach the steady state $[0, ..., 0]$.
- We will call any state that is neither a zero-state nor a one-state a *proper state.* Note that a state is proper iff $x_{2n+1} = 1$, $x_{2n+2} = 0$, $x_{2n+3} = 1$ and $x_{n+i} = \neg x_n$ for all $i \in [n]$.
- The system will reach a zero-state only from another zero-state or from a proper state such that $\psi(x_1, \ldots, x_n) = 0$.
- Any proper state such that $\psi(x_1, \ldots, x_n) = 1$ is a steady state.

It follows that the system above is a steady state system and that is has more than two steady states iff ψ is satisfiable. This proves Theorem 3. $\square$

Theorem 4 *(i) The steady state system problem is NP-hard for the class of quadratic monotone Boolean systems.*
(ii) The problem of deciding whether a given monotone quadratic Boolean system has a limit cycle of length two is NP-complete.

Proof: Many ideas of this proof are already contained in the proofs of Theorems 2 and 3. Suppose $\{x_1, \ldots x_n\}$ is a set of Boolean variables, suppose $\psi = \psi_1 \wedge \cdots \wedge \psi_m$, where $\psi_k = y_{k,1} \vee y_{k,2} \vee y_{k,3}$ for all $k \in [m]$ and each $y_{k,j}$ is either $x_{i_{k,j}}$ or the negation $\neg x_{i_{k,j}}$ for some variable $x_{i_{k,j}}$. We will construct a monotone quadratic Boolean system whose description length is polynomial in n and m that has a limit cycle of length two iff the system is not a steady state system iff ψ is satisfiable.

For better transparency of our argument, we will not aim at constructing a dynamical system with the smallest possible number of variables, and we will use a variety of letters for our variables, depending on their role in the system. To avoid ambiguity, we will use the notation f_{x_i} instead of f_i for the regulatory function of a variable x_i. Let L be the smallest even integer such that $L \geq \max\{n+5, m+5\}$. The variable set of our dynamical system will be a disjoint union $X \cup C \cup V \cup U \cup W \cup T \cup D \cup E$, where:

- $X = \{x_{i,\ell} : i \in [n], \ell \in [L]\}$. This set contains L input vectors for ψ.
- $C = \{c_{i,\ell} : i \in [n], \ell \in [L]\}$. If the system is in a *proper state,* this set will contain the negations of the variables in X.
- $V = \{v_{i,j} : i \in [n], j \in [i]\}$. This set will be used to compute and retain the values of $x_{i,1} \vee c_{i,1}$.
- $U = \{u_\ell : \ell \in [L-4]\}$. This set will be used to compute and retain the value of $(x_{1,1} \vee c_{1,1}) \wedge \cdots \wedge (x_{n,1} \vee c_{n,1})$.
- $W = \{w_{i,j} : i \in [n], j \in [i]\}$. This set will be used to compute and retain the values of $x_{i,L-2} \wedge c_{i,L-2}$.
- $T = \{t_i : i \in [n+1]\}$. This set will be used to compute and retain the value of $(x_{1,L-2} \wedge c_{1,L-2}) \vee \cdots \vee (x_{n,L-2} \wedge c_{n,L-2})$.
- $D = \{d_{k,\ell} : k \in [m], \ell \in [k+1]\}$. This set will be used to compute and retain the values of ψ_k.
- $E = \{e_\ell : \ell \in [L-4]\}$. This set will be used to compute and retain the values of ψ.

We construct a Boolean dynamical system in these variables as follows:

- $f_{x_{i,1}} = x_{i,L} \vee t_{n+1}$ for $i \in [n]$.
- $f_{c_{i,1}} = c_{i,L} \vee t_{n+1}$ for $i \in [n]$.
- $f_{x_{i,\ell+1}} = x_{i,\ell}$ for $i \in [n]$ and $\ell \in [L-3]$.
- $f_{c_{i,\ell+1}} = c_{i,\ell}$ for $i \in [n]$ and $\ell \in [L-3]$.
- $f_{x_{i,L-1}} = x_{i,L-2} \wedge u_{L-4}$ for $i \in [n]$.
- $f_{c_{i,L-1}} = c_{i,L-2} \wedge u_{L-4}$ for $i \in [n]$.
- $f_{x_{i,L}} = x_{i,L-1} \wedge e_{L-4}$ for $i \in [n]$.
- $f_{c_{i,L}} = c_{i,L-1} \wedge e_{L-4}$ for $i \in [n]$.

- $f_{v_{i,1}} = x_{i,1} \vee c_{i,1}$ for $i \in [n]$.
- $f_{v_{i,j+1}} = v_{i,\ell}$ for $i \in [n]$ and $j \in [i-1]$.
- $f_{u_1} = v_{1,1}$.
- $f_{u_{i+1}} = u_i \wedge v_{i+1,i+1}$ for $i \in [n-1]$.
- $f_{u_{n+r}} = u_{n+r-1}$ for $r \in [L-4-n]$.
- $f_{w_{i,1}} = x_{i,L-2} \wedge c_{i,L-2}$ for $i \in [n]$.
- $f_{w_{i,j+1}} = w_{i,j}$ for $i \in [n]$ and $j \in [i-1]$.
- $f_{t_1} = w_{1,1}$.
- $f_{t_{i+1}} = t_i \vee w_{i+1,i+1}$ for $i \in [n-1]$.
- $f_{t_{n+1}} = t_n \vee t_{n+1}$.
- $f_{d_{k,1}} = x_{i_{k,1},1} \vee x_{i_{k,2},1}$ if $k \in [m]$ and $y_{k,1} = x_{i_{k,1}}$ and $y_{k,2} = x_{i_{k,2}}$.
- $f_{d_{k,1}} = x_{i_{k,1},1} \vee c_{i_{k,2},1}$ if $k \in [m]$ and $y_{k,1} = x_{i_{k,1}}$ and $y_{k,2} = \neg x_{i_{k,2}}$.
- $f_{d_{k,1}} = c_{i_{k,1},1} \vee x_{i_{k,2},1}$ if $k \in [m]$ and $y_{k,1} = \neg x_{i_{k,1}}$ and $y_{k,2} = x_{i_{k,2}}$.
- $f_{d_{k,1}} = c_{i_{k,1},1} \vee c_{i_{k,2},1}$ if $k \in [m]$ and $y_{k,1} = \neg x_{i_{k,1}}$ and $y_{k,2} = \neg x_{i_{k,2}}$.
- $f_{d_{k,2}} = d_{k,1} \vee x_{i_{k,3},2}$ if $k \in [m]$ and $y_{k,3} = x_{i_{k,3}}$.
- $f_{d_{k,2}} = d_{k,1} \vee c_{i_{k,3},2}$ if $k \in [m]$ and $y_{k,3} = \neg x_{i_{k,3}}$.
- $f_{d_{k,\ell+2}} = d_{k,\ell+1}$ for $k \in [m]$ and $\ell \in [k-1]$.
- $f_{e_1} = d_{1,2}$.
- $f_{e_{k+1}} = e_k \wedge d_{k,k+1}$ for $k \in [m]$.
- $f_{e_{m+r+1}} = e_{m+r}$ for $r \in [L-5-m]$.

Clearly, this system is a quadratic monotone system. It remains to show that ψ is satisfiable iff the above system has a limit cycle of length two iff the above system is not a steady state system.

Let $f(\bar{z})$ denote the successor state of a state $\bar{z}$ of our system, and let $f^r(\bar{z})$ denote the r-th successor of $\bar{z}$.

We call a state of the system a *one-state* if at least one of the following holds:

1. $t_i = 1$ for some $i \in [n+1]$.
2. $w_{i,j} = 1$ for some $i \in [n]$, $j \in [i]$.

3. $x_{i,\ell} = c_{i,\ell} = 1$ for some $i \in [n]$, $\ell \in [L]\backslash\{L-1\}$.

4. $x_{i,L-1} = c_{i,L-1} = 1$ and $e_{L-4} = 1$.

Lemma 5 *(i) If the system starts from a one-state, then it will eventually reach the steady state* $[1, \ldots, 1]$.
(ii) A one-state can only be reached from another one-state.
(iii) The basin of attraction of the steady state $[1, \ldots, 1]$ *is the set of all one-states.*

Proof: If $\bar{z}$ satisfies condition 4, then $f(\bar{z})$ will satisfy condition 3. If $\bar{z}$ satisfies condition 3, then $f^r(\bar{z})$ will satisfy condition 2 for some $r \in [L]$. If $\bar{z}$ satisfies condition 2, then $f^r(\bar{z})$ will satisfy condition 1 for some $r \in [n]$. If $\bar{z}$ satisfies condition 1, then $f^r(\bar{z})$ will satisfy $t_{n+1} = 1$ for some $r \in [n]$, and this will continue to be true for all subsequent states. If $\bar{z}$ satisfies $t_{n+1} = 1$, then $f^r(\bar{z})$ will satisfy $x_{i,1} = c_{i,1} = 1$ for all $i \in [n]$ for all positive r. For sufficiently large r all remaining variables of $f^r(\bar{z})$ can be expressed as monotone functions of the values that $x_{i,1}$ and $c_{i,1} = 1$ took in some state $f^s(\bar{z})$ with $0 \leq s < r$, and thus must turn to one. These observations prove part (i) of the lemma.

For the proof of part (ii) note that if $f(\bar{z})$ satisfies condition 1, then $\bar{z}$ must satisfy condition 1 or condition 2; if $f(\bar{z})$ satisfies condition 2, then $\bar{z}$ must satisfy condition 2 or condition 3; if $f(\bar{z})$ satisfies condition 3, then $\bar{z}$ must satisfy condition 3, condition 4, or $t_{n+1} = 1$; if $f(\bar{z})$ satisfies condition 4, then $\bar{z}$ must satisfy condition 3.

Part (iii) is an immediate consequence of parts (i) and (ii). □

Let $\ell \in [L]$. We will say that a state of the system is a *zero-state of modulus* ℓ if it is not a one-state and $x_{i,\ell} = c_{i,\ell} = 0$ for some $i \in [n]$.

Lemma 6 *(i) If* $\ell \in [L]$ *and if* $\bar{z}$ *is a zero-state of modulus* ℓ, *then the state* $f^{2L}(\bar{z})$ *will satisfy* $x_{i,\ell} = c_{i,\ell}$ *for all* $i \in [n]$.
(ii) If $\bar{z}$ *is a zero-state of modulus* ℓ *simultaneously for all* $\ell \in L$, *then* $f^{3L}(\bar{z}) = [0, \ldots, 0]$.
(iii) If $\bar{z}$ *is any state from which the system reaches a limit cycle of length* > 1, *then there exists a nonnegative integer* r *such that for every nonnegative integer* s *the state* $f^{r+sL}(\bar{z})$ *is neither a one-state nor a zero-state of modulus* 1.

Proof: Suppose $\bar{z}_0$ is a zero-state of modulus ℓ. The values of $x_{i,\ell}$ will cycle in L steps through $x_{i,\ell+1}, x_{i,\ell+2}, \ldots, x_{i,\ell-1}$ back to $x_{i,\ell}$. Along the way, a value of $x_{i,j}$ may change from 1 to 0, but it cannot change from 0 to 1 unless the system is in a one-state, which is precluded by the definition of a zero-state of modulus ℓ and Lemma 5(ii). A similar observation holds for $c_{i,\ell}$. Thus in at most L steps, the system reaches a zero-state of modulus 1; let us call this state $\bar{z}$ and note that $\bar{z} = f^{L+1-\ell}(\bar{z}_0)$. It will be succeeded by a state where $v_{i,1} = 0$ for some $i \in [n]$, and this value will be propagated until it is used to calculate u_i. More precisely, the state $f^{i+1}(\bar{z})$ will satisfy $u_i = 0$; and for all $r \in [L-3-i]$ it will be the case that $f^{i+r}(\bar{z})$ satisfies $u_{i+r-1} = 0$. Thus $u_{L-4} = 0$ in state $f^{L-3}(\bar{z})$, which implies that $f^{L-2}(\bar{z})$ turns all $x_{i,L-1}, c_{i,L-1}$ into 0's, and their shifted copies $x_{i,\ell}, c_{i,\ell}$ will all be 0 in the state $f^{L-1+\ell}(\bar{z}) = f^{2L}(\bar{z}_0)$. This proves part (i).

Now let $\bar{z}$ be a state as in the assumption of part (ii). By part (i), the state $f^{2L}(\bar{z})$ has 0's in all variables $x_{i,\ell}, c_{i,\ell}$, and this will continue to hold in all subsequent states. Since $\bar{z}$ is not a one-state, the variable t_{n+1} must be zero in state $\bar{z}$ and must remain 0 in state $f^{2L}(\bar{z})$ by Lemma 5. Now notice that every variable of the system in state $f^{3L}(\bar{z})$ can be expressed as a monotone function of the values that $t_{n+1}, x_{i,\ell}, c_{i,\ell}$ for $i \in [n], \ell \in [L]$ take in state $f^{2L}(\bar{z})$, and thus will be 0. This proves part (ii).

Now assume $\bar{z}$ is as in the assumptions of part (iii). Then $\bar{z}$ and any of its successors is not a one-state by Lemma 5. Note that if $\bar{z}$ is a zero-state of modulus ℓ, then $f(\bar{z})$ is a zero-state of modulus $\ell + 1$ (or of modulus 1 if $\ell = L$). Thus it follows from part (ii) that if $f^t(\bar{z}) \neq [0, \ldots, 0]$ for all times t, the system must, for some nonnegative integer r, enter a state $f^r(\bar{z})$ that is not a zero-state of modulus 1, and neither is $f^{r+sL}(\bar{z})$ for any nonnegative integer s. □

Lemma 7 *If ψ is not satisfiable, then the system is a steady state system.*

Proof: Assume that $\bar{z}$ is a state from which a limit cycle of length > 1 is reached. By Lemma 6(iii), we may assume wlog that $\bar{z}$ is neither a one-state nor a zero-state of modulus 1, and neither is $f^L(\bar{z})$. Note that this implies that in state $\bar{z}$, for all $i \in [n]$ we have $c_{i,1} = \neg x_{i,1}$. It follows from the choice of the corresponding regulatory functions that in state $f(\bar{z})$ we have $d_{k,1} = y_{k,1} \vee y_{k,2}$ (where the y's are computed from the $x_{i,2}$'s of $f(\bar{z})$), and in state $f^2(\bar{z})$ we have $d_{k,2} = \psi_k(x_{1,3}, \ldots, x_{n,3})$. Furthermore, in state $f^3(\bar{z})$ we will have $e_1 = \psi_k(x_{1,4}, \ldots, x_{n,4})$, and straightforward induction shows that for $r \in [L - m - 4]$ the state $f^{m+2+r}(\bar{z})$ will satisfy $e_{m+r} = \psi(x_{1,m+3+r}, \ldots x_{n,m+3+r})$. The value e_{L-4} of state $f^{L-2}(\bar{z})$ will be used to compute $x_{i,L}$ and $c_{i,L}$ in $f^{L-1}(\bar{z})$ from $x_{i,L-1}, c_{i,L-1}$. Our choice of $\bar{z}$ implies that we must have $e_{L-4} = 1$ in state $f^{L-2}(\bar{z})$; otherwise $f^L(\bar{z})$ would become a zero-state of modulus 1. But e_{L-4} can be equal to 1 only if ψ is satisfiable. □

Lemma 8 *If ψ is satisfiable, then our system has a limit cycle of length two.*

Proof: Let $x_{1,1}, \ldots x_{n,1}$ be such that $\psi(x_{1,1}, \ldots x_{n,1})$ holds. Define an initial state $\bar{z}$ as follows: For odd ℓ, let $x_{i,\ell} = x_{i,1}$ and $c_{i,\ell} = \neg x_{i,\ell}$ for $i \in [n]$. For even ℓ, let $x_{i,\ell} = c_{i,\ell} = 0$. Set $w_{i,j} = 0$ for all $i \in [n]$ and $j \in [i]$, and also let $t_i = 0$ for all $i \in [n+1]$. Set $v_{i,j} = 1$ for all $i \in [n]$ and even $j \in [i]$, and $v_{i,j} = 0$ for all $i \in [n]$ and odd $j \in [i]$. Let $u_\ell = 1$ for all odd $\ell \in [L-4]$ and $u_\ell = 0$ for all even $\ell \in [L-4]$. Set $d_{k,\ell} = 0$ for all $k \in [m]$ and odd $\ell \in [k+1]$; for all $k \in [m]$ and even $\ell \in [k+1]$, set $d_{k,\ell} = 1$. Finally, set $e_\ell = 0$ for all odd $\ell \in [L-4]$ and $e_\ell = 1$ for all even $\ell \in [L-4]$.

Clearly, $f(\bar{z}) \neq \bar{z}$. It is straightforward, if somewhat tedious, to verify that $f^2(\bar{z}) = \bar{z}$. This proves Lemma 8 and concludes the proof of Theorem 4. □ □

Acknowledgements

This material is based upon work supported by the National Science Foundation under Agreement No. 0112050 while the author was a visitor at the Mathematical Biosciences

Institute. The author wishes to thank the MBI for providing an excellent research environment.

References

[1] Akutsu, T., Kuhara, S., Maruyama, O., and Miyano, S. (1998). Identification of Gene Regulatory Networks by Strategic Gene Disruptions and Gene Overexpressions. *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '98),* 695–702.

[2] Colón-Reyes, O., Laubenbacher, R., and Pareigis, B. (2004). Boolean Monomial Dynamical Systems. *Annals of Combinatorics,* **8**, 425–439.

[3] Elspas, B. (1959). The theory of autonomous linear sequential networks. *IRE Transactions on Circuit Theory* **CT-6**, 45–60.

[4] Harris, S. E., Sawhill, B. K., Wuensche, A., Kauffman, S. (2002). A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complexity* **7**(4), 23–40.

[5] Hernández-Toledo, R. A. (2005). Linear finite dynamical systems. *Communications in Algebra* **33**, 2977–2989.

[6] Jarrah, A. S., Laubenbacher, R., and Vera-Licona, P. (200?) An efficient algorithm for finding the phase space structure of linear finite dynamical systems. *In review.*

[7] Korshunov, A. D. and Shmulevich, I. (2002). On the distribution of the number of monotone Boolean functions relative to the number of lower units. *Discrete Mathematics* **257**, 463–479.

[8] Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution.* Oxford U Press.

[9] Milligan, D. K. and Wilson, M. J. D. (1993). *Connection Science* **5**, 153–167.