SCALAR EQUATIONS FOR SYNCHRONOUS BOOLEAN NETWORKS WITH BIOLOGICAL APPLICATIONS

CHRISTOPHER FARROW, JACK HEIDEL*, JOHN MALONEY, AND JIM ROGERS

Department of Mathematics, University of Nebraska at Omaha, Omaha, NE 68182-0243

*Contact author: jheidel@mail.unomaha.edu

Keywords: Boolean network; Nonlinear scalar equation; Reduced scalar equation

Summary

One way of coping with the complexity of biological systems is to use the simplest possible models which are able to reproduce at least some non-trivial features of reality.

Although two value Boolean models have a long history in technology (Boole, 1854), it is perhaps a little bit surprising that they can also represent important features of living organisms. In this paper, the scalar equation approach (Heidel *et al.*, 2003) to Boolean network models is further developed and then applied to two interesting biological models. In particular, a linear reduced scalar equation is derived from a more rudimentary nonlinear scalar equation. This simpler, but higher order, two term equation gives immediate information about both cycle and transient structure of the network.

Introduction

Boolean networks are proving to be quite useful in modeling cell regulation (Albert and Othmer, 2002; Huang, 2001; Huang 2002b; Huang & Ingber,

2000). The present authors are developing this approach in a long term project to understand the complexities of intra-cellular signal transduction (Heidel *et al.*, 2003). Describing a large biochemical reaction network with on-off Boolean variables is the simplest way to formalize large scale complexity in a realistic manner. The above mentioned papers, each with many references, give extensive motivation and both biological and mathematical detail. Here is presented a brief review of the essentials.

A Boolean network is a set of nodes (proteins or other molecules) A, B, C, ..., D which interact with each other in a synchronous manner. At each given time $t = 0, 1, 2, \ldots$ a node has only one of two different values: 1 (on) or 0 (off). Thus, the network can be described by a set of equations:

$$\begin{aligned} A_{t+1} &= f_A(A_t, B_t, C_t, \ldots, D_t) \\ B_{t+1} &= f_B(A_t, B_t, C_t, \ldots, D_t) \\ C_{t+1} &= C_A(A_t, B_t, C_t, \ldots, D_t) \\ &\vdots \\ D_{t+1} &= f_D(A_t, B_t, C_t, \ldots, D_t). \end{aligned}$$

Each function of f has only constant, linear, or product terms and is determined from the logical table for the behavior of a particular node. For example, the logical table for **OR**

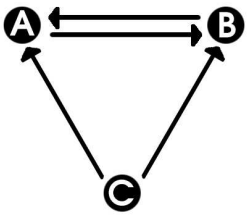
B	C	A
0	0	0
1	0	1
0	1	1
1	1	1

describes the behavior of node A in terms of B and C. This relationship can be expressed by the equation

$$A_{t+1} = B_t + C_t + B_t C_t$$

in mod 2 algebra, also referred to as AND/ExOR Boolean algebra (Warren-Smith, 2002).

Since each node has two possible values, 0 or 1, an n node network has 2^n different states. The three equivalent formulations of the network: logical tables, directed graph of states, or network equations are all useful. For example, the network



A	B
0	1
1	0

OFF

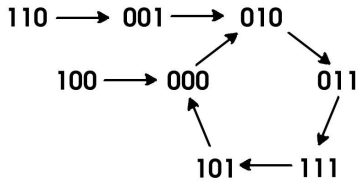
B	C
0	0
1	1

ON

B	C	A
0	0	0
0	1	0
1	0	0
1	1	1

AND

has the state space graph



and the network equations

$$A_{t+1} = B_t C_t$$
$$B_{t+1} = 1 + A_t$$
$$C_{t+1} = B_t.$$

The network has a single cycle of period five with three transient states.

This example clearly illustrates several important features of any Boolean network. Since there are only a finite number of states, every state eventually ends up on a cycle which may, of course, be a fixed point. Since the logic can be arbitrarily specified, in general, any state can move to any other state and there can be many different cycles in the same network. The most important feature of a Boolean network is this cycle structure. For any particular network, most likely defined by n logical tables which can

then be converted into n equations, the main task is to find all of the cycles. Because of the exponential growth in the number of states, this is a nontrivial task as soon as n becomes large.

It is a simple, intuitive observation that the behavior of any node A on a cycle of length k can be described by the equation $A_{t+k} = A_t$ for sufficiently large t (after all transients are traversed). If there are two cycles of length k_1 and k_2 then for k = least common multiple(k_1, k_2) again $A_{t+k} = A_t$ for sufficiently large t. What is interesting and nontrivial, however, is that such a "reduced scalar equation" can be found analytically from the network equations. Having such an equation $A_{t+k} = A_t$ for even one node A then shows that the length of all cycles in the network is a factor of k or a multiple of k. In fact, the reduced scalar equations found in this paper also provide information on the maximal transient length before a cycle is reached. Such information suggests a method for finding the actual cycles of a large network: iterate a random state beyond the maximal transient length and the resulting state will be a cycle point.

The reduced scalar equation, which has a simple linear form, is derived from a smaller order, often non-linear, scalar equation (or set of equations). The original scalar equation, itself, can be used to actually find the k states of any cycle of length k. However, the procedure for doing this may be "size 2^n" (Heidel *et al.*, 2003), and, therefore, practically not feasible. Thus, the scalar equation and reduced scalar equation have different, complimentary, uses even though the latter is derived from the former. These general concepts are illustrated in the many examples which follow.

Although synchronous Boolean networks can model very complicated systems, they do not, strictly speaking, exhibit chaos because they are of finite size. However if time intervals can be arbitrary, i.e. asynchronous, then

chaos may appear (Darby & Mysak, 1993; Robert, 1995; Thomas & Kaufman, 2001) . But how does one introduce asynchrony in a simple, natural way? This issue is now being successfully addressed for random Boolean networks (Harvey and Bossomaier, 1997; DiPaolo, 2000; DiPaolo, 2001). But until it can be resolved for the specific, deterministic networks we are interested in, it seems preferable to avoid this extra complication. Many of the results of this paper were developed in a recent thesis (Farrow, 2002).

Boolean networks are a highly specialized class of the more general neural networks. Feedback, or recurrent, neural networks may have cycles which can be found in particular cases (Zurada, 1992). Very little is known in general about cycles in neural networks. The one exception is fuzzy cognitive maps (Kosko, 1997). Here the signal or activation function is binary threshold or sigmoidal with an integer valued linearly computed input. This is the natural direction in which to extend the general cycle theory for Boolean networks discussed here. An intriguing possibility is to use three-valued logic (Adamatzky, 2003) to include uncertainty as well as true and false.

Scalar Equation(s)

A scalar equation is an ordinary recurrence equation for a particular node of a Boolean network. Where the logic equations of a network completely determine the exact structure of a network, a scalar equation may be better suited for analytically finding the general cyclic structure of the network. Consider the following three node network

$$A_{t+1} = 1 + C_t$$
$$B_{t+1} = A_t$$
$$C_{t+1} = B_t.$$

With a small network like this, one can easily determine the exact structure from these equations by plugging in each possible system state ($2^3 = 8$ of them) and mapping out their trajectories. However, this procedure is

practically impossible for large networks since an n node network has 2^n states. Scalar equations can sometimes be used to determine the cyclic structure of the network without determining the trajectories of all 2^n states in the state space.

By progressing the first of the above equations forward in time two steps (shifting by two) and making substitutions one finds $A_{t+3} = 1 + C_{t+2} = 1 + B_{t+1} = 1 + A_t$. The other nodes have identical scalar equations. From this scalar equation it follows immediately that all elements of the state space lie on an orbit of period six. It also follows that there are no period three cycles and no fixed points. Thus, the only possibility besides a full period six cycle is a period two cycle. It is easily verified that this network has a period two and a period six cycle (Heidel *et al.*, 2003).

As is generally the case with recurrence relations, linear examples such as the one above are easy to handle. And linear Boolean networks will produce linear scalar equations. There are also matrix methods available for analyzing linear Boolean networks (Cull, 1971; Milligan & Wilson, 1993; Wilson & Milligan, 1992). Systems with more nodes and non-linear logic equations, have more complex scalar equations. Take for example the following four node network with one non-linear term

$$\begin{aligned} A_{t+1} &= B_t D_t \\ B_{t+1} &= 1 + A_t \\ C_{t+1} &= B_t \\ D_{t+1} &= C_t. \end{aligned}$$

The scalar equation for node A of this network is $A_{t+4} = (1 + A_{t+2})(1 + A_t)$. Unlike the linear case, this nonlinear scalar equation does not immediately reveal anything about the cycle structure of the network. Furthermore, the other three nodes share a different scalar equation. Since A_4 is a function of A_0, A_1, A_2, A_3, the structure of this network could be determined by specifying all possible binary sequences of length four (16 of them) and inserting

them into the scalar equations, but this does not save any work. It turns out that for this example a simple analysis finds both a period three and a period six orbit (Heidel *et al.*, 2003).

Note that the scalar equations for the above two networks are of order three and four respectively, the same as the number of nodes in the networks. But this simple relationship may not always hold. For example the network

$$A_{t+1} = B_t + C_t$$
$$B_{t+1} = A_t C_t$$
$$C_{t+1} = A_t + B_t$$

has a fixed point and also orbits of period two and three. It does not (apparently) have a scalar equation of order three. However, it is easy to derive the relationship

$$A_{t+2} = A_t + C_t + A_t C_t + A_{t+1}$$

and then, noting the symmetry between nodes A and C,

$$C_{t+2} = A_t + C_t + A_t C_t + C_{t+1}.$$

Thus, specifying all of $A_t, A_{t+1}, C_t, C_{t+1}$, completely determines the trajectories. For example if $A_0 = 0$ and $A_1 = 1$ then all of the possibilities for A_t's and C_t's are

	A_t	C_t
$t = 0$	0	0 0 1 1
$t = 1$	1	0 1 0 1
$t = 2$	1 0	1 1 1 0
$t = 3$	0 1	0 0 1 1
$t = 4$	1 0	1 0 0 0
$t = 5$	1 1	0 1 1 1

which exhibits the periodicity for A_t and C_t. Other initial values of A_t and A_{t+1}, give the same cycles. It turns out to be the symmetry of the network equations which allows two cycles to exist instead of one and thereby makes the scalar equations more complicated. But the symmetry also provides a

simplification in deriving a reduced scalar equation, which is described in the next section. The question of the trade off between symmetry and simplicity of scalar equations is interesting in its own right and will be examined in a future paper.

The Reduced Scalar Equation

Even though the scalar equations for a synchronous Boolean network can be found in many cases, they may not prove to be useful if they take a complex form. The reduced scalar equation is a simplified form of the scalar equation that readily reveals information about the cyclic and transient structure of a network. An example will help introduce this technique.

Consider the scalar equation $A_{t+3} = 1 + A_t$ from above. This equation is used to show that the associated network has a cycle of length two and a cycle of length six. Shifting this equation by three yields the equation $A_{t+6} = 1 + A_{t+3}$. Substituting this into the original scalar equation (mod 2) gives a new equation $A_{t+6} = A_t$. This simplified but higher order equation is referred to as the reduced scalar equation. It only has two terms, and the order of the equation gives information about the cyclic structure of the network. The equation reveals that all states of node A lie on an orbit of period six, as did the original scalar equation. More specifically, the equation indicates that the length of any cycle must be a divisor of 6. So, node A can have a period of one, two, three, or six. In this example, the information gathered from the reduced scalar equation is less helpful than that of the original scalar equation. Indeed, the reduced scalar equation introduces the possibility of cycles of length one and three where cycles of these lengths are prohibited by the scalar equation. However, the reduced scalar equation proves very useful when a network has more complicated scalar equations.

Consider the network

$$A_{t+1} = B_t C_t$$
$$B_{t+1} = 1 + A_t$$
$$C_{t+1} = B_t$$

with non-linear scalar equations

$$A_{t+3} = 1 + A_{t+1} + A_t + A_{t+1}A_t = (1 + A_t)(1 + A_{t+1})$$
$$B_{t+3} = 1 + B_{t+1}B_t$$
$$C_{t+3} = 1 + C_{t+1}C_t.$$

Finding the structure of the network with these equations would require finding the output for all eight initial binary sequences of length three (Heidel *et al.*, 2003).

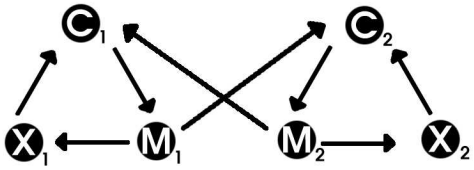
As an alternative to examining cycle structure directly from a third order scalar equation, iterate the equation for B (which appears simpler than the equation for A) to get

$$B_{t+3} = 1 + B_t B_{t+1}$$
$$B_{t+4} = 1 + B_{t+1}B_{t+2}$$
$$B_{t+5} = 1 + B_{t+2}(1 + B_t B_{t+1})$$
$$B_{t+6} = B_t B_{t+1} + (1 + B_t)B_{t+1}B_{t+2}$$
$$B_{t+7} = B_{t+2}.$$

Each of the other two nodes A and C have the exact same reduced scalar equations. We learn two things from these equations. First, all cycles of this network have period five. In addition, the equation shows that the longest possible transient trajectory of the network has length two. That is, given any point in the state space, after two iterations it will be on a cycle. A simple check shows that this network has no fixed points, so there must be a single cycle of length five and three transients. These simple and exact (see the example in the introduction) observations do not define the actual cycle, but they nonetheless give useful information about the structure of the network.

Biological Examples

The next example is a Boolean equation model of coupled oscillations in the cell cycle (Goodwin 1963; Heidel *et al.*, 2003). The connectivity graph, logic tables, and network equations are given below



..

.

X_1	M_2	C_1
0	0	1
0	1	0
1	0	0
1	1	0

NOR

C_1	M_1
0	0
1	1

ON

.

M_1	X_1
0	0
1	1

ON

.

X_2	M_1	C_2
0	0	1
0	1	0
1	0	0
1	1	0

NOR

C_2	M_2
0	0
1	1

ON

.

M_2	X_2
0	0
1	1

ON

$$C_{1(t+1)} = 1 + X_{1(t)} + M_{2(t)} + X_{1(t)} M_{2(t)}$$
$$M_{1(t+1)} = C_{1(t)}$$
$$X_{1(t+1)} = M_{1(t)}$$
$$C_{2(t+1)} = 1 + X_{2(t)} + M_{1(t)} + X_{2(t)} M_{1(t)}$$
$$M_{2(t+1)} = C_{2(t)}$$
$$X_{2(t+1)} = M_{2(t)}.$$

Noting the symmetry of the system, the network equations can easily be reduced to two equations

$$C_{1(t+3)} = (1 + C_{1(t)})(1 + C_{2(t+1)})$$
$$C_{2(t+3)} = (1 + C_{2(t)})(1 + C_{1(t+1)})$$

from which the period five and period ten orbits can be found directly (Heidel *et al.*, 2003). Alternatively, the reduced scalar equation can be found as follows. To simplify the equations make the substitution $A_t = 1 + C_{1(t)}$ and $B_t = 1 + C_{2(t)}$ to get the new equations

$$A_{t+3} = A_t B_{t+1} + 1$$
$$B_{t+3} = A_{t+1} B_t + 1.$$

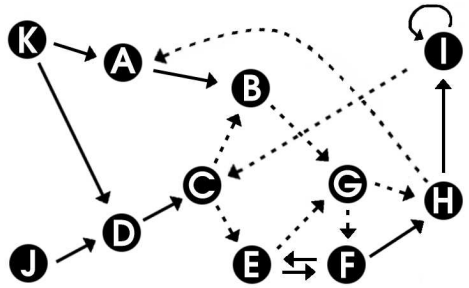
Iterating and substituting we obtain in turn

$$A_{t+4} = A_{t+1} B_{t+1} + 1$$
$$A_{t+5} = A_{t+2} + A_{t+1} A_{t+2} B_t + 1$$
$$A_{t+6} = (A_t + A_{t+2} + A_t A_{t+2}) B_{t+1}$$
$$A_{t+7} = [A_{t+1} + (A_{t+1} + 1)(A_t B_{t+1} + 1)] B_{t+2}$$
$$A_{t+8} = [A_{t+1}(A_{t+2} + 1) B_{t+2} + 1](A_{t+1} B_t + 1)$$
$$A_{t+9} = A_{t+2} B_{t+1} + 1 = B_{t+4}.$$

Converting back to C_1 and C_2 we have $C_{1(t+9)} = C_{2(t+4)}$. By symmetry in the logic equations, this implies $C_{2(t+9)} = C_{1(t+4)}$. Thus, the network is characterized by the reduced scalar equations $C_{1(t+14)} = C_{1(t+4)}$ and similarly for C_2. The network can contain fixed points or cycles of length two, five or ten. The longest transient chain is four system states in length, which is exact. It is easy to show that there are no fixed points. Again, the reduced scalar equations yield less information by themselves than the pair of order three nonlinear equations. But they show that no transient chain is longer than three, i. e. that any initial state will be moved to somewhere on a cycle by the fourth iteration.

Another example is the Boolean model of cell growth, differentiation, and apoptosis (programmed cell death) introduced by Huang and Ingber (2000). For simplicity, the 11 nodes of the network are represented by the letters

A through J. The connectivity graph, logic tables, and network equations are given below (**NIF** = "not if", **IMP** = "implication", **NAND** = "not and").



K	H	A
0	0	0
0	1	0
1	0	1
1	1	0

NIF

A	C	B
0	0	0
0	1	0
1	0	1
1	1	0

NIF

D	I	C
0	0	1
0	1	1
1	0	0
1	1	1

IMP

J	K	D
0	0	0
0	1	0
1	0	0
1	1	1

AND

C	F	E
0	0	1
0	1	1
1	0	0
1	1	1

IMP

E	G	F
0	0	0
0	1	1
1	0	0
1	1	1

NIF

B	E	G
0	0	1
0	1	1
1	0	1
1	1	0

NAND

F	G	H
0	0	0
0	1	0
1	0	1
1	1	0

NIF

H	I	H
0	0	0
0	1	0
1	0	1
1	1	0

NIF

J	J
0	0
1	1

ON

K	K
0	0
1	1

ON

$$A_{t+1} = K_t + K_t H_t$$
$$B_{t+1} = A_t + A_t C_t$$
$$C_{t+1} = 1 + D_t + D_t I_t$$
$$D_{t+1} = J_t K_t$$
$$E_{t+1} = 1 + C_t + C_t F_t$$
$$F_{t+1} = E_t + E_t G_t$$
$$G_{t+1} = 1 + B_t E_t$$
$$H_{t+1} = F_t + F_t G_t$$
$$I_{t+1} = H_t + H_t I_t$$
$$J_{t+1} = J_t$$
$$K_{t+1} = K_t.$$

In that article it is shown that a non-trivial growth attractor exists, assuming that the growth factor (node K) and cell spreading (node J) are both on. Iterating and substituting in the usual way we obtain (letting J=K=D=1)

$$I_{t+2} = F_t(1 + G_t)(1 + I_{t+1})$$
$$I_{t+3} = F_{t+1}(1 + I_{t+2})B_t E_t$$
$$I_{t=4} = F_{t+2}(1 + I_{t+3})B_{t+1}E_{t+1}$$
$$I_{t+5} = F_{t+3}(1 + I_{t+4})(1 + I_t + I_{t+1})$$

$$F_{t+2} = B_t E_t(1 + C_t + C_t F_t)$$
$$F_{t+3} = B_{t+1}E_{t+1}(1 + I_t + I_t F_{t+1})$$
$$F_{t+4} = (1 + I_t + I_{t+1})(1 + I_{t+1} + I_{t+1}F_{t+2}).$$

This gives two equations for F_t and I_t but we further notice that

$$I_{t+6} = (1 + I_t)(1 + I_{t+1})(1 + I_{t+2})(1 + I_{t+5}),$$

a single scalar equation for I_t. From the simple multiplicative symmetry of this equation it is seen that any initial binary sequence for I must lead to the sequence $(0, 0, 0, 0, 0, 0, 1, 0, 1)$ and then repeat. Furthermore, it is easy

to see that all nodes have the same periodicity as I_t and therefore the overall period nine cycle is

$$(0,0,1,1,1,0,1,0,0,1,1) \to (1,0,0,1,0,0,1,0,0,1,1) \to (1,1,0,1,1,0,1,0,0,1,1) \to$$
$$(1,1,0,1,1,0,0,0,0,1,1) \to (1,1,0,1,1,1,0,0,0,1,1) \to (1,1,0,1,1,1,0,1,0,1,1) \to$$
$$(0,1,0,1,1,1,0,1,1,1,1) \to (0,0,1,1,1,1,0,1,0,1,1) \to (0,0,0,1,1,1,1,1,1,1,1).$$

This agrees with the corrected form (Huang, 2002a) of the cycle found in Huang and Ingber (2000).

A reduced scalar equation can also be derived, proceeding in the following way

$$\begin{aligned} I_{t+6} &= (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+5}) \\ I_{t+7} &= (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+6}) \\ &= (1+I_{t+1})(1+I_{t+2})(1+I_{t+3}) + (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5}) \end{aligned}$$

(recall that $A^2 = A$ in mod 2 algebra)

$$\begin{aligned} I_{t+8} &= (1+I_{t+2})(1+I_{t+3})(1+I_{t+4}) + \\ &\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4}) + \\ &\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5}) \\ I_{t+9} &= (1+I_{t+3})(1+I_{t+4})(1+I_{t+5}) + \\ &\quad (1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5}) + \\ &\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5}) + \\ &\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+_{t+3})(1+I_{t+4})(1+I_{t+5}) \end{aligned}$$

$$
\begin{aligned}
I_{t+10} &= (1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})
\end{aligned}
$$

(recall that $A + A = 0$ in mod 2 algebra)

$$
\begin{aligned}
I_{t+11} &= (1+I_{t+5}) + (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+5})+ \\
&\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5})+ \\
&\quad (1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})
\end{aligned}
$$

$$
\begin{aligned}
I_{t+12} &= 1 + (1+I_{t+1})(1+I_{t+2})(1+I_{t+3}) + (1+I_{t+2})(1+I_{t+3})(1+I_{t+4})+ \\
&\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4}) + (1+I_{t+5})+ \\
&\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5})+ \\
&\quad (1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+4})(1+I_{t+5}) + (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+3})(1+I_{t+4})(1+I_{t+5})
\end{aligned}
$$

$$
\begin{aligned}
I_{t+13} &= (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+5}) + (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5})+ \\
&\quad (1+I_{t+4})(1+I_{t+5}) + (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+4})(1+I_{t+5})
\end{aligned}
$$

$$
\begin{aligned}
I_{t+14} &= (1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+1+ \\
&\quad (1+I_{t+2})(1+I_{t+3})(1+I_{t+4})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})+ \\
&\quad (1+I_{t+5})+(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5}) \\
I_{t+15} &= (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_{t+2})(1+I_{t+3})(1+I_{t+4})(1+I_{t+5})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+5})+ \\
&\quad (1+I_{t+2})(1+I_{t+3})(1+I_{t+4})+ \\
&\quad (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+4}) \\
I_{t+16} &= (1+I_{t+1})(1+I_{t+2})(1+I_{t+3})+ \\
&\quad (1+I_t)(1+I_{t+1})(1+I_{t+2})(1+I_{t+3})(1+I_{t+5}) \\
&\quad = I_{t+7}.
\end{aligned}
$$

The value of this reduced scalar equation is considerable. It says that not only is I_t (and hence every other node) on a cycle of period nine but that, starting from any initial condition, after at most seven transient states, the cycle is attained. Since there are $2^8 = 256$ initial states, this is a non-trivial conclusion. In fact, the initial trajectory (0,1,0,0,0,1) for I gives a transient of length seven. The reduced scalar equation means that iterating any randomly selected initial value 16 times, at most, will reveal the entire cycle structure.

We are currently working to create a computer algorithm to systematically find any and all possible types of scalar equations, nonlinear and reduced, of Boolean network equations. If an algorithm can be found, then the next question will be: Can it be implemented for "large" networks and how large do we really mean?

We also mention a paper (Poncet and Robert, 1999)which considers successive mappings of a whole Boolean network, not just a single node, to other, related networks. This mapping eventually returns the original network to itself while preserving fixed points at each step. The mapping has the simple form of a reduced scalar equation but its relation to the cycle structure of the original network is not clear.

ACKNOWLEDGEMENT

This research has been partially supported by NIH grant #1R016M672-01.

References

Adamatsky, A. (2003) On dynamical nontrivial three-valued logics: oscillatory and bifurcation species. *Chaos Solutions Fractals* **19**, 917-936.

Albert, R & Othmer, H. G. (2003) The topology and signature of the regulatory interactions predict the expression pattern of the segment polarity genes in Drosophila melanogaster, *J. Theor. Bio.* **223** (1), 1-18.

Boole, G. (1854) An investigation of the Laws of Thought. Walton and Maberly, London.

Cull, P. (1971), Linear analysis of switching nets. *Kybernetik* **8**(1), 31-39.

Darby, M. & Mysak, L. (1993) A Boolean delay equation model of an inter-decadal Arctic climate cycle. *Clim. Dyn.* **8**, 241-246.

DiPaolo, E. A. (2000) Searching for Rhythms in a Synchronous Random Boolean Network. Artificial Life VII: The Seventh International Conference on the Simulation and Synthesis of Living Systems, 1-6, August, 2000.

DiPaolo, E. A. (2001) Rhythmic and non-rhythmic attractors in asynchronous random Boolean networks. *BioSystems* **59** (2001), 185-195.

Farrow, C. (2002) *Reduction of Boolean Networks to Scalar Form.* Thesis, University of Nebraska at Omaha.

Goodwin B. C. *Temporal Organization of Cells.* Academic Press, New York.

Harvey I. & Bossomaier, T. (1997) Time out of joint: Attractors in asynchronous random Boolean networks. In: *Preceedings of the Fourth European Conference on Artificial Life* (Husbands, P. and Harvey, I. eds.) pp. 67–75, MIT Press, Cambridge.

Heidel, J., Maloney, J., Farrow, C. & Rogers, J. (2003) Finding cycles in synchronous Boolean networks with applications to biochemical systems. *Int. J. Bifurcat. Chaos.* **13**(3), 535-552.

Huang, S. (2001) Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation. *Pharmacogenomics* **2**, 203-222.

Huang, S. (2002a) private communication.

Huang, S. (2002b) Regulation of cellular states in mammalian cells from a genomewide view. In: *Gene Regulation and Metabolism* (Collado-Vides, J. & Hofestädt, R., eds.) pp. 181–220. MIT Press, Cambridge.

Huang, S. & Ingber, I. (2000) Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks. *Exp. Cell Res.* **261**, 91-103.

Kelley, W. & Peterson, A. (2001) *Difference Equations.* pp. 13–124. Harcourt, San Diego.

Kosko, B. (1997) *Fuzzy Engineering*, Prentice Hall, Upper Saddle River, New Jersey.

Milligan, D. K. & Wilson M. J. D. (1993) The behavior of affine Boolean sequential networks. *Connect. Sci.* **5**(2), 153-167.

Poncet. P. & Robert, F. (1999) About successive Gauss-Seidelizations. *Taiwanese J. Math.* **3**(4), 491-501.

Robert, F. (1986) *Discrete Iterations: A Metric Study.* pp. 10–11. Springer-Verlag, New York.

Robert, F. (1995) *Les Systèmes Dynamiques Discrets.* Springer, Berlin.

Thomas, R. & Kaufman, M. (2001) Multistationarity, the basis of cell differentiation and memory II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos* **11**(1), 180-195.

Warren-Smith, D. N. (2002) *Boolean Algebra Revisited*, Digital Logic Systems (http://users.senet.com/au/ dw-smith/boolean.htm).

Wilson, M. J. D. & Milligan, D. K. (1992) Cyclic behavior of autonomous, synchronous Boolean networks: some theorems and conjectures. *Connect. Sci.* **4**(2), 143-154.

Zurada, J. M. (1992) *Introduction to Artificial Neural Systems*, West, St. Paul.