Journal of Advances in Mathematics and Computer Science

35(6): 80-99, 2020; Article no.JAMCS.59926

ISSN: 2456-9968
(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

The Eigenvectors of the Transition Matrix as Predictors of the Dynamics of a Synchronous Boolean Network

Ali Muhammad Ali Rushdi[1*] and Adnan Ahmad Alsogati[1]

[1]Department of Electrical and Computer Engineering, King Abdulaziz University, P.O.Box 80200, Jeddah 21589, Saudi Arabia.

Authors' contributions

This work was carried out in collaboration between the two authors. Author AMAR envisioned and designed the study, performed the symbolic analysis, constructed the solutions, managed the literature search and wrote the entire manuscript. Author AAA performed the computational task, contributed to the literature search and drew the various figures. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2020/v35i630293
Editor(s):
(1) Dr. Zhenkun Huang, Jimei University (JMU), China.
Reviewers:
(1) Mukeba Kanyinda Jean Pierre, Institut Supérieur Pédagogique de Mbujimayi, Democratic Republic of the Congo.
(2) Waqas Anwar, Jiangsu University, China.
(3) Ranadheer Reddy Donthi, St. Martin's Engineering College (SMEC), India.
Complete Peer review History: http://www.sdiarticle4.com/review-history/59926

Received: 24 June 2020
Accepted: 29 August 2020
Published: 02 September 2020

Original Research Article

Abstract

The synchronous Boolean network model is a simple and powerful tool in describing, analyzing and simulating cellular biological networks. This paper seeks a complete understanding of the dynamics of such a model by utilizing conventional *matrix methods*, rather than scalar methods, or matrix methods employing the non-conventional semi-tensor products (STP) of matrices. The paper starts by relating the network *transition matrix* to its *function matrix* via a self-inverse (involutary) *state matrix*, which has a simple recursive expression, provided a *recursive ordering* is employed for the underlying basis vector. Once the network transition matrix is obtained, it can be used to generate a wealth of information including its powers, characteristic equation, minimal equation, 1-eigenvectors, and 0-eigenvectors. These might be used to correctly predict both the transient behavior and (more importantly) the cyclic behavior of the network. In a short-cut partial variant of the proposed approach, the step of computing the transition matrix might be by-passed. The reason for this is that the transition matrix and the function matrix are similar matrices that share the same characteristic equation and hence the function matrix might suffice when only the partial information supplied by the characteristic equation is all that is needed. We demonstrate the conceptual simplicity and practical utility of our approach via two illustrative examples. The first example illustrates the computation of 1-eigenvectors (that can be used to

**Corresponding author: E-mail: arushdi@kau.edu.sa, arushdi@yahoo.com;*

identify loops or attractors), while the second example deals with the evaluation of 0-eigenvectors (that can be used to explore transient chains). Since attractors are the main concern in the underlying model, then analysis of the Boolean network might be confined to the determination of 1-eigenvectors only.

Keywords: Synchronous Boolean networks; transition matrix; function matrix; characteristic equation; 1-eigenvectors; cyclic behavior; 0-eigenvectors; transient behavior; self-inverse (involutary) matrix.

1 Introduction

A synchronous Boolean network constitutes the simplest possible model that mimics or captures the essential features of general regulatory networks, gene expressions, and various biological systems comprising different types of cellular networks, proteins, DNA, and RNA [1-7]. Hence, this model became a powerful tool for describing, analyzing, and simulating these systems. Though the model is generally intended to capture coarse-grained dynamics of the biological system, it usually has a particular focus on features such as steady state behavior or limiting cycles, attractors or reverberations [8-13]. One advantage of Boolean models (and of similar discrete models) is that, for small models, the entire dynamics can be explored by exhaustive enumeration of all state transitions. Since the size of the state space of a Boolean model with n nodes is 2^n, this approach becomes unfeasible for larger networks, depending on the power of the computational resources available. In fact, the problem of finding, or even counting, steady states of Boolean networks is NP-hard [14-17], so that any algorithm for this problem will eventually encounter serious limitations.

A Synchronous Boolean network is a set of n interconnected nodes, each of which is either activated (in state 1, or On) or deactivated (in state 0, or Off) at any given instance of time t. The network operates on discrete (clock) time such that its state at a present-time point t uniquely determines its state at the next-time point $(t+1)$. Each node is updated at time $(t+1)$ by inputs from a fixed subset of the overall set of nodes (usually comprising neighboring nodes). The update takes place according to some specific logical rule, expressed as a set of next-state scalar functions. Since the number of network states is finite (2^n) and the network is a deterministic one, given an arbitrary initial state, the sequence of states succeeding this initial one must necessarily encounter a state it had previously occupied. Thereafter, the network must repeat the intermediate subsequence of states that it traversed from its earlier encounter with this repetitive state till its latter encounter with it. Such a subsequence of states is a cycle, and the number of distinct states in this cycle is its length. For any given network, some states (at least one state) are cyclic but others may be transient occurring during a run-in prior to cycling. For any given network, there may be different cycles of (possibly) varying lengths. If there are transient states, then some of them are first states (also called garden-of-Eden or isle-of-Eden states). Since a first state is a source state or a repelling state, it has no predecessor and does not serve as a next state for any current state. This means that if the network did not start initially in a first state, the network would never encounter that state at any time. In summary, all possible trajectories of the network consist of either cycles (loops or attractors) of any length from size one (a fixed point) to a maximum length of 2^n (when the whole trajectory is simply a single loop), or transient states leading eventually to a cycle.

A total description of a synchronous Boolean network is typically achieved by solving a matrix equation [18-20], or by algorithms in which a matrix equation is implicit [21,22]. Typically, matrix methods employed a novel matrix method utilizing a new matrix product, called the semi-tensor product (STP) of matrices [23-42]. A simpler description of a synchronous Boolean network is possible when the matrix equations of the network are replaced by scalar equations or reduced scalar equations [43,44]. Unfortunately, the scalar-equation technique suffers from several shortcomings and limitations [17,19,45-47]. Other candidate methods that might be employed in the analysis of synchronous Boolean networks include solution of Diophantine equations [17,46-51], solution of the Boolean satisfiability problem (SAT) [22,52-57], solution of Boolean equations [58-71], and integer linear programming [22].

The purpose of this paper is to derive the transition matrix $[T]$ of the synchronous Boolean network, independently from any STP formulations or derivations. This matrix is then analyzed to deduce full information about the transient behavior and (more importantly) the cyclic behavior of the network. In fact, subtle inferences about the network behavior are made without drawing the state diagram by exploring some mathematical properties of $[T]$, namely its characteristic equation, minimal equation, 1-eigenvectors and 0-eigenvectors.

The organization of the remainder of this paper is as follows. Section 2 presents our algorithm for computing the transition matrix $[T]$ of a synchronous Boolean network, while Section 3 explains how $[T]$ can be used to deduce the cyclic and transient behavior of the network without resort to a full construction of the network state diagram. Section 4 and 5 amplify our results, by presenting two illustrative examples of synchronous Boolean networks previously studied in the literature. Both Sections illustrate the mathematical techniques employed in deriving necessary related entities such as the characteristic equation, minimal equation, 1-eigenvectors, and 0-eigenvectors. The example of Section 4 deals mainly with the computation of 1-eigenvectors, as a prelude to predicting cyclic behavior. By contrast, the example of Section 5 deals mainly with the computation of 0-eigenvectors, as a step towards exploring transient behavior. Our matrix approach for computing the network transition matrix is superior to scalar techniques which demand cumbersome manipulations and might fail to predict the exact network behavior. In addition, our algorithm replicates in clear, visible and succinct steps the results obtained by the STP methodology. Our transition-matrix predications of network transient and cyclic behavior are also in full agreement with those obtained via a full construction of the network state diagram. In Section 6, we digress a little bit to make a brief discussion, while Section 7 concludes the paper.

2 Construction of the Transition Matrix

This section outlines an algorithm for constructing the transition matrix $[T]$. To understand the mathematical structure of this matrix, we first note that functions used in this paper are functions over the simplest finite or Galois field GF(2), also known as the binary field or mod-2 field [18,19,72,73]. The field has only two elements: the additive identity (0) and the multiplicative identity (1). The field addition $(+)$ and multiplication $(\times)$ operations are defined by the following axioms:

$$0 + 0 = 1 + 1 = 0, \tag{1a}$$

$$1 + 0 = 0 + 1 = 1, \tag{1b}$$

$$0 \times 0 = 0 \times 1 = 1 \times 0 = 0, \tag{1c}$$

$$1 \times 1 = 1. \tag{1d}$$

Note that the addition $(+)$ operation is a modulo-2 operation that resembles the *exclusive-OR* operation $(\oplus)$ in switching algebra (two-valued Boolean algebra) [74-78]. Any function of n variables over GF(2) is a polynomial of 2^n terms (Reed-Müller polynomial [19,58,72-74]), and hence can be represented by a vector of length 2^n, whose elements are the binary coefficients of the 2^n terms in the polynomial. This representation is also called a linear or Boolean-ring representation [58]. In the sequel, we will use two isomorphic representations (the one over GF(2) [72] and that of the Boolean ring [58]), interchangeably, and in particular our $(+)$ sign is equivalently understood to mean modulo-2 addition or *XOR* operation ($1a$ and $1b$), and our $(\times)$ sign (to be omitted and replaced simply by juxtapositioning) is equivalently understood to mean 1-bit multiplication or *AND* operation ($1c$ and $1d$).

The binary field GF(2) shares many of the familiar properties of other fields such as those of the rational numbers and real numbers, including the existence of an additive identity (0) and a multiplicative identity (1), the existence of an additive inverse for every element and a multiplicative inverse for every element except 0, that addition and multiplication are commutative and associative and that multiplication is distributive over addition. However, GF(2) possesses features that lack counterparts in the rational and real number systems, including two properties that can be deduced from (1) for every element x of GF(2), namely (a) additive annihilation ($x + x = 0$), and (b) multiplicative idempotency ($x \times x = x$). An important consequence of these two properties is that in GF(2) a vector with an even number of 1-elements is self-orthogonal. We stress that the binary matrices to be considered herein are called GF(2)-matrices and not called Boolean matrices. Brad [79] makes a clear distinction between the two types of matrices denoted by these two names. He bases this distinction on the notion that element multiplication is basically an exclusive-OR operation for GF(2)-matrices, while it is an inclusive-OR operation for Boolean matrices.

In the sequel, we employ a recursively-ordered vector of 2^n elements as a basis for all functions of n variables over GF(2) or equivalently all switching functions of n variables (viewed over the Boolean ring), in the sense that any of these functions equals the (scalar) dot product of the vector of binary coefficients representing it with the afore-mentioned basis vector. This basis vector is given explicitly in [19] as

$$\vec{B}_n = [1 \quad x_1 \quad x_2 \quad x_1x_2 \quad x_3 \quad x_1x_3 \quad x_2x_3$$

$$x_1x_2x_3 \quad \dots \quad x_n \quad x_1x_n \quad x_2x_n \quad x_1x_2x_n \dots \quad x_1x_2x_3 \dots x_n]^T. \tag{2}$$

Equivalently, it can be defined by the simple recursive relation:

$$\vec{B}_n = \begin{bmatrix} \vec{B}_{n-1} \\ \\ \vec{B}_{n-1} \wedge x_n \end{bmatrix}, \tag{3}$$

together with the boundary condition:

$$\vec{B}_0 = [1]. \tag{4}$$

Next, we define the exact state vector $\vec{Y}(t)$ as a binary column vector of dimension 2^n, whose elements are all 0 except one element that is of value 1. This particular element of value 1 corresponds to the position where the product depicting the network state occurs in the basis vector $\vec{B}_n$. We also define the transition matrix $[T]$ as a $2^n \times 2^n$ permutation matrix having exactly a single 1 in each column and with 0 elements otherwise. The transition matrix may be projective, i.e., there may be more than one element 1 in any row, and (hence) as many rows with all 0's. This transition matrix represents the state transition diagram of the network. The particular structure of $[T]$ asserts that the network is in only one state at a time and proceeds to a unique next state in the next instant of time. The transition matrix $[T]$ relates the exact next-state vector $\vec{Y}(t + 1)$ to the exact present-state one via [18,19]:

$$\vec{Y}(t + 1) = [T]\,\vec{Y}(t). \tag{5}$$

Equation (5) is a restatement of the fact that element t_{ij} in row i and column j of $[T]$ is 1 if and only if present state j leads to next state i (state i is successor to state j), and 0 otherwise. We also define the function matrix $[A]$ as a $2^n \times 2^n$ matrix that has as its rows the vector representations of the 2^n products of the n scalar next-state functions, taken k at a time $(0 \leq k \leq n)$. The transition matrix $[T]$ is related to the function matrix $[A]$ via the following similarity transformation, which involves the state matrix $[S_n]$ [18,19]:

$$[A][S_n] = [S_n][T], \tag{6}$$

$$[A] = [S_n][T][S_n]^{-1} = [S_n][T][S_n], \tag{7}$$

$$[T] = [S_n]^{-1}[A][S_n] = [S_n][A][S_n], \tag{8}$$

where the State Matrix $[S_n]$ is a $2^n \times 2^n$ upper triangular binary matrix. The recursive ordering of the basis $\vec{B}_n$ introduced in [19] allows the matrix $[S_n]$ to have the recursive definition [19]:

$$[S_0] = [1], \tag{9a}$$

$$[S_n] = \begin{bmatrix} [S_{n-1}] & [S_{n-1}] \\ [0_{n-1}] & [S_{n-1}] \end{bmatrix}, \qquad n > 0, \tag{9b}$$

where $[0_{n-1}]$ is the zero matrix of dimensions $2^{n-1} \times 2^{n-1}$. This recursive definition of $[S_n]$ is reminiscent of the Preparata Transformation between the linear (Reed-Müller) representation and the minterm (truth-table) representation of a switching function [80-84]. Matrix $[S_n]$ is an involutory (self-inverse) matrix, *i.e.*,

$$[S_n]^{-1} = [S_n], \tag{10}$$

$$[S_n]^2 = [S_n][S_n] = [S_n][S_n]^{-1} = [I_n], \tag{11}$$

where $[I_n]$ is the identity matrix of $2^n \times 2^n$ elements. As a check, the resulting $[T]$ should have a single element of value 1 in every column, with the remaining elements being all $0's$. The column vectors of $[T]$ constitute a subset (with possibly repetitive elements) of a set forming an ortho-normal basis of the 2^n-dimensional vector space. For the case $[S_n]$, the matrix $[S_3]$ is given by:

$$[S_3] = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \end{bmatrix}. \tag{12}$$

This section is summarized in Fig. 1, which presents a flow chart for the computation of the transition matrix of a synchronous Boolean network, given the scalar next-state functions of its nodes.

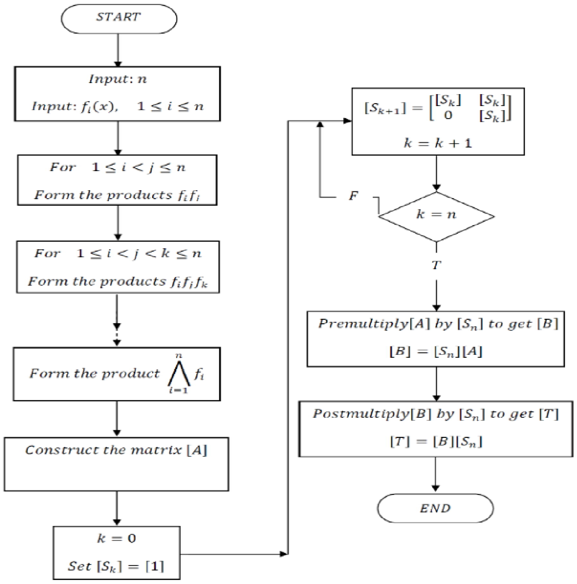

START
Input: n
Input: $f_i(x)$, $1 \le i \le n$
For $1 \le i < j \le n$
Form the products $f_i f_i$
For $1 \le i < j < k \le n$
Form the products $f_i f_j f_k$
Form the product $\bigwedge_{i=1}^{n} f_i$
Construct the matrix $[A]$
$k = 0$
Set $[S_k] = [1]$
$[S_{k+1}] = \begin{bmatrix} [S_k] & [S_k] \\ 0 & [S_k] \end{bmatrix}$
$k = k + 1$
F
$k = n$
T
Premultiply$[A]$ by $[S_n]$ to get $[B]$
$[B] = [S_n][A]$
Postmultiply$[B]$ by $[S_n]$ to get $[T]$
$[T] = [B][S_n]$
END

Fig. 1. A flow-chart for the construction of the matrix $[T]$

3 Utilization of the Transition Matrix in Predicting Network Dynamics

This section demonstrates how the transition matrix $[T]$ could be utilized in predicting network behavior via its powers, characteristic equation, minimal equation, 1-eigenvectors and 0-eigenvectors. A state of the network $\vec{Y}(t)$ is called cyclic if [18,19]:

$$\vec{Y}(t+m) = [T]^m\, \vec{Y}(t) = \vec{Y}(t), \quad for\ some\ m > 0. \tag{13a}$$

Otherwise it is called transient when:

$$\vec{Y}(t+m) = [T]^m\, \vec{Y}(t) \neq \vec{Y}(t), \quad for\ all\ m > 0. \tag{13b}$$

A set of cyclic states that map into one another are called a cycle. In particular, the k-element set defined by $C = \{\vec{Y}(t), [T]\, \vec{Y}(t), \cdots, [T]^{k-1}\, \vec{Y}(t)\}$ is a cycle of length k if $[T]^k\, \vec{Y}(t) = \vec{Y}(t)$ and the elements of C are distinct [28, p. 108]. For $k = 1$, the set C is the singleton $\{\, \vec{Y}(t)\}$, and the cycle becomes of length 1, *i.e.*, a fixed point. A cycle is therefore definable as a $[T]$-invariant subspace [28,85,86], since the matrix $[T]$ maps C to itself ($[T]C = C$). A state into which no state maps is called a first state (a garden-of-Eden or isle-of-Eden state), *i.e.*, a first state has no predecessor and does not serve as a next state for any current state, and hence the row corresponding to it in the transition matrix $[T]$ is an all-0 row. A set of transient states that can be reached from a certain first state are called a transient chain. Note that a state is transient if it is a first state or it can be reached only from a first state or from some other transient states.

By analogy to characteristic equations of real matrices [87,88], Cull [18] introduced the concept of the binary characteristic equation of a binary matrix $[T]$. He defined it as $det([T] - \lambda[I]) = 0$, where the operations are carried out in the binary field, so that subtraction $(-)$ and addition $(+)$ are indistinguishable. In fact, subtraction is not even defined on GF(2). Hence, the characteristic equation becomes $det([T] + \lambda[I]) = 0$. This equation will take the form [18,19]:

$$det([T] + \lambda[I]) = \lambda^m(\lambda^{r_1} + 1) \cdots (\lambda^{r_i} + 1) = 0, \tag{14}$$

where m is the total number of transient states, while the subscripted $r's$ are the lengths of the various cycles. We note that the eigenvalues of the matrix $[T]$ are the roots of its characteristic equation (14). In GF(2), the only possible eigenvalues are $\lambda = 1$ (corresponding to one-eigenvectors that represent cycles) and $\lambda = 0$ (corresponding to zero-eigenvectors that represent transient chains). The factorization of the binary characteristic equation is not always unique [18,19]. In fact,

$$(\lambda^r + 1) = (\lambda + 1)^r, \tag{15}$$

for r being a power of 2, so that one cannot specify all cycles exactly by specifying the characteristic equation [3]. Now, we observe that since matrices $[A]$ and $[T]$ are similar, they have the same characteristic equation [18,19]:

$$det([A] + \lambda[I]) = det([T] + \lambda[I]) = 0 \tag{16}$$

One reason why we prefer to transform to the matrix $[A]$ to $[T]$ is that the task of determining the characteristic polynomial of a permutation matrix such as $[T]$ is particularly simple, since it has a single non-zero entry per column. By contrast, the determination of the characteristic polynomial of $[A]$ is somewhat cumbersome since many entries of $[A]$ are non-zero. The Cayley-Hamilton theorem (which states in the continuous case that any matrix satisfies its own characteristic equation) is still applicable in this finite case, *i.e.*,

$$[T]^m([T]^{r_1} + [I]) \cdots ([T]^{r_i} + [I]) = [0], \tag{17}$$

$$[A]^m([T]^{r_1}+[I])\cdots([T]^{r_i}+[I])=[0]. \tag{18}$$

Cull [18] asserted that there is an equation of lowest degree that $[A]$ satisfies; which is the minimal equation [89-92] of $[A]$, namely:

$$[A]^{r_0}([A]^{r_h}+[I])\cdots([A]^{r_g}+[I])=[0]. \tag{19}$$

Here, r_0 is the length of the longest transient chain and $r_h,\cdots,r_g$ are lengths of distinct cycles such that all other cycle lengths divide one of the $r's$ with no remainder [18].

The cyclic behavior of a synchronous Boolean network can also be obtained from one-eigenvectors of the function matrix $[A]$ or the one-eigenvectors of the transition matrix $[T]$. In fact, the cycles of the network are linearly independent one-eigenvectors of $[T]$ and the one-eigenvectors of $[T]$ are cycles or the union of cycles [18,19].

Similarly, the zero-eigenvectors can be used in the study of the transient chains [18,19]. There are as many independent 0-eigenvectors as there are chains [18,19]. Each of these eigenvectors is of the form $\vec{Y_i}+\vec{Y_j}$, where $\vec{Y_i}\neq\vec{Y_j}$. At least one of the states $\vec{Y_i}$ and $\vec{Y_j}$ is the last state of a transient chain, *i.e.*, a state that is not on a cycle, but whose next state is on a cycle [19].

Gelfand [4] observed that the trace (usual arithmetic sum of diagonal elements) of $[T]^m$ is equal to the number of states on cycles of length k which is a divisor of the integer m. Hence, the number of fixed points is equal to the trace of $[T]$ itself. If we use N to denote 2^n , then $Tr(T^{N!})$ equals the number of states on cycles and $(n-Tr(T^{N!}))$ equals the number of transient states, where $N!$ denotes the factorial of $N=2^n$.

4 Example Dealing with 1-eigenvctors

This example was analyzed earlier by Cull [18] and Rushdi and Al-Otaibi [19]. Consider a network of 3 elements whose next-states are given by:

$$f_1=1+x_2+x_1x_2+x_1x_3+x_1x_2x_3, \tag{20a}$$

$$f_2=1+x_2+x_1x_2+x_3+x_1x_3+x_1x_2x_3, \tag{20b}$$

$$f_3=1+x_1+x_1x_2+x_1x_3+x_2x_3+x_1x_2x_3. \tag{20c}$$

Now, the products of these functions (two at a time and three (all) at a time) are:

$$f_1f_2=1+x_2+x_1x_2+x_3+x_2x_3+x_1x_2x_3, \tag{20d}$$

$$f_1f_3=1+x_1+x_2+x_1x_2x_3, \tag{20e}$$

$$f_2f_3=1+x_1+x_2+x_3+x_2x_3+x_1x_2x_3, \tag{20f}$$

$$f_1f_2f_3=1+x_1+x_2+x_3+x_1x_3+x_2x_3. \tag{20g}$$

Noting that the product of zero functions (conveniently called f_0) is the multiplication identity (1), we construct the function matrix $[A]$,

$$[A] = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \end{bmatrix}. \qquad (21)$$

In (21), we implicitly understand that the bases indexing the columns and rows of $[A]$ follow the recursive ordering of Rushdi & Al-Otaibi [19], namely

$$\vec{B}_{3(x)} = [1 \quad x_1 \quad x_2 \quad x_1x_2 \quad x_3 \quad x_1x_3 \quad x_2x_3 \quad x_1x_2x_3]^T, \qquad (22)$$

$$\vec{B}_{3(f)} = [1 \quad f_1 \quad f_2 \quad f_1f_2 \quad f_3 \quad f_1f_3 \quad f_2f_3 \quad f_1f_2f_3]^T, \qquad (23)$$

To understand the construction of the matrix $[A]$, let us consider how we form the entries in one of its rows, say the sixth row (numbered row 5, since numbering starts with 0). This row represents the output product f_1f_3 in (20e), which comprises four input products 1, x_1, x_2, and $x_1x_2x_3$. In this row, we place 1 in columns 0, 1, 2, and 7, i.e., beneath each of these input products, and place 0 in the remaining columns. The state matrix $[S_3]$ needed for computing the matrix $[T]$ is given by (12). Hence, the transition matrix $[T]$ is computed via its similarity relation to the function matrix $[A]$ as:

$$[T] = [S_3]\,[A]\,[S_3] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \end{bmatrix} = \vec{\delta}_8\,[7,\ 3,\ 4,\ 7,\ 5,\ 6,\ 2,\ 1]. \qquad (24)$$

The result in (24) is self-checking, since each matrix column is a unit of all-0 entries with exactly a single exception of a unique 1 element. The abbreviation $\vec{\delta}_8$ in (24) is a convenient way of representing the transition matrix by specifying the row number (from 0 to 7) in which the sole 1 (the only non-zero value) in a column is located. The characteristic equation for $[T]$ (or for $[A]$) is:

$$det([T] + \lambda[I]) = det([A] + \lambda[I]) = \lambda(\lambda^4 + 1)(\lambda^3 + 1). \qquad (25)$$

In this case the minimal equation is identical to the characteristic equation, *i.e.*, it is

$$[T]([T]^4 + [I])([T]^3 + [I]) = [0]. \qquad (26)$$

These facts tell us that the system has a transient chain of length 1, a cycle of length 4, and another cycle of length 3. We would have obtained these results by computing $[A]$ only without computing $[T]$, since the two matrices $[T]$ and $[A]$ are similar and share the same characteristic equation. This possibility hints at the existence of a short-cut partial variant of our proposed approach, in which the step of computing the transition matrix might be skipped. The function matrix might suffice when only the partial information supplied by the characteristic equation is all that is needed. The reason we opted to work with the transition matrix is that it provides a more efficient means for computing the characteristic equation, and it supplies extra information via its powers.

Now, we let $\vec{Y} = [y_0 \quad y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7]^T$ be a 1-eigenvector of $[T]$, *i.e.*, $[T]\vec{Y} = (1)\vec{Y}$, then: $y_0 = 0,\ y_1 = y_7,\ y_2 = y_6,\ y_3 = y_1,\ y_4 = y_2,\ y_5 = y_4,\ y_6 = y_5,\ y_7 = y_0 + y_3 = y_3$ which combine to give $y_0 = 0,\ y_1 = y_3 = y_7 = \alpha,\ y_2 = y_4 = y_5 = y_6 = \beta$. Hence, the generalized eigenvector of $[T]$ is:

$$\vec{Y} = [0 \quad \alpha \quad \beta \quad \alpha \quad \beta \quad \beta \quad \beta \quad \alpha]^T. \tag{27}$$

Table 1 shows all four possible values for $\vec{Y}$, out of which only two $(\vec{Y}_b\ and\ \vec{Y}_c)$ are linearly independent. The first linearly-independent 1-eigenvector $\vec{Y}_b$ identifies a cycle of period 4 with cycle states $2,\ 4,\ 5,\ and\ 6$ (referenced in the basis $\vec{B}_3$). These are states $x_2,\ x_3,\ x_1x_3,\ and\ x_2x_3$, or equivalently $(010), (001), (101),\ and\ (011)$ when identified by values of 3-tuple $(x_1x_2x_3)$. The second linearly-independent 1-eigenvector $\vec{Y}_c$ identifies another cycle of period 3 with cycle states $1,\ 3,\ and\ 7$ (again referenced in the basis $\vec{B}_3$). These are states $x_1,\ x_1x_2,\ and\ x_1x_2x_3$, or equivalently $(100), (110),\ and\ (111)$ again when representing $(x_1x_2x_3)$ values. Out of the 8 network states, 7 states are on cycles, and the remaining single state (000) is a transient state.

To finalize this example, we verify our findings via the Karnaugh-map representation of the three next-state functions $f_1, f_2, and\ f_3$ in Fig. 2, from which we draw the full state-diagram of the network in Fig. 3. In passing, we remark that we found it more convenient to study this example via 1-eigenvectors rather than 0-eigenvectors since most of its states are cycle states rather than transient ones.

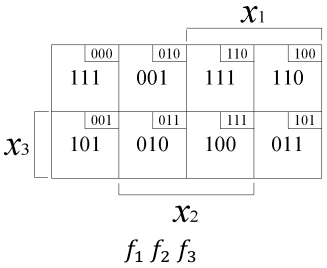


Fig. 2. A Karnaugh-map representation for the next-state functions $\{f_1\ f_2\ f_3\}$ of the network in the example of Sec. 3

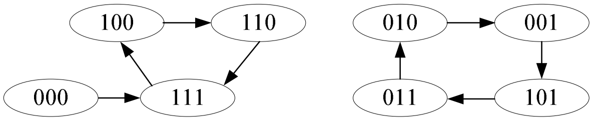


Fig. 3. The state diagram for the example of Sec. 3 with states designated by the binary values of the 3-tuple $\{x_1x_2x_3\}$.

Table 1. Possible values of the generalized 1-eigenvectors in the example of sec. 3.

α	β	$\vec{Y}$	Comment
0	0	$\vec{Y}_a = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$	trivial
0	1	$\vec{Y}_b = [0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0]^T$	independent
1	0	$\vec{Y}_c = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1]^T$	independent
1	1	$\vec{Y}_d = [0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]^T$	$\vec{Y}_d = \vec{Y}_b + \vec{Y}_c$

5 Example Dealing with 0-Eigenvectors

This example was analyzed earlier by Heidel et al. [43] and Rushdi and Al-Otaibi [19]. Consider a network of 3 elements whose next-states are given by:

$$f_1 = x_2 \vee x_3 = x_2 + x_3 + x_2 x_3, \tag{28a}$$

$$f_2 = x_1 \vee x_3 = x_1 + x_3 + x_1 x_3, \tag{28b}$$

$$f_3 = x_1 \vee x_2 = x_1 + x_2 + x_1 x_2. \tag{28c}$$

Now, the products of these functions are:

$$f_1 f_2 = x_1 x_2 + x_3 + x_1 x_2 x_3, \tag{28d}$$

$$f_1 f_3 = x_2 + x_1 x_3 + x_1 x_2 x_3, \tag{28e}$$

$$f_2 f_3 = x_1 + x_2 x_3 + x_1 x_2 x_3, \tag{28f}$$

$$f_1 f_2 f_3 = x_1 x_2 + x_1 x_3 + x_1 x_2 x_3. \tag{28g}$$

Then, we construct the function matrix $[A]$ using the strategy of the previous section:

$$[A] = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} \\ 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} \end{bmatrix}. \tag{29}$$

The transition matrix $[T]$ is given by:

$$[T] = [S_3][A][S_3] = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix} = \vec{\delta}_8[0,\ 6,\ 5,\ 7,\ 3,\ 7,\ 7,\ 7]. \tag{30}$$

This transition matrix $[T]$ has two non-0 diagonal elements, corresponding to fixed points at states 0 and 7 (expressed as (000) and (111)). Now, the two powers $[T]^2$ and $[T]^4$ are given by:

$$[T]^2 = [T]^4 = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \end{bmatrix} = \vec{\delta}_8[0,\ 7,\ 7,\ 7,\ 7,\ 7,\ 7,\ 7]. \tag{31}$$

Equation (31) says that the longest transient trajectory is of length 2. After two time periods all network states (except state 0) are attracted to state 7 (111). The characteristic equation of $[T]$ is:

$$det([T]+\lambda[I]) = det([A]+\lambda[I]) = \lambda^6(\lambda+1)^2, \tag{32}$$

and hence the system has 6 transient states and two cycles of length 1. The factor $(\lambda+1)^2$ should not be replaced by (λ^2+1) which represents a single cycle of length 2 (see Equation (15)). Here, the minimal equation is not identical to the characteristic equation, but is given by:

$$[A]^2([A]+[I])^2 = [0], \tag{33}$$

which means that the longest transient chain is of length 2. In fact, equations (7), (8), (11) and (31) can be used to obtain $[A]^4+[A]^2=[0]$, which can be recast in the form (33) since $([A]^2+[I])$ must be replaced by its equivalent form $([A]+[I])^2$, as required by the characteristic equation.

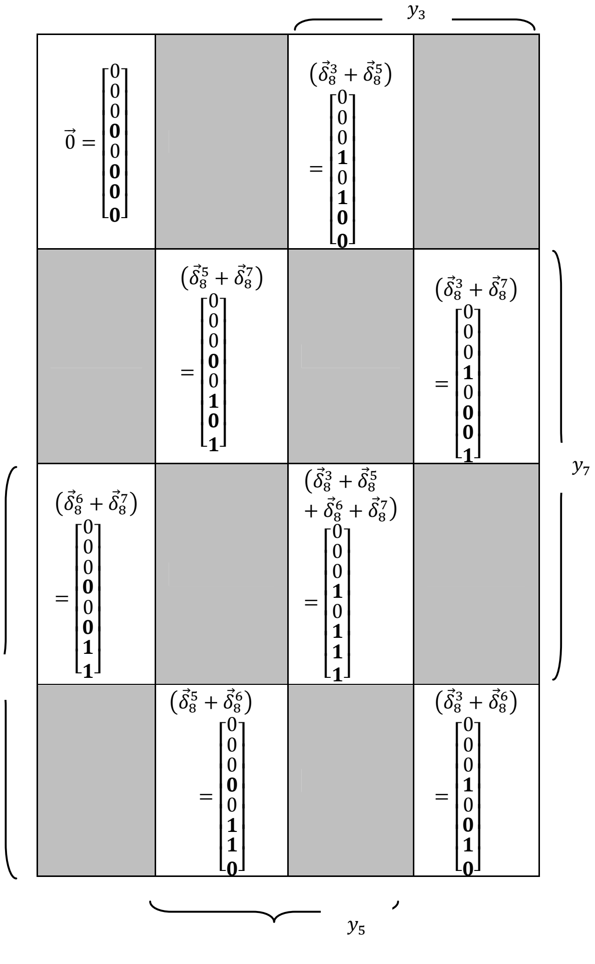

y_3
$\vec{0} = [0\ 0\ 0\ \mathbf{0}\ 0\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}]^T$
$(\vec{\delta}_8^3+\vec{\delta}_8^5) = [0\ 0\ 0\ \mathbf{1}\ 0\ \mathbf{1}\ \mathbf{0}\ \mathbf{0}]^T$
$(\vec{\delta}_8^5+\vec{\delta}_8^7) = [0\ 0\ 0\ \mathbf{0}\ 0\ \mathbf{1}\ \mathbf{0}\ \mathbf{1}]^T$
$(\vec{\delta}_8^3+\vec{\delta}_8^7) = [0\ 0\ 0\ \mathbf{1}\ 0\ \mathbf{0}\ \mathbf{0}\ \mathbf{1}]^T$
y_7
$(\vec{\delta}_8^6+\vec{\delta}_8^7) = [0\ 0\ 0\ \mathbf{0}\ 0\ \mathbf{0}\ \mathbf{1}\ \mathbf{1}]^T$
$(\vec{\delta}_8^3+\vec{\delta}_8^5+\vec{\delta}_8^6+\vec{\delta}_8^7) = [0\ 0\ 0\ \mathbf{1}\ 0\ \mathbf{1}\ \mathbf{1}\ \mathbf{1}]^T$
y_6
$(\vec{\delta}_8^5+\vec{\delta}_8^6) = [0\ 0\ 0\ \mathbf{0}\ 0\ \mathbf{1}\ \mathbf{1}\ \mathbf{0}]^T$
$(\vec{\delta}_8^3+\vec{\delta}_8^6) = [0\ 0\ 0\ \mathbf{1}\ 0\ \mathbf{0}\ \mathbf{1}\ \mathbf{0}]^T$
y_5

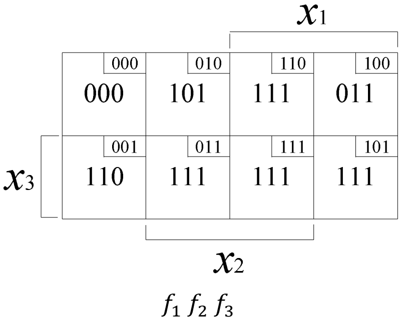
Fig. 4. Possible values for the 0-eigenvectors for the example of Sec. 4.

To study the transient behavior of the network, we find all the independent 0-eigenvectors of the network by solving $[T]\vec{Y}=\vec{0}$. This produces three identities $(0=0)$ plus the five constraints:

$$y_0 = y_1 = y_2 = y_4 = 0, \qquad y_3+y_5+y_6+y_7 = 0, \tag{34}$$

which reduces the eigenspace to a 3-dimensional one. This means that there are eight 0-eigenvectors of $[T]$, as shown in the Karnaugh-map of Fig. 4, where cells violating the even parity condition on y_3, y_5, $y_6, and\ y_7$ are rejected. Out of these eigenvectors only three are linearly independent, *e.g.*, the vectors $(\vec{\delta}_8^3 + \vec{\delta}_8^7), (\vec{\delta}_8^5 + \vec{\delta}_8^7), and\ (\vec{\delta}_8^6 + \vec{\delta}_8^7)$. This means that there are three transient chains in the network state diagram. In fact, we can identify the start states for these chains as the ones corresponding to all-0 rows in $[T]$. These are states $1\ (100),\ 2\ (010),\ and\ 4\ (001)$. Our knowledge about the linearly-independent 0-eigenvectors indicates that each of the four states $3,\ 5,\ 6,\ and\ 7$ is either the last state of a transient chain or on a cycle. Since we have three transient chains, then we know that three of the states $3,\ 5,\ 6,\ and\ 7$ are the last states of transients, with the fourth state being on a cycle. With our earlier observation that states $0\ and\ 7$ are 1-period cycles (fixed points), we realize that states $3,\ 5,\ and\ 6$ are transient end states.

To finalize this example, we verify our findings via the Karnaugh-map representation of the three next-state functions $f_1, f_2, and\ f_3$ in Fig. 5, from which we draw the full state-diagram of the network in Fig. 6.



x_1

000 000	010 101	110 111	100 011
001 110	011 111	111 111	101 111

x_3

x_2

$f_1\, f_2\, f_3$

Fig. 5. A Karnaugh-map representation for the next-state functions $\{f_1\, f_2\, f_3\}$ of the network for the example of Sec. 4

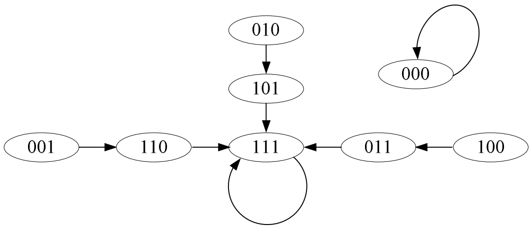


Fig. 6. The state diagram for the example of Sec. 4 with states designated by the binary values of the 3-tuple $\{x_1 x_2 x_3\}$

6 Discussion

Boolean network models are one of the simplest models to study complex dynamic behavior in biological systems. There are three major paradigms (synchronous, asynchronous and probabilistic) for the mode of transition of a Boolean network from a present state to a succeeding one [93]. We have only considered the synchronous paradigm, in which all nodes of the network take an equivalent amount of time (the clock period) to change their value. Consequently, the dynamics of the Boolean network are deterministic, and each state of the network has one successor. Our main concern herein was the study of these networks using the eigenvectors of the transition matrix. Some of our concepts translate to other paradigms of Boolean networks with other types of matrices.

Real networks exhibit heterogeneous nature with nodes playing different roles in structure and function. A non-trivial task is to identify vital nodes associated with some certain structural or functional objectives [94,95]. The concept of 'centrality' is forwarded to characterize the importance of a node according the network structure [96-99]. Generally speaking, a centrality measure assigns a real value to each node in the network, where the values produced are expected to provide a ranking of nodes subject to their importance. An interesting measure of centrality is that of eigenvector centrality [99,100,101], which has advantages over graph-theoretic centrality measures like degree, betweenness, and closeness centrality [99]. Eigenvector centrality supposes that the influence of a node is not only determined by the number of its neighbors, but also determined by the influence of each neighbor [97]. The centrality of a node is proportional to the summation of the centralities of the nodes to which it is connected [94]. We are particularly interested in the case where eigenvector centrality pertains to Boolean networks (see, e.g., [102,103]). Ghanbarnejad and Klemm [102] consider synchronous networks with deterministic update and asynchronous networks with stochastic update. They quantify the dynamical impact of a node as the probability of damage spreading after switching the node's state. They assert that the leading or principal eigenvector of the adjacency matrix is a good predictor of dynamical impact in the case of long-term spreading. This eigenvector centrality measure is a good predictor of how a node's initial state perturbs the attractor the system eventually arrives at. The quality of prediction is further improved when eigenvector centrality is based on the weighted matrix of activities rather than on the unweighted adjacency matrix.

7 Conclusions

To analyze a synchronous Boolean network, we start from the scalar next-state equations of the network so as to write its function matrix $[A]$. Next, we obtain the transition matrix $[T]$ via its similarity relation with $[A]$, a step which involves pre-multiplying and post-multiplying $[A]$ with the state matrix $[S_n]$. This step is self-checking, since it must produce a matrix $[T]$ whose column vectors are unit vectors. We utilize certain properties of $[T]$ to make useful deductions about network behavior without drawing the state diagram. We use the characteristic and minimal equations for $[T]$, its 1-eigenvectors, and 0-eigenvectors as advocated by Cull [18] and later by Rushdi and Al- Otaibi [19], as well as by Li et al. [10]. The predictions made by the various methods are self-consistent and are in full agreement with results obtained earlier, notably via the STP machinery. These predictions also coincide exactly with those deduced from the full state diagram. We note that there is a relation between our current matrix method and the method of describing the network via reduced scalar equations [43,44,47]. Contrary to earlier assumptions and assertions that appeared in [43,44], the reduced scalar equations are not necessarily the same for all variables, and the true or minimal scalar equation (for the worst-case variable) must be sought in order to give a correct information about the transient behavior of the network [46,47]. We stress that the method reported herein is, indeed, superior to the scalar techniques in [43,44]. In fact, scalar techniques use ad-hoc cumbersome heuristics, and might fail to predict the exact network behavior, especially its transient one [46,47]. Cheng et al. [28] reported a disparity in the assessment of the maximum length of a transient chain as obtained by their matrix method, and the scalar methods of [43,44]. Since the matrix method is a complete one, there was no doubt that its assessment was the correct one. Intrigued by the reason of fallibility of the scalar approach, Rushdi & Alsogati [46] and Rushdi [47] took pains to correct the way the scalar approach is implemented, so as to reconcile its output with that of matrix methods.

The current paper is a serious attempt to make the matrix analysis of synchronous Boolean networks independent of any particular implementation, such as that of the semi-tensor products (STP) of matrices. We produce results that exactly replicate those of the STP approach, but achieve that without sophisticated ambiguity or unwarranted redundancy. We stress that we perform binary (Boolean) computations on GF(2), while usual STP computations involve real numbers and real operations. However, we must admit that the STP methodology comprises an ingenious paradigm shift that might considerably mitigate the 'curse of dimensionality' in a variety of fields, including the present one of synchronous Boolean networks. All we are asking for, is that when STP handles logic or Boolean expressions, then it should be constrained to the use of Boolean operations.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] MeCulloch WS, Pitts WH. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 1943;5:115–133.

[2] Kauffman SA. Metabolic stability and Epigenesis in randomly connected nets. Journal of Theoretical Biology. 1969;22:437-467.

[3] Caianiello ER, Grimson WEL. Methods of analysis of neural nets. Biological Cybernetics. 1976;22(1):1-6.

[4] Gelfand AE. A behavioral summary for completely random nets. Bulletin of Mathematical Biology. 1982;44(3):309-320.

[5] Milligan DK, Wilson MJD. Fundamental structure/behaviour relationships in synchronous Boolean neural networks. In International Neural Network Conference Springer, Dordrecht. 1990;997-1000.

[6] Müssel C, Hopfensitz M, Kestler HA. BoolNet—an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics. 2010;26(10):1378-1380.

[7] Kochemazov S, Semenov A. Using synchronous Boolean networks to model several phenomena of collective behavior. PloS One. 2014;9(12):e115156.

[8] Wilson MJD, Milligan DK. Cyclic behavior of autonomous, synchronous Boolean networks: Some theorems and conjectures. Connection Science. 1992;4(2):143-154.

[9] Veliz-Cuba A, Laubenbacher R. On the computation of fixed points in Boolean networks. Journal of Applied Mathematics and Computing. 2012;39(1-2):145-153.

[10] Li Z, Song J, Xiao H. On the cycles of Boolean networks. In 2012 24th Chinese Control and Decision Conference (CCDC). IEEE. 2012;770-774.

[11] Suo J, Sun J. New algorithm for finding fixed points and cycles of Boolean network. In Proceedings of the 10th World Congress on Intelligent Control and Automation. IEEE. 2012;2691-2694.

[12] Berntenis N, Ebeling M. Detection of attractors of large Boolean networks via exhaustive enumeration of appropriate subspaces of the state space. BMC Bioinformatics. 2013;14(1):1-10.

[13] Hopfensitz M, Müssel C, Maucher M, Kestler HA. Attractors in Boolean networks: A tutorial. Computational Statistics. 2013;28(1):19-36.

[14] Akutsu T, Kuhara S, Maruyama O, Miyano S. A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions. Genome Informatics. 1998;9:151-160.

[15] Veliz-Cuba A, Aguilar B, Hinkelmann F, Laubenbacher R. Steady state analysis of Boolean molecular network models via model reduction and computational algebra. BMC Bioinformatics. 2014;15(221):1-8.

[16] Zhao Q. A remark on scalar equations for synchronous Boolean networks with biological applications. IEEE Transactions on Neural Networks. 2005;16(6):1715–1716.

[17] Rushdi AM, Al-Otaibi SO. On limitations of using scalar equations for analyzing synchronous Boolean networks. Journal of King Abdulaziz University: Engineering Sciences. 2008;19(2):41-49.

[18] Cull P. Linear analysis of switching nets. Kybernetik. 1971;8(1):31-39.

[19] Rushdi AM, Al-Otaibi SO. On the linear analysis of synchronous switching networks. Journal of King Abdulaziz University: Engineering Sciences. 2007;18(2):43-72.

[20] Yue J, Yan Y, Zhang Z, Li S. Matrix approach to simplify Boolean networks. In 2017 36th Chinese Control Conference (CCC), IEEE. 2017;2064-2069.

[21] Zheng D, Yang G, Li X, Wang Z. An algorithm to find cycles of biochemical systems. International Conference on Information, Business and Education Technology (ICIBET), Atlantis Press. 2013;14–18.

[22] Akutsu T. Algorithmic aspects of analysis and control of Boolean networks. Proceedings of The Évry Spring School on Modelling Complex Biological Systems in the Context of Genomics. 2010;47–60.

[23] Cheng D, Qi H, Xue A. A survey of semi-tensor product of matrices. Journal of System Science and Complexity. 2007;20:304-322.

[24] Cheng D. Semi-tensor product of matrices and its applications: A survey. Proceeding of the 6th ICCM'07, Zhejiang, China. 2007;3:641–668.

[25] Cheng D. Input-state approach to Boolean networks. IEEE Transactions on Neural Networks. 2009;20(3):512–521.

[26] Cheng D, Qi H. State–space analysis of Boolean networks. IEEE Transactions on Neural Networks. 2010;21(4):584-594.

[27] Cheng D, Qi H. A linear representation of dynamics of Boolean networks. IEEE Transactions on Automatic Control. 2010;55(10):2251–2258.

[28] Cheng D, Qi H, Li ZQ. Analysis and control of Boolean networks: A semi-tensor product approach. Springer-Verlag, London; 2011.

[29] Cheng D, Qi H, Zhao Y. On Boolean control networks-An algebraic approach. Proceedings of the 18th IFAC International Federation of Automatic Control" World Congress Milano (Italy). 2011;8366-8377.

[30] Laschov D, Margaliot M. A maximum principle for single-input Boolean control networks. IEEE Transactions on Automatic Control. 2011;56(4):913-917.

[31] Cheng D, Qi H, Zhiqiang L. Model construction of Boolean network via observed data. IEEE Transactions on Neural Networks. 2011;22(4):525-536.

[32] Wang Y, Li H. On definition and construction of Lyapunov functions for Boolean networks. IEEE, Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, China. 2012;1247-1252.

[33] Li R, Chu T. Complete synchronization of Boolean networks. IEEE Transactions on Neural Networks Learning Systems. 2012;23(5):840-846.

[34] Zhan J, Lu S, Yang G. Improved calculation scheme of structure matrix of Boolean network using semi-tensor product. In International Conference on Information Computing and Applications. Springer, Berlin, Heidelberg. 2012;242-248.

[35] Zhan J, Lu S, Yang G. Analysis of Boolean networks using an optimized algorithm of structure matrix based on semi-tensor product. JCP. 2013;8(6):1441-1448.

[36] Li R, Yang M, Chu T. Synchronization design of Boolean networks via the semi-tensor product method. IEEE Transactions on Neural Networks Learning Systems, On Line First; 2013.

[37] Rushdi AMA, Ghaleb FAM. A tutorial exposition of the analysis of synchronous Boolean networks via semi-tensor products. Journal of Qassim University: Engineering and Computer Sciences. 2015;8(1):45-64.

[38] Rushdi AM, Ghaleb FA. A tutorial exposition of semi-tensor products of matrices with a stress on their representation of Boolean functions. Journal of King Abdulaziz University: Computers and Information Technology. 2016;5(1):3-30.

[39] Zhang XH, Han HX, Sun ZJ, Zhang WD. Alternative approach to calculate the structure matrix of Boolean network with semi-tensor product. IET Control Theory & Applications. 2017;11(13):2048-2057.

[40] Lu J, Li H, Liu Y, Li F. Survey on semi-tensor product method with its applications in logical networks and other finite-valued systems. IET Control Theory & Applications. 2017;11(13):2040-2047.

[41] Li H, Zhao G, Meng M, Feng J. A survey on applications of semi-tensor product method in engineering. Science China Information Sciences. 2018;61(010202):1-17.

[42] Cheng D. From dimension-free matrix theory to cross-dimensional dynamic systems. Academic Press; 2019.

[43] Heidel J, Maloney J, Farrow J, Rogers J. Finding cycles in synchronous Boolean networks with applications to biochemical systems. International Journal Bifurcation and Chaos. 2013;13(3):535-552.

[44] Farrow C, Heidel J, Maloney H, Rogers J. Scalar equations for synchronous Boolean networks with biological applications. IEEE Transactions on Neural Networks. 2004;15(2):348–354.

[45] Alsogati AA. Analysis of synchronous Boolean networks by scalar and matrix techniques. Unpublished Master Thesis, ECE Department, King Abdulaziz University, Jeddah, Saudi Arabia; 2013.

[46] Rushdi AMA, Alsogati AA. On reduced scalar equations for synchronous Boolean networks. Journal of Mathematics and Statistics. 2013;9(3):262-276.

[47] Rushdi AMA. Derivation of reduced scalar equations for synchronous Boolean networks. Journal of King Abdulaziz University: Computer Science and Information Technology. 2015;4(2):39-68.

[48] Chou TWJ, Collins GE. Algorithms for the solution of systems of linear Diophantine equations. SIAM Journal on Computing. 1982;11(4):687-708.

[49] Erdös P, Steward CL, Tijdeman R. Some Diophantine equations with many solutions. Compositio Mathematica. 1988;66(1):37-56.

[50] Clausen M, Fortenbacher A. Efficient solution of linear Diophantine equations. Journal of Symbolic Computation. 1989;8(1-2):201-216.

[51] Chen F, Zhao G, He Q, Jin W. Dynamics classification of Boolean networks and solutions of Diophantine equation. Theoretical Mathematics and Applications. 2014;4:109-126.

[52] Dubrova E, Teslenko M. A SAT-based algorithm for finding attractors in synchronous Boolean networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2011;8(5):1393-1399.

[53] Rushdi AMA, Ahmad W. Finding all solutions of the Boolean satisfiability problem, if any, via Boolean-equation solving. Journal of King Abdulaziz University: Engineering Sciences. 2016;27(1): 19-34.

[54] Ahmad W, Rushdi AMA. A novel minimization method for sensor deployment via heuristic 2-sat solution. Sir Syed University Research Journal of Engineering & Technology. 2017;7(1):1-7.

[55] Rushdi AMA, Ahmad W. Satisfiability in big Boolean algebras via Boolean-equation solving. Journal of King Abdulaziz University. 2017;28(1):3-18.

[56] Rushdi MA, Rushdi AM, Zarouan M, Ahmad W. Satisfiability in intuitionistic fuzzy logic with realistic tautology. Kuwait Journal of Science. 2018;45(2):15-21.

[57] Ahmad W, Rushdi AMA. A new cryptographic scheme utilizing the difficulty of big Boolean satisfiability. International Journal of Mathematical, Engineering and Management Sciences (IJMEMS). 2018;3(1):47-61.

[58] Rudeanu S. Boolean functions and equations. North-Holland Publishing Company and American Elsevier, Amsterdam, the Netherlands; 1974.

[59] Brown FM. Boolean reasoning: The logic of Boolean equations. Kluwer Academic Publishers, Boston, MA, USA; 1990.

[60] Rushdi AM. Using variable-entered Karnaugh maps to solve Boolean equations. International Journal of Computer Mathematics. 2001;78(1):23-38.

[61] Rushdi AM, Ba-Rukab OM. Low-cost design of multiple-output switching circuits using map solutions of Boolean equations. Umm Al-Qura University Journal of Science–Medicine–Engineering. 2003;15(2):59-79.

[62] Rushdi AMA. Efficient solution of Boolean equations using variable-entered Karnaugh maps. Journal of King Abdulaziz University: Engineering Sciences. 2004;15(1):105-121.

[63] Rushdi AM, Amashah MH. Using variable–entered Karnaugh maps to produce compact parametric general solutions of Boolean equations. International Journal of Computer Mathematics. 2011;88(15): 3136-3149.

[64] Rushdi AMA, Albarakati HM. The inverse problem for Boolean equations. Journal of Computer Science. 2012;8(12):2098-2105.

[65] Rushdi AMA, Amashah MH. Purely-algebraic versus VEKM methods for solving big Boolean equations. Journal of King Abdulaziz University: Engineering Sciences. 2012;23(2):75-85.

[66] Rushdi AM. A comparison of algebraic and map methods for solving general Boolean equations. Journal of Qassim University: Engineering and Computer Sciences. 2012;5(2):147-173.

[67] Rushdi AMA, Albarakati HM. Prominent classes of the most general subsumptive solutions of Boolean equations. Information Sciences. 2004;281:53-65.

[68] Rushdi AMA, Ahmad W. A novel method for compact listing of all particular solutions of a system of Boolean equations. Journal of Advances in Mathematics and Computer Science. 2017;22(6):1-18.

[69] Rushdi AMA, Balamesh AS. Boolean curve fitting with the aid of variable-entered Karnaugh maps. International Journal of Mathematical, Engineering and Management Sciences. 2019;4(6):1287-1306.

[70] Rushdi AM, Zagzoog SS. On 'Big' Boolean-equation solving and its utility in combinatorial digital design. Chapter 3 in Advances in Applied Science and Technology. Book Publisher International, Hooghly, West Bengal, India. 2019;2:25-48.

[71] Balamesh AS, Rushdi AM. Solution of Boolean equations via atomic decomposition into independent switching equations. International Journal of Computer Mathematics: Computer Systems Theory. 2019;4(3-4):185-203.

[72] Lin S, Costello DJ. Error control coding. Second Edition, Prentice Hall, Englewood Cliffs, NJ; 2004.

[73] Garrett P. The mathematics of coding theory. Pearson Prentice Hall, Upper Saddle River, NJ; 2004.

[74] Lee SC. Modern switching theory and digital design. Prentice Hall, Englewood Cliffs, NJ; 1978.

[75] Muroga S. Logic design and switching theory. Wiley, New York, NY, USA; 1979.

[76] Rushdi AM, Rushdi MA. Switching-algebraic analysis of system reliability. Chapter 6 in Ram, M. and Davim, P. (Editors), Advances in Reliability and System Engineering, Management and Industrial Engineering Series, Springer International Publishing, Cham, Switzerland. 2017;139-161.

[77] Rushdi AMA. Utilization of Karnaugh maps in multi-value qualitative comparative analysis. International Journal of Mathematical, Engineering and Management Sciences. 2018;3(1):28-46.

[78] Rushdi RA, Rushdi AM. Karnaugh-map utility in medical studies: The case of fetal malnutrition. International Journal of Mathematical, Engineering and Management Sciences. 2018;3(3):220-244.

[79] Bard GV. Some basic facts about linear algebra over GF(2). In Algebraic Cryptanalysis. Springer, Boston, MA. 2009;81-88.

[80] Preparata FP. State-logic relations for autonomous sequential networks. IEEE Transactions on Electronic Computers. EC-13. 1964;5:542-548.

[81] Saluja KK, Ong EH. Minimization of Reed-Muller canonic expansion. IEEE Transactions on Computers. 1979;C-28(7):535-537.

[82] Quintana JM, Avedillo MJ. Reed-Muller descriptions of symmetric functions. Proceedings of the 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001). 2001;4:682-685.

[83] Rushdi AMA, Ghaleb FAM. On self-inverse binary matrices over the binary Galois field. Journal of Mathematics and Statistics. 2013;9(3):238-248.

[84] Rushdi AMA, Ghaleb FAM. The Walsh spectrum and the real transform of a switching function: A review with a Karnaugh-map perspective. Journal of Qassim University: Engineering and Computer Sciences. 2014;7(2):73-112.

[85] Beauzamy B. Introduction to operator theory and invariant subspaces. Elsevier; 1988.

[86] Gohberg I, Lancaster P, Rodman L. Invariant subspaces of matrices with applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, USA; 2006.

[87] Antsaklis JP, Michel AN. Linear systems. Birkhauser, Boston; 2005.

[88] Kreyszig E. Advanced engineering mathematics. Ninth Edition, Wiley, New York; 2006.

[89] Burrow MD. The minimal polynomial of a linear transformation. The American Mathematical Monthly. 1973;80(10):1129-1131.

[90] Garibaldi S. The characteristic polynomial and determinant are not ad hoc constructions. The American Mathematical Monthly. 2004;111(9):761-778.

[91] Hoang TM, Thierauf T. On the minimal polynomial of a matrix. In International Computing and Combinatorics Conference. Springer, Berlin, Heidelberg. 2002;37-46.

[92] Thomas DM, Stevens JG, Lettieri S. Characteristic and minimal polynomials of linear cellular automata. The Rocky Mountain Journal of Mathematics. 2006;36(3):1077-1092.

[93] Schwab JD, Kühlwein SD, Ikonomi N, Kühl M, Kestler HA. Concepts in Boolean network modeling: What do they all mean? Computational and Structural Biotechnology Journal. 2020;18:571-582.

[94] Lü L, Chen D, Ren XL, Zhang QM, Zhang YC, Zhou T. Vital nodes identification in complex networks. Physics Reports. 2006;650:1-63.

[95] Liu JG, Lin JH, Guo Q, Zhou T. Locating influential nodes via dynamics-sensitive centrality. Scientific Reports. 2016;21380(6):1-8.

[96] Freeman LC. A set of measures of centrality based on betweenness. Sociometry. 1977;40(1):35-41.

[97] Bonacich P. Power and centrality: A family of measures. American Journal of Sociology. 1987;92(5):1170-1182.

[98] Borgatti SP. Centrality and network flow. Social Networks. 2005;27(1):55-71.

[99] Borgatti SP, Everett MG. A graph-theoretic perspective on centrality. Social Networks. 2006;28(4): 466-484.

[100] Bonacich P. Some unique properties of eigenvector centrality. Social Networks. 2007;29(4):555-564.

[101] Grassi R, Stefani S, Torriero A. Some new results on the eigenvector centrality. Mathematical Sociology. 2007;31(3):237-248.

[102] Ghanbarnejad F, Klemm K. Impact of individual nodes in Boolean network dynamics. Europhysics Letters (EPL). 2012;99(5):1-7. Article 58006.

[103] Wang J, Feng X, Yin Z, Guo B, Zheng Z. Identifying influential vertices in Boolean networks through dynamical voter rank. In 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE. 2017;1016-1020.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://www.sdiarticle4.com/review-history/59926