Front. Electr. Electron. Eng. China 2010, 5(2): 163–172
DOI 10.1007/s11460-010-0011-4

RESEARCH ARTICLE

Jeffrey Zhi J. ZHENG, Christian H. ZHENG

A framework to express variant and invariant functional spaces for binary logic

Abstract A new framework has been developed to express variant and invariant properties of functions operating on a binary vector space. This framework allows for manipulation of dynamic logic using basic operations and permutations. Novel representations of binary functional spaces are presented. Current ideas of binary functional spaces are extended and additional conditions are added to describe new function representation schemes: F code and C code.

Sizes of the proposed functional space representation schemes were determined. It was found that the complete representation for any set of functions operating on a binary sequence of numbers is larger than previously thought. The complete representation can only be described using a structure having a space of size $2^{2^n} \times 2^n!$ for any given space of functions acting on a binary sequence of length n. The framework, along with the proposed coding schemes provides a foundational theory of variant and invariant logic in software and electric-electronic technology and engineering, and has uses in the analysis of the stability of rule-based, dynamic binary systems such as cellular automata.

Keywords two-dimensional (2D) organization, conjugate symmetry, cellular automata (CA), permutation, meta-state, vector space, truth table (TT), variant table (VT), invariant table (IVT), variant logic, optimization, global coding, vector function space

Received January 14, 2010; accepted February 23, 2010

Jeffrey Zhi J. ZHENG (✉)
Department of Information Security, School of Software, Yunnan University, Kunming 650200, China
Conjugate Systems Kunming Ltd. Co., Kunming 650032, China
E-mail: conjugatesys@gmail.com

Christian H. ZHENG
Department of Electrical and Electronic Engineering, School of Engineering, The University of Melbourne, Parkville, Victoria 3010, Australia

1 Boolean functions

1.1 Truth tables

In Boolean algebra, any Boolean function can be described as one of two canonical forms—a sum-of-products representation or product-of-sums representation—according to the values of its truth table (TT) [1–13]. A TT separates the values of all entries into two sets; a minimal item set containing all 0-valued entries, and a maximal item set containing all 1-valued entries. Table 1 shows a typical example of such representation.

Table 1 A Boolean function of two variables in a truth table

value	x_1x_0	TT(x_1,x_0)	M (max) or m (min)
0	00	0	m
1	01	0	m
2	10	1	M
3	11	0	m

Looking at Table 1, M represents an element of the maximal item set whilst m represents an element of the minimal item set. In this particular TT, $M=\{2\}$ and $m=\{0, 1, 3\}$. When TT is represented in this fashion, the two canonical forms of Boolean functions can be generated using either one of the item sets (m or M).

1.2 Binary function spaces

In the history of mathematical logic, the two canonical forms play a key role in the development of Boolean algebra. A further extension in this theory has been developed to describe the complexity of the complete function space of Boolean algebra. Boolean functions in this space are complete when all the possible functional representations have been described. This will be defined as "full coverage" of the binary functional space.

For any n variables, there are 2^n entries on a truth table. For all functions on the variable space to be described, the space of total number of functions on the entries will

contain 2^{2^n} elements. The complete representation for full coverage of the space for $n = 2$ is shown in Table 2.

Table 2 Complete function space for $n = 2$

function element	TT(x_1,x_0)			
	11	10	01	00
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
⋮	⋮	⋮	⋮	⋮
15	1	1	1	1

For a binary sequence of $n = 2$, there will be 2^n or 4 entries on TT. The total number of functions will then be 2^{2^n} or 16 elements. It should be noted that the example TT representation shown in Table 1 corresponds to the function element 4 shown in Table 2.

The number of elements in the function space for an n-length binary sequence increases at a greater than exponential rate with respect to n. However, there are symmetric forms that can be exploited to build a structure on top of the functional space so that elements can be analyzed for stability and recursive properties. Symmetry plays a key role in all academic disciplines including math and philosophy [14–20]. Symmetry is used in this paper to find the intrinsic properties of sets of binary functions.

2 Recursive representation of binary function spaces

2.1 Cellular automata

Cellular automata (CA) [21] is one example of a methodology that uses Boolean algebra to describe a set of function spaces acting on a binary sequence. Since J. von Neumann initially proposed cellular automata [21] in the 1950s, much progress has been made in the fifty years of development[1,2] [22–34].

The simplest non-trivial CA is a one-dimensional binary sequence. Each binary digit in the sequence represents a cell, which changes over time depending upon a rule that is applied recursively with input values being the state of the cell and its neighbors (defined to be the adjacent cells on either side of it). A cell and its two neighbors form a neighborhood of three cells, so there are 2^3 or 8 possible patterns for a neighborhood. There are then 2^8 or 256 possible rules. These 256 rules are generally referred to by their Wolfram code, a standard naming convention that gives each rule a number from 0 to 255[1] [25–27].

First proposed by and named after Stephan Wolfram in the 1990s, the code has been extensively studied. Any Boolean function can be classified into four categories according to its recursive properties. Reference [26] proposed a numeric value to indicate random or idle properties to identify some of the interesting effects on the edge of chaos to be observed around this type of special numbers. Zheng et al.[2] [31–34] proposed the conjugate transformation to map regular plane lattices on binary images. The conjugate transform has been used to study global behaviors for one-dimensional (1D) CA using two-dimensional (2D) mapping strategy[2] [33,34]. Much research in this direction have extended CA as one of the main branches of modern complexity science or as Wolfram himself emphatically states, "a new kind of science"[1].

2.2 Binary functions on CA

In the one-dimensional form of CA, an N-length binary sequence is defined as

$$X = X_{N-1}X_{N-2}\cdots X_j\cdots X_1X_0, \quad 0 \leqslant j < N.$$

The output Y of function $f : X \to Y$ is defined as

$$Y = Y_{N-1}Y_{N-2}\cdots Y_j\cdots Y_1Y_0, \quad 0 \leqslant j < N.$$

It is possible to use a moving window with a fixed length n to separate X into a local kernel with length n. A form of the kernel can be presented as

$$x_{n-1}x_{n-2}\cdots x_i\cdots x_1x_0 \in \{0,1\}, \quad 0 \leqslant i < n.$$

Any function f operating on the kernel is defined as

$$f : x_{n-1}x_{n-2}\cdots x_i\cdots x_1x_0 \to y, \quad 0 \leqslant i < n.$$

For the window sliding across X, the function can be applied to every local kernel sequence at position i. The function is defined for every element of the input (X_j) and the output (Y_j):

$$f : X_{j+n-i-1}\cdots X_j\cdots X_{j-1} \to Y_j, \quad j-i \leqslant j < j+n-i.$$

The relationship between the input of the sliding window $(x_{n-1}x_{n-2}\cdots x_i\cdots x_1x_0)$, the output of the sliding window (y), X_j, and Y_j is defined as follows:

$$\begin{aligned} y &= f(x_{n-1}x_{n-2}\cdots x_i\cdots x_1x_0) \\ &= f(X_{j+n-i-1}\cdots X_j\cdots X_{j-i}) = Y_j \end{aligned}$$

or

$$X_j = X_j^{t-1}, \quad Y_j = X_j^t.$$

In the simplest case of CA where the neighborhood of

1) Wolfram S. A New Kind of Science. Wolfram Media Inc., 2002. http://www.wolframscience.com/
2) Zheng Z J. Conjugate visualisation of global complex behaviour. Complexity International, Vol. 3, 1996. http://www.complexity.org.au/ci/vol03/zheng/

the cell is only itself and two adjacent cells, the window is of size $n = 3$.

The superscript t and $t-1$ on X_j relate to before and after values of an iteration of the recursive function $f : X_j^{t-1} \to X_j^t$ at position j of the sequence. Four types of the single element transform, $X_j^{t-1} \to X_j^t$, can be identified as follows:

a) $0 \to 0$,

b) $0 \to 1$,

c) $1 \to 0$,

d) $1 \to 1$.

In classic Boolean algebra, the set of maximal items are formed by grouping all the elements of b) and d). This corresponds to locating all output entries with a value of one. Alternatively, the grouping of a) and c) forms the set of minimal items, composed of all entries with a value of zero. Under this construction, only mixed join effects can be statically put in the two canonical forms. These classical canonical forms are composed of 2^n entries that provide for equivalent values for any function operating on an n-length binary sequence.

3 Variant and invariant tables

3.1 Construction of variant tables

To extend the theory of Boolean algebra and the representation of TT, definitions of the variant table (VT) and the invariant table (IVT) are given. The example TT shown in Table 1 is extended furthermore. Additional terms have been defined in Table 3 of the input and output sequences. The VT and IVT only relate a single input to the output. The tables display change in the pixel value between input and output. This is in contrast to the classic Boolean algebra scheme that classifies entries with either a 0-output or a 1-output.

Table 3 A truth table with variant and invariant tables

value	x_1x_0	TT(x_1,x_0)	VT(x_0)	IVT(x_0)	VT(x_1)	IVT(x_1)
0	00	0	0	1	0	1
1	01	0	1	0	0	1
2	10	1	1	0	0	1
3	11	0	1	0	1	0

Table 3 shows how VT and IVT are created for both x_1 and x_0 inputs. VT and IVT can be interpreted as XOR and NXOR operations respectively between the input and the output. VT entries contain a 1-value if there is a change from 0 to 1 or a change from 1 to 0. The entries contain a 0-value otherwise. IVT is the complement of VT.

3.2 Representation of variant properties

It can be seen that any binary function can be represented using three different types of configurations namely: TT, VT, and IVT.

In the early eleventh century, Chinese mathematician Yong Shao presented the concept of a binary map in the form of a black and white tree with more complexity as the tree branched away at every hierarchical layer [35–43]. Binary numbers did not appear in the West until the seventeenth century when Leibniz [35,41] created what is known today as the binary number sequences.

In reference to the work of Leibniz and Shao, the base ten equivalent to the binary number is defined henceforth as Shao-Leibniz (SL) code. This code forms a standard linear sequence of entries on binary function set (Table 2).

Definition 1 The operator $\mathrm{BN} : I \to B$ converts an integer to its relevant binary representation. The operator $\mathrm{DC} : B \to I$ converts a binary number to its decimal representation.

Definition 2 The SL coding scheme is an ordering of binary truth table outputs $\mathrm{TT} : B^n \to B$. An element $J_I \in \mathrm{SL}$ at position I where $I < 2^n$ represents function TT_I such that the binary representation of the sequence of inputs to TT_I is defined as

$$\mathrm{TT}_I[\mathrm{BN}(2^n-1)]\cdots\mathrm{TT}_I[\mathrm{BN}(1)]\mathrm{TT}_I[\mathrm{BN}(0)] = \mathrm{BN}(I).$$

For variant and invariant logic structure, a new ordering scheme is created. This is defined as general (G) code for one dimension binary functions, representing all expanded sets.

Definition 3 A G coding scheme is defined as an ordering of binary truth table outputs $\mathrm{TT} : B^n \to B$. An element $J_I \in G$ at position I where $I < 2^n$ represents function TT_I such that the binary representation of the sequence of inputs to TT_I is defined as

$$\mathrm{TT}_I[X_{2^n-1}]\cdots\mathrm{TT}_I[X_1]\mathrm{TT}_I[X_0] = \mathrm{BN}(I),$$
$$\forall X_0X_1\cdots X_{2^n-1} \in [0, 2^n-1],\ X_0 \neq X_1 \neq \cdots \neq X_{2^n-1}.$$

The complete set of G coding schemes contains all permutations of binary function outputs. The SL coding scheme is one method of ordering within the G code that orders the presentation of binary numbers in a certain order. As an example, the full representation of the total space for all G coding schemes for $n = 1$ is displayed in Table 4.

A more syntactically convenient scheme is to visualize each element within the scheme as in a two-dimensional table form. The two-dimensional coding form is defined as Wen (W) coding scheme that is based upon King Wen's expression for a discrete dynamic structure in the eleventh century BC [36–43].

Definition 4 A W coding scheme is defined as an ordering of binary truth table outputs $TT: B^n \to B$. An element $U_I \in W$ at position I where $I < 2^n$ represents function TT_I such that the binary representation of the sequence of inputs to TT_I is defined as

$$TT_I[X_{2^n-1}]\cdots TT_I[X_1]TT_I[X_0] = \langle J^1|J^0\rangle,$$
$$J^1 = BN\left(I/2^{n-1}\right),\ J^0 = BN\left(I\%2^{n-1}\right),$$
$$\forall X_0X_1\cdots X_{2^n-1} \in [0, 2^n-1],\ X_0 \neq X_1 \neq \cdots \neq X_{2^n-1}.$$

Let J be an SL element $0 \leqslant J < 2^{2^n}$, it can be divided into two parts as $\langle J^1|J^0\rangle$. This structure is the notation for W coding scheme. The notation maps a one-dimensional SL code as a two-dimensional representation W code. For example, J^1 defines the upper 2^{n-1} state sets whilst J^0 defines the lower 2^{n-1} state sets. The entire function space can be defined as $0 \leqslant J^1, J^0 < 2^{2^n-1}$. This representation forms a 2D structure to organize the entire 2^{2^n} functions on a plane. W coding scheme $\langle J^1|J^0\rangle$ is shown in Table 5.

3.3 Two operators in vector spaces

In addition to this type of separation, each vector of J^0, J^1 can be represented as itself or a revised vector to check any function values defined by TT, VT and IVT columns, respectively.

For this kind of results, it is feasible to apply a NOT operation on each vector function; there are 2^{2^n} distinct cases in total.

In addition to reverse operation on vector functions, 2^n meta-states as 2^n meta-feature vectors can be put in different order to generate different number sequences for $\langle J^1|J^0\rangle$ coding schemes. Different permutations can be represented as global effects via symmetric properties over the space [1,44–52]. In general condition, 2^n meta-vectors can be organized as 2^n elements of a symmetric permutation group [45–48]. This symmetry group is composed of $2^n!$ permutation respectively.

3.4 A series of global coding schemes

Variant and invariant properties can be represented by combining four sets {a,b,c,d}. Therefore, two complementary coding schemes can be identified according to their symmetric properties. Restrictions are made on W set to define Fu (F) coding scheme in honor of Xi Fu [35–43].

Definition 5 F coding scheme is defined as a subset W. For any W code, if any two meta states can be paired, such that $\forall j_1 - 2^{n-1} = j_0, 0 \leqslant j_0 < 2^{n-1}, 2^{n-1} \leqslant j_1 < 2^n$ and $I_{j_1} = \overline{I}_{j_0}$ indicates one state I_{j_1} being another state I_{j_0}'s complement.

F coding scheme can be further defined by the addition of more restrictive conditions to construct the Conjugate (C) coding scheme.

Definition 6 C coding scheme is defined as a set of F coding scheme whereby $\forall I_{j_0} \in I_{J^0}$, for the selected position i, $\forall a_i \in I_{J^0}$ and $a_i = 0(1)$, $0 \leqslant i < n$.

Under C code scheme, for all 2^{n-1} states in I_{j_0}, each state has a 0-value (or 1-value) on the ith position. In addition, all 2^{n-1} rest states in I_{j_1}, each state has a 1-value (or 0-value) on the ith position.

3.5 Sizes of vector space

It is possible to list the above global coding schemes in relation to their size of structures in a list of propositions.

Table 4 Full representation of total space for all G coding schemes for $n = 1$

variables	states	functions	exponential power products	SL code	W code	F code	C code
n	2^n	2^{2^n}	$2^n!$	1	$2^{2^n} \times 2^n!$	$2^{2^n\left(1+\frac{1}{2}\right)} \times 2^{n-1}!$	$8 \times 2^{n-1}!$
1	2	4	2	1	8	8	8

Table 5 2D scheme for W code*

$\langle J^1\|J^0\rangle$	0	1	…	J^0	…	$2^{2^n-1}-1$
0	$\langle 0\|0\rangle$	$\langle 0\|1\rangle$	…	$\langle 0\|J^0\rangle$	…	$\langle 0\|2^{2^n-1}-1\rangle$
1	$\langle 1\|0\rangle$	$\langle 1\|1\rangle$	…	$\langle 1\|J^0\rangle$	…	$\langle 1\|2^{2^n-1}-1\rangle$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
J^1	$\langle J^1\|0\rangle$	$\langle J^1\|1\rangle$	…	$\langle J^1\|J^0\rangle$	…	$\langle J^1\|2^{2^n-1}-1\rangle$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$2^{2^n-1}-1$	$\langle 2^{2^n-1}-1\|0\rangle$	$\langle 2^{2^n-1}-1\|1\rangle$	…	$\langle 2^{2^n-1}-1\|J^0\rangle$	…	$\langle 2^{2^n-1}-1\|2^{2^n-1}-1\rangle$

* One W code contains 2^{2^n} functions, one function has a certain position in the plan.

Proposition 1 For a W code scheme of n variables, if a Boolean functional space has 2^{2^n} members, then its W code has a total number of $2^{2^n} \times 2^n!$ cases.

Proof Each vector and their independent selection of meta-vector correspond to a separate symmetric group. A total number of cases are products of two structures.

Consider 2^{2^n} being a big number; further $2^{2^n} \times 2^n!$ times expansion contain much more complexity than 2^{2^n}.

Proposition 2 For n variables, F code scheme contains a total of $2^{2^n} \times 2^{2^{n-1}} \times 2^{n-1}! = 2^{2^n\left(1+\frac{1}{2}\right)} \times 2^{n-1}!$ cases.

Proof For each state position, there are 2^{2^n} cases to apply a selected NOT operation on 2^n vector functions. Undertaking permutation operation satisfying a pair condition, there are $2^n(2^n-2)(2^n-4)\cdots 2 = 2^{2^{n-1}} \times 2^{n-1}!$ cases distinguished. Since two operations are independent, we have a total number listed.

Proposition 3 For n variables, C code scheme contains a total of $8 \times 2^{n-1}!$ cases.

For these coding schemes, their sizes of variation space are summarized in Table 6.

3.6 Sample results

To show the proposed structure of all the proposed coding schemes, examples are given for the one variable and two variable cases.

3.6.1 One variable

The simplest case set can select one variable to generate all possible cases in both 1D and 2D distribution. A series of results are shown in Tables 7(a) –7(c) respectively:

1) G code and W code for eight cases;
2) Representing all cases in 2D form as G code;
3) Showing all possible arrangement for functions.

From this set of results, a complete set of coding schemes has been shown completely for the one variable case.

Table 6 Sizes comparison of different global coding schemes

variables	states	functions	exponential power products	SL code	W code	F code	C code
n	2^n	2^{2^n}	$2^n!$	1	$2^{2^n} \times 2^n!$	$2^{2^n\left(1+\frac{1}{2}\right)} \times 2^{n-1}!$	$8 \times 2^{n-1}!$
1	2	4	2	1	8	8	8
2	4	16	24	1	384	128	16
3	8	256	40320	1	10321920	98304	192
4	16	2^{16}	16!	1	$2^{16} \times 16!$	$2^{24} \times 8!$	$8 \times 8!$
5	32	2^{32}	32!	1	$2^{32} \times 32!$	$2^{48} \times 16!$	$8 \times 16!$

Table 7 For $n = 1$, $2^{2^n} \times 2^n! = 8$, showing all eight cases of G code and W code respectively

(a) G code and W code for different cases

G3				G2				G1				G0			
G code	$f(1)^1$	$f(0)^1$	W code	G code	$f(1)^1$	$f(0)^0$	W code	G code	$f(1)^0$	$f(0)^1$	W code	G code	$f(1)^0$	$f(0)^0$	W code
0	0	0	$\langle 0\|0\rangle$	1	0	1	$\langle 0\|1\rangle$	2	1	0	$\langle 1\|0\rangle$	3	1	1	$\langle 1\|1\rangle$
1	0	1	$\langle 0\|1\rangle$	0	0	0	$\langle 0\|0\rangle$	3	1	1	$\langle 1\|1\rangle$	2	1	0	$\langle 1\|0\rangle$
2	1	0	$\langle 1\|0\rangle$	3	1	1	$\langle 1\|1\rangle$	0	0	0	$\langle 0\|0\rangle$	1	0	1	$\langle 0\|1\rangle$
3	1	1	$\langle 1\|1\rangle$	2	1	0	$\langle 1\|0\rangle$	1	0	1	$\langle 0\|1\rangle$	0	0	0	$\langle 0\|0\rangle$
G7				G6				G5				G4			
G code	$f(0)^1$	$f(1)^1$	W code	G code	$f(0)^0$	$f(1)^1$	W code	G code	$f(0)^1$	$f(1)^0$	W code	G code	$f(0)^0$	$f(1)^0$	W code
0	0	0	$\langle 0\|0\rangle$	2	1	0	$\langle 1\|0\rangle$	1	0	1	$\langle 0\|1\rangle$	3	1	1	$\langle 1\|1\rangle$
2	1	0	$\langle 1\|0\rangle$	0	0	0	$\langle 0\|0\rangle$	3	1	1	$\langle 1\|1\rangle$	1	0	1	$\langle 0\|1\rangle$
1	0	1	$\langle 0\|1\rangle$	3	1	1	$\langle 1\|1\rangle$	0	0	0	$\langle 0\|0\rangle$	2	1	0	$\langle 1\|0\rangle$
3	1	1	$\langle 1\|1\rangle$	1	0	1	$\langle 0\|1\rangle$	2	1	0	$\langle 1\|0\rangle$	0	0	0	$\langle 0\|0\rangle$

(b) Representing G coding schemes in 2D forms

G3		G2		G1		G0		G7		G6		G5		G4	
0	1	1	0	2	3	3	2	0	2	2	0	1	3	3	1
2	3	3	2	0	1	1	0	1	3	3	1	0	2	2	0

(c) Including all original functions in the position

G3		G2		G1		G0		G7		G6		G5		G4	
0	$\bar{x}$	$\bar{x}$	0	x	1	1	x	0	x	x	0	$\bar{x}$	1	1	$\bar{x}$
x	1	1	x	0	$\bar{x}$	$\bar{x}$	0	$\bar{x}$	1	1	$\bar{x}$	0	x	x	0

3.6.2 Two variables

After another variable was added to the sequence, a large increase in the complexity of structure can be seen. Selected cases are shown in Tables 8(a)–8(f). All of the following show examples of schemes representing two variable coding in

(a) TT and VT
(b) False table (FT) and IVT
(c) Functions in VT
(d) Functions in IVT
(e) Functions in SL code for TT
(f) Functions in G code for FT

This set of results has shown very interesting symmetric properties. This can be seen from function distributions under relevant coding schemes with different symmetric properties. C and F coding schemes have much more symmetric properties than the more general schemes.

Table 8 Two variable cases in TT, FT, VT and IVT: Type A

(a) Two variable cases in TT and VT

SL code in a TT form (Shao-Leibniz code)						C code in a VT form Type A (I)						C code in a VT form Type A (II)						function
No.	11	10	01	00	$\langle J^2\|J^0\rangle$	No.	11	01	00	10	$\langle J^2\|J^0\rangle$	No.	01	11	10	00	$\langle J^2\|J^0\rangle$	
0	0	0	0	0	$\langle 0\|0\rangle$	12	1	1	0	0	$\langle 3\|\widetilde{0}\rangle$	12	1	1	0	0	$\langle 3\|\widetilde{0}\rangle$	0
1	0	0	0	1	$\langle 0\|1\rangle$	14	1	1	1	0	$\langle 3\|\widetilde{2}\rangle$	13	1	1	0	1	$\langle 3\|\widetilde{1}\rangle$	$\bar{z}\bar{x}$
2	0	0	1	0	$\langle 0\|2\rangle$	8	1	0	0	0	$\langle 2\|\widetilde{0}\rangle$	4	0	1	0	0	$\langle 1\|\widetilde{0}\rangle$	$\bar{z}x$
3	0	0	1	1	$\langle 0\|3\rangle$	10	1	0	1	0	$\langle 2\|\widetilde{2}\rangle$	5	0	1	0	1	$\langle 1\|\widetilde{1}\rangle$	$\bar{z}$
4	0	1	0	0	$\langle 1\|0\rangle$	13	1	1	0	1	$\langle 3\|\widetilde{1}\rangle$	14	1	1	1	0	$\langle 3\|\widetilde{2}\rangle$	$z\bar{x}$
5	0	1	0	1	$\langle 1\|1\rangle$	15	1	1	1	1	$\langle 3\|\widetilde{3}\rangle$	15	1	1	1	1	$\langle 3\|\widetilde{3}\rangle$	$\bar{x}$
6	0	1	1	0	$\langle 1\|2\rangle$	9	1	0	0	1	$\langle 2\|\widetilde{1}\rangle$	6	0	1	1	0	$\langle 1\|\widetilde{2}\rangle$	$\bar{z}x + z\bar{x}$
7	0	1	1	1	$\langle 1\|3\rangle$	11	1	0	1	1	$\langle 2\|\widetilde{3}\rangle$	7	0	1	1	1	$\langle 1\|\widetilde{3}\rangle$	$\bar{z} + \bar{x}$
8	1	0	0	0	$\langle 2\|0\rangle$	4	0	1	0	0	$\langle 1\|\widetilde{0}\rangle$	8	1	0	0	0	$\langle 2\|\widetilde{0}\rangle$	zx
9	1	0	0	1	$\langle 2\|1\rangle$	6	0	1	1	0	$\langle 1\|\widetilde{2}\rangle$	9	1	0	0	1	$\langle 2\|\widetilde{1}\rangle$	$zx + \bar{z}\bar{x}$
10	1	0	1	0	$\langle 2\|2\rangle$	0	0	0	0	0	$\langle 0\|\widetilde{0}\rangle$	0	0	0	0	0	$\langle 0\|\widetilde{0}\rangle$	x
11	1	0	1	1	$\langle 2\|3\rangle$	2	0	0	1	0	$\langle 0\|\widetilde{2}\rangle$	1	0	0	0	1	$\langle 0\|\widetilde{1}\rangle$	$\bar{z} + x$
12	1	1	0	0	$\langle 3\|0\rangle$	5	0	1	0	1	$\langle 1\|\widetilde{1}\rangle$	10	1	0	1	0	$\langle 2\|\widetilde{2}\rangle$	z
13	1	1	0	1	$\langle 3\|1\rangle$	7	0	1	1	1	$\langle 1\|\widetilde{3}\rangle$	11	1	0	1	1	$\langle 2\|\widetilde{3}\rangle$	$z + \bar{x}$
14	1	1	1	0	$\langle 3\|2\rangle$	1	0	0	0	1	$\langle 0\|\widetilde{1}\rangle$	2	0	0	1	0	$\langle 0\|\widetilde{2}\rangle$	$z + x$
15	1	1	1	1	$\langle 3\|3\rangle$	3	0	0	1	1	$\langle 0\|\widetilde{3}\rangle$	3	0	0	1	1	$\langle 0\|\widetilde{3}\rangle$	1

(b) Two variable cases in FT and IVT: Type A

G code in a FT form (General code)						C code in a IVT form Type A (I)						C code in a IVT form Type A (II)						function
No.	11	10	01	00	$\langle J^2\|J^0\rangle$	No.	11	01	00	10	$\langle J^2\|J^0\rangle$	No.	01	11	10	00	$\langle J^2\|J^0\rangle$	
15	1	1	1	1	$\langle 3\|3\rangle$	3	0	0	1	1	$\langle 0\|\widetilde{3}\rangle$	3	0	0	1	1	$\langle 0\|\widetilde{3}\rangle$	0
14	1	1	1	0	$\langle 3\|2\rangle$	1	0	0	0	1	$\langle 0\|\widetilde{1}\rangle$	2	0	0	1	0	$\langle 0\|\widetilde{2}\rangle$	$\bar{z}\bar{x}$
13	1	1	0	1	$\langle 3\|1\rangle$	7	0	1	1	1	$\langle 1\|\widetilde{3}\rangle$	11	1	0	1	1	$\langle 2\|\widetilde{3}\rangle$	$\bar{z}x$
12	1	1	0	0	$\langle 3\|0\rangle$	5	0	1	0	1	$\langle 1\|\widetilde{1}\rangle$	10	1	0	1	0	$\langle 2\|\widetilde{2}\rangle$	$\bar{z}$
11	1	0	1	1	$\langle 2\|3\rangle$	2	0	0	1	0	$\langle 0\|\widetilde{2}\rangle$	1	0	0	0	1	$\langle 0\|\widetilde{1}\rangle$	$z\bar{x}$

(*Continued*)

G code in a FT form (General code)						C code in a IVT form Type A (I)						C code in a IVT form Type A (II)						function
No.	11	10	01	00	$\langle J^2 \vert J^0 \rangle$	No.	11	01	00	10	$\langle J^2 \vert J^0 \rangle$	No.	01	11	10	00	$\langle J^2 \vert J^0 \rangle$	
10	1	0	1	0	$\langle 2 \vert 2 \rangle$	0	0	0	0	0	$\langle 0 \vert \widetilde{0} \rangle$	0	0	0	0	0	$\langle 0 \vert \widetilde{0} \rangle$	$\overline{x}$
9	1	0	0	1	$\langle 2 \vert 1 \rangle$	6	0	1	1	0	$\langle 1 \vert \widetilde{2} \rangle$	9	1	0	0	1	$\langle 2 \vert \widetilde{1} \rangle$	$\overline{z}x + z\overline{x}$
8	1	0	0	0	$\langle 2 \vert 0 \rangle$	4	0	1	0	0	$\langle 1 \vert \widetilde{0} \rangle$	8	1	0	0	0	$\langle 2 \vert \widetilde{0} \rangle$	$\overline{z} + \overline{x}$
7	0	1	1	1	$\langle 1 \vert 3 \rangle$	11	1	0	1	1	$\langle 2 \vert \widetilde{3} \rangle$	7	0	1	1	1	$\langle 1 \vert \widetilde{3} \rangle$	zx
6	0	1	1	0	$\langle 1 \vert 2 \rangle$	9	1	0	0	1	$\langle 2 \vert \widetilde{1} \rangle$	6	0	1	1	0	$\langle 1 \vert \widetilde{2} \rangle$	$zx + \overline{z}\overline{x}$
5	0	1	0	1	$\langle 1 \vert 1 \rangle$	15	1	1	1	1	$\langle 3 \vert \widetilde{3} \rangle$	15	1	1	1	1	$\langle 3 \vert \widetilde{3} \rangle$	x
4	0	1	0	0	$\langle 1 \vert 0 \rangle$	13	1	1	0	1	$\langle 3 \vert \widetilde{1} \rangle$	14	1	1	1	0	$\langle 3 \vert \widetilde{2} \rangle$	$\overline{z} + x$
3	0	0	1	1	$\langle 0 \vert 3 \rangle$	10	1	0	1	0	$\langle 2 \vert \widetilde{2} \rangle$	5	0	1	0	1	$\langle 1 \vert \widetilde{1} \rangle$	z
2	0	0	1	0	$\langle 0 \vert 2 \rangle$	8	1	0	0	0	$\langle 2 \vert \widetilde{0} \rangle$	4	0	1	0	0	$\langle 1 \vert \widetilde{0} \rangle$	$z + \overline{x}$
1	0	0	0	1	$\langle 0 \vert 1 \rangle$	14	1	1	1	0	$\langle 3 \vert \widetilde{2} \rangle$	13	1	1	0	1	$\langle 3 \vert \widetilde{1} \rangle$	$z + x$
0	0	0	0	0	$\langle 0 \vert 0 \rangle$	12	1	1	0	0	$\langle 3 \vert \widetilde{0} \rangle$	12	1	1	0	0	$\langle 3 \vert \widetilde{0} \rangle$	1

(c) Shown different functions in VT

C code ordering				Type A: I in VT				Type A: II in VT			
$\langle 0 \vert \widetilde{0} \rangle$	$\langle 0 \vert \widetilde{1} \rangle$	$\langle 0 \vert \widetilde{2} \rangle$	$\langle 0 \vert \widetilde{3} \rangle$	x	$z + x$	$\overline{z} + x$	1	x	$\overline{z} + x$	$z + x$	1
$\langle 1 \vert \widetilde{0} \rangle$	$\langle 1 \vert \widetilde{1} \rangle$	$\langle 1 \vert \widetilde{2} \rangle$	$\langle 1 \vert \widetilde{3} \rangle$	zx	z	$zx + \overline{z}\overline{x}$	$z + \overline{x}$	$\overline{z}x$	$\overline{z}$	$\overline{z}x + z\overline{x}$	$\overline{z} + \overline{x}$
$\langle 2 \vert \widetilde{0} \rangle$	$\langle 2 \vert \widetilde{1} \rangle$	$\langle 2 \vert \widetilde{2} \rangle$	$\langle 2 \vert \widetilde{3} \rangle$	$\overline{z}x$	$\overline{z}x + z\overline{x}$	$\overline{z}$	$\overline{z} + \overline{x}$	zx	$zx + \overline{z}\overline{x}$	z	$z + \overline{x}$
$\langle 3 \vert \widetilde{0} \rangle$	$\langle 3 \vert \widetilde{1} \rangle$	$\langle 3 \vert \widetilde{2} \rangle$	$\langle 3 \vert \widetilde{3} \rangle$	0	$z\overline{x}$	$\overline{z}\overline{x}$	$\overline{x}$	0	$\overline{z}\overline{x}$	$z\overline{x}$	$\overline{x}$

(d) Shown different functions in IVT

C code ordering				Type A: I in IVT				Type A: II in IVT			
$\langle 0 \vert \widetilde{0} \rangle$	$\langle 0 \vert \widetilde{1} \rangle$	$\langle 0 \vert \widetilde{2} \rangle$	$\langle 0 \vert \widetilde{3} \rangle$	$\overline{x}$	$\overline{z}\overline{x}$	$z\overline{x}$	0	$\overline{x}$	$z\overline{x}$	$\overline{z}\overline{x}$	0
$\langle 1 \vert \widetilde{0} \rangle$	$\langle 1 \vert \widetilde{1} \rangle$	$\langle 1 \vert \widetilde{2} \rangle$	$\langle 1 \vert \widetilde{3} \rangle$	$\overline{z} + \overline{x}$	$\overline{z}$	$\overline{z}x + z\overline{x}$	$\overline{z}x$	$z + \overline{x}$	z	$zx + \overline{z}\overline{x}$	zx
$\langle 2 \vert \widetilde{0} \rangle$	$\langle 2 \vert \widetilde{1} \rangle$	$\langle 2 \vert \widetilde{2} \rangle$	$\langle 2 \vert \widetilde{3} \rangle$	$z + \overline{x}$	$zx + \overline{z}\overline{x}$	z	zx	$\overline{z} + \overline{x}$	$\overline{z}x + z\overline{x}$	$\overline{z}$	$\overline{z}x$
$\langle 3 \vert \widetilde{0} \rangle$	$\langle 3 \vert \widetilde{1} \rangle$	$\langle 3 \vert \widetilde{2} \rangle$	$\langle 3 \vert \widetilde{3} \rangle$	1	$\overline{z} + x$	$z + x$	x	1	$z + x$	$\overline{z} + x$	x

(e) Functions in SL code for TT

SL code ordering 2D form				functions in SL code TT			
0	1	2	3	0	$\overline{z}\overline{x}$	$\overline{z}x$	$\overline{z}$
4	5	6	7	$z\overline{x}$	$\overline{x}$	$\overline{z}x + z\overline{x}$	$\overline{z} + \overline{x}$
8	9	10	11	zx	$zx + \overline{z}\overline{x}$	x	$\overline{z} + x$
12	13	14	15	z	$z + \overline{x}$	$z + x$	1

(f) Functions in G code for FT

G code ordering 2D form				functions in G code FT			
15	14	13	12	1	$z + x$	$z + \overline{x}$	z
11	10	9	8	$\overline{z} + x$	x	$zx + \overline{z}\overline{x}$	zx
7	6	5	4	$\overline{z} + \overline{x}$	$\overline{z}x + z\overline{x}$	$\overline{x}$	$z\overline{x}$
3	2	1	0	$\overline{z}$	$\overline{z}x$	$\overline{z}\overline{x}$	0

To illustrate other coding schemes, some results are shown in Tables 9(a)–9(d) for F and W codes. The schemes as well as their functions in VT and IVT are seen.

(a) VT

(b) IVT

(c) Functions in VT

(d) Functions in IVT

Compared with Tables 8 and 9, many symmetric properties on the C coding scheme can be observed. There are clear advantages of current 2D framework on visualizations of complex logic structure to represent variant and invariant properties.

4 Conclusion

In this paper, variant and invariant tables are proposed to extend truth table representation that describes different properties of binary sequences. This extension is required to expand traditional Boolean logic framework to a new

Table 9 F and W codes and their functions in VT and IVT

(a) VT

F code in VT						W code in VT						function
No.	00	01	11	10	$\langle J^2\|J^0\rangle$	No.	11	01	10	00	$\langle J^2\|J^0\rangle$	
6	0	1	1	0	$\langle 1\|2\rangle$	12	1	1	0	0	$\langle 3\|0\rangle$	0
14	1	1	1	0	$\langle 3\|2\rangle$	13	1	1	0	1	$\langle 3\|1\rangle$	$\bar{z}\bar{x}$
2	0	0	1	0	$\langle 0\|2\rangle$	8	1	0	0	0	$\langle 2\|0\rangle$	$\bar{z}x$
10	1	0	1	0	$\langle 2\|2\rangle$	9	1	0	0	1	$\langle 2\|1\rangle$	$\bar{z}$
7	0	1	1	1	$\langle 1\|3\rangle$	14	1	1	1	0	$\langle 3\|2\rangle$	$z\bar{x}$
15	1	1	1	1	$\langle 3\|3\rangle$	15	1	1	1	1	$\langle 3\|3\rangle$	$\bar{x}$
3	0	0	1	1	$\langle 0\|3\rangle$	10	1	0	1	0	$\langle 2\|2\rangle$	$\bar{z}x + z\bar{x}$
11	1	0	1	1	$\langle 2\|3\rangle$	11	1	0	1	1	$\langle 2\|3\rangle$	$\bar{z} + \bar{x}$
4	0	1	0	0	$\langle 1\|0\rangle$	4	0	1	0	0	$\langle 1\|0\rangle$	zx
12	1	1	0	0	$\langle 3\|0\rangle$	5	0	1	0	1	$\langle 1\|1\rangle$	$zx + \bar{z}\bar{x}$
0	0	0	0	0	$\langle 0\|0\rangle$	0	0	0	0	0	$\langle 0\|0\rangle$	x
8	1	0	0	0	$\langle 2\|0\rangle$	1	0	0	0	1	$\langle 0\|1\rangle$	$\bar{z} + x$
5	0	1	0	1	$\langle 1\|1\rangle$	6	0	1	1	0	$\langle 1\|2\rangle$	z
13	1	1	0	1	$\langle 3\|1\rangle$	7	0	1	1	1	$\langle 1\|3\rangle$	$z + \bar{x}$
1	0	0	0	1	$\langle 0\|1\rangle$	2	0	0	1	0	$\langle 0\|2\rangle$	$z + x$
9	1	0	0	1	$\langle 2\|1\rangle$	3	0	0	1	1	$\langle 0\|3\rangle$	1

(b) IVT

F code in IVT						W code in IVT						function
No.	00	01	11	10	$\langle J^2\|J^0\rangle$	No.	11	01	10	00	$\langle J^2\|J^0\rangle$	
9	1	0	0	1	$\langle 2\|1\rangle$	3	0	0	1	1	$\langle 0\|3\rangle$	0
1	0	0	0	1	$\langle 0\|1\rangle$	2	0	0	1	0	$\langle 0\|2\rangle$	$\bar{z}\bar{x}$
13	1	1	0	1	$\langle 3\|1\rangle$	7	0	1	1	1	$\langle 1\|3\rangle$	$\bar{z}x$
5	0	1	0	1	$\langle 1\|1\rangle$	6	0	1	1	0	$\langle 1\|2\rangle$	$\bar{z}$
8	1	0	0	0	$\langle 2\|0\rangle$	1	0	0	0	1	$\langle 0\|1\rangle$	$z\bar{x}$
0	0	0	0	0	$\langle 0\|0\rangle$	0	0	0	0	0	$\langle 0\|0\rangle$	$\bar{x}$
12	1	1	0	0	$\langle 3\|0\rangle$	5	0	1	0	1	$\langle 1\|1\rangle$	$\bar{z}x + z\bar{x}$
4	0	1	0	0	$\langle 1\|0\rangle$	4	0	1	0	0	$\langle 1\|0\rangle$	$\bar{z} + \bar{x}$
11	1	0	1	1	$\langle 2\|3\rangle$	11	1	0	1	1	$\langle 2\|3\rangle$	zx
3	0	0	1	1	$\langle 0\|3\rangle$	10	1	0	1	0	$\langle 2\|2\rangle$	$zx + \bar{z}\bar{x}$
15	1	1	1	1	$\langle 3\|3\rangle$	15	1	1	1	1	$\langle 3\|3\rangle$	x
7	0	1	1	1	$\langle 1\|3\rangle$	14	1	1	1	0	$\langle 3\|2\rangle$	$\bar{z} + \bar{x}$
10	1	0	1	0	$\langle 2\|2\rangle$	9	1	0	0	1	$\langle 2\|1\rangle$	z
2	0	0	1	0	$\langle 0\|2\rangle$	8	1	0	0	0	$\langle 2\|0\rangle$	$z + \bar{x}$
14	1	1	1	0	$\langle 3\|2\rangle$	13	1	1	0	1	$\langle 3\|1\rangle$	$z + x$
6	0	1	1	0	$\langle 1\|2\rangle$	12	1	1	0	0	$\langle 3\|0\rangle$	1

(c) Functions in VT

W code ordering				functions in F code VT				functions in W code VT			
⟨0,0⟩	⟨0,1⟩	⟨0,2⟩	⟨0,3⟩	x	$x+z$	$\bar{z}x$	$\bar{z}x+z\bar{x}$	x	$\bar{z}+x$	$z+x$	1
⟨1,0⟩	⟨1,1⟩	⟨1,2⟩	⟨1,3⟩	zx	z	0	$z\bar{x}$	zx	$zx+\bar{z}\bar{x}$	z	$z+\bar{x}$
⟨2,0⟩	⟨2,1⟩	⟨2,2⟩	⟨2,3⟩	$\bar{z}+x$	1	$\bar{z}$	$\bar{z}+\bar{x}$	$\bar{z}x$	$\bar{z}$	$\bar{z}x+z\bar{x}$	$\bar{z}+\bar{x}$
⟨3,0⟩	⟨3,1⟩	⟨3,2⟩	⟨3,3⟩	$zx+\bar{z}\bar{x}$	$z+\bar{x}$	$\bar{z}\bar{x}$	$\bar{x}$	0	$\bar{z}\bar{x}$	$z\bar{x}$	$\bar{x}$

(d) Functions in IVT

W code ordering				functions in F code IVT				functions in W code IVT			
⟨0,0⟩	⟨0,1⟩	⟨0,2⟩	⟨0,3⟩	$\bar{x}$	$\bar{z}\bar{x}$	$z+\bar{x}$	$zx+\bar{z}\bar{x}$	$\bar{x}$	$z\bar{x}$	$\bar{z}\bar{x}$	0
⟨1,0⟩	⟨1,1⟩	⟨1,2⟩	⟨1,3⟩	$\bar{z}+\bar{x}$	$\bar{z}$	1	$\bar{z}+x$	$\bar{z}+\bar{x}$	$\bar{z}x+z\bar{x}$	$\bar{z}$	$\bar{z}x$
⟨2,0⟩	⟨2,1⟩	⟨2,2⟩	⟨2,3⟩	$z\bar{x}$	0	z	zx	$z+\bar{x}$	z	$zx+\bar{z}\bar{x}$	zx
⟨3,0⟩	⟨3,1⟩	⟨3,2⟩	⟨3,3⟩	$\bar{z}x+z\bar{x}$	$\bar{z}x$	$z+x$	x	1	$z+x$	$\bar{z}+x$	x

variation space. Under two vector operations types, the new space has $2^{2^n}\times 2^n!$ times more complexity then traditional Boolean function space with 2^{2^n} members. In order to manage this complexity, the framework has proposed a novel coding scheme that enables the functions to be encoded through symmetric properties that come about from representing the elements as a 2D map. In this form of two-dimensional global representation, novel coding mechanisms can be constructed.

Boolean function space represents a hardware space and the new expanded space broadens the descriptions and coding schemes that can be used. Thus, a wide area of software coding and electric-electronic engineering for optimal IT system designs can be developed. In essence, the space of binary sequence functions can be thought of as a keyboard with $2^{2^n}\times 2^n!$ notes. Each note contains a complete Boolean function and its own TT/VT/IVT representation. The set of notes can be represented using a coding scheme that orders the notes in a particular sequence (SL and G codes) or map (W, F and C codes). These sequences have properties describing the dynamic properties of a binary system that forms a solid theoretical foundation for future analysis and design philosophies. This introductory paper outlines the construction and notation of such coding schemes. Future papers will show that the proposed scheme, with its foundation in symmetry, will have definite uses for predicting convergent and chaotic behavior in dynamic binary systems such as the analysis of cellular automata rules and quantum computations.

Acknowledgements This work was supported in part by Information Security Professional Education System Constructions from Yunnan University, and Yunnan Education Organization and Writing Environment in Conjugate Systems Kunming Ltd. Co.

References

1. Sikorski R. Boolean Algebras. Berlin: Springer-Verlag, 1960
2. Edwards F H. The Principles of Switching Circuits. Cambridge: MIT Press, 1973
3. Lee S C. Modern Switching Theory and Digital Design. New Jersey: Prentice-Hall Inc., 1978
4. Muroga S. Logic Design and Switching Theory. New York: Wiley-Interscience Publication, 1979
5. Thayse A. Boolean Calculus of Differences. Berlin: Springer-Verlag, 1981
6. Kim K H. Boolean Matrix Theory and Applications. New York: Marcel Dekker Inc., 1982
7. Markek W, Onyszkiewicz J. Elements of Logic and Foundations of Mathematics in Problems. Boston: Kluwer Academic Publishers, 1982
8. Dunne P E. The Complexity of Boolean Networks. London: Academic Press, 1988
9. Monk J D, Bonnet R. Handbook of Boolean Algebras. Amsterdam: North-Holland Publishing Company, 1989
10. Paterson M S. Boolean Function Complexity. Cambridge: Cambridge University Press, 1992
11. Vingron S P. Switching Theory: Insight Through Predicate Logic. Berlin: Springer, 2004
12. Von Neumann J. The General and Logical Theory of Automata. Collected Works. Vol. 5. New York: Pergamon Press, 1963
13. Noguchi S, Oizumi J. A survey of cellular logic. Journal of the Institute of Electronics and Communication Engineers of Japan, 1971: 54(2): 206–220
14. Fedorov E S. Symmetry of Crystals. New York: American Crystallographic Association, 1971 (Original published in Russian, 1890)
15. Weyl H. Symmetry. Princeton: Princeton University Press, 1952
16. Benacerraf P, Putnam H. Philosophy of Mathematics. New Jersey: Prentice-Hall Inc., 1964
17. MacGillavry C H. Symmetry Aspects of M. C. Escher's Periodic Drawings. Utrecht: A. Oosthoek's Uitgeversmaatschappij NV, 1965
18. Yale P B. Geometry and Symmetry. San Francisco: Holden Day Inc., 1968
19. Holden A. Shapes, Spaces and Symmetry. New York: Columbia University Press, 1971
20. Bool F H, Kist J R, Wierda F, Locher J L. M. C. Escher: His Life and Complete Graphic Work: With a Fully Illustrated Catalogue. London: Thames Hudson, 1982

21. Burks A W. Essays on Cellular Automata. Champaign: University of Illinois Press, 1970
22. Toffoli T. Cellular automata mechanics. Dissertation for the Doctoral Degree. Michigan: University of Michigan, 1977
23. Kandel A, Lee S C. Fuzzy Switching and Automata: Theory and Applications. New York: Crane Russak and Company Inc., 1979
24. Preston Jr K, Duff M J B. Modern Cellular Automata: Theory and Application. New York: Plenum Press, 1984
25. Wolfram S. Theory and Applications of Cellular Automata. Singapore: World Scientific, 1986
26. Langton C G. Life at the edge of Chaos. In: Lanton C G, Taylor C, Farmer J D, Rasmussen S, eds. Artificial Life II. New York: Addison-Wesley, 1992
27. Wolfram S. Cellular Automata and Complexity. New York: Addison-Wesley, 1994
28. Ilachinski A. Cellular Automata: A Discrete Universe. Singapore: World Scientific, 2001
29. Griffeath D, Moore C. New Constructions in Cellular Automata. Oxford: Oxford University Press, 2003
30. Umeo H, Morishita S, Nishinari K, Komatsuzaki T, Bandini S. Cellular Automata. Berlin: Springer, 2008
31. Zheng Z J, Maeder A J. The conjugate classification of the kernel form of the hexagonal grid. Modern Geometric Computing for Visualization. Berlin: Springer-Verlag, 1992, 73–89
32. Zheng Z J, Maeder A J. The elementary equation of the conjugate transformation for hexagonal grid. Modeling in Computer Graphics. Berlin: Springer-Verlag, 1993, 21–42
33. Zheng Z J. Conjugate transformation of regular plane lattices for binary images. Dissertation for the Doctoral Degree. Melbourne: Monash University, 1994
34. Zheng Z J, Leung C H C. Visualising global behaviour of 1D cellular automata image sequences in 2D maps. Physica A, 1996, 233(3): 785–800
35. Needham J. Science and Civilisation in China: Volume 2, History of Scientific Thought. Cambridge: Cambridge Press, 1956, 1954–1988
36. Hook D F. The I Ching and Mankind. London: Routledge and Kegan Paul Ltd., 1975
37. Chu W K, Sherrill W A. An Anthology of I Ching. London: Routledge and Kegan Paul Ltd., 1977
38. Shchutshii I K. Researches on the I Ching. Princeton: Princeton University Press, 1979
39. Wilhelm H. Chang: Eight Lectures on the I Ching. Princeton: Princeton University Press, 1979
40. Wilhelm R. Lectures on the I Ching: Constancy and Change. Princeton: Princeton University Press, 1979
41. Cooper J C. Yin and Yang, the Taoist Harmony of Opposites. Thorsons Date, 1982
42. Govinda L A. The Inner Structure of the I Ching: the Book of Transformation. San Francisco: Wheelwright Press, 1981
43. Whincup G. Rediscovering the I Ching. New York: St. Martin's Press, 1986
44. Birkhoff G. Lattice Theory. 3rd ed. American Mathematical Society Colloquium Publications, 1984, 25
45. Salii V N. Lattices with Unique Complements. American Mathematical Society, 1988, 69
46. Burn R P. Groups: A Path to Geometry. Cambridge: Cambridge University Press, 1985
47. Greutz M. Quarks, Gluons and Lattices. Cambridge: Cambridge University Press, 1983
48. Greenleaf F P. Invariant means on topological groups and their applications. Van Nostrand Mathematical Studies, 1969, 16
49. Fogarty J. Invariant Theory. New York: W. A. Benjamin Inc., 1969
50. Springer T A. Invariant Theory. Heidelberg: Springer-Verlag, 1977
51. Rosser J B, Turquette A R. Many-Valued Logics. Amsterdam: North-Holland Publishing Company, 1952
52. Vickers S. Topology via Logic. Cambridge: Cambridge University Press, 1989