Variant Logic Construction Under Permutation and Complementary Operations on Binary Logic

Jeffrey Zheng

Abstract This chapter presents a binary logic framework whose function elements are invariant under permutation and complementary operations. The entire framework is described using 4 levels of hierarchy: n variables, 2^n states, 2^{2^n} functions, and $2^n!2^n$ logic functionals. Under the proposed framework, it is possible to determine higher level function complexity by analysing lower levels of organisation characteristics. These characteristics can be determined quite accurately because the symmetry conditions of variable and state organisations have invariant logic functions and a corresponding logic functional organisation. More symmetrical arrangement at state level creates more symmetrical permutations within the function space. Lower level properties are highly influential on the higher level properties of function components within a logic functional space. The proposed framework provides a logic foundation to describe complex binary systems using lower level properties, making analysis of systems more efficient and less calculation intensive. Different global coding schemes are discussed and typical two-variable cases of logic functionals are illustrated.

Keywords Vector permutation · Complement · Variant logic · Functional space
Binary logic framework

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018Z1002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)

Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng

Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China

© The Author(s) 2019

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_1

1 Introduction

Mathematical invariance [1, 2] is key in the understanding and development of new scientific theories and technologies [3]. Most scientific theories rely on invariant properties of group behaviour and transformations [4] to describe the rules of the world we live in. Theories such as relativity and quantum mechanics all rely on invariance properties for their constructs [5]. In the field of mathematical logic, construction of theoretical frameworks [6, 7] focus upon three hierarchical levels: variables, states and function spaces. Boolean algebra and switching theory [8, 9] exploit combinatorial invariant properties, and use these foundational properties for implementing new theories and applications.

For reasons of consistency and symmetry of structure, logical operations are restricted to two types of canonical forms namely, the product-of-sums and the sum-of-products approach. Any complex logic function can be rewritten as these two canonical forms. The use of a truth table enables analysis and the transformation into the canonical representations [6].

Following the introduction of Conway's Game of Life [10], Stephan Wolfram from the 1980s [11, 12] started to apply Boolean algebra to describe the behaviour of Cellular Automata. His approach used a binary counting sequence to naming different rules of behaviour based upon the functions generating the next iteration in the game. Wolfram identified four classes of transformations within the rules of Cellular Automata (CA). Results of findings are published in his book [13]—"A New Kind of Science". The main method of analysis in this area of research chooses a CA operation, recursively applying the operation to different initial conditions to find emergent patterns from the process. This approach creates many interesting results that can be visually identified [14, 15].

In the analysis of dynamic systems, it is essential to identify transformation spaces with functional invariance [16, 17]. An example in physics is phase space [2]. The phase space plays an essential role to describe key properties of a given dynamic system. Phase characteristics are more difficult to construct under a logic framework. A mechanism for linking lower level characteristics with higher levels properties such as symmetry currently does not exist. Under combinatorial logic, different permutations add no additional information to access information in phase space [14].

1.1 Western and Eastern Logic Traditions

Beginning with Aristotle (384–322 B.C.), the foundations of Western logic have played a key role in the development of today's global society [18]. The modern theory of logic systems comprise of a series of outstanding individuals and their contributions to the theory of logic: G. Leibniz and the introduction of the Binary Number System (1646–1716) [19, 20]; G. Boole and the development of Boolean

Logic (1854) [21]; G. Cantor and Set Theory (1879); G. Frege and Conceptual Logic (1879) [22, 23]; B. Russell and Russell's Paradox (1910) [24]; J. Lukasiewicz and Multiple-Valued Logic (1920); D. Hilbert and Foundations of Geometric Logic (1923) [25], K. Gödel and his Incomplete Theorem (1931) [22], A. Turing and the Turing Machine (1936) [26]; C. Shannon and Switching Theory (1937) [27]; H. Reichenbach and Probability Logic (1949) [28]; as well as L. Zadeh and Fuzzy Logic (1965) [29]. Development of such theorems and mathematical frameworks have enabled Western culture to understand the operation of our world as a set of implementable rules. Logic and the development of rules for the expression of logic have provided a language that enabled the construction of today's scientific societies.

In contrast to the binary on–off nature of Western logic, Oriental culture have been influenced by spiritual traditions of balance and harmony. The theme of balance can be summarised in the I-Ching or 'The Book of Changes', one of the most influential books of classic Oriental literature [30–37]. The concept of Yin and Yang forces and the subtle interplay of the two opposing forces yield combinations and permutations of change. Orient philosophy believed that 'the only constant phenomena is change' and such a worldview emphasised the dynamic nature of a system; rather than focusing on the individual states of a system (on, off), prominence was instead placed on operations that yield change (on to off, off to on). The structure of thought introduced by the I-Ching allowed change to be systematically documented and analysed. Complex interactions, cyclic behaviour and the interplay of nature at all levels of oriental culture—sociology, literature, medicine, astrology and religion—were able to be described using the tools of dynamic logic provided by the I-Ching; the framework remains a complete philosophy as well as a universal language and has remained unchanged over the past two thousand years [38].

Leibniz in as early as 1690 realised that the balanced yin–yang structure proposed by Shao Yong (1050) was equivalent to the binary number system [33, 38]. However the Western scientific community have mostly disregarded the I-Ching; due mainly to cultural and language barriers as well as local superstitions that cloud the essence of the framework. In its ancient form of allegories and metaphors, the I-Ching is unable to satisfy the logician's requirement for completeness, consistence and other such properties. The challenge then is to be able present this philosophy for modern times, in the language of mathematics. Stripped of its colourful language, what insights does this ancient system contain? What are the essential differences between modern binary logic and the I-Ching's dynamic binary structures? The unification of these two schools of thought would bring greater understanding of the world we live in [35]. As the modern formulation of Cellular Automata generates complexity through binary logic whilst the I-Ching analyses complexity through binary logic, the modern language of the I-Ching can be found in the creation of a structural definition of CA.

1.2 Logic and Dynamic Systems

In the field of mathematical logic, construction of theoretical frameworks focus upon three spatial hierarchies: variables, states and function spaces [6, 7]. Boolean algebra and switching theory exploit such properties, using the combinatorial invariance of the framework for implementing new theories and applications [8, 9]. Logical operations are restricted to two types of canonical forms, namely the product-of-sums and the sum-of-products approaches. Any complex logic function can be rewritten as these two canonical forms. This is done for reasons of consistency, simplicity and symmetry of structure; as such the use of a truth table enables analysis and the transformation into the canonical representations [6].

In the analysis of dynamic systems, it is essential to identify transformation spaces with functional invariance [16, 17]. The Ising model is arguably the simplest binary system that undergoes a nontrivial phase transition [14]. In modern physics, this type of model uses a structure linked to phase space representation of a dynamic systems [2]. The phase space plays an essential role to describe key properties of any dynamic system, however under classical logic, phase characteristics are difficult to construct. A mechanism for linking low-level representations such as variables and states with higher level group properties such as symmetric conditions currently does not exist. This is more a limitation of the language and the operations allowed by the language. Classical logic is based on static combinatorial structures. Permutations, which are intrinsic to phase space, cannot be expressed under such a framework of classical combinatorial logic [14]. Cellular Automata frameworks [39], however, are fully dynamic and have been used to describe phase space [2]. Inspired by the traditional I-Ching hierarchical structures, new conditions, operations and relationships have been proposed on top of the Classical Logic framework to incorporate the dynamic nature of CA. The additional constructs provide support for CA using framework that is logically consistent and complete [40].

The [40] proposal builds upon earlier studies of logic systems from a structural viewpoint. Kunii and Takai [41] applied a n -cell structure for analysis, classification and generation of visual objects using topology and homotopy tools in computer graphics [42–46]. Zheng and Maeder [47] proposed a balanced classification on binary images for conjugate classification and transformation of binary images on regular plan lattices in 1990s to visualise different configurations [15, 48–50]. All such work used partial constructs of the [40] framework. The proposed framework supports classical logic, vector permutation and complementary operations. The new construction requires five spatial hierarchies containing $2^{2^n} \times 2^n!$ functional configurations for any n variables. This structure is much larger than classical logic having three spatial hierarchies supporting 2^{2^n} functions for n variables. Newly defined symmetric properties play an important role in predictions and classifications of possible recursive results. Using such properties, global behaviour can be identified and classified. A disadvantages of the new framework lies in its extreme complexity. It is possible to use parallel computers to do analysis of the configurations contained by $n = 3$ (the space already includes more than 10^7 configurations). It is impossible

using today's technology to process the $n = 5$ space due to the extreme growth of structural complexity ($2^{32} \times 32!$ configurations).

This chapter describes a logic framework, using invariant characteristics of permutations and complementary operations to identify an invariant structure under such mixed operations. This allows the definition of a phase space to be introduced into logic. The transformation does not change the relevant function space. A proposed 2D representation provides additional properties to predict different behaviours from permutations that influence higher level structures in a logic functional space.

2 Truth Table Representation for a Logic Function Space

The proposed framework describes three levels of a logic function space and the truth table representation of the space.

2.1 Basic Definitions

$$\begin{aligned} f : X \rightarrow Y; \quad Y = f(X); \quad X, Y \in B_2^N \\ X = X_{N-1}X_{N-2} \dots X_j \dots X_1X_0, \quad Y = Y_{N-1}Y_{N-2} \dots Y_j \dots Y_1Y_0 \quad (1) \\ X_j, Y_j \in B_2, 0 \leq j < N \end{aligned}$$

An example of a transform: the sequence $X = 0001110100$, $N = 10$ is an input for a function operation f , the output is a sequence of the same length $Y = 1101011001$; $X, Y \in B_2^{10}$.

Definition 1 Let $\dots X_j \dots$ be a n bit structure:

$$\begin{aligned} \dots X_j \dots = x_{n-1}x_{n-2} \dots x_i \dots x_1x_0 = \mathbf{x} \\ 0 \leq i < n, 0 \leq j < N, \mathbf{x} \in B_2^n \end{aligned} \quad (2)$$

where $X_j = x_i$ is a corresponding position.

$$Y_j = f(\dots X_j \dots) = f(x_{n-1}x_{n-2} \dots x_i \dots x_1x_0) = f(\mathbf{x}) \quad (3)$$

In Boolean logic, n variables correspond to a full truth table with $2^n \times 2^n$ entries. The I th meta-state $0 \leq I < 2^n$ has n -bit number to occupy the I th column position, the J th function $T(J)$ has the J th row with 2^n bits $0 \leq J < 2^n$, the function value of the I th entry is determined by $T(J)_I$. The full table can be represented as follows (Table 1):

Table 1 Truth Tables of n -variables

$0 \leq I < 2^n$	S_{2^n-1}	...	S_I	...	S_1	S_0
$I_{n-1} \dots I_i \dots I_0$	$1 \dots 1 \dots 1$	...	$I_{n-1} \dots I_i \dots I_0$	...	$0 \dots 0 \dots 1$	$0 \dots 0 \dots 0$
$0 \leq J < 2^{2^n}$	J_{2^n-1}	...	J_I	...	J_1	J_0
$T(0)$	0	...	0	...	0	0
$T(1)$	0	...	0	...	0	1
$T(2)$	0	...	0	...	1	0
...			...			
$T(J)$	J_{2^n-1}	...	J_I	...	J_1	J_0
...			...			
$T(2^{2^n} - 2)$	1	...	1	...	1	0
$T(2^{2^n} - 1)$	1	...	1	...	1	1

Method 1: Process Method of Truth Table

Input: $\mathbf{x}$: n variables in a $\{0, 1\}$ sequence, J : selected function number

Process: Using the input sequence $\mathbf{x}$, the meta-state number I is to select the I -th column of function $T(J)$

Output: Return $T(J)_I$'s value (1 for true and 0 for false) as output.

2.2 Permutation Invariants

Proposition 1 *Sequential Mapping Under sequential order, $T(J) = J$.*

Proof The relevant output entries of $T(J)$ are mapped to the binary number J having 2^n bits:

$$\begin{aligned}
 T(J) &= T(S_{2^n-1}(J_{2^n-1})) \dots T(S_I(J_I)) \dots T(S_0(J_0)) \\
 &= T(J)_{2^n-1} \dots T(J)_I \dots T(J)_0 = J \in B_2^{2^n} \\
 T(J)_I &= T(S_I(J_I)) = J_I \in B_2; 0 \leq I < 2^n, 0 \leq J < 2^{2^n}
 \end{aligned} \tag{4}$$

■

Definition 2 For any n binary logic variables, let $\Omega(N)$ be a symmetric group with N elements and P be a permutation operator, $P \in \Omega(2^n)$, then for any $J, \exists K, J, K \in B_2^{2^n}$, $P(T(J)) = K, 0 \leq J, K < 2^{2^n}$, the following permutation can be represented in Truth Table form:

$$\begin{aligned}
 P : J &\rightarrow K \\
 P(T(J)) &= P(T(S_{2^n-1}(J_{2^n-1}))) \dots P(T(S_I(J_I))) \dots P(T(S_0(J_0))) \\
 &= P(T(J)_{2^n-1}) \dots P(T(J)_I) \dots P(T(J)_0) \\
 &= K_{2^n-1} \dots K_I \dots K_0 = K \in B_2^{2^n} \\
 P(T(J)_I) &= P(T(S_I(J_I))) = T(S_{P(I)}(J_{P(I)})) \\
 &= T(J)_{P(I)} = J_{P(I)} = K_I \in B_2 \\
 0 \leq I &< 2^n, 0 \leq J, K < 2^{2^n}, P \in \Omega(2^n)
 \end{aligned} \tag{5}$$

Proposition 2 *The Truth Table under permutation operation on 2^n meta-states can generate $2^n!$ sequences for 2^{2^n} length of integers.*

Proof For any $P \in \Omega(2^n)$, 2^n are independent, it is composed of $\Omega(2^n)$ elements. ■

For the one-variable condition (i.e. $n = 1$), there are only two possible arrangements. The initial sequence is represented as $\mathbf{S} = S_1 S_0 = 10$, and a permutation operation generates the output $P(\mathbf{S}) = S_0 S_1 = 01$. The following shows two groups of results:

Mate-state	$\mathbf{S}$	1	0	$P(\mathbf{S})$	0	1
Function	J			$P(J)$		
0	0	0	0	0	0	0
$\bar{x}$	1	0	1	2	1	0
x	2	1	0	1	0	1
1	3	1	1	3	1	1

For any permutation operation, the function $T(J) = P(T(J))$ is always invariant. The inequality $J \neq K = P(J)$ holds in general.

3 Fourth Level of Organisation

Building upon the three levels (variables, states and functions), a fourth level of organisation is introduced.

3.1 Complementary Operation

Definition 3 Complementary Operator, for any binary (0–1) variable $y \in B_2$, let the relevant index $\delta \in B_2$ be a complementary operator:

$$y^\delta = \begin{cases} \bar{y} & \delta = 0 \\ y & \delta = 1 \end{cases} \quad (6)$$

Definition 4 Complementary Function Operation, for any n variable function of 2^n meta function vectors $\mathbf{S} = S_{2^n-1} \dots S_I \dots S_0$. Let $\Delta = \delta_{2^n-1} \dots \delta_I \dots \delta_0$, $0 \leq I < 2^n$, $\delta_I \in B_2$, $\Delta \in B_2^{2^n}$.

For this type of complementary operations on function, Δ is

$$\begin{aligned}
\Delta : T(J) &\rightarrow K; J, K \in B_2^{2^n}, 0 \leq J, K < 2^{2^n} \\
\mathbf{S}^\Delta &= S_{2^n-1}^{\delta_{2^n-1}} \dots S_I^{\delta_I} \dots S_0^{\delta_0}, S_I \in B_2^{2^n} \\
T(J)^\Delta &= T(S_{2^n-1}^{\delta_{2^n-1}}(J_{2^n-1})) \dots T(S_I^{\delta_I}(J_I)) \dots T(S_0^{\delta_0}(J_0)) \\
&= T(J)_{2^n-1}^{\delta_{2^n-1}} \dots T(J)_I^{\delta_I} \dots T(J)_0^{\delta_0} \\
&= K_{2^n-1} \dots K_I \dots K_0 = K \in B_2^{2^n} \\
T(J)_I^{\delta_I} &= T(S_I^{\delta_I}(J_I)) = J_I^{\delta_I} = K_I \in B_2 \\
0 \leq I &< 2^n, 0 \leq J, K < 2^{2^n}, \delta_I \in \Delta
\end{aligned} \tag{7}$$

3.2 Invariant Logic Functions Under Permutation and Complementary

Definition 5 Permutation and Complementary Operations. For any of the n variables expressed as 2^n meta vectors, Complementary Operations $\Delta \in B_2^{2^n}$ and Permutation Operations $P \in \Omega(2^n)$ are expressed as

$$\begin{aligned}
(P, \Delta) : T(J) &\rightarrow K; J, K \in B_2^{2^n}, P \in \Omega(2^n), \Delta \in B_2^{2^n} \\
P(T(J)^\Delta) &= P(T(S_{2^n-1}^{\delta_{2^n-1}}(J_{2^n-1}))) \dots P(T(S_I^{\delta_I}(J_I))) \dots P(T(S_0^{\delta_0}(J_0))) \\
&= P(T(J)_{2^n-1}^{\delta_{2^n-1}}) \dots P(T(J)_I^{\delta_I}) \dots P(T(J)_0^{\delta_0}) \\
&= K_{2^n-1} \dots K_I \dots K_0 = K \in B_2^{2^n} \\
P(T(J)_I^{\delta_I}) &= P(T(S_I^{\delta_I}(J_I))) = J_{P(I)}^{\delta_{P(I)}} = K_I \in B_2 \\
0 \leq I &< 2^n, 0 \leq J, K < 2^{2^n}, P \in \Omega(2^n), \delta_I \in \Delta
\end{aligned} \tag{8}$$

3.3 Logic Functional Spaces

Theorem 1 (Logic Function Invariants under Permutation & Complementary Operations) *For any logic function, the output of Method 2 provides an equivalent output as the original Truth Table under all conditions.*

Proof A J th row on the permutation and complementary table of $P(T^\Delta)$ for any $I \in B_2^n, J \in B_2^{2^n}$ is constructed by

$$P(T(J)_I^\Delta) = T(J)_{P(I)}^{\delta_{P(I)}} = \begin{cases} \neg T(J)_I & \delta_{P(I)} = 0 \\ T(J)_I & \delta_{P(I)} = 1 \end{cases} \tag{9}$$

Counting Order	7	6	5	4	3	2	1	0	
S	111	110	101	100	011	010	001	000	Binary counting
0	0	0	0	0	0	0	0	0	a full 0 vector
Δ	1	1	0	0	1	1	0	0	a Δ - vector
$\neg\Delta$	0	0	1	1	0	0	1	1	a not Δ - vector
1	1	1	1	1	1	1	1	1	a full 1 vector
$T(178)$	1	0	1	1	0	0	1	0	initial value
$T(178)^1$	1	0	1	1	0	0	1	0	$T(178)$ Truth
$T(178)^{\neg\Delta}$	0	1	1	1	1	1	1	0	$T(178)$ Δ -Variant
$T(178)^0$	0	1	0	0	1	1	0	1	$T(178)$ False
$T(178)^\Delta$	1	0	0	0	0	0	0	1	$T(178)$ Δ -Invariant

Method 2: Permutation and Complementary Methods Table $P(T^\Delta)$

Input: $\mathbf{x}$: n variables in a binary $\{0, 1\}$ sequence, J : is the selected function number, $P \in \Omega(2^n)$ and $\Delta \in B_2^{2^n}$ are Permutation and Complementary operators

Process: Input sequence $\mathbf{x}$ is established, the $P(I)$ -th column is selected using the meta-state number I . This represents the I -th column of the function $P(T(J)^\Delta)$

Output: If $\delta_{P(I)} = 1$, return the value of $T(J)_{P(I)}^{\delta_{P(I)}}$ (1 for true and 0 for false);
if $\delta_{P(I)} = 0$, return $\neg T(J)_{P(I)}^{\delta_{P(I)}}$.

After using Method 2, the results are shown:

$$P(T(J)_I^\Delta) = \begin{cases} \neg\neg T(J)_I = T(J)_I & \delta_{P(I)} = 0 \\ T(J)_I & \delta_{P(I)} = 1 \end{cases} \quad (10)$$

■

Theorem 2 (Permutation Group for Meta Function Vector) *For 2^n meta function vectors, a total of permutation numbers is $2^n!$.*

Theorem 3 (Permutation & Complementary Structure) *Under permutation and complementary operations, a total of $2^n!2^{2^n}$ permutations can be generated to form a logic functional space for the n variables.*

4 Different Coding Schemes: One- and Two-Dimensional Representations

The initial step to construct a series of logic functionals. Permutation and complementary differences can be shown in the proposed invariant function structures. Different coding schemes under different symmetric restrictions are established. Four schemes are described, in which one of them is in one-dimensional representation and other three schemes are two-dimensional representations. For binary sequences in sequential counting order, the scheme is known as the SL (Shao Yong & Leibniz) coding scheme.

4.1 G Coding

The General Code (G) is used to map permutation & complementary operations. For any state in the G coding scheme having 2^n bits,

$$G : (J, \Delta, P) \rightarrow K; J, K \in B_2^{2^n}; \Delta \in B_2^{2^n}, P \in \Omega. \quad (11)$$

4.2 W Coding

From the G coding scheme, their bit numbers are separated into two equal parts in the same bits to form a 2D representation. This mapping mechanism can represent a function space as a W coding scheme.

$$W : (J, \Delta, P) \rightarrow K = \langle J^1 | J^0 \rangle \quad (12)$$

$$J, K \in B_2^{2^n}; J^1, J^0 \in B_2^{2^{n-1}}; S^1, S^0 \in \mathbf{S}, \Delta \in B_2^{2^n}, P \in \Omega$$

Under this representation, a given logic functional for the function space is illustrated as a fixed matrix.

$$\{W(J)\}_{J=0}^{2^{2^n}} = \begin{array}{ccccc} \langle 0|0 \rangle & \dots & \langle 0|J^0 \rangle & \dots & \langle 0|2^{2^{n-1}} - 1 \rangle \\ \dots & & \dots & & \dots \\ \langle J^1|0 \rangle & \dots & \langle J^1|J^0 \rangle & \dots & \langle J^1|2^{2^{n-1}} - 1 \rangle \\ \dots & & \dots & & \dots \\ \langle 2^{2^{n-1}} - 1|0 \rangle & \dots & \langle 2^{2^{n-1}} - 1|J^0 \rangle & \dots & \langle 2^{2^{n-1}} - 1|2^{2^{n-1}} - 1 \rangle \end{array} \quad (13)$$

$$0 \leq J^0, J^1 < 2^{2^{n-1}}; 0 \leq J < 2^{2^n}$$

In the one-variable condition, there are eight cases in their logic functional spaces as follows:

f	J^{11}, T	W	$J^{01}, \Delta-V$	W	$J^{10}, \Delta-IV$	W	J^{00}, F	W
0	0	$\langle 0 0 \rangle$	2	$\langle 1 0 \rangle$	1	$\langle 0 1 \rangle$	3	$\langle 1 1 \rangle$
$\bar{x}$	1	$\langle 0 1 \rangle$	3	$\langle 1 1 \rangle$	0	$\langle 0 0 \rangle$	2	$\langle 1 0 \rangle$
x	2	$\langle 1 0 \rangle$	0	$\langle 0 0 \rangle$	3	$\langle 1 1 \rangle$	1	$\langle 0 1 \rangle$
1	3	$\langle 1 1 \rangle$	1	$\langle 0 1 \rangle$	2	$\langle 1 0 \rangle$	0	$\langle 0 0 \rangle$
f	$P(J^{11}), T$	W	$P(J^{01}), \Delta-V$	W	$P(J^{10}), \Delta-IV$	W	$P(J^{00}), F$	W
0	0	$\langle 0 0 \rangle$	1	$\langle 0 1 \rangle$	2	$\langle 1 0 \rangle$	3	$\langle 1 1 \rangle$
$\bar{x}$	2	$\langle 1 0 \rangle$	3	$\langle 1 1 \rangle$	0	$\langle 0 0 \rangle$	1	$\langle 0 1 \rangle$
x	1	$\langle 0 1 \rangle$	0	$\langle 0 0 \rangle$	3	$\langle 1 1 \rangle$	2	$\langle 1 0 \rangle$
1	3	$\langle 1 1 \rangle$	2	$\langle 1 0 \rangle$	1	$\langle 0 1 \rangle$	0	$\langle 0 0 \rangle$

For better visualisation and expression, the one-dimensional G coding scheme is converted into a two-dimensional W coding scheme.

$$W = \begin{array}{|c|c|} \hline \text{Truth} & \Delta\text{-Variant} \\ \hline 0 \bar{x} & x 1 \\ x 1 & 0 \bar{x} \\ \hline \bar{x} 0 & 1 x \\ 1 x & \bar{x} 0 \\ \hline \Delta\text{-Invariant} & \text{False} \\ \hline \end{array} \quad PW = \begin{array}{|c|c|} \hline \text{Truth} & \Delta\text{-Variant} \\ \hline 0 x & x 0 \\ \bar{x} 1 & 1 \bar{x} \\ \hline \bar{x} 1 & 1 \bar{x} \\ 0 x & x 0 \\ \hline \Delta\text{-Invariant} & \text{False} \\ \hline \end{array}$$

4.3 F Coding

Using 2D representation, symmetric condition can be added to arrange meta-states into specific order. For each pair of states in W, if they satisfy following condition, then a refined code: F coding scheme is determined.

$$\begin{array}{ccc} J^1 \text{ the } I\text{th meta-state} & \Leftrightarrow & J^0 \text{ the } I\text{th meta-state} \\ \updownarrow & \text{F coding scheme} & \updownarrow \\ X \in S^1 & \Rightarrow & \bar{X} \in S^0 \end{array}$$

4.4 C Coding

In addition to a pair of states in complementary relationship, further structure is introduced onto F code. When the pair of states in F have the same values in their i th position, they form a C coding scheme.

$$\begin{array}{ccc} S^1 \text{ the } I\text{th} & \Leftrightarrow & S^0 \text{ the } I\text{th} \quad \text{F coding scheme} \\ \updownarrow & \text{C coding scheme} & \updownarrow \\ \forall x_i \in S^1, x_i = 1(0) & \Rightarrow & \forall x_i \in S^0, x_i = 0(1) \quad \text{general conjugate} \end{array}$$

The C coding scheme, have the strongest symmetric conditions available. Only a relatively small number among the three invariant groups can be identified within this scheme.

5 Two-Variable Cases

Four groups of the proposed schemes are selected as examples. Each group of a logic functional represents 16 logic functions as 4×4 images. 4 groups are arranged as 2×2 blocks to arrange as Truth/False, Δ -Variant/ Δ -Invariant properties. The 2×2 blocks correspond to:

Truth Block	Δ -Variant
Δ - Invariant	False Block

Each block contains 16 entries of function images as a 4×4 ($2^2 \times 2^2$) configuration. Each image entry denotes a transformed number and its

function number in the form: $\begin{bmatrix} \langle J^1 | J^0 \rangle \\ J \end{bmatrix}$ where $K = \langle J^1 | J^0 \rangle$ is a transformed number and J is the function number. In all four figures, (a) 2×2 base blocks to represent function images and (b) 2×2 vector blocks to represent relevant coding schemes respectively.

In Fig. 1, the counting order of meta-states has been arranged as W coding (SL code): $P = (3210)$, $P(\Delta) = 1010$. In this group, only Functions 6 and 9 can be observed in complementary symmetric condition in main diagonal direction.

In Fig. 2, variation the configurations among W coding: $P = (2301)$, $P(\Delta) = 0101$ creates similar effects seen in Fig. 1.

In Fig. 3, the F coding scheme is shown: under this configuration, $P = (2310)$, $P(\Delta) = 0110$. Six pairs (0:15, 1:7, 2:11, 4:13, 6:9, 8:14) of complementary functions can be identified. The group has four blocks containing the same pairs of configurations.

In Fig. 4, C coding has represented: $P = (3102)$, $P(\Delta) = 1100$. In addition to six pairs as same as F coding, four corners are 4 functions (0, 5, 10, 15) in all blocks. This makes most regular structures compared to all other coding schemes.

Fig. 1 W coding (SL code): $P = (3210)$, $P(\Delta) = 1010$; **a** 2×2 base blocks **b** 2×2 vector blocks

Fig. 2 W coding: $P = (2301)$, $P(\Delta) = 0101$; **a** 2×2 base blocks **b** 2×2 vector blocks

Fig. 3 F coding: $P = (2310)$, $P(\Delta) = 0110$; **a** 2×2 base blocks **b** 2×2 vector blocks

Fig. 4 C coding: $P = (3102)$, $P(\Delta) = 1100$; **a** 2×2 base blocks **b** 2×2 vector blocks

6 Conclusion

It is shown in this chapter that the arrangement of binary function space using four levels of classification can be used to add symmetry and regular structure onto the entire space of binary functions. For ease of visualisation, it is convenient to apply 2D representation mechanism that enables symmetric configurations of the system to be analysed via different coding schemes. Binary functional spaces provide additional optimal information to generate large numbers of potential configurations in order to arrange and organise logic phase spaces.

The mechanism can be developed further to establish a solid logic foundation on logic functional levels for theoretical explorations and practical applications. We aim to make refined investigation on different coding schemes within the highest levels of organisation in our future work.

Acknowledgements Thanks Mr. J. Wan for generation all sample images and configurations and Dr. D. Heim for editing the chapter. Financial support was given by School of Software, Yunnan University.

References

1. T.A. Springer, *Invariant Theory* (Springer, Berlin, 1977)
2. A. Holden, *Shapes, Spaces and Symmetry* (Columbia University Press, New York, 1971)
3. G. Birkhoff, *Lattice Theory*, vol. 25, 3rd edn. (American Mathematical Society Colloquium Publications, 1984)
4. R.P. Burn, *Groups: A Path to Geometry* (Cambridge University Press, Cambridge, 1985)
5. H. Weyl, *Symmetry* (Princeton, 1952)
6. M. Bonnet, *Handbook of Boolean Algebras* (North Holland, 1989)
7. R. Sikorski, *Boolean Algebra* (Springer, Berlin, 1960)
8. S. Lee, *Modern Switching Theory and Digital Design* (Prentice-Hall Inc., Englewood Cliffs, 1978)
9. S. Vingron, *Switching Theory: Insight Through Predicate Logic* (Springer, Berlin, 2004)
10. H. Umeo, S. Morishita, K. Nishinari, *Cellular Automata* (Springer, Berlin, 2008)
11. S. Wolfram, *Theory and Applications of Cellular Automata* (World Scientific, Singapore, 1986)
12. S. Wolfram, *Cellular Automata and Complexity* (Addison-Wesley, New York, 1994)
13. S. Wolfram, *A New Kind of Science* (Wolfram Media Inc., Champaign, 2002). <http://www.wolframscience.com/>
14. A. Ilachinski, *Cellular Automata—A Discrete Universe* (World Scientific, Singapore, 2001)
15. Z. Zheng, C. Leung, Visualising global behaviour of 1d cellular automata image sequences in 2d maps. *Phys. A* **233**, 785–800 (1996)
16. P. Dunn, *The Complexity of Boolean Networks* (Academic Press, New York, 1988)
17. M. Paterson, *Boolean Function Complexity* (Cambridge University Press, Cambridge, 1992)
18. M. Kline, *Mathematical Thought from Ancient to Modern Times* (Oxford University Press, Oxford, 1972)
19. G. Leibniz, *Philosophical Papers and Letters* (Springer, Berlin, 1976)
20. G. Leibniz, R. Ariew, D. Garber, *Philosophical Essays* (Hackett Publishing, 1989)
21. G. Boole, *An Investigation of the Laws of Thought* (Dover, 1850/1940/1958)
22. J. Dawson, *Logical Dilemmas—The Life and Work of Kurt gödel* (A.K. Peters, Ltd., Wellesley, 2005)

23. W. Demopoulos, *Frege's Philosophy of Mathematics* (Harvard University Press, Cambridge, 1995)
24. B. Russell, *Principles of Mathematics* (Forgotten Books, 1942)
25. D. Hilbert, *Grundlagen der Geometrie* (Göttingen, 1899)
26. A. Turing, On computable numbers, with an application to the entscheidungs problem. Proc. Lond. Math. Soc. Ser. 2 **42**, 230–265 (1936)
27. C. Shannon, N. Sloane, A. Wyner, *Claude Elwood Shannon: Collected Papers* (IEEE Press, IEEE Information Theory Society, 1993)
28. H. Reichenbach, *The Theory of Probability* (The University of California Press, Berkeley, 1949)
29. L. Zadeh, Fuzzy sets. Information and Control **8**, 338–357 (1965)
30. W. Chu, W. Sherrill, *An Anthology of I Ching* (Routledge and Kegan Paul Ltd., London, 1977)
31. J. Cooper, Yin and Yang, The Taoist Harmony of Opposites (The Thetford Press, 1981)
32. L. Govinda, *The Inner Structure of I Ching: The Book of Transformation* (Wheelwright Press, 1981)
33. D. Hook, *The I Ching and Mankind* (Routledge and Kegan Paul Ltd., London, 1975)
34. I. Shchutshii, *Researches on the I Ching* (Princeton University Press, Princeton, 1979)
35. G. Whincup, *Rediscovering of I Ching* (GreyWhincup, 1986)
36. H. Wilhelmi, *Change: Eight Lectures on I Ching* (Princeton University Press, Princeton, 1979)
37. R. Wilhemi, *Lectures on the I Ching: Constancy and Change* (Princeton University Press, Princeton, 1979)
38. J. Needham, L. Wang, *Science and Civilisation in China: History of Scientific Thought* (Cambridge University Press, Cambridge, 1954–1988), p. 2
39. D. Griffeth, C. Moor, *New constructions in cellular automata*, Santa Fe Institute Studies in the Sciences of Complexity (2003)
40. J. Zheng, C. Zheng, A framework to express variant and invariant functional spaces for binary logic. Front. Electr. Electron. Eng. China **5**(2), 163–172 (2010) (Higher Education Press and Springer). <http://www.springerlink.com/content/91474403127n446u/>
41. T. Kunii, Y. Takai, Cellular self-reproducing automata as a parallel processing model for botanical colony growth pattern simulation, in *New Advances in Computer Graphics* (Springer, Berlin, 1989), pp. 7–22
42. T. Kunii, H. Hioki, Y. Shinagawa, Visualizing Highly Abstract Mathematical Concepts: A Case Study in Animation of Homology Groups, in *Multimedia Modeling* (World Scientific, Singapore, 1993), pp. 3–30
43. T. Kunii, H. Kunii, A cellular model for information systems on the web—integrating local and global information, in *Proceedings of 1999 International Symposium on Database Applications in Non-Traditional Environments* (IEEE CS Press, 1999)
44. T. Kunii, Y. Shinagawa, *Modern Geometric Computing for Visualization* (Springer, Berlin, 1992)
45. T. Kunii, S. Takahashi, Area guide map modeling by manifolds and CW-complexes, in *Modeling in Computer Graphics* (Springer, Berlin, 1993), pp. 5–20
46. T. Kunii, Y. Tsuchida, Y. Arai, H. Matsuda, M. Shirahama, S. Miura, A model of hands and arms based on manifold mappings, in *Communicating with Virtual Worlds* (Springer, Berlin, 1993), pp. 381–398
47. Z. Zheng, A. Maeder, The conjugate classification of the kernel form of the hexagonal grid, in *Modern Geometric Computing for Visualization* (Springer, Berlin, 1992), pp. 73–89
48. Z. Zheng, Conjugate transformation of regular plan lattices for binary images, Ph.D. Thesis, Monash University, 1994
49. Z. Zheng, Conjugate visualisation of global complex behaviour. Complex. Int. **3** (1996). <http://www.complexity.org.au/ci/vol03/zheng/>
50. Z. Zheng, A. Maeder, The elementary equation of the conjugate transformation for hexagonal grid, in *Modeling in Computer Graphics* (Springer, Berlin, 1993), pp. 21–42

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.