16

A Framework of Variant Logic Construction for Cellular Automata

Jeffrey Z.J. Zheng[1], Christian H.H. Zheng[2] and Tosiyasu L. Kunii[3]

[1]*Yunnan University*
[2]*University of Melbourne*
[3]*University of Tokyo*
[1]*P.R. China*
[2]*Australia*
[3]*Japan*

1. Introduction

1.1 Cellular automata

Cellular Automata are regular uniform networks of locally-connected finite-state machines invented by von Neumann (1966) to describe dynamic properties of finite state logic machines. Following the introduction of Conway's Game of Life (Umeo et al., 2008), Wolfram (1986; 1994) applied Boolean algebra to describe the behaviour of Cellular Automata (CA) as a series of dynamic images. His approach used a binary counting sequence to code different rules of behaviour based upon the functions generating the next iteration in the game. Wolfram (2002) identified four classes of transformations within the rules of CA, proclaiming their discovery as "A New Kind of Science", the title of the book. The typical analysis of behaviour in this area of research would begin by choosing a CA operation; by recursively applying the operation to different initial conditions, emergent patterns are identified, creating interesting visuals that are identifiable by behavioural type (Ilachinski, 2001).

After sixty years of research, Cellular Automata are now ubiquitous with non-trivial behaviour; they have been incorporated into mathematical computational models as well as models of natural systems. Cutting edge research using CA include: digital physics and modeling of spatially extended non-linear systems; complex systems, dynamic systems, massive-parallel computing, parallel implementations, language acceptance, and computability; reversibility of computation, system biology, modeling for real phenomena, natural computing, graph-theoretic analysis and logic; chaos and undecidability (Adamatzky, 2008; Schiff, 2007; Wuensche & Lesser, 1992).

Today's scientists rely on computation models as analytical tools for studying reality; using the computational framework provided by packages such as Mathematica and MATLAB to build recursive models that are iterated countless times before emergent behaviour is observed (Wolfram, 2002). Wolfram (1985) and other researchers have systematically applied logical equations to CA in order to systemise such complex dynamics into a science of complexity. Wolfram (1985) summarised twenty questions enquiring into the determination of mathematical properties for CA such as: entropy and Lyapunov exponents, geometric analogue, statistical behaviour, scaling, language complexity, universality and undecidability, irreducibility, & high-level descriptions. All of the questions are excellent from computational viewpoints, but they have not focused on the foundational aspects of the framework. An additional question remains to be answered: What is the governing relationship between the Cellular Automata framework and that of the Classical Logic framework?

1.2 Western and eastern logic traditions

Begining with Aristotle (384-322 B.C.), the foundations of western logic have played a key role in the development of today's global society (Kline, 1972). 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) [Leibniz (1976); Leibniz et al. (1989)]; G. Boole and the development of Boolean Logic (1854) [Boole (1850/1940/1958)]; G. Cantor and Set Theory (1879); G. Frege and Conceptual Logic (1879) [Dawson (2005); Demopoulos (1995)]; B. Russell and Russell's Paradox (1910) [Russell (1942)]; J. Lukasiewicz and Multiple-Valued Logic (1920); D. Hilbert and Foundations of Geometric Logic (1923 [Hilbert (1899)], K. Gödel and his Incomplete Theorem (1931) [Dawson (2005)], A. Turing and the Turing Machine (1936) [Turing (1936)]; C. Shannon and Switching Theory (1937) [Shannon et al. (1993)]; H. Reichenbach and Probability Logic(1949) [Reichenbach (1949)]; as well as L. Zadeh and Fuzzy Logic (1965) [Zadeh (1965)]. 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 summarized in the I-Ching or 'The Book of Changes', one of the most influential books of classic oriental literature (Chu & Sherrill, 1977; Cooper, 1981; Govinda, 1981; Hook, 1975; Shchutshii, 1979; Whincup, 1986; Wilhelmi, 1979; Wilhemi, 1979). 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 world view emphasised the dynamic nature of a system; rather than focusing in 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 (Needham & Wang, 1954-1988).

Leibniz in as early as 1690 realized that the balanced yin-yang structure proposed by Shao Yong (1050) was equivalent to the binary number system (Hook, 1975; Needham & Wang, 1954-1988). 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 colorful 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 (Whincup, 1986). As the modern formulation of Cellular Automata generates complexity through binary logic whilst the I-Ching analyses complexity though binary logic, the modern language of the I-Ching can be found in the creation of a structural definition of CA.

1.3 Logic and dynamic systems

In the field of mathematical logic, construction of theoretical frameworks focus upon three spatial hierarchies: variables, states and function spaces (Bonnet, 1989; Sikorski, 1960). Boolean algebra and switching theory exploit such properties, using the combinatorial invariance of the framework for implementing new theories and applications (Lee, 1978; Vingron, 2004). 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 (Bonnet, 1989).

In the analysis of dynamic systems, it is essential to identify transformation spaces with functional invariance (Dunn, 1988; Paterson, 1992). The Ising model is arguably the simplest binary system that undergoes a nontrivial phase transition (Ilachinski, 2001). In modern physics, this type of model uses a structure linked to phase space representation of a dynamic systems (Griffeath & Moor, 2003). 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 (Ilachinski, 2001). Cellular Automata frameworks however, are fully dynamic and has been used to describe phase space (Griffeath & Moor, 2003). Inspired by the traditional I-Ching hierarchical structures, new conditions, operations and relationships have been proposed on top of the Classical Logic framework to encorporate the dynamic nature of CA. The additional constructs provide support for CA using framework that is logically consistent and complete (Zheng & Zheng, 2010).

The Zheng & Zheng (2010) proposal builds upon earlier studies of logic systems from a structural viewpoint. Kunii & Takai (1989) applied a n-cell structure for analysis, classification and generation of visual objects using topology and homotopy tools in computer graphics (Kunii, Hioki & Shinagawa, 1993; Kunii & Kunii, 1999; Kunii & Shinagawa, 1992; Kunii & Takahashi, 1993; Kunii, Tsuchida, Arai, Matsuda, Shirahama & Miura, 1993). Zheng & Maeder (1992) 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 (Zheng, 1994; 1996; Zheng & Leung, 1996; Zheng & Maeder, 1993). All such work used partial constructs of the Zheng & Zheng (2010) 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 the variant logic framework proposed by Zheng & Zheng (2010), identifying variant and invariant characteristics of logic under permutations and complementary operations from CA. This allows the definition of a variant space to be introduced into logic. Using an extended truth-valued table, vector permutations and complementary operations can be applied to form a giant structure with equivalent properties in a spatial hierarchy. The framework supports additional operations whithout changing the logic function space. A proposed 2D matrix representation provides additional support to visualise globally symmetric patterns from permutations of generated using the proposed extensions.

2. Truth table in boolean logic

The truth-table plays a vital role in traditional logic construction; It provides a static structure, using Yes|No (1|0) to indicate possible conditions from variables, states to any function. The proposed framework includes traditional logic function space and the truth-table representation of the space.

2.1 Basic definitions

$$\begin{gathered} X = X_{N-1}X_{N-2}...X_j...X_1X_0, \quad Y = Y_{N-1}Y_{N-2}...Y_j...Y_1Y_0 \\ X_j, Y_j \in B_2 = \{0,1\}, 0 \leqslant j < N \\ f : X \to Y; \quad Y = f(X); \quad X, Y \in B_2^N = \{0,1\}^N \end{gathered} \tag{1}$$

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 2.1 Let $[...X_j...]$ be a n bit structure as a kernel form:

$$\begin{gathered} [...X_j...] = x_{n-1}x_{n-2}...x_i...x_1x_0 = \mathbf{x} \\ 0 \leqslant i < n, 0 \leq j < N, \mathbf{x} \in B_2^n \end{gathered} \tag{2}$$

where $X_j = x_i$ is a corresponding position.

$$Y_j = f([...X_j...]) = f(x_{n-1}x_{n-2}...x_i...x_1x_0) = f(\mathbf{x}) \tag{3}$$

In Boolean logic, n variables in a kernel form correspond to a full truth table with $2^n \times 2^{2^n}$ 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 $0 \leq J < 2^{2^n}$, the function value of the I-th entry is determined by $T(J)_I$. The full table can be represented as follows:
From this type of tables, it is feasible to establish accessing method to look up corresponding table values.
Method 2.1: Process Method of Truth Table:
Input: $\mathbf{x}$: n variables in a $\{0,1\}$ sequence, J: selected function number

$0 \leq I < 2^n$	S_{2^n-1}	...	S_I	...	S_1	S_0
$I_{n-1}...I_i...I_0$	1...1...1	...	$I_{n-1}...I_i...I_0$	...	0...0...1	0...0...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

Table 1. Truth Tables of n-variables

Process: Using the input sequence **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.

3. Cellular automata representations

3.1 Basic representation

Cellular Automata - CA uses a recursive mechanism to represent a given function with a time direction. Dynamic properties of CA can be supported in further expansions. In a one dimensional form of CA, a N length binary sequence is

$$X = X_{N-1}X_{N-2}...X_j...X_1X_0, 0 \leqslant j < N, X_j \in \{0,1\} = B_2$$

For a given function f, the output sequence is defined: $f : X \to Y, Y = f(X)$,

$$Y = Y_{N-1}Y_{N-2}...Y_j...Y_1Y_0, 0 \leqslant j < N, Y_j \in B_2$$

It is feasible to use a moving window with a fixed length n to separate X into a local kernel in length n. The kernel can be presented as

$$[...X_j...] = x_{n-1}...x_i...x_0, x_i \in B_2$$

For a given function f

$$y = f(x_{n-1}...x_i...x_0)$$

It is necessary to assign a certain position i in the kernel for special care to associated with j position of both sequences. All above relations are exactly same as traditional Boolean equation with n variables.
It is possible to distinguish current time and next time sequences, following equations relevant to cellular automata can be identified:

$$y = f(x_{n-1}...x_i...x_0) = f([...X_j...]) = Y_j$$

or $X_j = X_j^{t-1}, Y_j = X_j^t$ i.e.

$$f : X_j^{t-1} \to X_j^t, X_j^{t-1}, X_j^t \in B_2 \tag{4}$$

3.2 Four variation forms

Time direction is a significant property to distinguish a Cellular Automata logic function from a traditional logic function. Considering $f : X_j^{t-1} \to X_j^t$ for any function of boolean logic system to analyze their variation properties, it is normal to have following proposition.

Proposition 3.1 For any $f : X_j^{t-1} \to X_j^t$ transformation, four forms of transforming classes are identified: TA: $0 \to 0$, TB: $0 \to 1$, TC: $1 \to 0$, TD: $1 \to 1$.

Proof: X_j, Y_j are 0-1 variables, only four classes listed are possible. ■

Definition 3.1 Four Transforming forms are corresponding to following sets: TA: Invariant-valued class for 0 value, TB: Variant-valued class for 0 value, TC:Variant-valued class for 1 value, TD: Invariant-valued class for 1 value. Under such definition, following proposition can be established.

Proposition 3.2 Using four classes of transformation, four variant operations are defined.

Type	X_j	$\to Y_j$	Truth	Variant	Invariant	False
TA	0	0	0	0	1	1
TB	0	1	1	1	0	0
TC	1	0	0	1	0	1
TD	1	1	1	0	1	0

Proof: Truth (False) values are determined by $Y_j(\bar{Y}_j)$ and Variant(Invariant) values are determined by {TB, TC} for 1(0) and {TA, TD} for 0(1) respectively.■

4. Permutation invariants

From an accessing viewpoint, many invariant properties can be observed from table operation.

Proposition 4.1 Under sequential mapping in sequential order of Method 2.1, there is $T(J) = J$.

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

$$\begin{aligned} T(J) &= T(S_{2^n-1}(J_{2^n-1}))...T(S_I(J_I))...T(S_0(J_0)) \\ &= T(J)_{2^n-1}...T(J)_I...T(J)_0 = J \in B_2^{2^n} \\ T(J)_I &= T(S_I(J_I)) = J_I \in B_2; 0 \le I < 2^n, 0 \le J < 2^{2^n} \end{aligned} \tag{5}$$

■

It is possible to apply permutation operation on the table to generate a transformed table following a certain rule.

Definition 4.1 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 $J, \exists K, J, K \in B_2^{2^n}, P(T(J)) = K, 0 \le J, K < 2^{2^n}$, the following permutation can be represented in Truth Table form:

$$
\begin{aligned}
P &: J \rightarrow K \\
P(T(J)) &= P(T(S_{2^n-1}(J_{2^n-1})))...P(T(S_I(J_I)))...P(T(S_0(J_0))) \\
&= P(T(J)_{2^n-1})...P(T(J)_I)...P(T(J)_0) \\
&= K_{2^n-1}...K_I...K_0 = K \in B_2^{2^n} \\
P(T(J)_I) &= P(T(S_I(J_I))) = T(S_{P(I)}(J_{P(I)})) \\
&= T(J)_{P(I)} = J_{P(I)} = K_I \in B_2 \\
&0 \leq I < 2^n, 0 \leq J, K < 2^{2^n}, P \in \Omega(2^n)
\end{aligned}
\tag{6}
$$

Proposition 4.2 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 (ie. $n = 1$) 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	**S**	1	0	P(**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.

5. Organisational space

Building upon the three spaces (variables, states, and functions), an additional space of organisation is composed of permutation operations provided by permutation invariance properties defined in the previous section.

5.1 Complementary operation

Definition 5.1 (Complementary Operator) For any binary (0-1) variable $y \in B_2$, let the relevant index $\delta \in B_2$ be a complementary operator:

$$
y^\delta = \begin{cases} \bar{y} & \delta = 0 \\ y & \delta = 1 \end{cases}
\tag{7}
$$

Definition 5.2 (Complementary Function Operation) For any n variable function of 2^n meta function vectors $\mathbf{S} = S_{2^n-1}...S_I...S_0$ Let $\Delta = \delta_{2^n-1}...\delta_I...\delta_0, 0 \leq I < 2^n, \delta_I \in B_2, \Delta \in B_2^{2^n}$.
For this type of complementary operations on function,

Δ is :

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

5.2 Invariant logic functions under permutation and complementary

Definition 5.3 (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:

$$\begin{aligned}
(P, \Delta) : T(J) &\rightarrow K; J, K \in B_2^{2^n}, P \in \Omega(2^n), \Delta \in B_2^{2^n} \\
P(T(J)^{\Delta}) &= P(T(S_{2^n-1}^{\delta_{2n-1}}(J_{2^n-1})))...P(T(S_I^{\delta_I}(J_I)))...P(T(S_0^{\delta_0}(J_0))) \\
&= P(T(J)_{2^n-1}^{\delta_{2n-1}})...P(T(J)_I^{\delta_I})...P(T(J)_0^{\delta_0}) \\
&= K_{2^n-1}...K_I...K_0 = K \in B_2^{2^n} \\
P(T(J)_I^{\delta_I}) &= P(T(S_I^{\delta_I}(J_I))) = J_{P(I)}^{\delta_{P(I)}} = K_I \in B_2 \\
0 \leq I &< 2^n, 0 \leq J, K < 2^{2^n}, P \in \Omega(2^n), \delta_I \in \Delta
\end{aligned} \tag{9}$$

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
$T(178)^{\mathbf{1}}$	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
$T(178)^{\mathbf{0}}$	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 5.1: 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 $\neg T(J)_{P(I)}^{\delta_{P(I)}}$.

5.3 Logic functional spaces

Theorem 5.1 (Logic Function Invariants under Permutation & Complementary Operations) For any logic function, the output of Method 5.1 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

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

After using Method 5.1, the results are shown:

$$P(T(J)_I^{\Delta}) = \begin{cases} \neg\neg T(J)_I = T(J)_I & \delta_{P(I)} = 0 \\ T(J)_I & \delta_{P(I)} = 1 \end{cases} \tag{11}$$

■

Theorem 5.2 (Permutation Group for Meta Function Vector) For 2^n meta function vectors, a total of permutation numbers is $2^n!$.

Theorem 5.3 (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.

6. 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 1-Dimensional representation and other three schemes are 2-Dimensional representations. For binary sequences in sequential counting order, the scheme is known as the SL (Shao Yong & Leibniz) coding scheme .

6.1 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,

$$G : (J, \Delta, P) \to K; J, K \in B_2^{2^n}; \Delta \in B_2^{2^n}, P \in \Omega. \tag{12}$$

6.2 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.

$$\begin{aligned} &W : (J, \Delta, P) \to K = \langle J^1 | J^0 \rangle \\ &\quad J, K \in B_2^{2^n}; J^1, J^0 \in B_2^{2^{n-1}}; S^1, S^0 \in \mathbf{S}, \Delta \in B_2^{2^n}, P \in \Omega \end{aligned} \tag{13}$$

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

$$\{W(J)\}_{J=0}^{2^{2^n}} = \begin{array}{|c|c|c|c|c|}\hline \langle 0|0\rangle & ... & \langle 0|J^0\rangle & ... & \langle 0|2^{2^{n-1}}-1\rangle \\ \hline ... & & ... & & ... \\ \hline \langle J^1|0\rangle & ... & \langle J^1|J^0\rangle & ... & \langle J^1|2^{2^{n-1}}-1\rangle \\ \hline ... & & ... & & ... \\ \hline \langle 2^{2^{n-1}}-1|0\rangle & ... & \langle 2^{2^{n-1}}-1|J^0\rangle & ... & \langle 2^{2^{n-1}}-1|2^{2^{n-1}}-1\rangle \\ \hline \end{array} \tag{14}$$

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

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

f	J^{11}, T	W	J^{10}, IV	W	J^{01}, V	W	J^{00}, F	W
0	0	$\langle 0\|0\rangle$	1	$\langle 0\|1\rangle$	2	$\langle 1\|0\rangle$	3	$\langle 1\|1\rangle$
$\bar{x}$	1	$\langle 0\|1\rangle$	0	$\langle 0\|0\rangle$	3	$\langle 1\|1\rangle$	2	$\langle 1\|0\rangle$
x	2	$\langle 1\|0\rangle$	3	$\langle 1\|1\rangle$	0	$\langle 0\|0\rangle$	1	$\langle 0\|1\rangle$
1	3	$\langle 1\|1\rangle$	2	$\langle 1\|0\rangle$	1	$\langle 0\|1\rangle$	0	$\langle 0\|0\rangle$
f	$P(J)^{11}$, T	W	$P(J)^{10}$, IV	W	$P(J)^{01}$, V	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 of multiple complementary results, the 1-Dimensional G coding-scheme can be converted into a 2-Dimensional W coding-scheme as an extending matrix. It is convenient to extend different Δ variant matrices into a composed matrix. Four typical Δ variant values: $\{\Delta\} = \{11, 10, 01, 00\} = \begin{array}{|c|c|}\hline 11 & 10 \\ \hline 01 & 00 \\ \hline \end{array}$ corresponding to {Truth, Invariant, Variant, False} matrices respectively. A 2x2 matrix is composed of four block matrices in the order: $\{\Delta\} = \begin{array}{|c|c|}\hline 11 & 10 \\ \hline 01 & 00 \\ \hline \end{array}$.

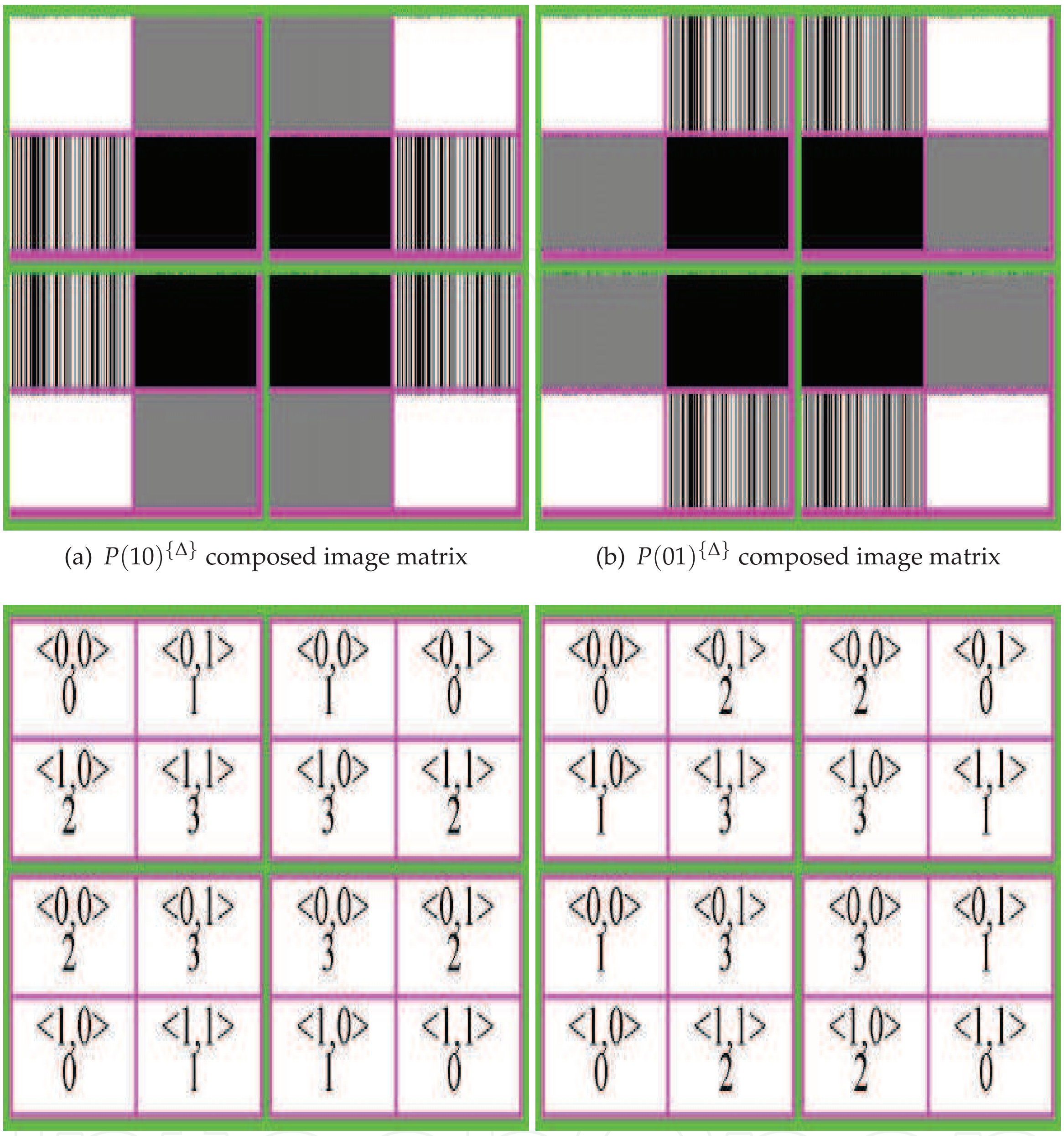
$$W^{\{\Delta\}} = P(10)^{\{\Delta\}} = \begin{array}{|cc|cc|}\hline \text{Truth} & & & \text{Invariant} \\ & 0\ \bar{x} & \bar{x}\ 0 & \\ & x\ 1 & 1\ x & \\ \hline & x\ 1 & 1\ x & \\ & 0\ \bar{x} & \bar{x}\ 0 & \\ \text{Variant} & & & \text{False} \\ \hline \end{array}$$

$$PW^{\{\Delta\}} = P(01)^{\{\Delta\}} = \begin{array}{|cc|cc|}\hline \text{Truth} & & & \text{Invariant} \\ & 0\ x & x\ 0 & \\ & \bar{x}\ 1 & 1\ \bar{x} & \\ \hline & \bar{x}\ 1 & 1\ \bar{x} & \\ & 0\ x & x\ 0 & \\ \text{Variant} & & & \text{False} \\ \hline \end{array}$$

On each 2x2 matrix, a pair of functions can be identified as follows:

$$0 \cap 1 = 1 \cap 0 = x \cap \bar{x} = \bar{x} \cap x = 0 \tag{15}$$

$$0 \cup 1 = 1 \cup 0 = x \cup \bar{x} = \bar{x} \cup x = 1 \tag{16}$$



(a) $P(10)^{\{\Delta\}}$ composed image matrix (b) $P(01)^{\{\Delta\}}$ composed image matrix

(c) $P(10)^{\{\Delta\}}$ composed number matrix (d) $P(01)^{\{\Delta\}}$ composed number matrix

Fig. 1. One variable variant logic matrices: $P = \{10, 01\}, \{\Delta\} = \{11, 10, 01, 00\}$; (a-b)2x2 base blocks (c-d)2x2 vector blocks

This makes following matrix equations:

$$\begin{pmatrix} 0 & \bar{x} \\ x & 1 \end{pmatrix} \cap \begin{pmatrix} 1 & x \\ \bar{x} & 0 \end{pmatrix} = ... = \begin{pmatrix} x & 1 \\ 0 & \bar{x} \end{pmatrix} \cap \begin{pmatrix} \bar{x} & 0 \\ 1 & x \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{17}$$

$$\begin{pmatrix} 0 & \bar{x} \\ x & 1 \end{pmatrix} \cup \begin{pmatrix} 1 & x \\ \bar{x} & 0 \end{pmatrix} = ... = \begin{pmatrix} x & 1 \\ 0 & \bar{x} \end{pmatrix} \cup \begin{pmatrix} \bar{x} & 0 \\ 1 & x \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \tag{18}$$

However this type of complementary properties cannot be directly observed on $W^{\{\Delta\}}$ and $PW^{\{\Delta\}}$ matrices under logic operations.

$$W^{\{\Delta\}} \cap PW^{\{\Delta\}} = \left(\begin{array}{cc|cc} 0 & \bar{x} & \bar{x} & 0 \\ x & 1 & 1 & x \\ \hline x & 1 & 1 & x \\ 0 & \bar{x} & \bar{x} & 0 \end{array}\right) \bigcap \left(\begin{array}{cc|cc} 0 & x & x & 0 \\ \bar{x} & 1 & 1 & \bar{x} \\ \hline \bar{x} & 1 & 1 & \bar{x} \\ 0 & x & x & 0 \end{array}\right) = \left(\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ \hline 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right) \tag{19}$$

$$W^{\{\Delta\}} \cup PW^{\{\Delta\}} = \left(\begin{array}{cc|cc} 0 & \bar{x} & \bar{x} & 0 \\ x & 1 & 1 & x \\ \hline x & 1 & 1 & x \\ 0 & \bar{x} & \bar{x} & 0 \end{array}\right) \bigcup \left(\begin{array}{cc|cc} 0 & x & x & 0 \\ \bar{x} & 1 & 1 & \bar{x} \\ \hline \bar{x} & 1 & 1 & \bar{x} \\ 0 & x & x & 0 \end{array}\right) = \left(\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right) \tag{20}$$

All 8 configurations of one variable conditions for variant logic can be exhaustively listed in two matrices. A total of 8 functional configurations in two groups can be seen in Fig. 1. There are 16 blocks in a comprised matrix in which thin lines separate different recursive images as blocks and each block under a function recursively, thick lines separate four 2x2 blocks and each 2x2 blocks under a set of Δ complementary operations. Four Δ operations apply to four 2x2 blocks respectively and each Δ complementary operation applied to a matrix contained 2x2 block functions. It is convenient to replace corresponding function symbols to recursive images respectively. Fig. 1(a) shows the $W^{\{\Delta\}} = P(10)^{\{\Delta\}}$ matrix; its four sub-blocks use $\{f_0 = 0, f_1 = \bar{x}, f_2 = x, f_3 = 1\}$ for its four operations; in Fig. 1(b) shows the $PW^{\{}\Delta\} = P(01)^{\{\Delta\}}$ matrix; its four sub-blocks use $\{0, x, \bar{x}, 1\}$ functions on each 2x2 block for its four operations: $\{\Delta\} = \{11, 10, 01, 00\}$ respectively to generate truth, variant, invariant and false Δ operations. The four outer corners and inner corners of $P(10)^{\{\Delta\}}$ and $P(01)^{\{\Delta\}}$ composed matrix in Fig. 1 (a) and (b) are one function as four white and four black blocks, $\{\Delta\}$ operations provide both horizontal and vertical reflection effects on two matrices.
From an operational viewpoint, under the same $\{\Delta\}$ operator, $P(01)^{\{\Delta\}}$ and $P(10)^{\{\Delta\}}$ have similar visual effects of rotation 90 degrees each other that can be clearly observed via selected sample images. These structure provide visual mechanism to present all possible configurations of the one variable functional space without repeat exhaustively.

6.3 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.

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

6.3.1 Pairs of conjugate functions

one special corresponding relationship can be identified as a pair of conjugate functions. For a given function f, its conjugate function $\tilde{f}$ is determined by undertaken following transformation:

$$\{0 \leftrightarrow 1; \cap \leftrightarrow \cup\} \tag{21}$$

all other variables keep invariant, do not transformed into their complementary variables. e.g. $f = x \cap \bar{y}$ and $\tilde{f} = x \cup \bar{y}$; $f = 0$ and $\tilde{f} = 1$ are two typical pairs of conjugate functions.
Under F coding scheme, it is natural to have pairs of conjugate functions distributed on diagonal directions. Such special arrangements are much easier to be observed with complementary symmetric properties via pairs of recursive images.

6.4 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.

$$\begin{array}{cccc} S^1 \text{ the } I\text{-th} & \leftrightharpoons & S^0 \text{ the } I\text{-th} & \text{F coding scheme} \\ \updownarrow & \text{C coding scheme} & \updownarrow & + \text{ Four corners} \in \{0, \bar{x}, x, 1\} \\ \forall x_i \in S^1, x_i = 1(0) & \rightleftharpoons & \forall x_i \in S^0, x_i = 0(1) & + \text{ General conjugate} \end{array}$$

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. Under this coding scheme, four corner positions of a matrix are composed of four functions of one variable matrix respectively.

7. Two-variable cases

There are a total of $384 = 24 \times 16$ configurations in functional spaces of two variable configurations. Similar exhaustive mechanism of one variable condition, different configurations of functionals can be illustrated by combining them into a comprised matrix on which satisfy complementary relationships on block matrix condition. Complete arrangements can be assigned as 24 matrices each matrix for a permutation to contain 16 complementary operations to be arranged as 4x4 blocks and each block is linked to a given function. Small sized blocks such as 2x2 are also selected to show special visual configurations. Each block must have complementary relationship with its opposite block on diagonal directions.
In convenient illustration, six groups of examples are selected. Four figures 2-5 contain a logic functional represents 16 logic functions as 4x4 images separated by thin lines. Four functionals are arranged as 2x2 block matrices separated by thick lines in Truth/False, Invariant/Variant properties. Relevant 2x2 block matrices of complementary operations correspond to:

$$\{\Delta\} = \begin{array}{|c|c|} \hline \text{Truth} = 1111 & \text{Invariant} = 1100 \\ \hline \text{Variant} = 0011 & \text{False} = 0000 \\ \hline \end{array}$$

Each matrix contains 16 entries of function images as a 4x4 ($2^2 \times 2^2$) configuration. Each image entry denotes a transformed number and its function number in the form: $\boxed{\begin{array}{c} \langle J^1 | J^0 \rangle \\ J \end{array}}$
where $K = \langle J^1 | J^0 \rangle$ is a transformed number and J is the function number. In all four figures, (a)2x2 base block matrices to represent function images and (b)2x2 vector blocks to represent relevant coding schemes respectively.
To show a complete matrix on a functional configuration, two permutation groups of Figures 6-7 are selected. Each figure contain a total of 256 images arranged as 16x16 matrices to

represent 16 block matrices for corresponding complementary operations. Each block matrix has 4x4 images. The 4x4 block matrices contain following $\{\Delta\}$ values respectively:

$$\{\Delta\} = \begin{array}{|c|c|c|c|}\hline 1111 & 1110 & 1101 & 1100 \\ \hline 1011 & 1010 & 1001 & 1000 \\ \hline 0111 & 0110 & 0101 & 0100 \\ \hline 0011 & 0010 & 0001 & 0000 \\ \hline \end{array}$$

Each image entry has a corresponding transformed number and its function number respectively.

In Figure 2, the counting order of meta-states has been arranged as W coding (SL code): $P = (3210), \{\Delta\} = \begin{array}{|c|c|}\hline 1111 & 1100 \\ \hline 0011 & 0000 \\ \hline \end{array}$. In this group, only functions 6 & 9 and 0 & 15 can be visualised in complementary symmetric condition in two diagonal directions. Visual symmetric effects of other pairs cannot be easily observed. However under four $\{\Delta\}$ operations generate a composed matrix that has clear horizontal and vertical reflect symmetries. It is directly to use equation to check their complementary properties:

$$\begin{aligned} & \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{pmatrix}_{\Delta=1111} \bigcap \begin{pmatrix} 15 & 14 & 13 & 12 \\ 11 & 10 & 9 & 8 \\ 7 & 6 & 5 & 4 \\ 3 & 2 & 1 & 0 \end{pmatrix}_{\Delta=0000} \\ = & \begin{pmatrix} 3 & 2 & 1 & 0 \\ 7 & 6 & 5 & 4 \\ 11 & 10 & 9 & 8 \\ 15 & 14 & 13 & 12 \end{pmatrix}_{\Delta=1100} \bigcap \begin{pmatrix} 12 & 13 & 14 & 15 \\ 8 & 9 & 10 & 11 \\ 4 & 5 & 6 & 7 \\ 0 & 1 & 2 & 3 \end{pmatrix}_{\Delta=0011} \\ = & \begin{pmatrix} 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \end{pmatrix} = (\ldots f_0 = 0 \ldots) \end{aligned} \tag{22}$$

$$\begin{aligned} & \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{pmatrix}_{\Delta=1111} \bigcup \begin{pmatrix} 15 & 14 & 13 & 12 \\ 11 & 10 & 9 & 8 \\ 7 & 6 & 5 & 4 \\ 3 & 2 & 1 & 0 \end{pmatrix}_{\Delta=0000} \\ = & \begin{pmatrix} 3 & 2 & 1 & 0 \\ 7 & 6 & 5 & 4 \\ 11 & 10 & 9 & 8 \\ 15 & 14 & 13 & 12 \end{pmatrix}_{\Delta=1100} \bigcup \begin{pmatrix} 12 & 13 & 14 & 15 \\ 8 & 8 & 10 & 11 \\ 4 & 5 & 6 & 7 \\ 0 & 1 & 2 & 3 \end{pmatrix}_{\Delta=0011} \\ = & \begin{pmatrix} 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \end{pmatrix} = (\ldots f_{15} = 1 \ldots) \end{aligned} \tag{23}$$

In Figure 3, variation the configurations among W coding: $P = (2301), \{\Delta\} = \begin{array}{|c|c|}\hline 1111 & 1100 \\ \hline 0011 & 0000 \\ \hline \end{array}$

$$
\begin{aligned}
&\begin{pmatrix} 0 & 2 & 1 & 3 \\ 8 & 10 & 9 & 11 \\ 4 & 6 & 5 & 7 \\ 12 & 14 & 13 & 15 \end{pmatrix}_{\Delta=1111} \bigcap \begin{pmatrix} 15 & 13 & 14 & 12 \\ 7 & 5 & 6 & 4 \\ 11 & 9 & 10 & 8 \\ 3 & 1 & 2 & 0 \end{pmatrix}_{\Delta=0000} \\
= &\begin{pmatrix} 3 & 1 & 2 & 0 \\ 11 & 9 & 10 & 8 \\ 7 & 5 & 6 & 4 \\ 15 & 13 & 14 & 12 \end{pmatrix}_{\Delta=1100} \bigcap \begin{pmatrix} 12 & 14 & 13 & 15 \\ 4 & 6 & 5 & 7 \\ 8 & 10 & 9 & 11 \\ 0 & 2 & 1 & 3 \end{pmatrix}_{\Delta=0011} \\
= &\begin{pmatrix} 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \end{pmatrix} = (\,...f_0 = 0...\,)
\end{aligned}
\tag{24}
$$

$$
\begin{aligned}
&\begin{pmatrix} 0 & 2 & 1 & 3 \\ 8 & 10 & 9 & 11 \\ 4 & 6 & 5 & 7 \\ 12 & 14 & 13 & 15 \end{pmatrix}_{\Delta=1111} \bigcup \begin{pmatrix} 15 & 13 & 14 & 12 \\ 7 & 5 & 6 & 4 \\ 11 & 9 & 10 & 8 \\ 3 & 1 & 2 & 0 \end{pmatrix}_{\Delta=0000} \\
= &\begin{pmatrix} 3 & 1 & 2 & 0 \\ 11 & 9 & 10 & 8 \\ 7 & 5 & 6 & 4 \\ 15 & 13 & 14 & 12 \end{pmatrix}_{\Delta=1100} \bigcup \begin{pmatrix} 12 & 14 & 13 & 15 \\ 4 & 6 & 5 & 7 \\ 8 & 10 & 9 & 11 \\ 0 & 2 & 1 & 3 \end{pmatrix}_{\Delta=0011} \\
= &\begin{pmatrix} 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \end{pmatrix} = (\,...f_{15} = 1...\,)
\end{aligned}
\tag{25}
$$

This group contains visual symmetric effects in each block similar to Figure 2. Due to different permutation applied, detailed arrangements of each block are significantly different. The composed matrix under $\{\Delta\}$ operations also has horizontal and vertical reflect symmetries.

In Figure 4, the F coding-scheme is selected: under this configuration, $P = (2310), \{\Delta\} = \begin{array}{|c|c|}\hline 1111 & 1100 \\ \hline 0011 & 0000 \\ \hline \end{array}$.

$$\begin{aligned}
&\begin{pmatrix} 0 & 1 & 2 & 3 \\ 8 & 9 & 10 & 11 \\ 4 & 5 & 6 & 7 \\ 12 & 13 & 14 & 15 \end{pmatrix}_{\Delta=1111} \bigcap \begin{pmatrix} 15 & 14 & 13 & 12 \\ 7 & 6 & 5 & 4 \\ 11 & 10 & 9 & 8 \\ 3 & 2 & 1 & 0 \end{pmatrix}_{\Delta=0000} \\
&= \begin{pmatrix} 3 & 2 & 1 & 0 \\ 11 & 10 & 9 & 8 \\ 7 & 6 & 5 & 4 \\ 15 & 14 & 13 & 12 \end{pmatrix}_{\Delta=1100} \bigcap \begin{pmatrix} 12 & 13 & 14 & 15 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 0 & 1 & 2 & 3 \end{pmatrix}_{\Delta=0011} \\
&= \begin{pmatrix} 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 0000 \end{pmatrix} = (\ldots f_0 = 0\ldots)
\end{aligned} \tag{26}$$

$$\begin{aligned}
&\begin{pmatrix} 0 & 1 & 2 & 3 \\ 8 & 9 & 10 & 11 \\ 4 & 5 & 6 & 7 \\ 12 & 13 & 14 & 15 \end{pmatrix}_{\Delta=1111} \bigcup \begin{pmatrix} 15 & 14 & 13 & 12 \\ 7 & 6 & 5 & 4 \\ 11 & 10 & 9 & 8 \\ 3 & 2 & 1 & 0 \end{pmatrix}_{\Delta=0000} \\
&= \begin{pmatrix} 3 & 2 & 1 & 0 \\ 11 & 10 & 9 & 8 \\ 7 & 6 & 5 & 4 \\ 15 & 14 & 13 & 12 \end{pmatrix}_{\Delta=1100} \bigcup \begin{pmatrix} 12 & 13 & 14 & 15 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 0 & 1 & 2 & 3 \end{pmatrix}_{\Delta=0011} \\
&= \begin{pmatrix} 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \end{pmatrix} = (\ldots f_{15} = 1\ldots)
\end{aligned} \tag{27}$$

There are six pairs (0:15, 1:7, 2:11, 4:13, 6:9, 8:14) of complementary functions that can be visually identified in pair conjugate symmetric conditions for each complementary block. The group has four block matrices in which containing the same pairs of configurations. There are horizontal and vertical symmetries too.

In Figure 5, C coding has represented: $P = (0231), \{\Delta\} = \begin{array}{|c|c|}\hline 1111 & 1100 \\ \hline 0011 & 0000 \\ \hline\end{array}$. Checking four blocks under $\cup$ operations:

$$\begin{pmatrix} 0 & 4 & 1 & 5 \\ 2 & 6 & 3 & 7 \\ 8 & 12 & 9 & 13 \\ 10 & 14 & 11 & 15 \end{pmatrix}_{\Delta=1111} \bigcup \begin{pmatrix} 15 & 11 & 14 & 10 \\ 13 & 9 & 12 & 8 \\ 7 & 3 & 6 & 2 \\ 5 & 1 & 4 & 0 \end{pmatrix}_{\Delta=0000}$$
$$= \begin{pmatrix} 5 & 1 & 4 & 0 \\ 7 & 3 & 6 & 2 \\ 13 & 9 & 12 & 8 \\ 15 & 11 & 14 & 10 \end{pmatrix}_{\Delta=1100} \bigcup \begin{pmatrix} 10 & 14 & 11 & 15 \\ 8 & 12 & 9 & 13 \\ 2 & 6 & 3 & 7 \\ 0 & 4 & 1 & 5 \end{pmatrix}_{\Delta=0011} \quad (28)$$
$$= \begin{pmatrix} 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \\ 1111 & 1111 & 1111 & 1111 \end{pmatrix} = (\ ...f_{15} = 1...\ )$$

In addition to six pairs similar to F coding, four corners of the 4x4 image matrix are fixed by the 4 functions (0, 5, 10, 15) in four block matrices. Four functions of each block are $\{f_0 = 0, f_5 = \bar{x}, f_{10} = x, f_{15} = 1\}$. In addition, pairs of horizontal reflection results can be observed via $\{\Delta\} = \{1111, 1100\}$ operations and pairs of vertical reflection results can be observed via $\{\Delta\} = \{1111, 0011\}$ operations. This property makes this coding scheme be the most regular structures among all coding schemes.

In Figure 6, W coding is represented as: $P = (3210), \{\Delta\} =$

1111	1110	1101	1100
1011	1010	1001	1000
0111	0110	0101	0100
0011	0010	0001	0000

.

Four corner blocks are 4 block matrices under {1111, 1100, 0011, 0000} operations contain the same figures in Figure 2. All block matrices have reflection symmetric distributions via horizontal and vertical reflection symmetry distributions. This extending matrix is showing further symmetric properties in this construction, through a lot of image block matrices without clear pairs of local symmetry, a global reflection symmetric matrix can be observed under the complementary operations.

In Figure 7, W coding has represented: $P = (3102), \{\Delta\} =$

1111	1110	1101	1100
1011	1010	1001	1000
0111	0110	0101	0100
0011	0010	0001	0000

.

From a global viewpoint, this configuration has a global horizontal and vertical reflection symmetry in vertical and horizontal directions. From a local viewpoint, this configuration has more local symmetry than Figure 6. Four corners are 4 blocks {1111, 1100, 0011, 0000} are composed of figures to be typical C coding scheme shown in Figure 5. In addition, four inner corners of block matrices {1010, 1001, 0110, 0101} contain a F coding structure. Under a local observation, other 8 block matrices and their relevant functions in each matrix are composed of images without clear pairs of conjugate symmetric properties.

8. Comparison

It is convenient to list numeric parameters to compare for different coding schemes in following table.

Var	State	Function	ExPower	SL	W code	F code	C code
n	2^n	2^{2^n}	$2^n!$	1	$2^{2^n} \cdot 2^n!$	$2^{2^n(1+1/2)} \cdot 2^{n-1}!$	$8 \cdot 2^{n-1}!$
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	$2^{16} \cdot 16!$	$2^{24} \cdot 8!$	$8 \cdot 8!$
5	32	2^{32}	32!	1	$2^{32} \cdot 32!$	$2^{48} \cdot 16!$	$8 \cdot 16!$

where we use Var: variable number; State: state number; Function: function number; ExPower: exponent power products; SL: SL coding number; W code: W coding number under vector operations; F code: F coding number under vector operations; C code: C coding number under vector operations in the table respectively.

9. Conclusion

The arrangement of binary function space using a hierarchy of four spaces of classifications can be used to add symmetry and regular structure onto the entire space of binary-functions. This construction has capacities to support vector permutations and complementary operations. For ease of visualization, it is convenient to apply 2D matrix-type representation mechanism that enables symmetric configurations of the system to be analysed via different coding schemes from a local or global viewpoint. Binary functional spaces provide additional optimal information to generate large numbers of potential configurations in order to arrange and organise variant logic spaces. Complementary operations are made further extension easier and visualising in a larger matrix. Sample matrices are shown their configurations in different functionals and complementary operations. From these examples, exhaustive approaches for functional space are illustrated.

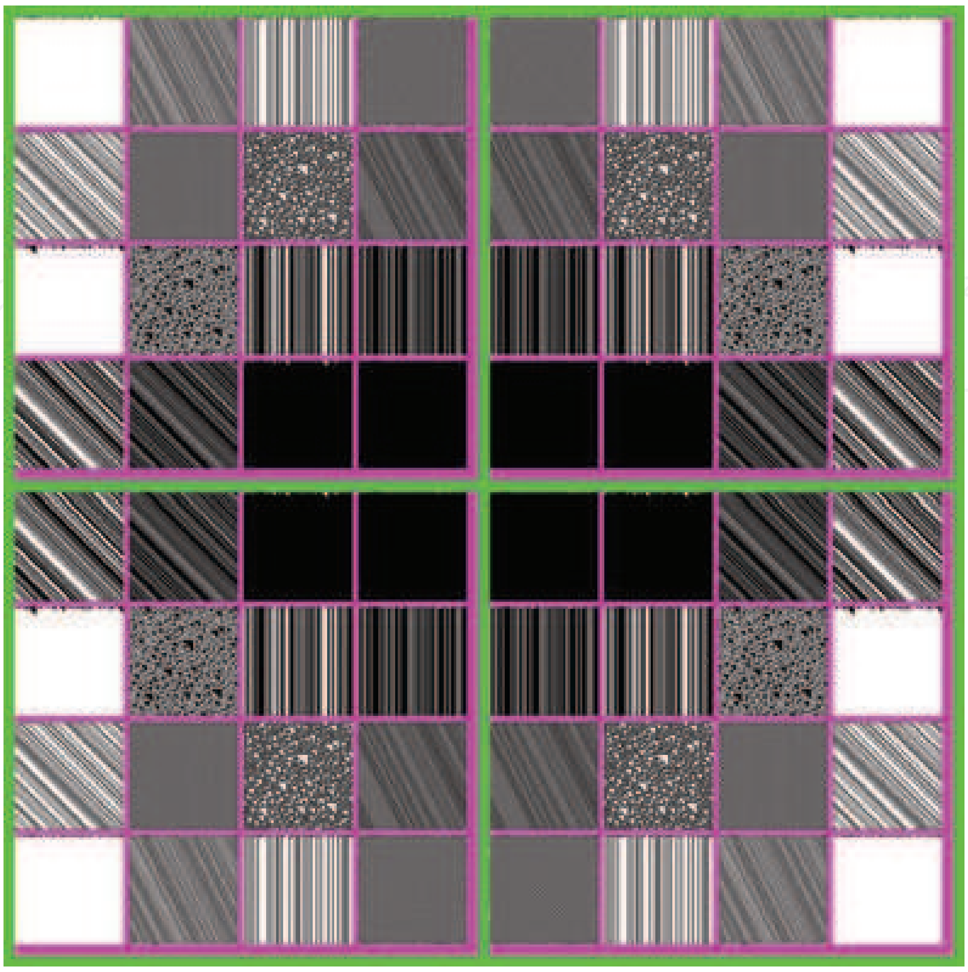
From a series of definitions, propositions and theorems, solid foundation of variant logic framework has been constructed. Under selected sample images and operational matrices, a set of typical results are illustrated. This construction can be observed from different viewpoints under symmetric considerations, in addition to detect emerging patterns from each recursively operations, further global transforming patterns can be identified from a functional space viewpoint. Under such expanding mechanism, a beautiful nature of mathematics has appeared. True natural effects are interesting for modern developments.

9.1 Future work

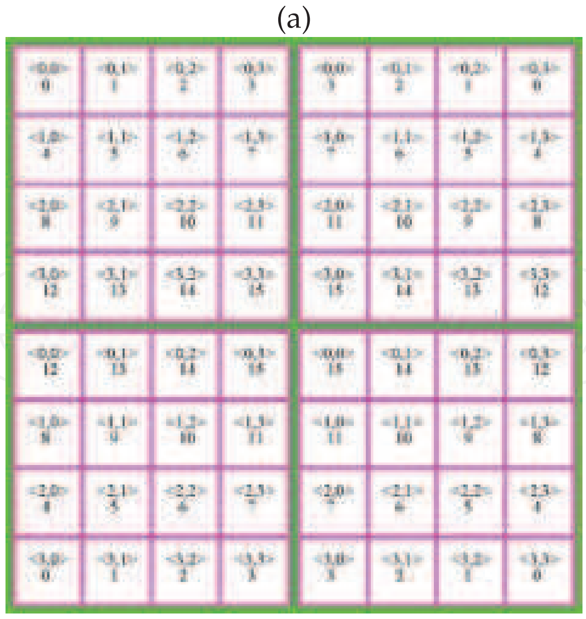
The mechanism can be developed further to establish foundations for logical construction of applications for computational models and structural optimisation requirements. Investigation on different coding schemes within the higher levels of organisation will be described in future work. This new mathematical logic foundation will support further theoretical descriptions to explore dynamic logics using modern mathematical language.

9.2 Acknowledgments

Thanks Mr. J. Wan for generation all sample images and configurations for the paper. Financial support was given by School of Software, Yunnan University.

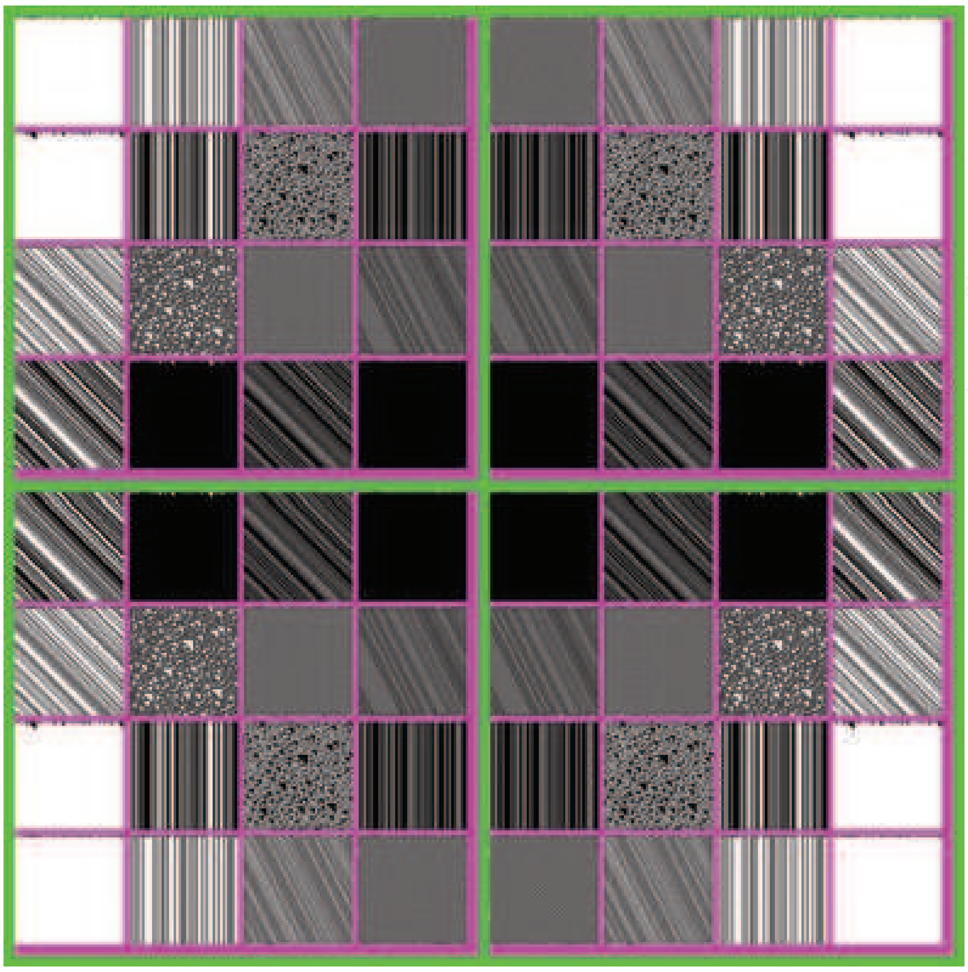


(a)

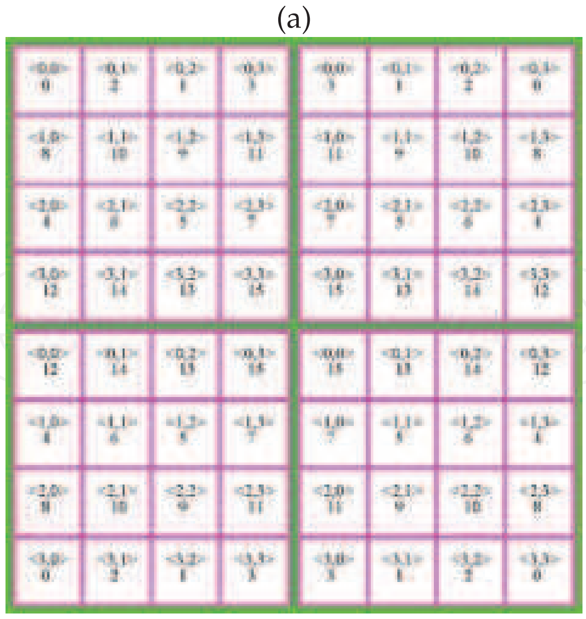


(b)

Fig. 2. W coding (SL code): $P = (3210), P(\Delta) = \{1111, 1100, 0011, 0000\}$; (a)2x2 base blocks (b)2x2 vector blocks

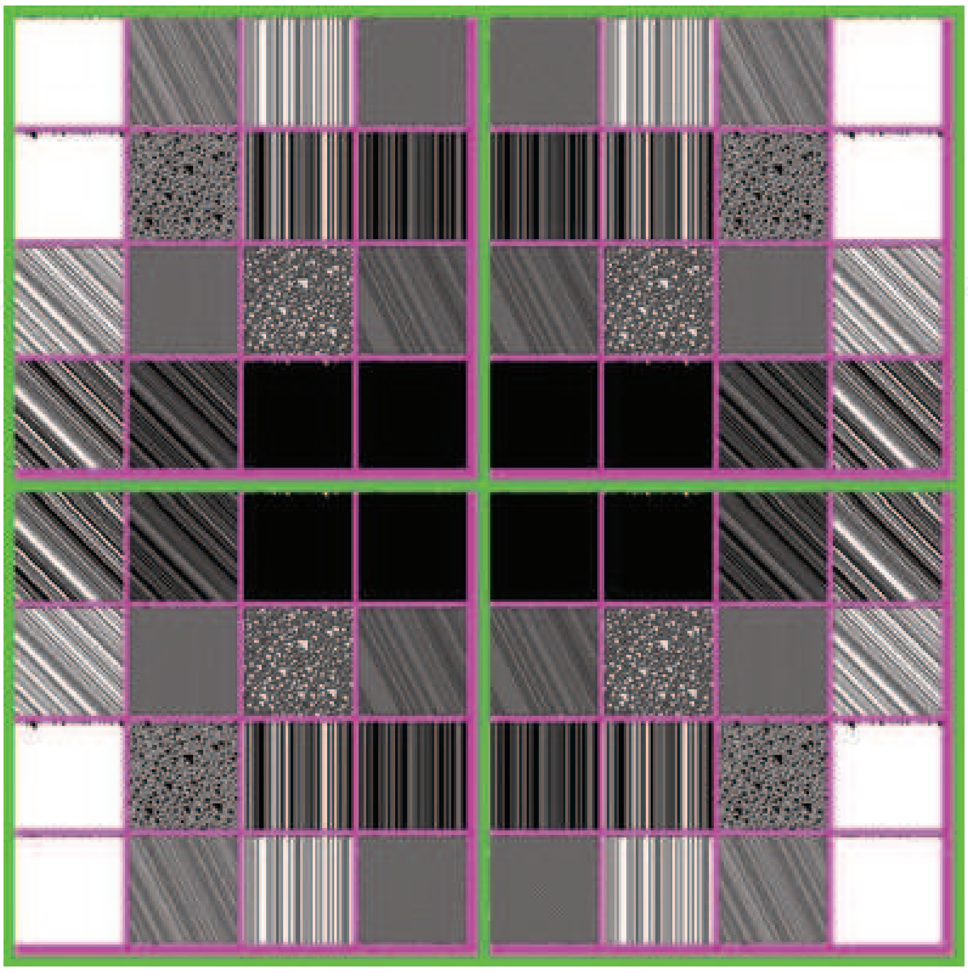


(a)

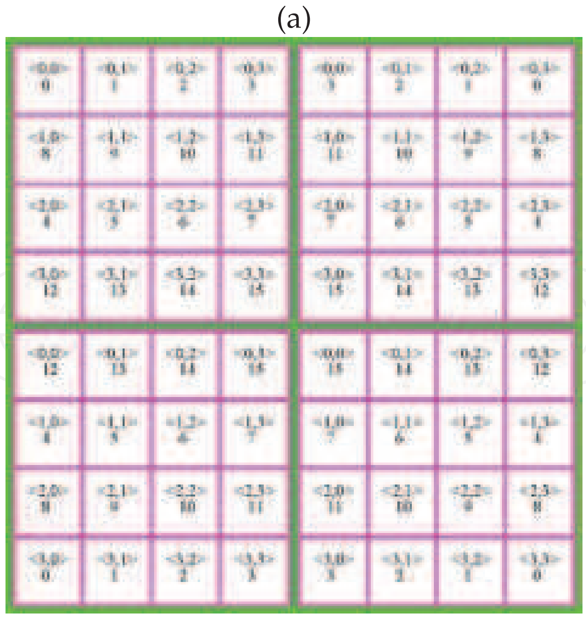


(b)

Fig. 3. W coding: $P = (2301), P(\Delta) = \{1111, 1100, 0011, 0000\}$; (a)2x2 base blocks (b)2x2 vector blocks

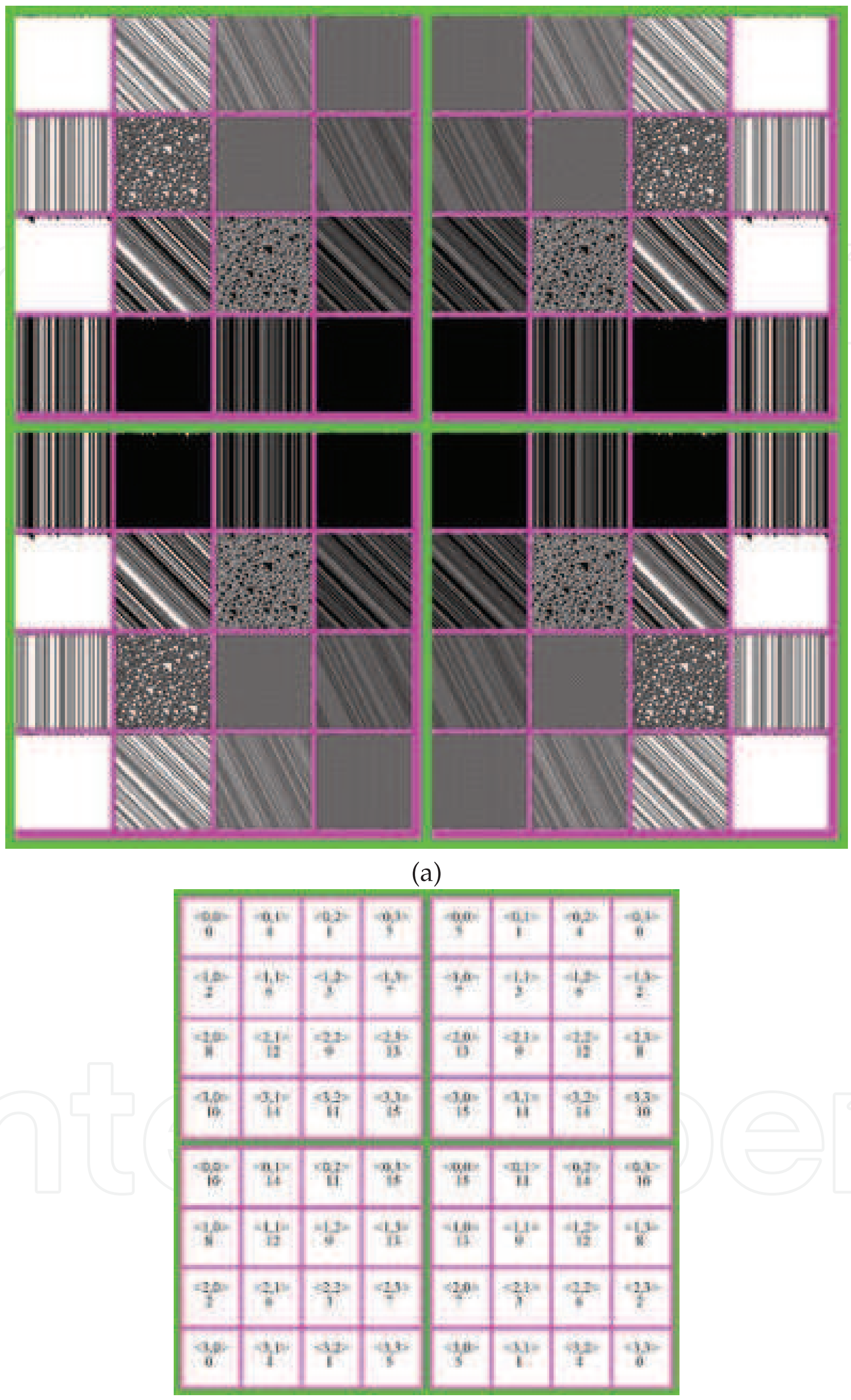


(a)



(b)

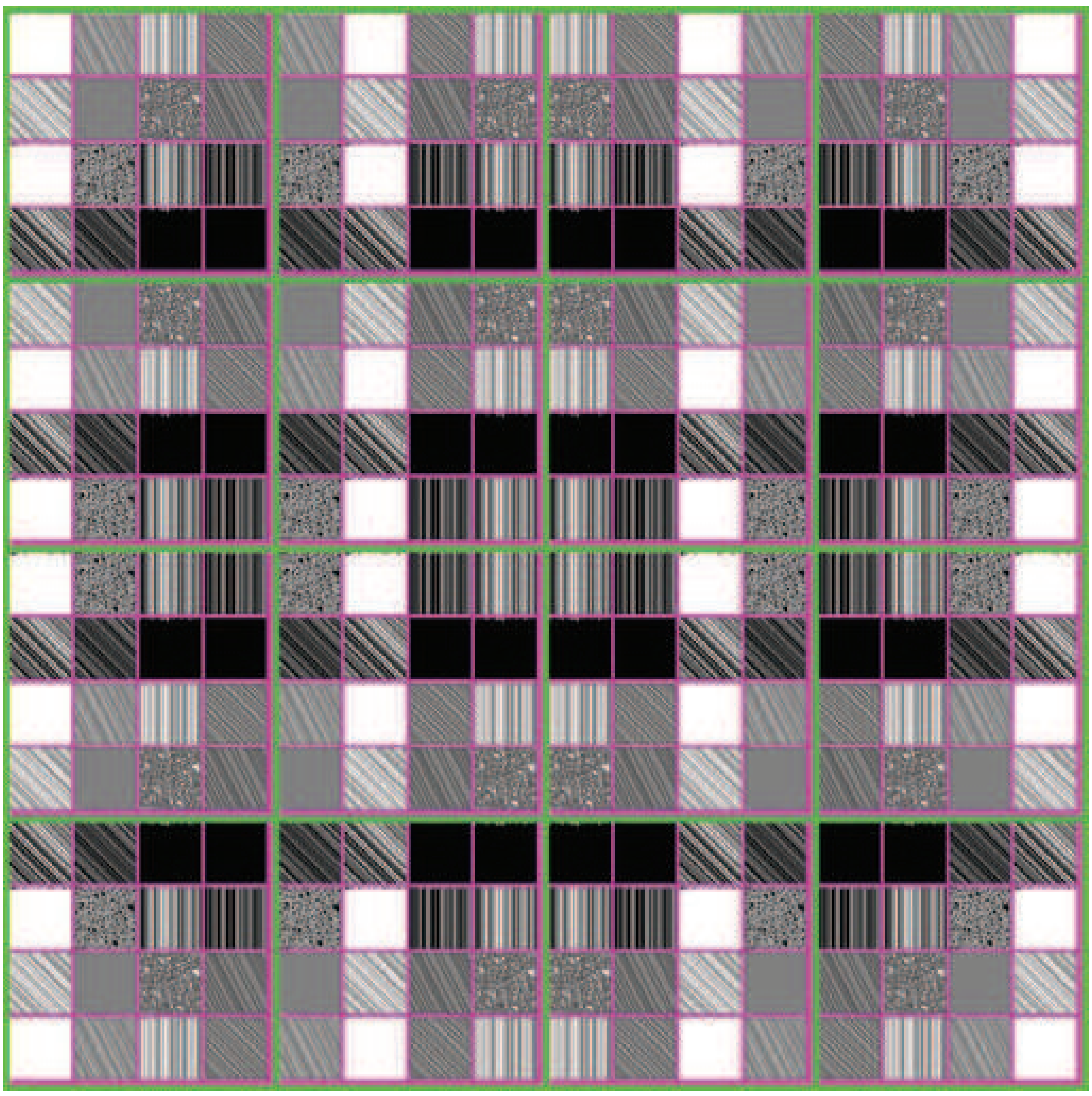
Fig. 4. F coding: $P = (2310), P(\Delta) = \{1111, 1100, 0011, 0000\}$; (a)2x2 base blocks (b)2x2 vector blocks



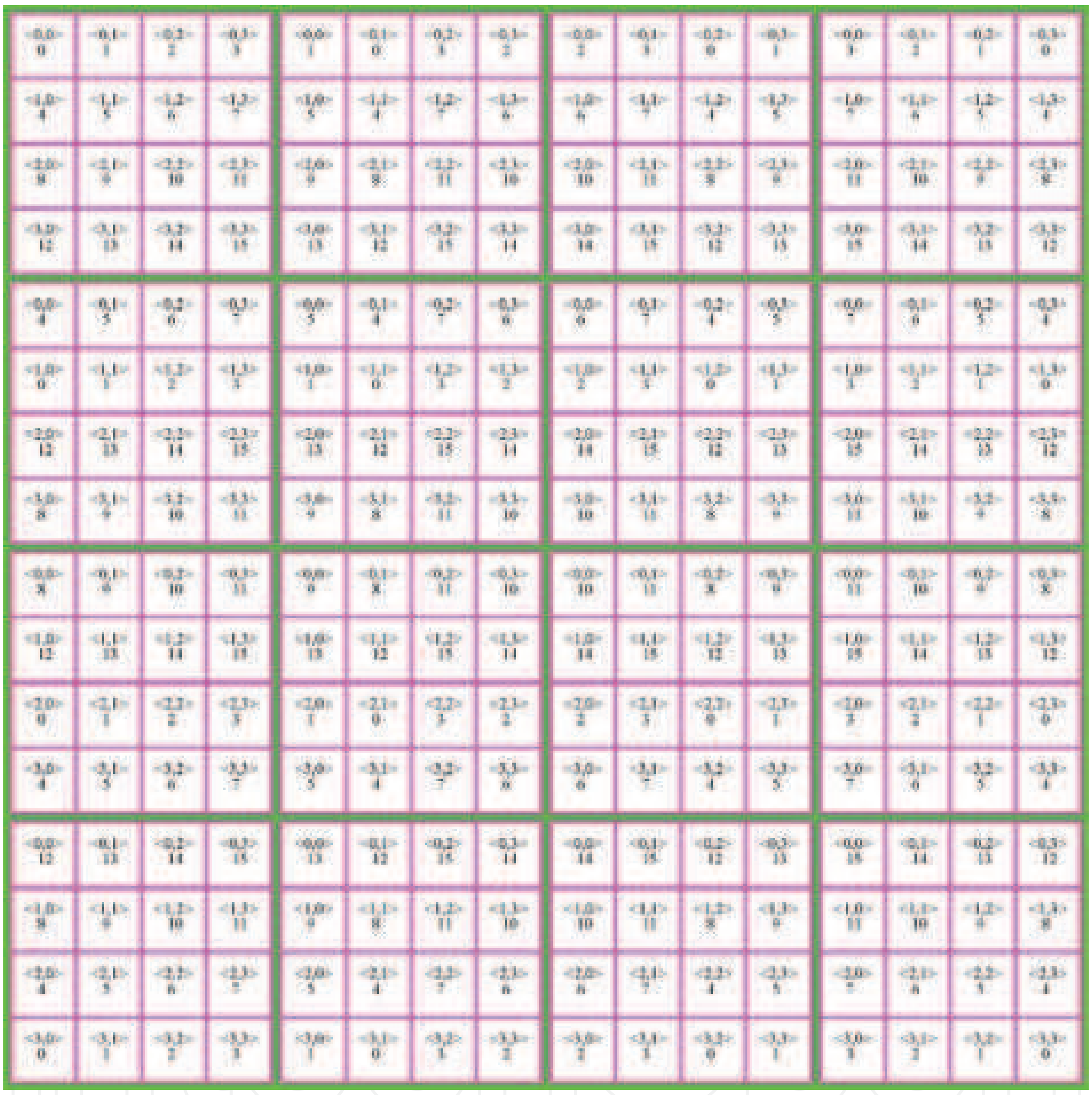
(a)

(b)

Fig. 5. C coding: $P = (0231), P(\Delta) = \{1111, 1100, 0011, 0000\}$; (a)2x2 base blocks (b)2x2 vector blocks

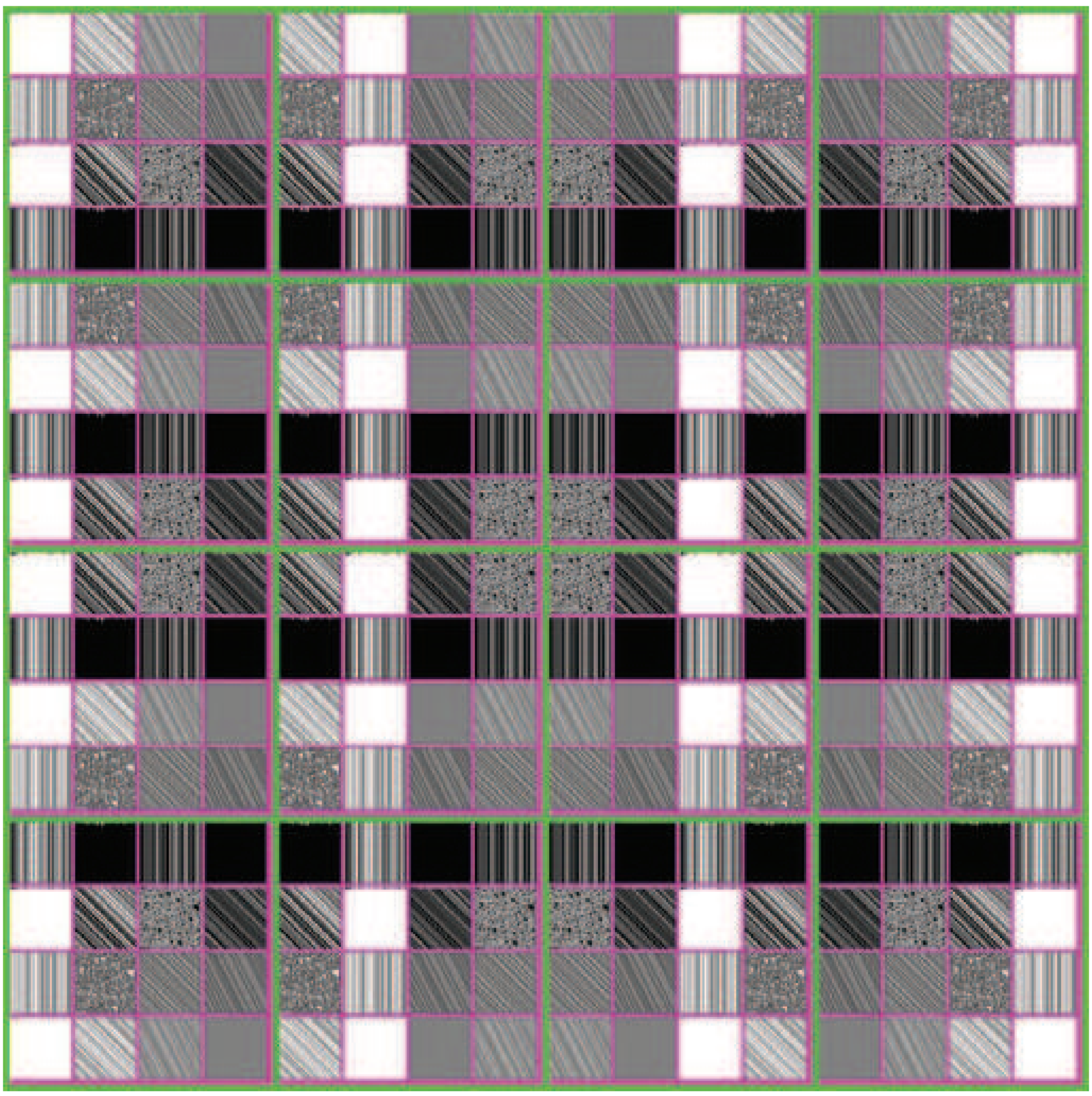


(a)

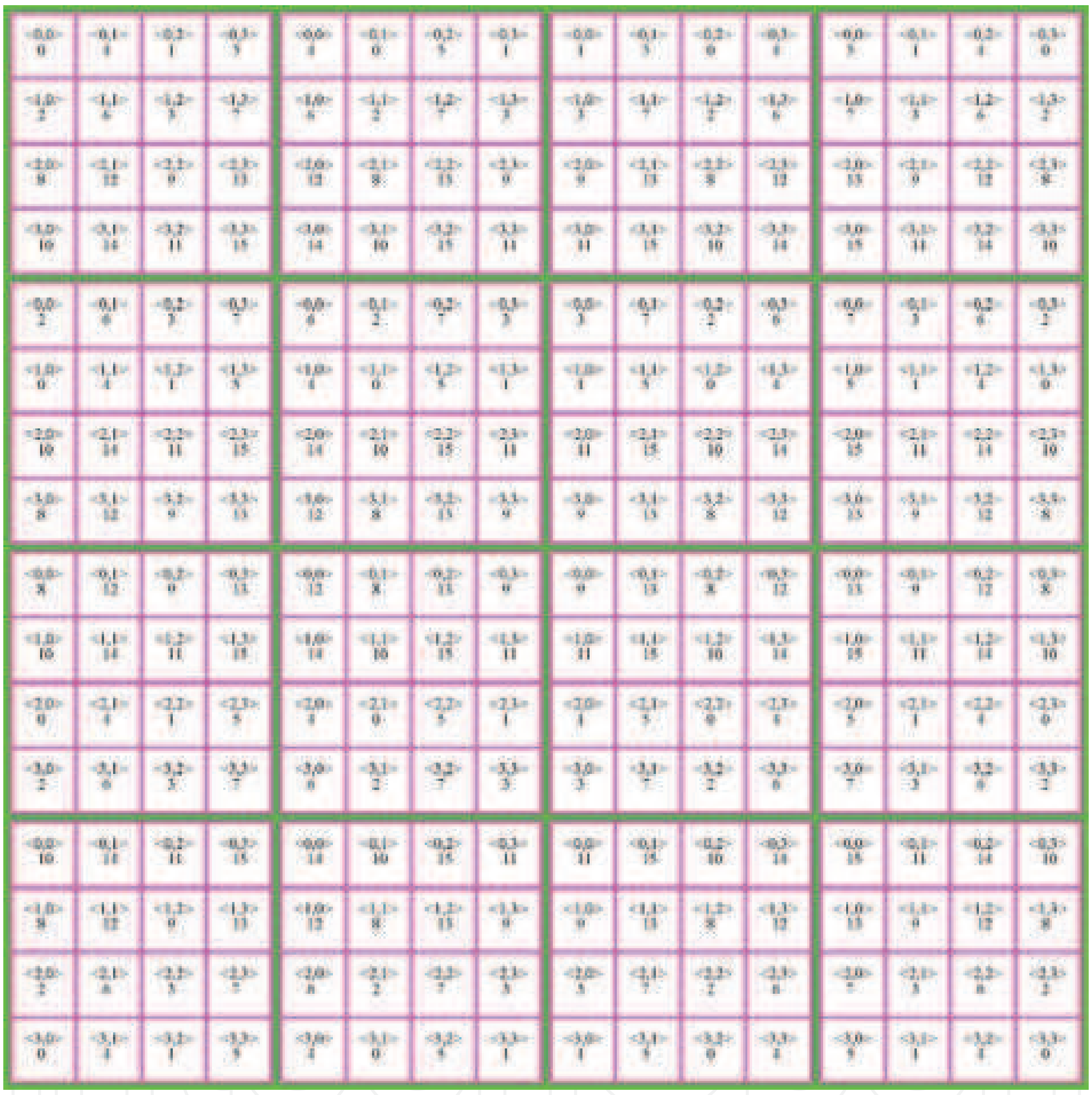


(b)

Fig. 6. W coding: $P = (3210)$,
$P(\Delta) =$
$\{1111, 1110, 1101, 1100, 1011, 1010, 1001, 1000, 0111, 0110, 0101, 0100, 0011, 0010, 0001, 0000\}$;
(a)4x4 base blocks (b)4x4 vector blocks



(a)



(b)

Fig. 7. W coding: $P = (3102)$,
$P(\Delta) =$
$\{1111, 1110, 1101, 1100, 1011, 1010, 1001, 1000, 0111, 0110, 0101, 0100, 0011, 0010, 0001, 0000\}$;
(a)4x4 base blocks (b)4x4 vector blocks

10. References

Adamatzky, A. (2008). Automata-2008: Theory and applications of cellular automata, *Luniver Press* .

Bonnet, M. (1989). Handbook of boolean algebras, *North Holland* .

Boole, G. (1850/1940/1958). An investigation of the laws of thought, *Dover* .

Chu, W. & Sherrill, W. (1977). An anthology of i ching, *Routledge and Kegan Paul Ltd* .

Cooper, J. (1981). Yin and yang, the taoist harmony of opposites, *The Thetford Press* .

Dawson, J. (2005). Logical dilemmas - the life and work of kurt gödel, *A.K. Peters, Ltd.* .

Demopoulos, W. (1995). Frege's philosophy of mathematics, *Harvard University Press* .

Dunn, P. (1988). The complexity of boolean networks, *Academic Press* .

Govinda, L. (1981). The inner structure of i ching: The book of transformation, *Wheelwright Press* .

Griffeath, D. & Moor, C. (2003). New constructions in cellular automata, *Santa Fe Institute Studies in the Sciences of Complexity* .

Hilbert, D. (1899). Grundlagen der geometrie, *Goettingen* .

Hook, D. (1975). The i ching and mankind, *Routledge and Kegan Paul Ltd* .

Ilachinski, A. (2001). Cellular automata-a discrete universe, *World Scientific* .

Kline, M. (1972). Mathematical thought from ancient to modern times, *Oxford University Press* .

Kunii, T., Hioki, H. & Shinagawa, Y. (1993). Visualizing highly abstract mathematical concepts: a case study in animation of homology groups, *Multimedia Modeling, World Scientific* pp. 3–30.

Kunii, T. & Kunii, H. (1999). A cellular model for information systems on the web - integrating local and global information, *Proceedings of 1999 International Symposium on Database Applications in Non-Traditional Environments, IEEE CS Press* .

Kunii, T. & Shinagawa, Y. (1992). Modern geometric computing for visualization, *Springer Verlag* .

Kunii, T. & Takahashi, S. (1993). Area guide map modeling by manifolds and cw-complexes, *Modeling in Computer Graphics, Springer Verlag* pp. 5–20.

Kunii, T. & Takai, Y. (1989). Cellular self-reproducing automata as a parallel processing model for botanical colony growth pattern simulation, *New Advances in Computer Graphics, Springer Verlag* pp. 7–22.

Kunii, T., Tsuchida, Y., Arai, Y., Matsuda, H., Shirahama, M. & Miura, S. (1993). A model of hands and arms based on manifold mappings, *Communicating with Virtual Worlds, Springer Verlag* pp. 381–398.

Lee, S. (1978). Modern switching theory and digital design, *Prentice-Hall Inc.* .

Leibniz, G. (1976). Philosophical papers and letters, *Springer* .

Leibniz, G., Ariew, R. & Garber, D. (1989). Philosophical essays, *Hackett Publishing* .

Needham, J. & Wang, L. (1954-1988). Science and civilisation in china: History of scientific thought., *Cambridge Press* 2.

Paterson, M. (1992). Boolean function complexity, *Cambridge University Press* .

Reichenbach, H. (1949). The theory of probability, *The University of California Press* .

Russell, B. (1942). Principles of mathematics, *Forgotten Books* .

Schiff, J. (2007). Cellular automata: A discrete view of the world, *Wiley Series in Discrete Mathematics and Optimization* .

Shannon, C., Sloane, N. & Wyner, A. (1993). Claude elwood shannon: Collected papers, *IEEE Press, IEEE Information Theory Society* .

Shchutshii, I. (1979). Researches on the i ching, *Princeton University Press* .

Sikorski, R. (1960). Boolean algebra, *Springer-Verlag* .

Turing, A. (1936). On computable numbers, with an application to the entscheidungs problem, *Proceedings of London Mathematical Society, Ser.2* 42: 230–265.

Umeo, H., Morishita, S. & Nishinari, K. (2008). Cellular automata, *Springer* .

Vingron, S. (2004). Switching theory: Insight through predicate logic, *Springer* .

von Neumann, J. (1966). Theory of self-reproducing automata, *University of Illionois Press* .

Whincup, G. (1986). Rediscovering of i ching, *GreyWhincup* .

Wilhelmi, H. (1979). Chang: Eight lectures on i ching, *Princeton Press* .

Wilhemi, R. (1979). Lectures on the i ching: Constancy and change, *Princeton University Press* .

Wolfram, S. (1985). Twenty problems in the theory of cellular automata, *Physica Scripta* T9: 170–183.

Wolfram, S. (1986). Theory and applications of cellular automata, *World Scientific* .

Wolfram, S. (1994). Cellular automata and complexity, *Addison-Wesley* .

Wolfram, S. (2002). A new kind of science, *Wolfram Media Inc.* . http://www.wolframscience.com/.

Wuensche, A. & Lesser, M. (1992). The global dynamics of cellular automata, *Santa Fe Institute studies in the sciences of complexity: Reference volumes, Addison-Wesley Publishing Company* 1.

Zadeh, L. (1965). Fazzy sets, *Information and Control* 8. 338-357.

Zheng, J. & Zheng, C. (2010). A framework to express variant and invariant functional spaces for binary logic, *Frontiers of Electrical and Electronic Engineering in China, Higher Education Press and Springer* 5(2): 163–172. http://www.springerlink.com/content/91474403127n446u/.

Zheng, Z. (1994). Conjugate transformation of regular plan lattices for binary images, *PhD Thesis, Monash University* .

Zheng, Z. (1996). Conjugate visualisation of global complex behaviour, *Complexity International* 3. http://www.complexity.org.au/ci/vol03/zheng/.

Zheng, Z. & Leung, C. (1996). Visualising global behaviour of 1d cellular automata image sequences in 2d maps, *Physica A* 233: 785–800.

Zheng, Z. & Maeder, A. (1992). The conjugate classification of the kernel form of the hexagonal grid, *Modern Geometric Computing for Visualization. Springer-Verlag* pp. 73–89.

Zheng, Z. & Maeder, A. (1993). The elementary equation of the conjugate transformation for hexagonal grid, *Modeling in Computer Graphics. Springer-Verlag* pp. 21–42.

Cellular Automata - Innovative Modelling for Science and Engineering
Edited by Dr. Alejandro Salcido

ISBN 978-953-307-172-5
Hard cover, 426 pages
Publisher InTech
Published online 11, April, 2011
Published in print edition April, 2011

Modelling and simulation are disciplines of major importance for science and engineering. There is no science without models, and simulation has nowadays become a very useful tool, sometimes unavoidable, for development of both science and engineering. The main attractive feature of cellular automata is that, in spite of their conceptual simplicity which allows an easiness of implementation for computer simulation, as a detailed and complete mathematical analysis in principle, they are able to exhibit a wide variety of amazingly complex behaviour. This feature of cellular automata has attracted the researchers' attention from a wide variety of divergent fields of the exact disciplines of science and engineering, but also of the social sciences, and sometimes beyond. The collective complex behaviour of numerous systems, which emerge from the interaction of a multitude of simple individuals, is being conveniently modelled and simulated with cellular automata for very different purposes. In this book, a number of innovative applications of cellular automata models in the fields of Quantum Computing, Materials Science, Cryptography and Coding, and Robotics and Image Processing are presented.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821