Hierarchical Organization of Variant Logic

Jeffrey Zheng

Abstract In modern logic, various systems have been proposed extending classical Boolean logic & switching theory. Such logic frameworks include multiple-valued logic, probability logic, fuzzy logic, module logic, quantum logic and various other frameworks. Although these extensions have been applied to many applications in mathematics, in science and in engineering, all extensions to Boolean logic invalidates at least one of the six fundamental rules of Boolean logic shown in L1 to L6. We propose a new framework of logic, variant logic, extending Boolean logic whilst satisfying the six fundamental rules (L1–L6). By defining the Variant–Invariant behaviour of logical operations, this framework can be constructed using four types of general operators. Main results of the chapter are summarized in **Theorems 8–10**, respectively. To show significant differences between classical logic and new variant logic, invariant properties of this hierarchical organization are discussed. Simplest cases of one-variable conditions are illustrated. Variant logic can provide the necessary framework to support analysis and description of Cellular Automata, Fractal Theory, Chaos Theory and other systems dealing with complexity. Such applications of this framework will be explored in future papers.

Keywords Switching theory · Boolean/multiple valued/probability/fuzzy logic Variant/invariant property · Hierarchical organization · Variant logic

---

This work was supported by the Key Project on Electric Information and Next Generation IT Technology of Yunnan (2018ZI002), NSF of China (61362014), Yunnan Advanced Overseas Scholar Project.

J. Zheng (✉)
Key Laboratory of Software Engineering of Yunnan, Yunnan University, Kunming, China
e-mail: conjugatelogic@yahoo.com

J. Zheng
Key Laboratory of Quantum Information of Yunnan, Yunnan University, Kunming, China

J. Zheng (ed.), *Variant Construction from Theoretical Foundation to Applications*,
https://doi.org/10.1007/978-981-13-2282-2_2

1 Laws of Logic Systems

1.1 Laws in Classical Logic Systems

Classical logic identifies a class of formal logic that are characterized by a number of properties [1–17].

Definition 1 For any logic system if all CL1–CL5 are satisfied, then it is a classical logic system. The five properties of classical logic (CL1–CL5) are listed as follows:

CL1: Law of the excluded middle and double negative elimination
CL2: Law of non-contradiction
CL3: Monotonicity and idempotency of entailment
CL4: Commutativity of conjunction
CL5: De Morgan duality

Examples of such classical logic systems include works of philosophy and religion (Aristotle's Organon; Nagarjuna's tetralemma; and Avicenna's temporal modal logic) as well as foundational logic systems such as reformulations by George Bool and Gottlob Frege [4–17]. These properties can be rewritten as simplified equations describing basic properties of a logic system using characteristics of the five classical properties. The following equations (L1–L6) describe such a system.

L1: $P \cup P = P$ Idempotency
L2: $P \cap P = P \ldots$
L3: $\neg P \cup P = P$ Excluded Middle
L4: $\neg P \cap P = P \ldots$
L5: $\neg\neg P = P$ Double Negative Elimination
L6: $P, P \rightarrow Q$

The set of equations can be applied in the analysis of modern logic systems to determine if they are all satisfied. The equations will be defined as canonical properties and a logic system satisfying all six properties will be defined as a canonical system. If any logic system does not, they are categorized as non-canonical.

1.2 Current Logic Systems

Many modern logic systems cannot satisfy the six canonical properties. Three-valued logic proposed by Luckasiewicz 1920 can satisfy L3–L6, cannot satisfy L1–L4. Probability logic proposed by Reichenbach 1949 can satisfy L5–L6, cannot satisfy L1–L4. Fuzzy logic proposed by Zadeh 1965 satisfy L1, L2, L5, L6, cannot satisfy L3–L4. Since they cannot satisfy canonical properties, they are all non-canonical logic systems [1–22].

2 Truth Valued Representation in Boolean Logic Systems

For any n-variable Boolean logic system, it is natural to establish 2^n states. Under either selected or not selected operation, it can be building up a truth table for a given Boolean function. Collecting all possible selections, a full truth table is constructed in 2^n columns and 2^{2^n} rows in presentation. We can list this table as follows:

$0 \leq I < 2^n$ $0 \leq i < n$	$2^n - 1$ 1...1...1	... ...	I $I_{n-1}...I_i...I_0$	... ...	1 0...0...1	0 0...0...0
$0 \leq J < 2^{2^n}$						
0	0	...	0	...	0	0
1	0	...	0	...	0	1
2	0	...	0	...	1	0
...			...			
J	J_{2^n-1}	...	J_I	...	J_1	J_0
...			...			
$2^{2^n} - 2$	1	...	1	...	1	0
$2^{2^n} - 1$	1	...	1	...	1	1

where there are three parameters: $i, I, J : 0 \leqslant i < n, 0 \leqslant I < 2^n, 0 \leqslant J < 2^{2^n}$ corresponding to variable, state and function numbers, respectively. Under such conditions, for any J, it is convenient to use Karnaugh map or relevant logic tools to construct the given Boolean function in combination [6–17].

3 Cellular Automata Representations

Cellular Automata—CA uses a different mechanism [23–35] to represent a given function. In a one-dimensional form of CA, a N-length binary sequence is

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

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

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

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

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

For a given function f

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

It is necessary to assign a certain position i in the kernel for special care to associated with j position of both sequences. We have

$$y = f(x_{n-1} \dots x_i \dots x_0) = f(\dots X_j \dots) == Y_j$$

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

$$f : X_j^{t-1} \rightarrow X_j^t, X_j^{t-1}, X_j^t \in B_2$$

4 Variant Construction

4.1 Four Variation Forms

Considering $f : X_j^{t-1} \rightarrow X_j^t$ for any function of Boolean logic system to analyse their variation properties [36–40], it is normal to have following proposition.

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

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

Definition 2 Four transforming forms are corresponding to following sets: TA: Invariant class for 0 value, TB: Variant class for 0 value, TC:Variant class for 1 value, TD: Invariant class for 1 value.

Under such definition, the following proposition can be established.

Proposition 2 *Using four classes of transformation, four variant operations are defined.*

Type	X_j	$\rightarrow Y_j$	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. ∎

Theorem 1 *In { Truth, Variant, Invariant, False} groups, only two pairs of groups: {Truth, False} and {Variant, Invariant} satisfy L1–L6 to form a canonic logic system.*

Proof Both groups are composed of 0-1 variables, in addition, Truth/False, Variant/Invariant are formed complement relationships. Other combinations contain common parts, it is not possible for them to satisfy logic canonic conditions L1–L6. ∎

Definition 3 Sequential number of binary is defined as SL coding to remember Y. Shao and Leibniz contribution [41–49] on binary logic.

Definition 4 The operator $BN : J \to B$ converts an integer to its binary representation. The operator $DC : B \to J$ converts a binary number to its decimal representation.

Definition 5 The SL coding scheme is an ordering of binary table outputs $T : B_2^{2^n} \to J$. An element $J_I \in SL$ at position I, where $0 \leqslant I < 2^n$ represents function T_I such that the binary representation of T_I is defined as

$$BN(J) = T_{2^n-1}[J_{2^n-1}] \dots T_I[J_I] \dots T_0[J_0]$$

For any n variable structure, J is composed of 2^n bits to represent $0 \leqslant J < 2^{2^n}$ numbers.

Definition 6 A G coding scheme is defined as an ordering of binary table outputs $T : B_2^{2^n} \to J$. An element $J_I \in SL$ at position I where $0 \leqslant I < 2^n$ represents function T_I such that the binary representation of T_I is defined as

$$G = \{\forall J | T(J), 0 \leqslant J < 2^{2^n}\};$$

$$T(J) = T_{2^n-1}[Y(J_{2^n-1})] \dots T_I[Y(J_I)] \dots T_0[Y(J_0)], 0 \leqslant I < 2^n$$

Where $\{Y(J_I), 0 \leqslant I < 2^n\}$ are 2^{2^n} length 0-1 vectors, $Y(J_{2^n-1}) \neq \dots \neq Y(J_I) \neq \dots \neq Y(J_0)$, respectively.

Under G coding scheme, ordering number is an integer sequence with 2^{2^n} positions. Different transformations will make this sequence extremely complex. In convenient to do representation, a two-dimensional W coding scheme is proposed.

Definition 7 A W coding scheme is defined as an ordering pair of binary table outputs $T : B_2^{2^n} \to \langle J^1 | J^0 \rangle$. Each component is composed of 2^{n-1} bits in representation:

$$\langle J^1 | J^0 \rangle = T_{2^n-1}[Y(J_{2^n-1})] \dots T_I[Y(J_I)] \dots T_0[Y(J_0)], 0 \leqslant I < 2^n$$

$$J^0 = \{\forall I | BN(J_I mod 2^{n-1}), 0 \leq I < 2^{n-1}\}$$

$$J^1 = \{\forall I | BN(J_I mod 2^{n-1}), 2^{n-1} \leq I < 2^n\}$$

Under this construction, a G coding scheme is transformed into a W coding scheme to represent two-dimensional structure for different permutation results. In general, J^0 represents lower 2^{n-1} bits and J^1 represents higher 2^{n-1} bits, respectively. A general structure of W coding is a $2^{2^{n-1}} \times 2^{2^{n-1}}$ matrix shown in the following figure.

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

$0 \leq J^0, J^1 < 2^{2^{n-1}}$ $\{\langle J^1|J^0\rangle\}$: 2D Space for 2^{2^n} Functions

4.2 Complement and Variant Operators

Definition 8 In B_2^n, the generalized complement Y^Q, $Q \in B_2^{2^n}$ of a variable Y is defined to be the element obtained from complementing the components of Y according to the value of corresponding component of Q; Y_I is complemented or un-complemented if Q_I is 0 or 1, respectively, where Y_I and Q_I designate the Ith component of Y and Q.

For example, given B_2^4 for $Q = \{0101, 0110\}$ are as follows:

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

To apply Q operator on 2^n meta vectors, a vector family can be generated.

Proposition 3 *In $B_2^{2^n}$, generalized complement operator $Q \in B_2^{2^n}$ has 2^{2^n} different cases.*

Proof Q is a 2^n bits vector, each position can be selected as 0 or 1, so a total of selections is equal to 2^{2^n}. ■

Definition 9 For 2^n meta states composed of vector Ψ, the ith vector $\Psi(i), 0 \leq i < n$ has 2^n bits. Four vectors: $\{\mathbf{0}, \Psi(i), \neg\Psi(i), \mathbf{1}\}$ in 2^n bits can be selected as Q operators. This special form of Q type operations is defined as QV operation.

Proposition 4 *For a QV operator, $QV \in \{\mathbf{0}, \Psi(i), \neg\Psi(i), \mathbf{1}\}$, four QV vectors provide following complement results respectively in transformation:*

$$\begin{aligned} \mathbf{0} &: \textit{False Operator} \\ \mathbf{1} &: \textit{Truth Operator} \\ \Psi(i) &: \textit{Invariant Operator} \\ \neg\Psi(i) &: \textit{Variant Operator} \end{aligned}$$

Proof $\mathbf{1}$ operator keeps original truth table values; $\mathbf{0}$ operator reverses all values; $\Psi(i)$ operator makes invariant condition and $\neg\Psi(i)$ operator generates variant property. ■

Proposition 5 *Undertaken QV operations, 2^{n+1} cases are generated as a complement variant group.*

Proof Only $0 \leq i < n$ selected, each position have two selections associated with i plus two constant vectors. So a total of $2 \times 2^n = 2^{n+1}$ cases can be generated. ■

Definition 10 For 2^n meta vectors Y, its Ith component $Y(I) \in B_2^{2^{2^n}}$, $Y(I)$ has 2^{2^n} bits. A permutation operator P makes the Ith component into $P(I)$th component for $\forall I, 0 \leq I < 2^n$, respectively.

Proposition 6 *Undertaken P operation to 2^n meta vectors in Y, a total of $2^n!$ permutations can be generated.*

Proof P operator is equal to permutation on 2^n integers. This generates a symmetric group contained $2^n!$ members. ■

Proposition 7 *Undertaken Q and P operators in Y, a total of $2^{2^n} \cdot 2^n!$ cases can be created. This creates a Complement Permutation Structure—CPS.*

Proof Q and P operators are independent of each other. Their results can be multiplied together. ■

Proposition 8 *Undertaken QV and P operators in Y, a total of $2^{n+1} \cdot 2^n!$ cases can be created. This creates a Complement Variant Structure—CVS.*

Proof QV and P operators are independent each other. Their results can be multiplied together. ■

4.3 Other Global Coding Schemes

Under $QV + P$ and $Q + P$ operations, more coding schemes can be defined.

Definition 11 The F coding scheme is defined as a subset W. For any W code, if any two meta state can be paired, such that $\forall j_1, j_1 - 2^{n-1} = j_0, 0 \leq j_0 < 2^{n-1} \leq j_1 < 2^n, I_{j_1} = \bar{I}_{j_0}$ indicate state I_{j_1} be I_{j_0}'s complement.

F coding provides restricted pair conditions to the structure. Its corresponding forms are as follows:

$$\begin{array}{ccc} J^1 \; j\text{-th meta state} & \leftrightharpoons & J^0 \; j\text{-th mate state} \\ \updownarrow & \text{F coding base} & \updownarrow \\ X & \rightleftharpoons & \bar{X} \end{array}$$

Definition 12 A coding scheme satisfies general conjugate condition if $\forall I_{j_0} \in I_{J^0}$, for the selected position $i, \forall a_i \in I_{j_0}, a_i = 0, 0 \leq i < n$.

In other words, the general conjugate condition makes selected position on lower part in 0 valued and higher part in 1-valued, respectively.

Definition 13 The C coding scheme is defined as a set of the F coding whereby $\forall I_{j_0} \in I_{J^0}$, for the selected position i, $\forall a_i \in I_{j_0}, a_i = 0, 0 \le i < n$.

C coding provides more strong restrictions to separate all 0-valued meta states in lower part and all 1-valued meta states in higher part.

$$\begin{array}{ccccc} J^1\ j\text{-th mate state} & \leftrightharpoons & J^0\ j\text{-th} & & \text{F coding} \\ \updownarrow & \text{C coding base} & \updownarrow & & + \\ \forall x_i \in J^1, x_i = 1 & \rightleftharpoons & \forall x_i \in J^0, x_j = 0 & & \text{General Conjugate} \end{array}$$

Some coding samples are listed in following table:

No.	7	6	5	4	3	2	1	0	Normal sequential number
SL	111	110	101	100	011	010	001	000	Ordering sequence
Truth	0	0	0	1	1	1	1	0	G: $J = 30$; W: $\langle 1\|12\rangle$
Variant	1	1	0	1	0	0	1	0	G: $J = 210$; W: $\langle 13\|2\rangle$
W	111	110	010	011	001	000	100	101	General Conjugate, without pairs
Truth	0	0	1	1	1	0	1	0	G: $J = 58$; W: $\langle 3\|10\rangle$
Variant	1	1	0	0	1	0	1	0	G: $J = 202$; W: $\langle 12\|10\rangle$
F	111	110	101	100	000	001	010	011	Meta states in pairs
Truth	0	0	0	1	0	1	1	1	G: $J = 23$; F: $\langle 1\|7\rangle$
Variant	1	1	0	1	0	1	0	0	G: $J = 212$; F: $\langle 13\|4\rangle$
C	111	110	010	011	000	001	101	100	General Conjugate + pairs
Truth	0	0	1	1	0	1	0	1	G: $J = 54$; C: $\langle 3\|5\rangle$
Variant	1	1	0	0	0	1	0	1	G: $J = 197$; C: $\langle 12\|5\rangle$

4.4 *Sizes of Variant Spaces*

Definition 14 Under $QV + P$ operations, W, F and C coding schemes are defined as WV, FV and CV coding schemes, respectively.

Theorem 2 *For a W coding scheme of n variables, it has a total of* $2^{2^n} \cdot 2^n!$ *cases distinguished.*

Theorem 3 *For a WV coding scheme of n variables, it has a total of* $2^{n+1} \cdot 2^n!$ *cases distinguished.*

Theorem 4 *For a F coding scheme of n variables, it has a total of* $2^{2^n} \cdot 2^{2^{n-1}} \cdot 2^{n-1}! = 2^{2^n(1+1/2)} \cdot 2^{n-1}!$ *cases distinguished.*

Theorem 5 *For a FV coding scheme of n variables, it has a total of* $2^{n+1} \cdot 2^{2^{n-1}} \cdot 2^{n-1}! = 2^{2^n+n+1} \cdot 2^{n-1}!$ *cases distinguished.*

Theorem 6 *For a C coding scheme of n variables, it has a total of* $2^{2^n} \cdot 2^{n-1}!$ *cases distinguished.*

Theorem 7 *For a CV coding scheme of n variables, it has a total of* $2^{n+1} \cdot 2^{n-1}!$ *cases distinguished.*

Using definitions of different coding schemes, shown in various sequences of one variable cases in the following table:

Function	Truth W coding		Variant W coding		Invariant WV coding		False WV coding	
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	$\langle 0\|0\rangle$
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	$\langle 0\|0\rangle$

using 2D W coding to arrange 1D sequences into 2D matrices:

Original:

Truth	Variant
$0\ \bar{x}$	$x\ 1$
$x\ 1$	$0\ \bar{x}$
$\bar{x}\ 0$	$1\ x$
$1\ x$	$\bar{x}\ 0$
Invariant	False

Permutation:

Truth	Variant
$0\ x$	$x\ 0$
$\bar{x}\ 1$	$1\ \bar{x}$
$\bar{x}\ 1$	$1\ \bar{x}$
$0\ x$	$x\ 0$
Invariant	False

5 Invariant Properties of Variant Constructions

It is interesting to notice that under QV operations, there are $2n + 2$ vectors available to generate QVS. This makes significant differences among classical logic and Variant logic construction [50–56]. The main results of this chapter are summarized in the following theorems.

Theorem 8 (Four Invariant Points for One Variable Condition) *For a W coding scheme under one variable condition, four points of the structure correspond to four functions:* $\{0, x, \bar{x}, 1\}$*, respectively.*

Proof When $n = 1$, four vectors are available for any Q or QV operations. ■

Theorem 9 (Two Invariant Points for Truth and False Schemes) *For any* $n > 1$, *W(WV) coding schemes, for any truth or false representation, only full 0 or full 1 valued vectors can be invariant undertaken* P *operations.*

Proof Undertaken P operation, if there is any not full 0 or 1 vectors, its binary number sequences will be changed. ■

Theorem 10 (Four Invariant Points for C Coding Scheme) *For any C (CV) coding scheme in variant construction, four corner positions of 2D function matrix have extreme invariant properties.*

Proof Under C(CV) coding scheme, four functions:$\{0, x, \bar{x}, 1\}$ correspond as follows: $x = \langle 0|0\rangle$; $0 = \langle 2^{2^{n-1}} - 1|0\rangle$; $1 = \langle 0|2^{2^{n-1}} - 1\rangle$; $\bar{x} = \langle 2^{2^{n-1}} - 1|2^{2^{n-1}} - 1\rangle$. Four positions are all corner points of the variant matrix. ■

6 Comparison

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

Var	State	Function	ExPower	SL	W coding	WV coding	C coding	CV coding
n	2^n	2^{2^n}	$2^n!$	1	$2^{2^n}2^n!$	$2^{n+1}2^n!$	$2^{2^n}2^{n-1}!$	$2^{n+1}2^{n-1}!$
1	2	4	2	1	8	8	4	4
2	4	16	24	1	384	192	32	32
3	8	256	40320	1	10321920	645120	6144	384
4	16	2^{16}	16!	1	$2^{16}16!$	$32 \cdot 16!$	$2^{16} \cdot 8!$	$32 \cdot 8!$
5	32	2^{32}	32!	1	$2^{32}32!$	$64 \cdot 32!$	$2^{32} \cdot 16!$	$64 \cdot 16!$

where we use Var: variable number; State: state number; Function: function number; ExPower: exponent power products; SL: SL coding number; W coding: W coding number under $Q + P$ operations; WV coding: WV coding number under $QV + P$ operations; C coding: C coding number under $Q + P$ operations; CV coding: CV coding number under $QV + P$ operations in the table, respectively.

7 Conclusion

In this chapter, variant logic has been proposed to extend truth table representation that describes variant properties of binary sequences. This extension is requiredto expand traditional Boolean logic framework to a new variation space. Under two types of vector operations, the new space has $2^{2^n} 2^n!$ times more complexity than traditional Boolean function space with 2^{2^n} members. In order to manage this complexity, the framework has proposed a series of global coding schemes encoded through symmetric properties representing the elements in a matrix as a 2D map. Under this two-dimensional model, coding mechanism can be constructed and their invariant properties can be discussed.

Boolean function space represents a core invariant functional space and the newly expanded space broadens the descriptions and coding schemes used. Thus, a wide area of variation coding can be developed. In essence, the space of binary sequence functions can be thought of as a keyboard with 2^{2^n} notes. Each note contains a complete Boolean function set and its own representation. The set of notes can be represented using a coding scheme that orders the notes in a particular sequence (SL and G codes) or their 2D maps (W, F and C codes).

Under W coding representation mechanism, 2D matrix is suitable to visualize permutation sequences of n variable logic structures. Using invariant properties, classical logic and variant logic can be clearly identified. Further work on dynamic behaviours of complex dynamic systems can be explored. This chapter outlines the construction and notation of variant logic only. Future papers will show that the proposed scheme, with its foundation in symmetry, will have definite uses for predicting convergent and chaotic behaviour in dynamic binary systems such as the analysis of cellular automata rules using various visual methodologies.

References

1. G. Birkhoff, *Lattice Theory*, vol. 25, 3rd edn. (American Mathematical Society Colloquium Publications, 1984)
2. V.N. Salii, *Lattices with Unique Complements* (Amer. Math. Soc. 1988)
3. F. Maeda, S. Maeda, *Theory of Symmetric Lattices* (Springer, Berlin, 1970)
4. A.B. Rosser, A.R. Turquette, *Many-Valued Logics* (North-Holland Publishing Company, Amsterdam, 1952)
5. S. Vickers, *Topology via Logic* (Cambridge University Press, Cambridge, MA, 1989)
6. R. Sikorski, *Boolean Algebra* (Springer, Berlin, 1960)
7. F.H. Edwards, *The Principles of Switching Circuits* (MIT Press, Cambridge, MA, 1973)
8. S.C. Lee, Vector Boolean algebra and calculus. IEEE Trans. Comput. **C–25**(9), 865–874 (1976)
9. S.C. Lee, *Modern Switching Theory and Digital Design* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1978)
10. S. Muroga, *Logic Design and Switching Theory* (Wiley-Interscience Publication, 1979)
11. A. Thayse, *Boolean Calculus of Differences* (Springer, Berlin, 1981)
12. K.H. Kim, *Boolean Matrix Theory and Applications* (Marcel Dekker Inc., New York, 1982)
13. W. Markek, J. Onyszkiewicz, *Elements of Logic and Foundations of Mathematics in Problems* (Kluwer Academic Publishers Group, Dordrecht, 1982)
14. P.E. Dunn, *The Complexity of Boolean Networks* (Academic Press, London, 1988)
15. M. Bonnet, *Handbook of Boolean Algebras* (North Holland, New York, 1989)
16. M.S. Paterson (ed.), *Boolean Function Complexity* (Cambridge University Press, Cambridge, MA, 1992)
17. S.P. Vingron, *Switching Theory: Insight Through Predicate Logic* (Springer, Berlin, 2004)

18. R.P. Burn, *Groups: A Path to Geometry* (Cambridge University Press, Cambridge, MA, 1985)
19. M. Greutz, *Quarks, Gluons and Lattices* (Cambridge University Press, Cambridge, MA, 1983)
20. F.P. Greenleaf, *Invariant Means on Topological Groups* (Van Nostrand Reinhold Company, 1969)
21. J. Fogarty, *Invariant Theory* (W.A. Benjamin Inc., 1969)
22. T.A. Springer, *Invariant Theory* (Springer, Berlin, 1977)
23. J. Von Neumann, The general and logical theory of automata, in *Collected Works* vol. 5, ed. by J. von Neumann (1963)
24. S. Noguchi, Oizumi, A survey of cellular logic. J. Inst. Electron. Commun. Engrg. **54**(2), 206–220 (1971)
25. A.W. Burks, Cellular automata and natural systems, in *Proceedings of the 5th Congress of Deutsche Gessellschaft für Kybernetik* (1973)
26. T. Toffoli, Cellular Automata Mechanics. Ph.D. Thesis (University of Michigan, 1977)
27. A. Kandel, S.C. Lee, *Fuzzy Switching and Automata: Theory and Applications* (Crane Russak & Company Inc., New York, 1979)
28. K. Preston Jr., M.J.B. Duff, *Modern Cellular Automata: Theory and Application* (Plenum Press, New York, 1984)
29. S. Wolfram, *Theory and Applications of Cellular Automata* (World Scientific, Singapore, 1986)
30. C.G. Langton, Life at the edge of chaos, in *Artificial Life II* (Addison-Wesley, Reading, MA, 1992)
31. S. Wolfram, *Cellular Automata and Complexity* (Addison-Wesley, Reading, MA, 1994)
32. S. Wolfram, *A New Kind of Science* (Wolfram Media Inc., 2002). http://www.wolframscience.com/
33. A. Ilachinski, *Cellular Automata-A Discrete Universe* (World Scientific, Singapore, 2001)
34. D. Griffeath, C. Moor, *New Constructions in Cellular Automata* (Santa Fe Institute Studies in the Sciences of Complexity, 2003)
35. H. Umeo, S. Morishita, K. Nishinari, *Cellular Automata* (Springer, Berlin, 2008)
36. Z.J. Zheng, A.J. Maeder, *The Conjugate Classification of the Kernel Form of the Hexagonal Grid, Modern Geometric Computing for Visualization* (Springer, Berlin, 1992), pp. 3–89
37. Z.J. Zheng, A.J. Maeder, *The Elementary Equation of the Conjugate Transformation for Hexagonal Grid, Modeling in Computer Graphics* (Springer, Berlin, 1993), pp. 21–42
38. Z.J. Zheng, Conjugate Transformation of Regular Plan Lattices for Binary Images. Ph.D. Thesis (Monash University, 1994)
39. Z.J. Zheng, *Conjugate Visualisation of Global Complex Behaviour, Complexity International*, vol. 3 (1996). http://www.complexity.org.au/ci/vol03/zheng/
40. Z.J. Zheng, C.H.C. Leung, Visualising global behaviour of 1D cellular automata image sequences in 2D maps. Phys. A **233**, 785–800 (1996)
41. J. Needham, *Science and Civilisation of China vol. 2 History of Scientific Thought* (Cambridge Press, Cambridge, 1954–1988)
42. D.F. Hook, *The I Ching and Mankind* (Routledge and Kegan Paul Ltd, London, 1975)
43. W.K. Chu, W.A. Sherrill, *An Anthology of I Ching* (Routledge and Kegan Paul Ltd, London, 1977)
44. I.K. Shchutshii, *Researches on the I Ching* (Princeton University Press, Princeton, 1979)
45. H. Wilhelmi, *Chang: Eight Lectures on I Ching* (Princeton Press, Princeton, 1979)
46. R. Wilhemi, *Lectures on the I Ching: Constancy and Change* (Princeton University Press, Princeton, 1979)
47. J.C. Cooper, *Yin and Yang, The Taoist Harmony of Opposites* (The Thetford Press, 1981)
48. L. A. Govinda, *The Inner Structure of I Ching: The Book of Transformation* (Wheelwright Press, 1981)
49. G. Whincup, *Rediscovering of I Ching* (GreyWhincup, 1986)
50. E.S. Fedorov, Symmetry of Crystals (1890) (American Crystallographic Association, 1971)
51. H. Weyl, *Symmetry* (Princeton, 1952)
52. P. Benacerraf, H. Putnam, *Philosophy of Mathematics* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1964)

53. C.H. McGillavry, *Symmetry Aspects of M.C. Escher's Periodic Drawings* (A. Oosthoek's Uitgeversmaatschappij N.V., Utrecht, 1965)
54. P.B. Yale, *Geometry and Symmetry* (Holden Day, 1968)
55. A. Holden, *Shapes, Spaces and Symmetry* (Columbia Uni. Press, New York, NY, 1971)
56. F.H. Bool, *Escher: with A Complete Catalogue of Graphic Works* (Thames Hudson, 1982)