Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 357 (2020) 9871–9884

www.elsevier.com/locate/jfranklin

Completeness and normal form of multi-valued logical functions☆

Daizhan Cheng[a,*], Zequn Liu[a,b], Hongsheng Qi[a,b]

[a] *Key Laboratory of Systems and Control, Academy of Mathematics and Systems Sciences, Chinese Academy of Sciences, Beijing 100190, PR China*

[b] *School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, PR China*

Received 14 May 2019; received in revised form 8 June 2020; accepted 30 June 2020
Available online 9 July 2020

Abstract

Theory of completeness is essential for multi-valued logical functions. Using semi-tensor product (STP) of matrices, the algebraic form of k-valued logical functions is presented. Using algebraic form, a method is proposed to construct an adequate set of connectives (ASC), consisting of unary operators with conjunction/disjunction for k-valued logical functions, which can be used to express any k-valued logical functions. Based on it, two normal forms of k-valued logical functions are presented, which are extensions of the disjunctive normal form and conjunctive normal form of Boolean functions respectively. The ASC is then simplified to a condensed set. Finally, the normal forms are further extended to mix-valued logical functions.

1. Preliminaries

"Logic and set theory are sometimes called the foundations of mathematics, because they are used as a basis for other branches of mathematics." [11]. As a natural extension of standard 2-valued logic (or Boolean logic), multi-valued logic (or k-valued logic) has been widely applied to computer science, automata, and circuit design, etc. [5,12].

☆ This work is supported partly by the National Natural Science Foundation of China (NSFC) under Grants 61773371, 61873262, and 61733018, and the Youth Innovation Promotion Association of the CAS.

* Corresponding author.

E-mail addresses: dcheng@iss.ac.cn (D. Cheng), liuzequn@amss.ac.cn (Z. Liu), qihongsh@amss.ac.cn (H. Qi).

https://doi.org/10.1016/j.jfranklin.2020.06.026

A logical variable χ can take a value from $\mathcal{D} = \{0, 1\}$. A unary logical operator is a mapping: $\mathcal{D} \to \mathcal{D}$, and a binary logical operator is a mapping $\mathcal{D} \times \mathcal{D} \to \mathcal{D}$. Usually logical operators with more than two logical variables are called logical functions [9].

Two problems about k-valued logic are considered in this paper: (i) completeness; and (ii) normal form. To make the problems addressed in this paper clear, we explain the corresponding problems for classical logic (i.e., 2-valued logic) first. We refer to [4] or any standard textbook for notations and concepts of classical logic.

(1) Completeness problem:

Find a set of logic generators, which is called an adequate set of connectives (ASC) [4], such that any logical function can be expressed as a compounded function of this set of generators. ASC is not unique. For instance, (i) $A_c := \{\neg, \wedge\}$, (ii) $A_d := \{\neg, \vee\}$, (iii) $A_{cd} = \{\neg, \wedge, \vee\}$ are commonly used ASCs.

It was pointed out that [7]: "In the theory of multiple-valued logic, the problem of verifying the completeness of a set of logical functions is a fundamental and important problem. This problem must be solved when the multiple-valued logic is applied to automata, multiple-valued logic circuit, etc." However, to our best knowledge, despite of quite a number of contributions have been published on the completeness of multiple logic, this long standing problem has not been solved completely.

(2) Normal form:

It is well known that [4] a Boolean function always has a disjunctive normal form and a conjunctive normal form. A disjunctive (or conjunctive) normal form of Boolean functions is based on the ASC $\{\neg, \wedge, \vee\}$. Since the ASC of k-valued logical functions is still not clear, the normal form of k-valued logical functions is also an open problem.

Completeness and normal form are two closely related problems, which are important in analysis and property investigation of logical functions [11], logical circuit arrangement, decomposition, and design [7], control design of logical networks [2], etc. The contribution of this paper consists of (i) finding a convenient ASC, and (ii) providing normal forms for k-valued logic, as well as mix-valued logic.

Recently, a new mathematical tool, called semi-tensor product (STP) of matrices, has been proposed. It is the fundamental tool for the approach of this paper. By expressing a Boolean or multi-valued logical function in a matrix form, STP is successfully applied to the analysis and control of Boolean networks as well as of multi-valued logical networks [1,2]. We also refer to some STP survey papers for current developments, say, [3,6,8].

The main technology road map in this paper is as follows: Using STP, a k-valued logical function is converted into its algebraic form. From the algebraic form of a logical function, its normal forms are obtained. From the normal forms an ASC of k-valued logic is obtained, which is the set of unary operators with conjunction and disjunction. To reduce the size of this set of generators, the set of unary operators is divided into two parts: the non-singular set (Ψ_k^n), and the singular set (Ψ_k^s). Using group isomorphism, Ψ_k^n can be reduced to its set of generators G_k^n. Using equivalence, Ψ_k^s can be reduced to its set of generators G_k^s. Eventually, a much smaller compact adequate set is obtained.

Finally, the normal form expressions are also extended to mix-valued logical functions.

Before ending this section, some notations are presented as follows:

1. $\mathbb{R}^n$: the n-dimensional Euclidean space.
2. $\mathcal{M}_{m\times n}$: the set of $m\times n$ real matrices.
3. $Col(M)$ ($Row(M)$): the set of columns (rows) of M. $Col_i(M)$ ($Row_i(M)$): the ith column (row) of M.
4. $\mathcal{D} := \{0, 1\}$.
5. ${\delta_n}^i$: the ith column of the identity matrix I_n.
6. $\Delta_n := \left\{\delta_n^i | i = 1, \dots, n\right\}$; $\Delta := \Delta_2$.
7. $\mathbf{1}_\ell := (\underbrace{1, 1, \dots, 1}_{\ell})^{\mathrm{T}}$.
8. $\mathbf{1}_{p\times q} := (\underbrace{\mathbf{1}_p, \mathbf{1}_p, \dots, \mathbf{1}_p}_{q})$.
9. A matrix $L \in \mathcal{M}_{m\times n}$ is called a logical matrix if $Col(L) \subset \Delta_m$. We denote by $\mathcal{L}_{m\times n}$ the set of $m\times n$ logical matrices.
10. If $L \in \mathcal{L}_{n\times r}$, by definition it can be expressed as $L = [\delta_n^{i_1}, \delta_n^{i_2}, \dots, \delta_n^{i_r}]$. For the sake of compactness, it is briefly denoted by $L = \delta_n[i_1, i_2, \dots, i_r]$.
11. $\mathbf{S}_k$: the k-th order symmetric group, consisting of all permutations of k objects.

The rest of this paper is organized as follows: Section 2 introduces the algebraic form of multi-valued logical functions. Certain related properties are investigated. In Section 3 we consider ASC of multi-valued logic. Using symmetric group isomorphism and equivalent set of generators for nonsingular and singular unary operators respectively, a compact ASC is obtained. Moreover, the conjunctive and disjunctive normal forms for multi-valued logical functions are also obtained. Finally, the normal forms are also extended to mix-valued logic in Section 4. Section 5 is a brief conclusion.

2. Algebraic form of multi-valued logic

We first recall STP, which is the fundamental tool for deriving the algebraic form of logical functions.

Definition 2.1 [1]**.** Let $A \in \mathcal{M}_{m\times n}$, $B \in \mathcal{M}_{p\times q}$, and the least common multiple $lcm(n, p) = t$. Then the STP of A and B is defined as

$$A \ltimes B := \left(A \otimes I_{t/n}\right)\left(B \otimes I_{t/p}\right), \tag{1}$$

where $\otimes$ is Kronecker product.

Note that STP is a generalization of classical matrix product. That is, when $n = p$ STP degenerates to classical matrix product. Throughout this paper the default matrix product is STP, and in most cases the symbol $\ltimes$ is omitted.

Definition 2.2. A k-valued logical variable χ takes its values in

$$\mathcal{D}_k := \left\{1, \frac{k-2}{k-1}, \frac{k-3}{k-1}, \dots, \frac{1}{k-1}, 0\right\}, \quad k \geq 3. \tag{2}$$

We identify each logical value (α_i) with a vector (a_i) as

$$\alpha_i := \frac{k-i}{k-1} \Leftrightarrow a_i = \delta_k^i, \quad i = 1, 2, \dots, k. \tag{3}$$

Denote by Φ_k the set of unary operators on $\mathcal{D}_k$. Let $\sigma \in \Phi_k$. Then there exists a unique $\beta_\sigma = [\alpha_{i_1}, \ldots, \alpha_{i_k}]$, such that

$$\sigma(\alpha_i) = \alpha_{i_j}, \quad 1 \le j \le k.$$

Moreover, if χ is expressed in vector form as $x \in \Delta_k$, then $\sigma \in \Phi_k$ has matrix form

$$M_\sigma = \delta_k[i_1, i_2, \ldots, i_k].$$

Now if $\chi \in \mathcal{D}_k$, $\sigma \in \Phi_k$, and assume

$$\xi = \sigma(\chi).$$

Then in vector form we have

$$y = M_\sigma x,$$

where x and y are vector forms of χ and ξ respectively.

We give a simple example to describe this.

Example 2.3. Consider a 3-valued logic. Then

$$\begin{array}{lll} \alpha_1 = 1 & \Leftrightarrow & a_1 = \delta_3^1, \\ \alpha_2 = 0.5 & \Leftrightarrow & a_2 = \delta_3^2, \\ \alpha_3 = 0 & \Leftrightarrow & a_3 = \delta_3^3. \end{array}$$

Let $\sigma \in \Phi_3$, be defined as follow:

$$\sigma(1) = 0, \ \sigma(0.5) = 1, \ \sigma(0) = 1.$$

Then

$$\sigma(\alpha_1) = \alpha_3, \ \sigma(\alpha_2) = \alpha_1, \ \sigma(\alpha_3) = \alpha_1.$$

Hence,

$$M_\sigma = \delta_3[3, 1, 1].$$

Now define a product on Φ_k as the composition of two operators. That is, let $\sigma, \mu \in \Phi_k$. Then their product is defined by

$$\sigma \circ \mu(\chi) := \sigma(\mu(\chi)), \quad \chi \in \mathcal{D}_k.$$

It is obvious that $(\Phi_k, \circ)$ is a monoid (i.e., a semi-group with identity). (We refer to [10] for group (semi-group, monoid), homomorphism (isomorphism)).

Note that each $\sigma \in \Phi_k$ has matrix form $M_\sigma \in \mathcal{L}_k$. Let $\times$ be the classical matrix product. Then the following result is obvious.

Proposition 2.4. *The mapping $\pi : \Phi_k \to \mathcal{L}_k$, defined by $\sigma \mapsto M_\sigma$, is a monoid isomorphism.*

Proof. Let $\sigma, \mu \in \Phi_k$. A straightforward computation shows that

$$M_{\sigma \circ \mu} = M_\sigma \times M_\mu.$$

That is, π is a homomorphism. It is also easy to check that π is one-to-one and onto. So π is an isomorphism. $\square$

Note that this proposition allows us to investigate $(\Phi_k, \circ)$ through $(\mathcal{L}_k, \times)$. Since the latter has matrices as its elements, to investigate $\mathcal{L}_k$ is more convenient.

Some unary operators are of particular importance.

Definition 2.5.

(i) Constant Operator:
$\sigma \in \Phi_k$ is called a constant operator, if

$$\sigma(\chi) = \alpha_i, \forall \chi \in \mathcal{D}_k. \tag{4}$$

Equivalently,

$$M_\sigma = \delta_k[\underbrace{i, \ldots, i}_{k}]. \tag{5}$$

The set of constant operators is denoted by *Const*.

(ii) Negation:

$$\neg^{(k)}\chi = 1 - \chi, \quad \chi \in \mathcal{D}_k. \tag{6}$$

(iii) Dirac Operator ($\triangleright_k^i$):

$$\triangleright_k^i(\chi) = \begin{cases} 1, & \text{if } \chi = \alpha_i \\ 0, & \text{if } \chi \neq \alpha_i. \end{cases} \tag{7}$$

(iv) Dual Dirac Operator ($\triangleleft_k^i$):

$$\triangleleft_k^i(\chi) := \neg\triangleright_k^i = \begin{cases} 0, & \text{if } \chi = \alpha_i \\ 1, & \text{if } \chi \neq \alpha_i. \end{cases} \tag{8}$$

Next, we consider some binary operators.

Definition 2.6. Let $\xi, \ \eta \in \mathcal{D}_k$ be two k-valued variables. Then

(i) Conjunction:

$$\xi \wedge^{(k)} \eta = \min\{\xi, \ \eta\}. \tag{9}$$

(ii) Disjunction:

$$\xi \vee^{(k)} \eta = \max\{\xi, \ \eta\}. \tag{10}$$

Remark 2.7. The structure matrices of some operators on k-valued logic are shown as follows. They can be easily verified.

(i) $\neg^{(k)}$ is a unary operator. Its structure matrix is

$$M_n^{(k)} = \delta_k[k, k-1, \ldots, 1]. \tag{11}$$

(ii) Conjunction $\wedge^{(k)}$ is a binary operator, its structure matrix is

$$M_c^{(k)} = \delta_k[M_1, M_2, \ldots, M_k], \tag{12}$$

where

$$M_i = [\underbrace{i, \ldots, i}_{i}, i+1, i+2, \ldots, k], \quad i = 1, \ldots, k.$$

(iii) Disjunction $\vee^{(k)}$ is a binary operator, its structure matrix is

$$M_d^{(k)} = \delta_k[N_1, N_2, \ldots, N_k], \tag{13}$$

where

$$N_i = [1, 2, \ldots, i, \underbrace{i, \ldots, i}_{k-i}], \quad i = 1, \ldots, k.$$

All commonly used binary operators for 2-valued logic can be extended to k-valued logic by using $\neg^{(k)}$, $\wedge^{(k)}$, and $\vee^{(k)}$. For instance, it is well known that in the 2-valued case the conditional operator ($\rightarrow$) and the biconditional operator ($\leftrightarrow$) can be expressed as

$$\xi \rightarrow \eta = \neg\xi \vee \eta, \tag{14a}$$

$$\xi \leftrightarrow \eta = (\xi \rightarrow \eta) \wedge (\eta \rightarrow \xi). \tag{14b}$$

Using (14), we can define $\rightarrow^{(k)}$ and $\leftrightarrow^{(k)}$ as follows:

Definition 2.8. In the k-valued logic, we define

(i) $$\xi \rightarrow^{(k)} \eta := \neg^{(k)}\xi \vee^{(k)} \eta; \tag{15}$$

(ii) $$\xi \leftrightarrow^{(k)} \eta := (\xi \rightarrow^{(k)} \eta) \wedge^{(k)} (\eta \rightarrow^{(k)} \xi). \tag{16}$$

We use the following example to show how to derive the structure matrix of a multi-valued logical function.

We need some tools[1]:

(i) Power-reducing Matrix:

$$PR_k := diag(\delta_k^1, \delta_k^2, \ldots, \delta_k^k), \quad k = 2, 3, \ldots. \tag{17}$$

Then we have the following result:

Proposition 2.9. *Assume $x \in \Delta_k$, then*

$$x^2 = PR_k x. \tag{18}$$

(ii) Vector-Vector Swap: Define the $[m, n]$ swap matrix as

$$W_{[m,n]} := [I_n \otimes \delta_m^1, I_n \otimes \delta_m^2, \ldots, I_n \otimes \delta_m^m]. \tag{19}$$

Then we have the following:

Proposition 2.10. *Assume $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, then*

$$W_{[m,n]}xy = yx. \tag{20}$$

(iii) Vector-Matrix Swap:

Proposition 2.11. *Assume $x \in \mathbb{R}^t$, then for any matrix A, we have*

$$xA = (I_t \otimes A)x. \tag{21}$$

Example 2.12. Consider the 3-valued logic, and assume $\xi,\ \eta \in \mathcal{D}_3$, and x, y are the vector forms of ξ and η respectively. Then in vector form we have

(i)

$$\begin{aligned} x \rightarrow^{(3)} y &= M_i^{(3)} xy = M_d^{(3)} M_n^{(3)} xy \\ &= \delta_3[1, 1, 1, 1, 2, 2, 1, 2, 3]\delta_3[3, 2, 1]xy \\ &= \delta_3[1, 2, 3, 1, 2, 2, 1, 1, 1]xy. \end{aligned}$$

Hence

$$M_i^{(3)} = \delta_3[1, 2, 3, 1, 2, 2, 1, 1, 1]. \tag{22}$$

(ii)

$$\begin{aligned} x \leftrightarrow^{(3)} \ y &= M_e^{(3)} xy \\ &= M_c^{(3)} (M_d^{(3)} M_n^{(3)} xy) M_d^{(3)} M_n^{(3)} yx) \\ &= M_c^{(3)} M_d^{(3)} M_n^{(3)} (I_9 \otimes M_d^{(3)} M_n^{(3)}) xy^2 x \\ &= M_c^{(3)} M_d^{(3)} M_n^{(3)} (I_9 \otimes M_d^{(3)} M_n^{(3)}) xPR_3 yx \\ &= M_c^{(3)} M_d^{(3)} M_n^{(3)} (I_9 \otimes M_d^{(3)} M_n^{(3)}) (I_3 \otimes PR_3) xW_{[3,3]} xy \\ &= M_c^{(3)} M_d^{(3)} M_n^{(3)} (I_9 \otimes M_d^{(3)} M_n^{(3)}) (I_3 \otimes PR_3) (I_3 \otimes W_{[3,3]}) PR_3 xy \end{aligned}$$

Hence,

$$\begin{aligned} M_e^{(3)} &= M_c^{(3)} M_d^{(3)} M_n^{(3)} (I_9 \otimes M_d^{(3)} M_n^{(3)}) (I_3 \otimes PR_3) (I_3 \otimes W_{[3,3]}) PR_3 \\ &= \delta_3[1, 2, 3, 2, 2, 2, 3, 2, 1]. \end{aligned} \tag{23}$$

3. Normal form and ASC

Assume $F : \mathcal{D}_k^n \rightarrow \mathcal{D}_k$ is an n-variable k-valued logical function. Assume its algebraic form is

$$F(x_1, \ldots, x_n) := M_F \ltimes_{i=1}^n x_i, \tag{24}$$

where $M_F \in \mathcal{L}_{k \times k^n}$ is the structure matrix of F. Then we split M_F into k^{n-1} blocks as

$$M_F := [M_1, M_2, \ldots, M_{k^{n-1}}], \tag{25}$$

where $M_j \in \mathcal{L}_{k \times k},\ j = 1, \ldots, k^{n-1}$.

Then we define a set of unary operators as

$$\phi_j \in \Phi_k, \quad j = 1, \ldots, k^{n-1}, \tag{26}$$

which have M_j as their structure matrices respectively.

Similarly to the Boolean case, we have the following result.

Theorem 3.1. *Every k-valued logical function has its disjunctive normal form and its conjunctive normal form.*

Proof. We give a constructive proof for them. Assume a k-valued logical function $F(\chi_1, \ldots, \chi_k) : \mathcal{D}_k^n \rightarrow \mathcal{D}_k$ is given. Moreover, assume that the structure matrix of F is

$$M_F = [N_1, N_2, \ldots, N_k],$$

where $N_i \in \mathcal{L}_{k\times k^{n-1}}$, $i = 1, \ldots, k$. Similar to Boolean case, it is easy to prove that

$$\begin{aligned} F(\chi_1, \chi_2, \ldots, \chi_k) = & \left[\triangleright_k^1(\chi_1) \wedge F_1(\chi_2, \ldots, \chi_k)\right] \\ & \vee\left[\triangleright_k^2(\chi_1) \wedge F_2(\chi_2, \ldots, \chi_k)\right] \\ & \vee \ldots \\ & \vee\left[\triangleright_k^k(\chi_1) \wedge F_k(\chi_2, \ldots, \chi_k)\right], \end{aligned} \tag{27}$$

where F_i has N_i as its structure matrix, $i = 1, \ldots, k$.

By applying this procedure to each F_i and their sub-functions and using Eq. (26) at the last step, we have

- Disjunctive normal form:

$$\begin{aligned} F(\chi_1, \ldots, \chi_n) = & \bigvee_{i_1=1}^k \bigvee_{i_2=1}^k \cdots \bigvee_{i_{n-1}=1}^k \Big[\triangleright_k^{i_1}(\chi_1) \\ & \bigwedge \triangleright_k^{i_2}(\chi_2) \bigwedge \ldots \bigwedge \triangleright_k^{i_{n-1}}(x_{n-1}) \bigwedge \phi^{i_1,i_2,\ldots,i_k}(\chi_n)\Big]. \end{aligned} \tag{28}$$

  where $\phi^{i_1,i_2,\ldots,i_{n-1}} = \phi_j$ with

  $j = (i_1 - 1)k^{n-2} + (i_2 - 1)k^{n-3} + \ldots + (i_{n-2} - 1)k + i_{n-1}$.

- Conjunctive normal form:

Assume $\neg F(\chi_1, \ldots, \chi_n)$ has disjunctive normal form as Eq. (28). Then using De Morgan formula, we can have

$$\begin{aligned} F(\chi_1, \ldots, \chi_n) = & \bigwedge_{i_1=1}^k \bigwedge_{i_2=1}^k \cdots \bigwedge_{i_{n-1}=1}^k \\ & \Big[\triangleleft_k^{i_1}(\chi_1) \bigvee \triangleleft_k^{i_2}(\chi_2) \bigvee \ldots \bigvee \triangleleft_k^{i_{n-1}}(\chi_{n-1}) \ \bigvee \phi^{k+1-i_1,k+1-i_2,\ldots,k+1-i_k}(\chi_n)\Big]. \end{aligned} \tag{29}$$

Note that in Eqs. (28) and (29), $\vee$, $\wedge$, and $\neg$ are brief forms for $\vee^{(k)}$, $\wedge^{(k)}$, and $\neg^{(k)}$ respectively. □

Corollary 3.2. *For k-valued logic*

$$A_{cd}^k := \{\wedge^{(k)}, \vee^{(k)}, \Phi_k\} \tag{30}$$

is an ASC.

Since $|\Phi_k| = k^k$, it is an important and challenging task to find a condensed ASC. In the following we consider how to reduce the size of this ASC.

Define

$$\Psi_k^n := \{\sigma \in \Phi_k \mid \det(M_\sigma) \neq 0\},$$

$$\Psi_k^s := \{\sigma \in \Phi_k \mid \det(M_\sigma) = 0\}.$$

Then

$$\Phi_k = \Psi_k^n \cup \Psi_k^s.$$

Denote by

$$M_k^n := \{M_\sigma \mid \sigma \in \Psi_k^n\},$$

$$M_k^s := \{M_\sigma \mid \sigma \in \Phi_k^s\}.$$

We define a mapping $\pi\colon \Phi_k \to M_k$ as

$$\pi : \sigma \mapsto M_\sigma.$$

By restricting it on Ψ_k^n and Ψ_k^s we have $\pi : \Psi_k^n \to M_k^n$ and $\pi : \Psi_k^s \to M_k^s$ respectively.

Then the following relations are obvious.

Proposition 3.3.

(i) $\pi : \Psi_k^n \to M_k^n$ *is a group isomorphism.*
(ii) $\pi : \Psi_k^s \to M_k^s$ *is a semigroup isomorphism.*

Because of these isomorphisms, instead of Ψ_k^n and Ψ_k^s, we can investigate M_k^n and M_k^s respectively.

It is well known that $\{(1, t) \mid t = 2, \dots, k\}$ is a set of generators of $\mathbf{S}_k$ [10]. Then M_k^n has a set of generators

$$G_k^n := \{\pi(\sigma_t) \mid \sigma_t = (1, t),\ t = 2, \dots, n\}.$$

Note that $\left|\Psi_k^n\right| = k!$ and $\left|G_k^n\right| = k - 1$. Hence, the number of this part of elements in the ASC has been reduced from $k!$ to $k - 1$.

Through G_k^n we can easily obtain a basis for M_k^n. We give an example to explain this.

Example 3.4. In Ψ_3^n, we have a basis $\{(1,\ 2),\ (1,\ 3)\}$. The corresponding basis of M_3^n is $\{M_{(1,2)}, M_{(1,3)}\}$, where

$$M_{(1,2)} = \delta_3[2, 1, 3], \quad M_{(1,3)} = \delta_3[3, 2, 1].$$

Next, we consider Ψ_k^s. We first define an equivalence on Ψ_k^s as follows:

Definition 3.5.

(i) Two unary operators $\sigma_1,\ \sigma_2 \in \Psi_k^s$ are said to be equivalent, and this is denoted by $\sigma_1 \sim \sigma_2$, if there exist $\mu_1,\ \mu_2 \in \Psi_k^n$, such that

$$\sigma_1 \circ \mu_1 = \mu_2 \circ \sigma_2. \tag{31}$$

(ii) Two matrices $M_1,\ M_2 \in M_k^s$ are said to be equivalent, and this is denoted by $M_1 \sim M_2$, if there exist $P_1,\ P_2 \in M_k^n$, such that

$$M_1 P_1 = P_2 M_2. \tag{32}$$

The following proposition is obvious.

Proposition 3.6. *Assume* $\sigma,\ \mu \in \Psi_k^s$. *Then*

$$\sigma \sim \mu \Leftrightarrow M_\sigma \sim M_\mu.$$

Since Ψ_k^n can be generated by G_k^n, if two operators in Ψ_k^s are equivalent, then one can be generated from the other one (under the left and right actions of Ψ_k^n). Then we only need to choose one representative operator from each equivalence class as an element of the generator set, denoted by G_k^s.

According to Proposition 3.6, we need only to consider the equivalence class on M_k^s. Two rows are said to be equivalent (denoted by $\sim$), if they have same numbers of "1". For example, $(1, 1, 0, 0) \sim (0, 1, 0, 1)$.

Let $M_1,\ M_2 \in M_k^s$. M_1 and M_2 are said to be equivalent, denoted by $M_1 \sim M_2$, if they have one-one corresponding equivalent rows. For example, we have

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \sim M_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \tag{33}$$

because

$$Row_1(M_1) \sim Row_3(M_2),$$
$$Row_2(M_1) \sim Row_1(M_2),$$
$$Row_3(M_1) \sim Row_2(M_2).$$

Consider "1" as a dove, "row" as a cage, the number of equivalence classes, denoted by $n(k)$, is equivalent to the following classical dove-cage problem: put k doves into $k-1$ cages. This is because $\sigma \in \Psi_k^s$, which means M_σ is singular. In addition, M_σ is a logical matrix, so at least there is one zero row. Note that the cages are not labeled, because all the rows are not distinguished. (To see this, we recall (33), where three doves have been put into two cases: in M_1, two doves are put into cage 1 and one dove is put into cage 2; in M_2, two doves are put into cage 3 and one dove is put into cage 1. These two cases are considered as the same. That is, we consider cages 1, 2, 3 are indistinguishable.)

Then we can choose a representative from each equivalence class to form a set of generators, denoted by G_k^s. Hence, we have $\left|G_k^s\right| = n(k)$.

The representative is chosen as follows: Denote the number of doves in row i by a_i, we require

$$a_1 \geq a_2 \geq \cdots \geq a_{k-1} \geq a_k = 0. \tag{34}$$

Then each equivalence class has a unique representative M, such that $Row_i(M)$ has the number of 1 equal to a_i. We call this class as $(a_1, a_2, \ldots, a_k)$ class.

Denote by $S(m, s)$ the set of decreasing nonnegative integer sequences, where the starting element is less than or equal to s, and the total sum is m. That is,

$$S(m, s) = \left\{ (a_1 \geq a_2 \geq \ldots \geq a_m) \mid a_1 \leq s, \sum_{i=1}^{m} a_i = m \right\}.$$

Denote the number of such sequences by

$$N(m, s) = |S(m, s)|.$$

Now we consider $n(k)$. Assume k doves are put into $k-1$ cases with the numbers of doves in decreasing order (such as in (34)). Then $a_1 \geq 2$. a_1 cannot be one, because in this case $k-1$ cages can have at most $k-1$ doves.

Note that if $a_1 = i$, then the number of possible (representative) matrices is $N(k-i, \min\{i, k-i\})$. Hence we have

$$n(k) = \sum_{i=2}^{k} N(k-i, \min\{i, k-i\}). \tag{35}$$

To calculate $N(m, s)$, we set $t = \left[\frac{n}{s}\right]$, where $[a]$ is the integral part of a. Hence, t means at most we can have how many nonzero elements for each sequence in $S(m, s)$. Then a recursive formula can be obtained easily as

$$N(n, s) = \begin{cases} 1, & \text{if } \min\{n, s\} = 1 \\ \sum_{j=0}^{t} N(n-js, s-1), & \text{otherwise.} \end{cases} \tag{36}$$

Example 3.7. Assume $m = 5$, $s = 3$, we calculate $N(5, 3)$. Since $t = \left[\frac{5}{3}\right] = 1$, then we have

$N(5, 3) = N(5, 2) + N(2, 2).$

Similarly, we have

$N(5, 2) = N(5, 1) + N(3, 1) + N(1, 1) = 3;$

$N(2, 2) = N(2, 1) + N(0, 1) = 2.$

That is: $N(5, 3) = 5$. It is easy to verity that the corresponding sequences are

$s_1 = \{1, 1, 1, 1, 1, 0, \dots\}$

$s_2 = \{2, 1, 1, 1, 0, \dots\}$

$s_3 = \{2, 2, 1, 0 \dots\}$

$s_4 = \{3, 1, 1, 0, \dots\}$

$s_5 = \{3, 2, 0 \dots\}.$

Using formula (35)–(36), it is easy to calculate that

$n(2) = 1, \quad n(3) = 2, \quad n(4) = 4, \quad n(5) = 6,$

$n(6) = 10, \quad n(7) = 14, \quad n(8) = 21, \quad \dots$

Note that $s(k) := \left|\Psi_k^s\right| = k^k - k!$, then we have

$s(2) = 2, \quad s(3) = 21, \quad s(4) = 232, \quad s(5) = 3005,$

$s(6) = 45936, \quad s(7) = 818503, \quad s(8) = 16736896, \quad \dots$

One sees easily that the number of generators has been tremendously reduced.

Using *Const*. as the equivalence class of constant mappings, then we do not need these mappings in an ASC because they can be replaced by constant numbers. So we have the following result:

Proposition 3.8.

$$G_k := \left\{G_k^n, G_k^s \backslash Const., \vee^{(k)}, \wedge^{(k)}\right\} \tag{37}$$

is an adequate set of k-valued logic.

In fact, using De Morgan formula, $\vee^{(k)}$ can be expressed by $\wedge^{(k)}$ and $\neg^{(k)}$ and vice versa. So we do not need to keep both of them.

Example 3.9. Consider the 3-valued logic.

(i) We have

$$G_3^n = \{\delta_3[2,1,3]; \delta_3[3,2,1]\};$$

$$G_3^s \backslash Const. = \{\delta_3[1,1,2]\}$$

(ii) We may construct an ASC as:

$$\{\delta_3[2,1,3], \delta_3[3,2,1], \delta_3[1,1,2], \wedge^{(3)}\}. \tag{38}$$

(iii) Is (38) an ASC of smallest size? In fact, it is not. Similar to the 2-valued case, we may replace $\wedge^{(3)}$ by

$$\uparrow^{(3)}(\xi,\eta) := \neg^{(3)}\left[\wedge^{(3)}(\xi,\eta)\right].$$

Setting $\xi = 1$, we have $\neg^{(3)}(\eta)$, that is, the $\neg^{(3)}$ is obtained.
We also have

$$\neg\uparrow^{(3)}(\xi,\eta) = \wedge^{(3)}(\xi,\eta).$$

It follows that

$$\{\delta_3[2,1,3], \delta_3[1,1,2], \uparrow^{(3)}\}$$

is also an adequate set.

(iv) G_k^s is the set of equivalence classes, which may not be the smallest set of generators for Φ_k^s, because some elements in G_k^n can be generated by others. For instance, consider the 4-valued logic. Let $A = \delta_4[1,1,2,3]$ and $B = \delta_4[1,2,2,4]$. $A \sim B$ because both of them are in $s = (2,1,1,0)$ class. But $AB = \delta_4[1,1,1,3]$, which is in $s = (3,1,0,0)$ class. Hence (3,1,0,0) class can be generated by (2,1,1,0) class. Hence (3,1,0,0) class is not necessary to be involved into G_4^s.
Hence finding an ASC of smallest size is, in general, a challenging task. So we may accept G_k, defined by (35), as a non-minimum but acceptable ASC for k-valued logic.

4. Mix-valued logic

Definition 4.1. Assume $\chi_i \in \mathcal{D}_{k_i}$, $i = 1, \dots, n$, a mapping $F : \prod_{i=1}^n \mathcal{D}_{k_i} \to \mathcal{D}_{k_0}$, denoted by $F(\chi_1, \dots, \chi_n) \in \mathcal{D}_{k_0}$, is called a mix-valued logical function.

Proposition 4.2 [2]. *Given a mix-valued logical function $F : \prod_{i=1}^n \mathcal{D}_{k_i} \to \mathcal{D}_{k_0}$, assume $\chi_i \in \mathcal{D}_{k_i}$ has vector form $x_i \in \Delta_{k_i}$, $i = 1, \dots, n$. Then, there exists a unique logical matrix $M_F \in \mathcal{L}_{k_0 \times k}$ ($k = \prod_{i=1}^n k_i$), such that*

$$F(x_1, \dots, x_n) = M_F \ltimes_{i=1}^n x_i, \quad x_i \in \Delta_{k_i}. \tag{39}$$

Definition 4.3. Assume $\chi \in \mathcal{D}_p$, $\xi \in \mathcal{D}_q$, where $p \neq q$, we define

$$\begin{cases} \chi \wedge \xi = \xi \wedge \chi = \chi, & \text{if } \xi = 1 \\ \chi \vee \xi = \xi \vee \chi = \chi, & \text{if } \xi = 0. \end{cases}$$

Splitting M_F into $t = k/k_n$ equal blocks as

$$M_f = [L^{1,\ldots,1}, \ldots, L^{1,\ldots,k_{n-1}}, \ldots, L^{k_1,k_2,\ldots,k_{n-1}}],$$

we define a set of unary operators $\phi^{i_1,i_2,\ldots,i_{n-1}} : \mathcal{D}_{k_n} \to \mathcal{D}_{k_0}$ with structure matrix $L^{i_1,i_2,\ldots,i_{n-1}}$, $i_s = 1, \ldots, k_s,\ s = 1, \ldots n-1$. Then we have the following normal form.

Theorem 4.4. *Assume $F : \prod_{i=1}^n \mathcal{D}_{k_i} \to \mathcal{D}_{k_0}$ is an n-variable mix-valued logical function with algebraic form*

$$F(x_1, \ldots, x_n) := M_F \ltimes_{i=1}^n x_i, \tag{40}$$

where $M_F \in \mathcal{L}_{k_0 \times k}$ is the structure matrix of F.

Then it has

- *Disjunctive normal form:*

$$\begin{aligned} F(\chi_1, \ldots, \chi_n) = \bigvee_{i_1=1}^{k_1} \bigvee_{i_2=1}^{k_2} \cdots \bigvee_{i_{n-1}=1}^{k_{n-1}} \\ \Big[\triangleright_{k_1}^{i_1}(\chi_1) \bigwedge \triangleright_{k_2}^{i_2}(\chi_2) \bigwedge \ldots \bigwedge \triangleright_{k_{n-1}}^{i_{n-1}}(\chi_{n-1}) \bigwedge \phi^{i_1,\ldots,i_{k-1}}(\chi_n) \Big]. \end{aligned} \tag{41}$$

- *Conjunctive normal form:*
  Assume $\neg F(\chi_1, \ldots, \chi_n)$ has disjunctive normal form as (41). Then using De Morgan formula, the conjunctive normal form of $F(\chi_1, \ldots, \chi_n)$ can be obtained as

$$\begin{aligned} F(\chi_1, \ldots, \chi_n) = \bigwedge_{i_1=1}^{k_1} \bigwedge_{i_2=1}^{k_2} \cdots \bigwedge_{i_{n-1}=1}^{k_{n-1}} \\ \Big[\triangleleft_{k_1}^{i_1}(\chi_1) \bigvee \triangleleft_{k_2}^{i_2}(\chi_2) \bigvee \ldots \bigvee \triangleleft_{k_{n-1}}^{i_{n-1}}(\chi_{n-1}) \ \bigvee \phi^{k_0+1-i_1,\ldots,k_0+1-i_{k-1}}(\chi_n) \Big]. \end{aligned} \tag{42}$$

5. Conclusion

In this paper, we first reviewed the algebraic form of k-valued logical functions. Then the conjunctive and disjunctive normal forms for k-valued logic were firstly presented. The completeness of k-valued logic was discussed via constructing ASC, which was then tremendously compressed by constructing a proper set of generators. Finally, the normal forms are also extended to mix-valued logic.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Cheng, H. Qi, Z. Li, Analysis and Control of Boolean Networks: A Semi-tensor Product Approach, Springer, London, 2011.

[2] D. Cheng, H. Qi, Y. Zhao, An Introduction to Semi-tensor Product of Matrices and Its Applications, World Scientific, Singapore, 2012.

[3] E. Fornasini, M.E. Valcher, Recent developments in boolean networks control, J. Control Dec. 3 (1) (2016) 1–18.

[4] A.G. Hamilton, Logic for Mathematicians, Revised Ed, Cambridge Univ. Press, Cambridge, 1988.

[5] M. Hu, H.T. Mouftah, Fault-tolerant system using 3-value logic circuits, IEEE Trans. Reliab. R-36 (2) (1987) 227–231.

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

[7] Z. Luo, M. Hu, T. Chen, Multiple-Valued Logic: Theory and Applications, Science Press, Beijing, 1992. (in Chinese)

[8] A. Muhammad, A. Rushdi, F.A.M. Ghaleb, A tutorial exposition of semi-tensor products of matrices with a stress on their representation of boolean function, JKAU Comp. Sci 5 (2016) 3–30.

[9] L. Rade, B. Westergren, Mathematics Handbook for Science and Engineering, 4th Ed, Studentlitteratur, Lund, 1998.

[10] D.S. Robinson, A Course in the Theory of Groups, Springer-Verlag, New York, Berlin, 1982

[11] E. Schechter, Classical and Nonclassical Logics: An Introduction to the Mathematics of Propositions, Princeton Univ. Press, New Jersey, 2015.

[12] Z.G. Vranesic, K.C. Smith, Electronic circuits for multi-valued digital systems, in: D.C. Rine (Ed.), Chapter 13 of Computer Science and Multiple-valued Logic: Theory and Applications, Elsevier Science Pub. B.V., 1984.