Chapter 6
Mix-valued Logic

In this chapter we first introduce the normal form of logic and k-valued logic. Using the normal form a generalized new logic, called the mix-valued logic, is proposed. Then the properties of general logical mappings are explored. Some applications are briefly introduced.

6.1 Normal Form of Logical Operators

Consider the set of r-ary logical operators (i.e., logical functions). Since there are r arguments and each argument can take 2 different values, so an r-ary logical operator is a mapping from a set of cardinality 2^r to a set of cardinality 2. Hence, there are 2^{2^r} different operators.

When $r = 0$, there are two nullary (0-ary) operators, which are $f \equiv 0$ and $f \equiv 1$. Where $r = 1$ there are 4 unary (1-ary) operators, which are $f(x) = x$, $f(x) = \neg x$, and the two nullary operators as its particular cases. When $r = 2$, there are 16 binary (2-ary) operators, which are listed in Table 5.6, including 4 unary operators as its particular cases. When $r = 3$ we have $2^{2^3} = 256$ ternary (3-ary) operators, and so on.

If we need all of these different forms to express all possible logical functions, it will be a terrible mess. Hence a natural question is: Is it possible to find a finite set of operators, which can be used to describe all the operators?

Definition 6.1. A set of logical operators is said to be an adequate set, if any operator can be expressed as a combination of them.

Proposition 6.1 ([2]). *The pairs* $\{\neg, \wedge\}$, $\{\neg, \vee\}$ *are adequate sets.*

In fact $\{\neg, \wedge, \vee\}$ is a commonly used adequate set. From Proposition 6.1 it is clear that $\{\neg, \wedge, \vee\}$ is an adequate set. Conversely, if $\{\neg, \wedge, \vee\}$ is an adequate set, then using De Morgan's law, Proposition 6.1 is clear.

It is easy to see that all the binary operators can be constructed from $\{\neg,\wedge,\vee\}$. (We leave the verification to the reader as an exercise.) Then how about r-ary operators when $r \geq 3$. Verifying all of them is impossible. But the following theorem solves this problem.

Theorem 6.1. *Assume $f(x_1,\cdots,x_r)$ is an r-ary operator with its structure matrix $M_f \in \mathscr{L}_{2\times 2^r}$, $r \geq 2$. Split M_f into two equal-size blocks as*

$$M_f = [B_1, B_2].$$

Then

$$f(x_1,\cdots,x_r) = (x_1 \wedge f_1(x_2,\cdots,x_r)) \vee (\neg x_1 \wedge f_2(x_2,\cdots,x_r)), \tag{6.1}$$

where $f_i(x_2,\cdots,x_r)$ has B_i as its structure matrix, $i = 1,2$.

Proof. First, we prove

$$f(x_1,\cdots,x_r) = (x_1 \wedge f(1,x_2,\cdots,x_r)) \vee (\neg x_1 \wedge f(0,x_2,\cdots,x_r)). \tag{6.2}$$

If $x_1 = 1$,

$$\begin{aligned} RHS &= (1 \wedge f(1,x_2,\cdots,x_r)) \vee (0 \wedge f(0,x_2,\cdots,x_r)) \\ &= f(1,x_2,\cdots,x_r) \vee 0 \\ &= f(1,x_2,\cdots,x_r) = LHS, \end{aligned}$$

if $x_1 = 0$,

$$\begin{aligned} RHS &= (0 \wedge f(1,x_2,\cdots,x_r)) \vee (1 \wedge f(0,x_2,\cdots,x_r)) \\ &= 0 \vee f(0,x_2,\cdots,x_r) \\ &= f(0,x_2,\cdots,x_r) = LHS, \end{aligned}$$

thus (6.2) follows.

Denote by $f_1(x_2,\cdots,x_r) = f(1,x_2,\cdots,x_r)$, $f_2(x_2,\cdots,x_r) = f(0,x_2,\cdots,x_r)$, it is easy to see where $f_i(x_2,\cdots,x_r)$ has B_i as its structure matrix. □

Using Theorem 6.1, an r-ary operator can be expressed by $\{\neg,\wedge,\vee\}$ and two $(r-1)$-ary operators. Continuing this process, it can be expressed by $\{\neg,\wedge,\vee\}$ and some unary operators. We call such a form the normal form of logical operators. In fact, this normal form is called the disjunctive normal form [2].

Next, we consider the k-valued logic. Similar to $k = 2$ case, it is easy to see that there are k^{k^r} r-ary logical operators. Particularly, there are k trivial nullary operators. We consider unary operators. There are k^k unary operators. We name some of them [1] (The operators are defined in their scalar values)

(i) Negation

$$\neg P := 1 - P. \tag{6.3}$$

Its structure matrix is

$$M_{n,k} = \delta_k[k \;\; k-1 \;\; \cdots \;\; 1]. \tag{6.4}$$

(ii) Rotator $\oslash_k$ is defined as

$$\oslash_k(P) := \begin{cases} P - \frac{1}{k-1}, & P \neq 0, \\ 1, & P = 0. \end{cases} \tag{6.5}$$

Its structure matrix, $M_{o,k}$, is

$$M_{o,k} = \delta_k\,[2 \;\; 3 \;\; \cdots \;\; k \;\; 1]\,. \tag{6.6}$$

For instance, we have

$$M_{o,3} = \delta_3[2\;3\;1], \quad M_{o,4} = \delta_4[2\;3\;4\;1]. \tag{6.7}$$

(iii) i-confirmor, $\triangledown_{i,k}$, $i = 1, \cdots, k$, are defined as

$$\nabla_{i,k}(P) = \begin{cases} 1, & P = \frac{k-i}{k-1}, \text{ (equivalently } P = \delta_k^i) \\ 0, & \text{otherwise.} \end{cases} \tag{6.8}$$

Its structure matrix (using same notation)

$$\nabla_{i,k} = \delta_k[\underbrace{k \cdots k}_{i-1} \; 1 \; \underbrace{k \cdots k}_{k-i}], \quad i = 1, 2, \cdots, k. \tag{6.9}$$

For instance, we have

$$\triangledown_{2,3} = \delta_3[3\;1\;3], \quad \triangledown_{2,4} = \delta_4[4\;1\;4\;4], \quad \triangledown_{3,4} = \delta_4[4\;4\;1\;4]. \tag{6.10}$$

(iv) Conjunction

$$P \wedge Q := \min\{P, Q\}. \tag{6.11}$$

Its structure matrix is (to save space let $n = 3$)

$$M_{c,3} = \delta_3[1\;2\;3\;2\;2\;3\;3\;3\;3]. \tag{6.12}$$

(v) Disjunction

$$P \vee Q := \max\{P, Q\}. \tag{6.13}$$

Its structure matrix is ($n = 3$)

$$M_{d,3} = \delta_3[1\;1\;1\;1\;2\;2\;1\;2\;3]. \tag{6.14}$$

In fact, we did not name all and the above set of operators are also not enough to construct all the unary operators. For statement ease, we use a general notation for all unary operators.

Definition 6.2. Let $i_1,\cdots,i_k \in \{1,2,\cdots,k\}$. The operator $\oslash_{i_1,i_2,\cdots,i_k}$ is a unary operator, defined by (in vector form)

$$\oslash_{i_1,i_2,\cdots,i_k}(\delta_k^j) = \delta_k^{i_j}, \quad j = 1,\cdots,k. \tag{6.15}$$

If $i_s = s$, which means when $x = \delta_k^s$, then $\oslash_{i_1,i_2,\cdots,i_k}(\delta_k^s) = \delta_k^s$. That is, x is invariant with respect to this operator. In this case, we replace i_s by $*$, which makes the operators more clear.

Example 6.1. Using the general expression, we have

$$\begin{aligned}
\neg_k &= \oslash_{k,k-1,\cdots,1};\\
\oslash_k &= \oslash_{2,3,\cdots,k,1};\\
\nabla_{i,k} &= \oslash_{\underbrace{k,\cdots,k}_{i-1},1,\underbrace{k,\cdots,k}_{k-i}}.
\end{aligned}$$

Next, we consider whether there is an expression of k-valued logical function similar to Theorem 6.1. In fact, we have the following.

Theorem 6.2. *Assume $f(x_1,\cdots,x_r)$ is an r-ary k-valued operator with its structure matrix $M_f \in \mathscr{L}_{k\times k^r}$, $r \geq 2$. Split M_f into k equal-size parts as*

$$M_f = [B_1,B_2,\cdots,B_k].$$

Then

$$\begin{aligned}
f(x_1,\cdots,x_r) = &\big(\nabla_{1,k}(x_1)\wedge f_1(x_2,\cdots,x_r)\big)\vee\big(\nabla_{2,k}(x_1)\wedge f_2(x_2,\cdots,x_r)\cdots\\
&\nabla_{k,k}(x_1)\wedge f_k(x_2,\cdots,x_r)\big),
\end{aligned} \tag{6.16}$$

where $f_i(x_2,\cdots,x_r)$ has B_i as its structure matrix, $i = 1,2,\cdots,k$.

Proof. Similar to the proof of Theorem 6.1, we only need to prove

$$f(x_1,\cdots,x_r) = \vee_{i=1}^k\left(\nabla_{i,k}\wedge f(\frac{k-i}{k-1},x_2,\cdots,x_r)\right) \tag{6.17}$$

If $x_1 = \frac{k-j}{k-1}$,

$$\begin{aligned}
RHS =& \vee_{i=1}^{j-1}\left(0\wedge f(\frac{k-j}{k-1},x_2,\cdots,x_r)\right)\vee\left(1\wedge f(\frac{k-j}{k-1},x_2,\cdots,x_r)\right)\vee\\
&\vee_{i=j+1}^{k}\left(0\wedge f(\frac{k-j}{k-1},x_2,\cdots,x_r)\right)\\
=& f(\frac{k-j}{k-1},x_2,\cdots,x_r) = LHS,
\end{aligned}$$

thus Theorem 6.2 follows. □

Denote by $\mathscr{U}_k$ the set of unary k-valued operators. That is,

$$\mathscr{U}_k = \left\{ \oslash_{i_1,\cdots,i_k} \middle| 1 \le i_j \le k; j = 1,\cdots,k \right\}.$$

Then it is easy to see the following corollary.

Corollary 6.1. *For k-valued logic, the set $\mathscr{U}_k \cup \{\vee\} \cup \{\wedge\}$ is an adequate set.*

We use the following example to depict this.

Example 6.2. Assume $k = 3$ and

$$f(x,y) = \delta_3[3\ 2\ 1\ 2\ 2\ 3\ 3\ 1\ 2]xy.$$

Find the logical expression of $f(x,y)$.
It is easy to calculate that

$$f(x,y) = (\nabla_{1,3}(x) \wedge \sigma_1(y)) \vee (\nabla_{2,3}(x) \wedge \sigma_2(y)) \vee (\nabla_{3,3}(x) \wedge \sigma_3(y)),$$

where

$$\begin{aligned} \sigma_1(y) &= \oslash_{3,2,1}(y) = \neg y; \\ \sigma_2(y) &= \oslash_{2,2,3}(y) = \oslash_{2,*,*}(y); \\ \sigma_3(y) &= \oslash_{3,1,2}(y) = \oslash^2(y) (\text{or } \oslash^{-1}(y)). \end{aligned}$$

6.2 Mix-valued Logic

Assume we have a set of logical variables $x_0, x_1, \cdots, x_n$, where

$$x_i \in \mathscr{D}_{k_i}, \quad k_i \ge 2, \quad i = 0,1,\cdots,n. \tag{6.18}$$

The mix-valued logic considers the operators over logical variables which belong to different logical regions. Precisely,

Definition 6.3. Let x_i, $i = 0,1,\cdots,n$ be as in (6.18). An n-ary mix-valued logical operator (function) f is a mapping $f : \mathscr{D}_{k_1} \times \cdots \times \mathscr{D}_{k_n} \to \mathscr{D}_{k_0}$.

First, set $k = \prod_{i=1} nk_i$, we consider unary mix-valued logical operators, which are mappings from $\mathscr{D}_k$ to $\mathscr{D}_{k_0}$. Similar to (6.15), we define

$$\oslash^{k_0}_{i_1,i_2,\cdots,i_k}(\delta_k^j) = \delta_{k_0}^{i_j}, \quad j = 1,\cdots,k. \tag{6.19}$$

Now the set

$$\mathscr{U}_k^{k_0} := \left\{ \oslash^{k_0}_{i_1,i_2,\cdots,i_k} \middle| 1 \le i_j \le k_0,\ j = 1,\cdots,k \right\}$$

is the set of all unary logical operators from $\mathscr{D}_k$ to $\mathscr{D}_{k_0}$.

Particularly, we consider the identifiers:

$$\nabla^{k_0}_{j,k}(x) := \begin{cases} \delta^1_{k_0}, & x = \delta^j_k \\ \delta^{k_0}_{k_0}, & \text{otherwise.} \end{cases} \tag{6.20}$$

In general form, we have

$$\nabla^{k_0}_{j,k} = \oslash_{\underbrace{k_0,\cdots,k_0}_{j-1},1,\underbrace{k_0,\cdots,k_0}_{k-j}}.$$

For notational compactness, when there is no possible confusion we simply denote

$$\nabla_i(\delta^j_k) := \begin{cases} \delta^1_k, & i = j \\ \delta^k_k, & mboxotherwise. \end{cases} \tag{6.21}$$

In (**??**) the operator ∇_i can be considered as a general operator from Δ_k to Δ_{k_0}, where k_0 can either the same as k or different from k.

Next, we deduce the (disjunctive) normal form for mix-valued logical functions. Assume $f(x_1,\cdots,x_n)$ is a mix-valued logical function as defined in Definition 6.3. Using truth table, it is easy to construct the structure matrix of f as

$$M_f \in \mathscr{L}_{k_0\times\prod_{j=1}^n k_j}.$$

Split M_f into k_1 equal blocks as

$$M_f = [B_1\ B_2\ \cdots\ B_{k_1}].$$

Then we have the following result, which is parallel to Theorem 6.1 for standard logic, and Theorem 6.2 for k-valued logic.

Theorem 6.3. *Let $f(x_1,\cdots,x_n)$ be the mix-valued function defined in Definition 6.3. Then $f(x_1,\cdots,x_n)$ can be expressed as*

$$\begin{aligned} f(x_1,\cdots,x_n) = {} & (\nabla_1(x_1)\wedge f_1(x_2,\cdots,x_n))\vee(\nabla_2(x_1)\wedge f_2(x_2,\cdots,x_n))\vee\cdots \\ & \vee\left(\nabla_{k_1}(x_1)\wedge f_{k_1}(x_2,\cdots,x_n)\right), \end{aligned} \tag{6.22}$$

where $f_i(x_2,\cdots,x_n)$ has B_i as its structure matrix, $i = 1,\cdots,k_1$.

Using Theorem 6.3 repetitively, we finally can get the (disjunctive) normal form of mix-valued logical operators.

Corollary 6.2. *Let $f(x_1,\cdots,x_n)$ be the mix-valued function defined in Definition 6.3. Split M_f into k/k_n equal-size blocks*

$$M_f = [B_1\ B_2\ \cdots\ B_{k/k_n}],$$

denote

$$B_j = \delta_{k_0}[i_1^j, i_2^j, \cdots, i_{k_n}^j].$$

Then $f(x_1,\cdots,x_n)$ can be expressed as

$$\begin{aligned} f(x_1,\cdots,x_n) = & \vee_{j_1=1}^{k_1} \vee_{j_2=1}^{k_2} \cdots \vee_{j_{n-1}=1}^{k_{n-1}} \Big(\nabla_{j_1}(x_1) \wedge \nabla_{j_2}(x_2) \wedge \cdots \\ & \wedge \nabla_{j_{n-1}}(x_{n-1}) \wedge \oslash^{k_0}_{i_1^J, i_2^J, \cdots, i_{k_n}^J}(x_n)\Big), \end{aligned} \tag{6.23}$$

where $J = \sum_{i=1}^{n-2} \left((j_i - 1)\prod_{p=i+1}^{n-1} k_p\right) + j_{n-1}$.

Remark 6.1. 1. (6.23) is called the (disjunctive) normal form of $f(x_1,\cdots,x_n)$. Both (6.1) (for conventional logic) and (6.16) (for k-valued logic) can be considered as its special cases.

2. The set

$$\left\{\mathscr{U}_{k_j}^{k_0}, j = 1,\cdots,n\right\} \cup \{\vee_{k_0}\} \cup \{\wedge_{k_0}\}$$

is adequate for the set of operators $\sigma : \mathscr{D}_{k_1} \times \cdots \times \mathscr{D}_{k_n} \to \mathscr{D}_{k_0}$.

Example 6.3. A mix-valued logical function $f : \mathscr{D}_2 \times \mathscr{D}_3 \times \mathscr{D}_2 \to \mathscr{D}_3$ has its structure matrix as

$$M_f = \delta_3[3\ 1\ 2\ 3\ 1\ 2\ 1\ 2\ 3\ 3\ 1\ 1].$$

Find its logical expression.

It is easy to calculate that

$$\begin{aligned} f(x_1,x_2,x_3) = & \left(\Delta_1(x_1) \wedge \Delta_1(x_2) \wedge \oslash^3_{3,1}(x_3)\right) \vee \left(\Delta_1(x_1) \wedge \Delta_2(x_2) \wedge \oslash^3_{2,3}(x_3)\right) \vee \\ & \left(\Delta_1(x_1) \wedge \Delta_3(x_2) \wedge \oslash^3_{1,2}(x_3)\right) \vee \left(\Delta_2(x_1) \wedge \Delta_1(x_2) \wedge \oslash^3_{1,2}(x_3)\right) \vee \\ & \left(\Delta_2(x_1) \wedge \Delta_2(x_2) \wedge \oslash^3_{3,3}(x_3)\right) \vee \left(\Delta_2(x_1) \wedge \Delta_3(x_2) \wedge \oslash^3_{1,1}(x_3)\right). \end{aligned}$$

6.3 General Logical Mappings

Let $x_i \in \mathscr{D}_{k_i}$, $i = 1,\cdots,n$ and $z_j \in \mathscr{D}_{s_j}$, $j = 1,\cdots,m$. Set $k = \prod_{i=1}^n k_i$ and $s = \prod_{j=1}^m s_j$. Assume a logical mapping

$$F : \prod_{i=1}^n \mathscr{D}_{k_i} \to \prod_{j=1}^m \mathscr{D}_{s_j} \tag{6.24}$$

is determined by

$$\begin{cases} z_1 = f_1(x_1,\cdots,x_n) \\ z_2 = f_2(x_1,\cdots,x_n) \\ \vdots \\ z_m = f_m(x_1,\cdots,x_n). \end{cases} \tag{6.25}$$

In vector form, we set $x = \ltimes_{i=1}^{n} x_i \in \Delta_k$ and $z = \ltimes_{j=1}^{m} z_j \in \Delta_s$, and denote by $M_j := M_{f_j} \in \mathscr{L}_{s_j \times k}$ the structure matrices of f_j. Then in vector form (6.25) becomes

$$\begin{cases} z_1 = M_1 x \\ z_2 = M_2 x \\ \vdots \\ z_m = M_m x. \end{cases} \tag{6.26}$$

We look for the structure matrices of F. We need some preparations.

Define a matrix, called the dummy matrix, as

$$D_{p,q} := \mathbf{1}_p^T \otimes I_q. \tag{6.27}$$

Then, via a straightforward computation, we have

Proposition 6.2. *Let $x \in \Delta_p$ and $y \in \Delta_q$. Then*

$$\begin{aligned} D_{p,q} xy &= y \\ D_{q,p} W_{[p,q]} xy &= x. \end{aligned} \tag{6.28}$$

Using (6.28), we can add some fabricated variables into a logical expression.

Next, we define a matrix, called the order-reducing matrix, as

$$M_r^k := \operatorname{diag}(\delta_k^1, \delta_k^2, \cdots, \delta_k^k). \tag{6.29}$$

Then it is easy to prove the following

Proposition 6.3. *Let $x \in \Delta_k$. Then*

$$x^2 = M_r^k x. \tag{6.30}$$

Using (6.30), we can reduce the power of a logical variable to 1. That is,

$$x^t = \left(M_r^k\right)^{t-1} x. \tag{6.31}$$

Now we are ready to provide the structure matrix of F.

Theorem 6.4. *Consider a mapping, F, defined by (6.24) and (6.25). There is a unique matrix $M_F \in \mathscr{L}_{s \times k}$, called the structure matrix of F, such that*

$$z = M_F x. \tag{6.32}$$

Proof. Multiplying both sides of (6.26) yields

$$\begin{aligned} z &= M_1 x M_2 x \cdots M_m x \\ &= M_1 (I_k \otimes M_2) x^2 M_3 x \cdots M_m x \\ &= M_1 (I_k \otimes M_2) \cdots (I_{k^{m-1}} \otimes M_m) x^m \\ &= M_1 (I_k \otimes M_2) \cdots (I_{k^{m-1}} \otimes M_m) \left(M_r^k\right)^{m-1} x. \end{aligned}$$

Hence

$$M_F = M_1(I_k \otimes M_2)\cdots(I_{k^{m-1}} \otimes M_m)\left(M_r^k\right)^{m-1}. \tag{6.33}$$

To see M_F is a logical matrix, it comes from the following easily verifiable claim that "the product of two logical matrices is a logical matrix".

To see M_F is unique, assume there is another such structure matrix, called M'_F, and the i-th columns of these two matrices are different. That is, $\mathrm{Col}_i(M_F) \neq \mathrm{Col}_i(M'_F)$. Choose x_i, $i = 1,\cdots,n$ such that $x = \delta_k^i$. Then we have $z = F(x)$ equals to $\mathrm{Col}_i(M_F)$ or $\mathrm{Col}_i(M'_F)$ by using M_F or M'_F respectively. This is absurd. □

In fact, Equation (6.33) may be considered as a formula to calculate the structure matrix of a mix-valued logical mapping. But the following property may provide a more convenient way for numerical calculation.

Let $x_i \in \mathscr{D}_{k_i}$, $i = 1,\cdots,n$, $y_p \in \mathscr{D}_{s_p}$, $p = 1,\cdots,m$, $z_q \in \mathscr{D}_{t_q}$, $q = 1,\cdots,r$. Set $k = \prod_{i=1}^n k_i$, $s = \prod_{p=1}^m s_p$, $t = \prod_{q=1}^r t_q$. Assume $F : \prod_{i=1}^n \mathscr{D}_{k_i} \to \prod_{p=1}^m \mathscr{D}_{s_p}$ and $G : \prod_{i=1}^n \mathscr{D}_{k_i} \to \prod_{q=1}^r \mathscr{D}_{t_q}$ have their structure matrices $M_F \in \mathscr{L}_{s\times k}$ and $M_G \in \mathscr{L}_{t\times k}$ respectively. The product mapping

$$\pi = F \times G : \prod_{i=1}^n \mathscr{D}_{k_i} \to \prod_{p=1}^m \mathscr{D}_{s_p} \prod_{q=1}^r \mathscr{D}_{t_q}$$

is defined as

$$\pi(x) = F(x) \ltimes G(x).$$

Proposition 6.4. *Assume $\pi(x) = F(x) \ltimes G(x)$ and the structure matrices of F and G are $M_F \in \mathscr{L}_{s\times k}$ and $M_G \in \mathscr{L}_{t\times k}$. Denote the structure matrix of π by $M_\pi \in \mathscr{L}_{st\times k}$. Then*

$$M_\pi = M_F * M_G, \tag{6.34}$$

where "$*$" is the Khatri-Rao product. That is

$$\mathrm{Col}_\alpha(M_\pi) = \mathrm{Col}_\alpha(M_F)\mathrm{Col}_\alpha(M_G), \quad \alpha = 1,2,\cdots,k.$$

Proof. Let $x = \delta_k^\alpha$. Then we have

$$F(x) = \mathrm{Col}_\alpha(M_F);\ G(x) = \mathrm{Col}_\alpha(M_G);\ \text{and } \pi(x) = \mathrm{Col}_\alpha(M_\pi).$$

The conclusion follows. □

Using this proposition to all the components, we have the following corollary.

Corollary 6.3. *Consider a mapping, F, defined by (6.24) and (6.25). Assume the structure matrices of f_i, $i = 1,\cdots,m$ are M_i. Then the structure matrix M_F of F can be calculated by*

$$M_F = M_1 * M_2 * \cdots * M_m. \tag{6.35}$$

Example 6.4. Assume $x_1,x_3,z_1,z_2 \in \mathscr{D}$, $x_2,z_3 \in \mathscr{D}_3$, and the mapping $F:(x_1,x_2,x_3)\mapsto(z_1,z_2,z_3)$ is decided by

$$\begin{cases} z_1 = x_1 \wedge (\oslash^2_{2,1,2} x_2) \\ z_2 = (\oslash^2_{2,1,1} x_2) \vee x_3 \\ z_3 = \oslash^3_{1,3}(x_1 \leftrightarrow x_3). \end{cases} \tag{6.36}$$

(6.36) can be converted to algebraic form

$$\begin{cases} z_1 = \delta_2[1\ 2\ 2\ 2]x_1\delta_2[2\ 1\ 2]x_2 \\ z_2 = \delta_2[1\ 1\ 1\ 2]\delta_2[2,1,1]x_2x_3 \\ z_3 = \delta_3[1,3]\delta_2[1\ 2\ 2\ 1]x_1x_3. \end{cases} \tag{6.37}$$

Consider z_1, we have

$$\begin{aligned} z_1 &= \delta_2[1\ 2\ 2\ 2]x_1\delta_2[2\ 1\ 2][I_3\ I_3]x_3x_2 \\ &= \delta_2[1\ 2\ 2\ 2]x_1\delta_2[2\ 1\ 2][I_3\ I_3]W_{[3,2]}x_2x_3 \\ &= \delta_2[1\ 2\ 2\ 2]x_1\delta_4[2\ 2\ 1\ 1\ 2\ 2\ 4\ 4\ 3\ 3\ 4\ 4]x_2x_3 \\ &= \delta_2[1\ 2\ 2\ 2]\left(I_2\otimes\delta_4[2\ 2\ 1\ 1\ 2\ 2\ 4\ 4\ 3\ 3\ 4\ 4]\right)x_1x_2x_3 \\ &= \delta_2[2\ 2\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2]x_1x_2x_3. \end{aligned}$$

Similar calculation yields

$$\begin{cases} z_1 = \delta_2[2\ 2\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2]x_1x_2x_3 \\ z_2 = \delta_2[1\ 2\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 1]x_1x_2x_3 \\ z_3 = \delta_3[1\ 3\ 1\ 3\ 1\ 3\ 3\ 1\ 3\ 1\ 3\ 1]x_1x_2x_3. \end{cases}$$

Using (6.33) or Corollary 6.3, we can get

$$M_F = \delta_{12}[7\ 12\ 1\ 3\ 7\ 9\ 9\ 10\ 9\ 7\ 9\ 7].$$

Next, we consider how to convert M_F to the logical expression of F. Assume F is defined by (6.24) with its structure matrix $M_F \in \mathscr{L}_{s\times k}$. For this, we firstly define a set of logical matrices $S_j \in \mathscr{L}_{s_j\times s}$, called the retrievers.

$$S_1 = \delta_{s_1}[\underbrace{1\ \cdots\ 1}_{s/s_1}\ \underbrace{2\ \cdots\ 2}_{s/s_1}\ \cdots\ \underbrace{s_1\ \cdots\ s_1}_{s/s_1}] \tag{6.38}$$

$$S_2 = \delta_{s_2}[\underbrace{1\ \cdots\ 1}_{s/s_1s_2}\ \cdots\ \underbrace{s_2\ \cdots\ s_2}_{s/s_1s_2}\ \cdots\ \underbrace{1\ \cdots\ 1}_{s/s_1s_2}\ \cdots\ \underbrace{s_2\ \cdots\ s_2}_{s/s_1s_2}] \tag{6.39}$$

$$\vdots \tag{6.40}$$

$$S_n = \delta_{s_n}[1\ 2\ \cdots\ s_n\ \cdots\ 1\ 2\ \cdots\ s_n]. \tag{6.41}$$

We have the following result.

Proposition 6.5. *Assume* $z_j \in \mathscr{D}_{s_j}$, $j = 1,2,\cdots,n$, *denote by* $s = \prod_{j=1}^{n} s_j$, *let* $z = \ltimes_{j=1}^{n} z_j$, *then*

$$z_j = S_j z, \quad j = 1,2,\cdots,n. \tag{6.42}$$

Proof. Since S_j has $\prod_{i=1}^{j} s_i$ equal-size blocks, if $z_j = \delta_{s_j}^{p}$, we have

$$\begin{aligned}
S_j z &= S_j z_1 \cdots z_{j-1} z_j z_{j+1} \cdots z_n \\
&= \delta_{s_j}[\underbrace{1 \cdots 1}_{s/\prod_{i=1}^{j} s_i} \ \underbrace{2 \cdots 2}_{s/\prod_{i=1}^{j} s_i} \cdots \underbrace{s_j \cdots s_j}_{s/\prod_{i=1}^{j} s_i}] z_j z_{j+1} \cdots z_n \\
&= \delta_{s_j}[\underbrace{p \cdots p}_{s/\prod_{i=1}^{j} s_i}] z_{j+1} \cdots z_n \\
&= \delta_{s_j}^{p} = z_j.
\end{aligned}$$

□

By Corollary 6.3 and Proposition 6.5, we have

Corollary 6.4. *The structure matrices* M_j *of* f_j *in (6.25) can be retrieved as follows:*

$$M_j = S_j M_F, \quad j = 1,2,\cdots,n. \tag{6.43}$$

Using Corollary 6.4 we can get the structure matrix M_i which has $x_1, x_2, \cdots, x_n$ as its variables. But in usual, some variables may not affect the value of the f_i, we call these variables the fabricated variables. For removing these variables, we have the following:

Proposition 6.6. *Consider system (6.26). For arbitrary* $1 \le j \le n$, *split* $M_i W_{[k_j, \prod_{p=1}^{j-1} k_p]}$ *into* k_j *equal-size blocks as*

$$\begin{aligned}
&M_i W_{[k_j, \prod_{p=1}^{j-1} k_p]} \\
&= \left[\mathrm{Blk}_1(M_i W_{[k_j, \prod_{p=1}^{j-1} k_p]}), \mathrm{Blk}_2(M_i W_{[k_j, \prod_{p=1}^{j-1} k_p]}), \cdots, \mathrm{Blk}_{k_j}(M_i W_{[k_j, \prod_{p=1}^{j-1} k_p]}) \right].
\end{aligned}$$

If all the blocks are the same, then x_j *is a fabricated variable. Moreover, the equation of* z_i *can be replaced by*

$$z_i = M_i' x_1 \cdots x_{j-1} x_{j+1} \cdots x_n, \tag{6.44}$$

where

$$M_i' = \mathrm{Blk}_1(M_i W_{[k_j, \prod_{p=1}^{j-1} k_p]}) = M_i W_{[k_j, \prod_{p=1}^{j-1} k_p]} \delta_{k_j}^{1}.$$

Proof. Note that

$$
\begin{aligned}
z_i &= M_i x_1 \cdots x_{j-1} x_j x_{j+1} \cdots x_n \\
&= M_i W_{[k_j, \prod_{p=1}^{j-1} k_p]} x_j x_1 \cdots x_{j-1} x_{j+1} \cdots x_n,
\end{aligned}
$$

if x_j does not affect z_i, then z_i is invariant whatever the value of x_j is. Then we can simply set $x_j = \delta_{k_j}^1$ to simplify the expression. □

We give an example to depict this.

Example 6.5. Assume the structure matrix of a logical mapping with $x_1, x_3, z_1, z_2 \in \mathscr{D}$, and $x_2, z_3 \in \mathscr{D}_3$ is

$$M_F = \delta_{12}[7\ 12\ 1\ 3\ 7\ 9\ 9\ 10\ 9\ 7\ 9\ 7].$$

Then using Corollary 6.4, we have

$$
\begin{aligned}
M_1 = S_1 M_F &= \delta_2[2\ 2\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2] \\
M_2 = S_2 M_F &= \delta_2[1\ 2\ 1\ 1\ 1\ 1\ 1\ 2\ 1\ 1\ 1\ 1] \\
M_3 = S_3 M_F &= \delta_3[1\ 3\ 1\ 3\ 1\ 3\ 3\ 1\ 3\ 1\ 3\ 1].
\end{aligned}
$$

Net, consider M_1, it is easy to verify that

$$
\begin{aligned}
M_1 &= \delta_2[2\ 2\ 1\ 1\ 2\ 2 \mid 2\ 2\ 2\ 2\ 2\ 2] \\
M_1 W_{[2]} &= \delta_2[2\ 2\ 2\ 2 \mid 1\ 1\ 2\ 2 \mid 2\ 2\ 2\ 2] \\
M_1 W_{[3,4]} &= \delta_2[2\ 1\ 2\ 2\ 2\ 2 \mid 2\ 1\ 2\ 2\ 2\ 2].
\end{aligned}
$$

We conclude that z_1 depends on x_1 and x_2 only. Then z_1 can be simplified as

$$z_1 = \delta_2[2\ 1\ 2\ 2\ 2\ 2] x_1 x_2.$$

Similarly, we can remove the dummy variables from other equations. We have

$$
\begin{cases}
z_1 = \delta_2[2\ 1\ 2\ 2\ 2\ 2] x_1 x_2 \\
z_2 = \delta_2[1\ 2\ 1\ 1\ 1\ 1] x_2 x_3 \\
z_3 = \delta_3[1\ 3\ 3\ 1] x_1 x_3.
\end{cases}
$$

which is same to (6.37).

Using Theorem 6.3 we finally have

$$
\begin{cases}
z_1 = x_1 \wedge (\oslash_{2,1,2}^2 x_2) \\
z_2 = \Delta_{1,3}^2(x_2) \wedge x_3 \vee \Delta_{2,3}^2(x_2) \vee \Delta_{3,3}^2(x_2) = (\oslash_{2,1,1}^2 x_2) \vee x_3 \\
z_3 = \Big(\Delta_{1,2}^3(x_1) \wedge \oslash_{1,3}^3(x_3)\Big) \vee \Big(\Delta_{2,2}^3(x_1) \wedge \oslash_{3,1}^3(x_3)\Big) = \oslash_{1,3}^3(x_1 \leftrightarrow x_3).
\end{cases}
$$

6.4 Some Applications

6.4.1 Rules in Fuzzy Control

As one of the most successful intelligent control technologies, fuzzy control has attracted much attention from control community, and it has been used widely in industries. It is a suitable tool for a variety of challenging control engineering problems. As pointed out in [3] "In the fuzzy control design methodology, we ask this operator to write down a set of rules on how to control the process, then we incorporate these into a fuzzy controller that emulates the decision-making process of the human."

Rules play a key role in Fuzzy control. We describe this through the following example.

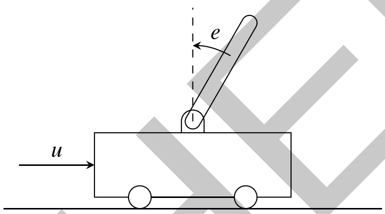

Fig. 6.1 An inverted pendulum

Example 6.6. Fig. 6.1 depicts an inverted pendulum. Denote by e the error, which is the angle leaving from the vertical position (with left side as positive value), $\dot{e}$ is the time derivative of e.

Quantizing the error, change-in-error and the force (control) into 5 levels as: positive-large (denoted by 2), positive-small (denoted by 1), zero (denoted by 0), negative-small (denoted by -1), and negative-large (denoted by -2). The control rules are presented as expert's linguistic description of how to perform the control. Say,

- **If** error is zero and change-in-error is zero **Then** force is zero
- **If** error is zero and change-in-error is positive-small **Then** force is negative small
- $\cdots$

Then the linguistic statements form a set of control rules, which are listed in Table 6.1[3].

Simply identifying $-2 \sim \delta_5^1$, $-1 \sim \delta_5^2$, $0 \sim \delta_5^3$, $1 \sim \delta_5^4$, and $2 \sim \delta_5^5$, we can see that $u(e,\dot{e}) : \mathscr{D}_5 \times \mathscr{D}_5 \to \mathscr{D}_5$ is a logical function. Its algebraic form is

$$u = M_u e\dot{e}, \tag{6.45}$$

where the structure matrix of u can be easily calculated as

Table 6.1 Rule Table for the Inverted Pendulum

$e\backslash u\backslash \dot{e}$	-2	-1	0	1	2
-2	2	2	2	1	0
-1	2	2	1	0	-1
0	2	1	0	-1	-2
1	1	0	-1	-2	-2
2	0	-1	-2	-2	-2

$$M_u = \delta_5[5\ 5\ 5\ 4\ 3\ 5\ 5\ 4\ 3\ 2\ 5\ 4\ 3\ 2\ 1\ 4\ 3\ 2\ 1\ 1\ 3\ 2\ 1\ 1\ 1]. \tag{6.46}$$

In general, for a control system the controls may depend on several variables which can take different numbers of discrete numbers, hence as u considered as logical functions, they are in general mix-valued logical functions. But to the authors' surprisal, most application examples in reference books the control functions are k-valued ones. A possible reason for this is the k-valued logical functions have corresponding circuit realization.

We consider another example.

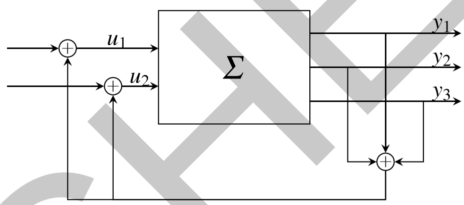

Fig. 6.2 Output-feedback control system

Example 6.7. Given a control system Σ as depicted in Fig. 6.2, which has two inputs u_1, u_2, and three outputs y_i, $i = 1,2,3$. Assume the controls can take values as:

$$u_1 : 1,\ 0,\ -1;$$
$$u_2 : 2,\ 1,\ 0,\ -1,\ -2;$$

and the outputs are classified as

y_1 : high, low;
y_2 : high, middle, low;
y_3 : very high, high, middle, low, very low;

We simply let $u_1 \in \mathscr{D}_3$ and $u_2 \in \mathscr{D}_5$. Similarly, we have $y_1 \in \mathscr{D}_2$, $y_2 \in \mathscr{D}_3$, and $y_3 \in \mathscr{D}_5$ Consider the output feedback controls, which means the controls depend on outputs. Then the control becomes a mapping

$$\pi : \mathscr{D} \times \mathscr{D}_3 \times \mathscr{D}_5 \to \mathscr{D}_3 \times \mathscr{D}_5.$$

We need 30 statements to describe the control rules as, for instance,:

"If y_1 is hight and y_2 is hight and y_3 is very high, then $u_1 = 1$, and $u_2 = 2$."

Denote by $u = u_1 \ltimes u_2$ and $y = \ltimes_{i=1}^{3} y_i$. Then u can be described as a mix-valued logical mapping, determined by its structure matrix M_v as

$$u = M_u y, \tag{6.47}$$

where $M_u \in \mathscr{L}_{15\times 30}$. As M_u is given, the mix-valued logical expressions for u_i, $i = 1,2$ can also be constructed. For example, assume

$$M_u = \delta_{15}[6\ 7\ 8\ 9\ 9\ 6\ 6\ 6\ 6\ 6\ 11\ 12\ 12\ 12\ 12\ 1\ 2\ 3\ 4\ 4\ 6\ 6\ 6\ 6\ 6\ 11\ 12\ 12\ 12\ 12].$$

Then the logical expressions can be obtained as

$$\begin{cases} u_1 = \left(\oslash^3_{2,1} y_1\right) \wedge y_2 \\ u_2 = \left(\oslash^5_{4,1,2} y_2\right) \vee y_3). \end{cases}$$

6.4.2 *Strategy of Dynamic Games*

As another applications of mix-valued logic, we consider the strategies of infinitely repeated game, which is a kind of dynamic games. This problem will be discussed in detail later, here we give an example to depict it.

Example 6.8. In a game assume there are two players: P_1 and P_2. P_1 has 2 possible actions $S_1 = \{\alpha_1, \alpha_2\}$, and P_2. P_2 has 3 possible actions $S_2 = \{\beta_1, \beta_2, \beta_3\}$. Assume the game is infinitely repeated, and the strategies of each players at time $t+1$ depend on the strategies of the players at time t. Denote by $x(t)$ and $y(t)$ the strategies of P_1 and P_2 at time t respectively, then we have

$$\begin{cases} x(t+1) = f_1(x(t), y(t)) \\ y(t+1) = f_2(x(t), y(t)). \end{cases} \tag{6.48}$$

To use vector expression, we identify

$$\begin{aligned} &\alpha_1 \sim \delta_2^1, \quad \alpha_2 \sim \delta_2^2; \\ &\beta_1 \sim \delta_3^1, \quad \beta_2 \sim \delta_3^2, \quad \beta_3 = \delta_3^3. \end{aligned}$$

Then $x(t) \in \Delta_2$ and $y(t) \in \Delta_3$, and we can find the structure matrices $M_1 \in \mathscr{L}_{2\times 6}$ and $M_2 \in \mathscr{L}_{3\times 6}$ of f_1 and f_2 respectively, such that (6.48) can be expressed as

$$\begin{cases} x(t+1) = M_1 x(t) \ltimes y(t) \\ y(t+1) = M_2 x(t) \ltimes y(t). \end{cases} \tag{6.49}$$

Furthermore, setting $z(t) = x(t) \ltimes y(t)$, we have

$$z(t+1) = Lz(t), \tag{6.50}$$

where $L = M_1 * M_2 \in \mathscr{L}_{6\times 6}$.

To a numerical expression, we assume

$$L = \delta_6[1\ 3\ 5\ 2\ 4\ 6].$$

Then we have

$$M_1 = ?M_2 = ?$$

and

$$f_1 = ?f_2 = ?$$

(Please complete this example.)

Exercise 6

1. Use $\{\neg, \wedge, \vee\}$ to express the 16 binary operators.
2. Use Proposition 6.1 to prove that $\{\neg, \rightarrow\}$ is an adequate set.
3. A mix-valued logical function $f : \mathscr{D}_2 \times \mathscr{D}_3 \times \mathscr{D}_2 \to \mathscr{D}_2$ is defined as

$$f(x_1, x_2, x_3) = \left[x_1 \wedge \oslash^2_{121}(x_2)\right] \leftrightarrow x_3.$$

   Calculate the structure matrix of f.
4. A mix-valued logical function $f : \mathscr{D}_2 \times \mathscr{D}_3 \to \mathscr{D}_3$ has the structure matrix as

$$M_f = \delta_3[1\ 3\ 2\ 2\ 2\ 1].$$

   Find its logical expression.
5. A mix-valued logical mapping F is defined by

$$\begin{cases} z_1 = f_1(x_1, x_2, x_3) \\ z_2 = f_2(x_1, x_2, x_3), \end{cases}$$

   where $x_1, x_3, z_1 \in \mathscr{D}$ and $x_2, z_2 \in \mathscr{D}_3$. Let $z = z_1 z_2$ and $x = x_1 x_2 x_3$. Then the structure matrix of F is

$$M_F = \delta_6[1\ 3\ 5\ 2\ 4\ 6\ 2\ 4\ 6\ 1\ 3\ 5].$$

   Find the logical expressions of f_1 and f_2.
6. Prove Proposition 6.2.
7. Prove Proposition 6.4.

References

1. Cheng, D., Qi, H., Li, Z.: Analysis and Control of Boolean Networks: A Semi-tensor Product Approach. Springer, London (2011)
2. Hamilton, A.: Logic for Mathematicians, revised edn. Cambridge Univ Press, Cambridge (1988)
3. Passino, K., Yurkovich, S.: Fuzzy Control. Addison Wesley Longman (1998)