Matrix Approach to Boolean Calculus

Daizhan Cheng, Yin Zhao, Xiangru Xu

***Abstract*— Using semi-tensor product of matrices and the matrix expression of logic, formulas for calculating Boolean derivatives are obtained. Using this form, the solvability of Boolean algebraic equations and Boolean differential equations is considered. Its application to fault detection of combinational circuits is investigated. Then we define the Boolean integrals as the inverse of the Boolean derivative in certain sense. Three kinds of integrals are proposed. The inverse of a partial derivative with respect to x_i is called the ith primitive function. The inverse of a differential form is called the indefinite integral. A necessary and sufficient condition for the existence of the indefinite integral is proved. Using the unique indefinite integral (up to complement equivalence), definite integral is also defined. Simply computable formulas are provided for solving each kind of integrals.**

I. Introduction

Right after G. Boole invented an algebra in 1847, which is lately called the Boolean algebra, an effort rose, which attempts to establish Boolean analogues of concepts and results from Calculus. The first version of Boolean Differential Calculus was proposed by Daniell in 1917 [14]. Some forty years later after Shannon proposed the switching algebra in the evaluation of switching circuit designing, it was discovered that the partial derivatives of Boolean functions are particularly useful in switching theory [23], [3]. Since then, the Boolean derivative has been developed quickly, both in view of applications and for its own algebraic interest [22], [5], [34], [27], [30], [15].

There are several definitions on Boolean derivative, we adopt the common definition of Boolean derivative, which can be found, for instance, in [33], [29]. A general definition and basic properties and some applications can be found in [29]. The fundamental requirements and satisfactory of Boolean derivatives are discussed in[25].

Many applications of Boolean derivatives have been reported. The applications include control of Boolean networks [20], synthesis of discrete event systems [26], logical circuit analysis [5], [19], asynchronous circuit design [28], image edge detection [2], selection probabilities of stack filters [16], cellular automata and finite state machine [21], [32], etc. These evidence that Boolean derivative is a useful tool.

Recently, a new matrix product, called the semi-tensor product, has been proposed and it has been successfully applied to the analysis and control of Boolean networks [13]. We also refer to [6], [7], [8], [9], [10], [11], [12] for its applications to the topological structure, controlability, observability, stabilization, disturbance decoupling, etc. of Boolean (control) networks.

The key point of this approach is to convert a logical expression into an algebraic form, and then the known methods for analyzing conventional static and/or dynamic systems are applicable to logical (dynamic) systems. This new technique is the crucial tool for the investigation in this paper.

The first interesting topic is the calculation of Boolean derivatives [35], [18]. Using semi-tensor product, [19] attempts to provide the general formulas for calculating the Boolean derivatives. Following [19], we provide a formula for calculating the structure matrix of a Boolean derivative. It is essentially equivalent to the fast implementation presented by [1], which is coined the jth partial derivative transform, using matrix multiplication [2]. But our vector form is convenient in later use.

Then the Boolean algebraic and differential equations are investigated. Algorithms are provided to solve the equations. As an application, the fault detection of combinational circuits is investigated.

Another interesting topic is the counterpart of the Boolean derivative, that is, the the Boolean integral. There are much less literatures on Boolean integral. [31] provides some interesting insights for Boolean integral. But it seems to the author that there is no convergent definition yet.

Using the formula for calculating Boolean derivatives, we formulate the Boolean integral, as the inverse of the Boolean derivative. Three kinds of the inverses of Boolean derivatives are defined. The inverse of a partial derivative, is called the primitive function; the inverse of a differential form, called the indefinite integral. Using the derivative algorithm, the indefinite integral can be calculated easily. The uniqueness of the indefinite integral (up to a complement equivalence) is proved. Using this uniqueness, the evaluation of the indefinite integral with an integral function over a given reagin, which is called the definite integral, is defined. Easily computable formulas are also provided for each kind of integrals.

The rest of this paper is organized as follows: Section 2 consists of some preliminaries which provide fundamental tools for later investigation, including the semi-tensor product of matrices, the matrix expression of logical equations, and the Boolean product of Boolean matrices. Section 3 discusses the calculation of Boolean derivatives. Some easily computable formulas are developed. Section 4 is devoted to solving Boolean algebraic equations and Boolean differential equations. After providing general algorithms for solving them, the application to fault detection of combinational

This work was Supported partly by NNSF 61074112, 60736022, and 60821091 of China.

Daizhan Cheng, Yin Zhao and Xiangru Xu are with Key Lab. of Systems and Control, AMSS, Chinese Academy of Sciences, Beijing 100190, P.R.China. E-mail: dcheng@iss.ac.cn, zhaoyin@amss.ac.cn, xuxiangru@amss.ac.cn

circuits is investigated. The Boolean integral is proposed in Section 5. Three kinds of Boolean integrals are built and the related formulas are also provided to calculate them. Some examples are presented to illustrate the concepts and algorithms. Section 6 is a brief conclusion.

II. Preliminaries

First, we introduce some notations.

- $\mathcal{M}_{m\times n}$: the set of $m\times n$ real matrices.
- $\mathbf{1}_{m\times n}$ ($\mathbf{0}_{m\times n}$): a matrix in $\mathcal{M}_{m\times n}$ with all entries equal 1 (correspondingly, 0).
  If no ambiguity is possible, we simply use $\mathbf{1}_n$ for $\mathbf{1}_{n\times n}$ $\mathbf{0}$ for $\mathbf{0}_n$, or $\mathbf{0}_n^T$, or $\mathbf{0}_{m\times n}$.
- $\mathcal{D}=\{1,0\}$.
- δ_n^k is the k-th column of the identity matrix I_n.
- $\Delta_n:=\{\delta_n^1,\cdots,\delta_n^n\}$. For compactness, $\Delta:=\Delta_2$.
- $\mathrm{Col}_i(A)$ ($\mathrm{Row}_i(A)$) is the i-th column (i-th row) of a matrix A, the set of all the columns (rows) of A is denoted by $\mathrm{Col}(A)$ ($\mathrm{Row}(A)$).
- A matrix $L\in M_{n\times m}$ is called a logical matrix if its columns, $\mathrm{Col}(M)\subset\Delta_n$.
  The set of $n\times m$ logical matrices is denoted by $\mathcal{L}_{n\times m}$.
- Let $L\in\mathcal{L}_{n\times m}$. Then
  $$L=[\delta_n^{i_1},\delta_n^{i_2},\cdots,\delta_n^{i_m}].$$
  For the sake of briefness, it is denoted as
  $$L=\delta_n[i_1,i_2,\cdots,i_m].$$
- A matrix $A=(a_{i,j})\in\mathcal{M}_{m\times n}$ is called a Boolean matrix if its entries $a_{i,j}\in\mathcal{D}$. The set of $m\times n$ Boolean matrices is denoted by $\mathcal{B}_{m\times n}$.
- Let $A=(a_{i,j}),B=(b_{i,j})\in\mathcal{B}_{m\times n}$. Then $\neg A=(\neg a_{i,j})$; and $A\wedge B=(a_{i,j}\wedge b_{i,j})$, etc.
- A swap matrix $W_{[n,m]}\in\mathcal{M}_{mn\times mn}$ is designed to swap two vector factors in their "product". We refer to [9] or [13] for its definition and properties, and to the following (3) for its basic function.

Definition 2.1: [9], [13] Let $M\in\mathcal{M}_{m\times n}$ and $N\in\mathcal{M}_{p\times q}$. The semi-tensor product of matrices, denoted by $M\ltimes N$, is defined as

$$M\ltimes N:=\left(M\otimes I_{s/n}\right)\left(N\otimes I_{s/p}\right),\qquad(1)$$

where $s=\mathrm{lcm}\{n,p\}$ is the least common multiple of n and p, $\otimes$ is the Kronecher product of matrices.

Remark 2.2: Throughout this paper, unless else product symbol is used, the matrix product is assumed to be semi-tensor product, which contains the conventional matrix product as its particular case when $n=p$. Hence, the symbol $\ltimes$ can be omitted. We do this in the sequel. Since all the product properties of the conventional matrix product remain correct, we can perform the semi-tensor product as conventional product without worrying about the dimensions.

Definition 2.3: A k-ary logical function (or operator) is a mapping $f:\mathcal{D}^k\to\mathcal{D}$. It is commonly expressed as $f(x_1,\cdots,x_k)$, where $x_i\in\mathcal{D}$, $i=1,\cdots,k$.

To use the matrix expression of logic, we identify $1\sim\delta_2^1$ and $0\sim\delta_2^2$. Under this vector form, a logical function $f:\mathcal{D}^k\to\mathcal{D}$ becomes a function $f:\Delta^k\to\Delta$, (or equivalently, $f:\Delta_{2^k}\to\Delta$).

TABLE I
Structure Matrices of Operators

Operator	Structure Matrix
$\neg$	$M_\neg=\delta_2[2\ 1]$
$\wedge$	$M_\wedge=\delta_2[1\ 2\ 2\ 2]$
$\vee$	$M_\vee=\delta_2[1\ 1\ 1\ 2]$
$\to$	$M_\to=\delta_2[1\ 2\ 1\ 1]$
$\leftrightarrow$	$M_\leftrightarrow=\delta_2[1\ 2\ 2\ 1]$
$\bar\vee$(or $\oplus$)	$M_\oplus=\delta_2[2\ 1\ 1\ 2]$

Theorem 2.4: [13] Let $f:\mathcal{D}^k\to\mathcal{D}$. Then there exists a unique logical matrix $M_f\in\mathcal{L}_{2\times 2^k}$, called the structure matrix of f, such that in vector form we have

$$f(x_1,\cdots,x_k)=M_f x,\qquad(2)$$

where $x=\ltimes_{i=1}^k x_i\in\Delta_{2^k}$.

For convenience, we give the structure matrices of some commonly used logical operators in Table I.

Remark 2.5: Let $M_f\in\mathcal{L}_{2\times 2^k}$ be the structure matrix of $f:\mathcal{D}^k\to\mathcal{D}$. Then $\mathrm{Row}_1(M_f)$ is the truth table of f (in row form) and $\mathrm{Row}_2(M_f)=\neg\,\mathrm{Row}_1(M_f)$. We simply denote

$$m_f:=\mathrm{Row}_1^T(M_f).$$

Finally, we need a lemma, which is useful in the sequel.

Lemma 2.6: [13]

1) Let $x\in\mathbb{R}^m$ and $y\in\mathbb{R}^n$ be two column vectors. Then
$$W_{[m,n]}xy=yx.\qquad(3)$$
2) Let $x\in\mathbb{R}^t$ and A is a given matrix. Then
$$xA=(I_t\otimes A)x.\qquad(4)$$
3) Let $x=\ltimes_{i=1}^k x_i$. Then
$$x^2=M_r^{2^k}x,\qquad(5)$$
where
$$M_r^n:=\mathrm{diag}[\delta_n^1\ \delta_n^2\ \cdots\ \delta_n^n].$$

A Boolean algebra on $\mathcal{D}$ is a quadruple $(\mathcal{D},+_\mathcal{B},\times_\mathcal{B},\neg)$. We refer to [24] for an elementary definition. For our purpose, we only consider Galois algebra in which $+_\mathcal{B}=\oplus$ and $\times_\mathcal{B}=\wedge$. We firstly define the Boolean product of matrices under such algebra.

Definition 2.7: For a Boolean algebra $\mathcal{B}=(\mathcal{D},\oplus,\wedge,\neg)$, we define

1) The Boolean product of $A=(a_{i,j})\in\mathcal{B}_{m\times n}$ and $B=(b_{i,j})\in\mathcal{B}_{n\times s}$ is defined as
$$A\ltimes_\mathcal{B}B:=(c_{i,j})\in\mathcal{B}_{m\times s},\qquad(6)$$
where
$$c_{i,j}=a_{i,1}\wedge b_{1,j}\oplus a_{i,2}\wedge b_{2,j}\oplus\cdots\oplus a_{i,n}\wedge b_{n,j}.$$
2) The Boolean semi-tensor product of $A=(a_{i,j})\in\mathcal{B}_{m\times n}$ and $B=(b_{i,j})\in\mathcal{B}_{p\times q}$ is defined as
$$A\ltimes_\mathcal{B}B=\left(A\otimes I_{s/n}\right)\ltimes_\mathcal{B}\left(A\otimes I_{s/p}\right),\qquad(7)$$
where $s=\mathrm{lcm}(n,p)$.

Note that (6) is a particular case of (7).

The following properties are immediate consequence of the definition.

Proposition 2.8: Assume $A, B \in \mathcal{B}_{m\times n}$, and C, D are of arbitrary dimensions. Then

1) (Commutative Law)
$$A \oplus B = B \oplus A. \tag{8}$$
2) (Associative Law)
$$A \ltimes_{\mathcal{B}} (C \ltimes_{\mathcal{B}} D) = (A \ltimes_{\mathcal{B}} C) \ltimes_{\mathcal{B}} D. \tag{9}$$
3) (Distributive Law)
$$(A \oplus B) \ltimes_{\mathcal{B}} C = (A \ltimes_{\mathcal{B}} C) \oplus (B \ltimes_{B} C). \tag{10}$$
4)
$$[A \ltimes_{\mathcal{B}} B]^T = B^T \ltimes_{\mathcal{B}} A^T. \tag{11}$$

We give a simple example to depict it.

Example 2.9: Let
$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}.$$
$$A \oplus B = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad A \ltimes_{\mathcal{B}} C = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix},$$
$$\begin{aligned} C^T \ltimes_{\mathcal{B}} A^T &= \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \ltimes_{\mathcal{B}} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = (A \ltimes_{\mathcal{B}} C)^T. \end{aligned}$$

III. Boolean Derivatives

The Boolean derivative in this paper is defined as [4]

Definition 3.1: Let $f(x_1, \cdots, x_n) : \mathcal{D}^n \to \mathcal{D}$ be a logical function.

1) The Boolean derivative of f with respect to x_i is defined as
$$\frac{\partial f}{\partial x_i} = f(x_1, \cdots, x_i, \cdots, x_n) \oplus f(x_1, \cdots, \neg x_i, \cdots, x_n). \tag{12}$$
2) The higher order derivative of f with respect to x_{i_1}, $\cdots$, x_{i_k} is defined recursively as
$$\frac{\partial^k f}{\partial x_{i_1} \cdots \partial x_{i_k}} = \frac{\partial}{\partial x_{i_1}} \left(\frac{\partial}{\partial x_{i_2}} \left(\cdots \left(\frac{\partial f}{\partial x_{i_k}} \right) \right) \right). \tag{13}$$

We cite some basic properties in the following.

Proposition 3.2: [33]

1) $\frac{\partial f}{\partial x_i}$ is independent of x_i, and hence
$$\frac{\partial^2 f}{\partial^2 x_i} = 0.$$
2)
$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i}. \tag{14}$$
3)
$$\frac{\partial (f_1 \oplus f_2)}{\partial x_i} = \frac{\partial f_1}{\partial x_i} \oplus \frac{\partial f_2}{\partial x_i}. \tag{15}$$
4)
$$\frac{\partial (f_1 f_2)}{\partial x_i} = \frac{\partial f_1}{\partial x_i} f_2 \oplus f_1 \frac{\partial f_2}{\partial x_i} \oplus \frac{\partial f_1}{\partial x_i} \frac{\partial f_2}{\partial x_i}. \tag{16}$$
5) Denote $\bar{f} := \neg f$, and $\bar{x} := \neg x$, then
$$\frac{\partial \bar{f}}{\partial x_i} = \frac{\partial f}{\partial x_i}, \quad \text{and} \quad \frac{\partial f}{\partial \bar{x}_i} = \frac{\partial f}{\partial x_i}. \tag{17}$$

Using the vector form of logical expression as introduced in Section 2, we can easily obtain the matrix expression of $\frac{\partial}{\partial x_i}$, denoted by $M_{\partial_i f}$. Denote by M_f the structure matrix of f and $x := \ltimes_{i=1}^n x_i$. Using (12), we have
$$\frac{\partial f}{\partial x_i} = M_{\partial_i f} x = M_f x \oplus M_f x_1 \cdots \bar{x}_i \cdots x_n. \tag{18}$$
Then using Lemma 2.6 to simplify the right hand side of (18), it is easy to have that [19]
$$M_{\partial_i f} = M_{\oplus} M_f \left[I_{2^n} \otimes M_f \left(I_{2^{i-1}} \otimes M_{\neg} \right) \right] M_r^{2^n}. \tag{19}$$
Then the higher order derivatives can also be calculated recursively [19].

In the following we shall give an explicit form of the structure matrices of the derivatives. Consider the structure matrix of $g(x_1, \cdots, x_n) := f(x_1, \cdots, \bar{x}_i, \cdots, x_n)$. Assume the structure matrices of f and g are M_f and M_g respectively. Using (4), we have
$$\begin{aligned} M_g x &= M_f x_1 \cdots M_{\neg} x_i \cdots x_n \\ &= M_f \left(I_{2^{i-1}} \otimes M_{\neg} \right) x. \end{aligned}$$
That is,
$$M_g = M_f \left(I_{2^{i-1}} \otimes M_{\neg} \right). \tag{20}$$
The following proposition is obvious.

Proposition 3.3: Assume $f(x_1, \cdots, x_n)$ and $g(x_1, \cdots, x_n)$ have their truth tables as $m_f, m_g \in \mathcal{B}_{2^n}$ respectively, and σ is a binary logical operator. Then
$$m_{f \sigma g} = m_f \sigma m_g. \tag{21}$$
Using Remark 2.5 and Proposition 3.3, we have
$$m_{\partial_i f}^T = m_f^T \oplus m_f^T (I_{2^{i-1}} \otimes M_{\neg}). \tag{22}$$

Using the distributive law (10), we can calculate that
$$\begin{aligned} m_{\partial_i f}^T &= m_f^T \oplus m_f^T (I_{2^{i-1}} \otimes M_{\neg}) \\ &= m_f^T \ltimes_{\mathcal{B}} I_{2^i} \oplus m_f^T \ltimes_{\oplus} (I_{2^{i-1}} \otimes M_{\neg}) \\ &= m_f^T \ltimes_{\mathcal{B}} \left(I_{2^i} \oplus (I_{2^{i-1}} \otimes M_{\neg}) \right) \\ &= m_f^T \ltimes_{\mathcal{B}} \left(I_{2^{i-1}} \otimes (I_2 \oplus M_{\neg}) \right) \\ &= m_f^T \ltimes_{\mathcal{B}} \left(I_{2^{i-1}} \otimes \mathbf{1}_{2\times 2} \right). \end{aligned}$$

We conclude that

Theorem 3.4: Let $f(x_1,\cdots,x_n)$ be a Boolean function with structure matrix M_f. Then The structure matrix of $\frac{\partial f}{\partial x_i}$, denoted by $M_{\partial_i f}$, is

$$M_{\partial_i f}=\begin{bmatrix}\mathrm{Row}_1(M_f)\ltimes_{\mathcal{B}}\varXi_n^i\\ \neg\,\mathrm{Row}_1(M_f)\ltimes_{\mathcal{B}}\varXi_n^i\end{bmatrix} \tag{23}$$

where

$$\varXi_n^i=I_{2^{i-1}}\otimes \mathbf{1}_{2\times 2}.$$

Hence, in vector form,

$$\frac{\partial f}{\partial x_i}=M_{\partial_i f}x, \tag{24}$$

where $x=\ltimes_{i=1}^n x_i$. Moreover,

$$m_{\partial_i f}=\left[\varXi_n^i\right]^T m_f. \tag{25}$$

As we know that $\frac{\partial f}{\partial x_i}$ is independent of x_i, so one may be interested in an alternative expression as

$$\frac{\partial f}{\partial x_i}=M_{\partial_{[i]}f}x_1\cdots x_{i-1}\hat{x}_i x_{i+1}\cdots x_n, \tag{26}$$

where notation "$\hat{x}_i$" means x_i is omitted.

To calculate $M_{\partial_{[i]}f}$, we dividing $M_{\partial_i f}$ into 2^i equal blocks as

$$M_{\partial_i f}=[C_1\ C_2\ \cdots\ C_{2^i}].$$

One sees easily that to get $M_{\partial_{[i]}f}$ from $M_{\partial_i f}$, we need only to pick out all odd (or even) blocks. It can be done by right-multiply

$$\left(I_{2^{i-1}}\otimes\begin{bmatrix}I_{2^{n-i}}\\ \mathbf{0}_{2^{n-i}\times 2^{n-i}}\end{bmatrix}\right).$$

That is,

$$M_{\partial_{[i]}f}=\begin{bmatrix}\mathrm{Row}_1(M_f)\ltimes_{\mathcal{B}}[\Psi_n^i]^T\\ \neg\,\mathrm{Row}_1(M_f)\ltimes_{\mathcal{B}}[\Psi_n^i]^T\end{bmatrix} \tag{27}$$

where

$$\begin{aligned}\Psi_n^i=&\left(I_{2^{i-1}}\otimes\left[I_{2^{n-i}}\quad \mathbf{0}_{2^{n-i}\times 2^{n-i}}\right]\right)\ltimes_{\mathcal{B}}\left(I_{2^{i-1}}\otimes \mathbf{1}_{2\times 2}\right)\\ =&I_{i-1}\otimes \mathbf{1}_2^T\otimes I_{n-i}.\end{aligned}$$

Note that the transpose of the first row $\mathrm{Row}_1^T(M_f)$ is the truth table of f. We have the following

Corollary 3.5: Assume the truth table of a logical function $f(x_1,\cdots,x_n)$ is m_f. Then the truth table of $\frac{\partial f}{\partial x_i}$, in condensed form, is

$$m_{\partial_{[i]}f}=\Psi_n^i m_f. \tag{28}$$

Corollary 3.5 coincides with the result in [1], [2].

The following Corollaries 3.6 and 3.7, which are convenient in numerical computation, are obvious.

Corollary 3.6: Divide m_f^T into 2^i blocks

$$m_f^T=(c_{1,1}\ c_{1,2}\ c_{2,1}\ c_{2,2}\ \cdots\ c_{2^{i-1},1}\ c_{2^{i-1},2}).$$

Then $m^T_{\partial_{[i]}f}$ can be calculated directly by

$$m^T_{\partial_{[i]}f}=(c_{1,1}\oplus c_{1,2}\ c_{2,1}\oplus c_{2,2}\ \cdots\ c_{2^{i-1},1}\oplus c_{2^{i-1},2}). \tag{29}$$

Corollary 3.7: The truth table of $\frac{\partial^k f}{\partial x_{i_1}\cdots\partial x_{i_k}}$ is (assume $i_1>i_2>\cdots>i_k$):

$$m_{\partial_{[i_k,\cdots,i_1]}f}=\Psi_{n-k+1}^{i_k}\Psi_{n-k+2}^{i_{k-1}}\cdots\Psi_n^{i_1}m_f. \tag{30}$$

Note that in (30) we require $i_1>i_2>\cdots>i_k$ because otherwise, the later positions need to be adjusted. For instance, say, $i_1<i_2$, then after differentiate with respect to x_{i_1} the position for i_2 becomes i_2-1. So we need this order. Because of (14), we can assume this without loss of generality.

We give an example to show how to calculate the derivatives. To this end, we introduce the MacLaurin expansion of a Boolean function.

Theorem 3.8: [3] A Boolean function $f(x_1,\cdots,x_n)$ has its MacLaurin expansion as

$$\begin{aligned}f(x_1,\cdots,x_n)=&f(\mathbf{0})\oplus\bigoplus_{i=1}^n\left.\frac{\partial f}{\partial x_i}\right|_{\mathbf{0}}\wedge x_i\\ &\oplus\bigoplus_{1\le i_1<i_2\le n}\left.\frac{\partial^2 f}{\partial x_{i_1}\partial x_{i_2}}\right|_{\mathbf{0}}\wedge x_{i_1}\wedge x_{i_2}\oplus\cdots\\ &\oplus\left.\frac{\partial^n f}{\partial x_1\partial x_2\cdots\partial x_n}\right|_{\mathbf{0}}\wedge x_1\wedge\cdots\wedge x_n.\end{aligned} \tag{31}$$

Example 3.9: Assume $f(x_1,x_2,x_3,x_4)=(x_1\bar{\vee}x_2)\to(x_3\vee x_4)$. Its truth table is*

$$m_f=[1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1]^T.$$

Using (28),

$$\Psi_4^1=\begin{bmatrix}I_4 & I_4\end{bmatrix},$$

the truth table of $\frac{\partial f}{\partial x_1}$ is

$$m_{\partial_{[1]}f}=\Psi_4^1\ltimes_{\mathcal{B}}m_f=[0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T.$$

Similarly,

$$\Psi_4^2=\begin{bmatrix}I_3 & I_3 & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & I_3 & I_3\end{bmatrix}.$$

$$m_{\partial_{[2]}f}=\Psi_4^2\ltimes_{\mathcal{B}}m_f=[0\ 0\ 0\ 1\ 0\ 0\ 0\ 1]^T.$$

$$\Psi_4^3=\begin{bmatrix}I_2 & I_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & I_2 & I_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I_2 & I_2 & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & I_2 & I_2\end{bmatrix}.$$

$$m_{\partial_{[3]}f}=\Psi_4^3\ltimes_{\mathcal{B}}m_f=[0\ 0\ 0\ 1\ 0\ 1\ 0\ 0]^T.$$

$$\Psi_4^4=\begin{bmatrix}1 & 1 & 0 & 0 & \cdots & 0 & 0\\ 0 & 0 & 1 & 1 & \cdots & 0 & 0\\ & & & & \ddots & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1\end{bmatrix}.$$

$$m_{\partial_{[4]}f}=\Psi_4^4\ltimes_{\mathcal{B}}m_f=[0\ 0\ 0\ 1\ 0\ 1\ 0\ 0]^T.$$

*A toolbox for all the related computations is available at http://lsc.amss.ac.cn/~dcheng/

Using (30), we can also easily calculate that

$$\begin{array}{ll} m_{\partial_{[1,2]}f}=[0\ 0\ 0\ 0]^T; & m_{\partial_{[1,3]}f}=[0\ 1\ 0\ 1]^T; \\ m_{\partial_{[1,4]}f}=[0\ 1\ 0\ 1]^T; & m_{\partial_{[2,3]}f}=[0\ 1\ 0\ 1]^T; \\ m_{\partial_{[2,4]}f}=[0\ 1\ 0\ 1]^T; & m_{\partial_{[3,4]}f}=[0\ 1\ 1\ 0]^T; \\ m_{\partial_{[1,2,3]}f}=[0\ 0]^T; & m_{\partial_{[1,2,4]}f}=[0\ 0]^T; \\ m_{\partial_{[1,3,4]}f}=[1\ 1]^T; & m_{\partial_{[2,3,4]}f}=[1\ 1]^T; \\ m_{\partial_{[1,2,3,4]f}}=[0]. & \end{array}$$

Note that evaluating the Boolean derivatives at $\mathbf{0}$ is equivalent to taking last element of its corresponding true table. Hence we havethe MacLaurin expansion of $f(x)$ as

$$\begin{aligned} & f(x_1,x_2,x_3,x_4) \\ =&1\oplus x_1\oplus x_2\oplus x_1\wedge x_3\oplus x_1\wedge x_4\oplus x_2\wedge x_3 \\ &\oplus x_2\wedge x_4\oplus x_1\wedge x_3\wedge x_4\oplus x_2\wedge x_3\wedge x_4. \end{aligned} \tag{32}$$

IV. Application to Some Related Problems

Firstly, we consider the solution of Boolean equations which involves a known Boolean function $f(x_1,\cdots,x_n)$ and its Boolean derivatives as

$$G_j\left(x_i,f,\frac{\partial f}{\partial x_i},\cdots,\frac{\partial^k f}{\partial x_{i_1}\cdots\partial x_{i_k}}\right)=c_j, \quad j=1,\cdots,s, i=1,\cdots,n. \tag{33}$$

Using (27), solving the equations (33) is standard [13]. We describe it as an algorithm.

Algorithm 4.1:

- *Step 1:* Convert each logical equation into its algebraic form as

$$M_jx=c_j,\quad j=1,\cdots,s, \tag{34}$$

where $M_j\in\mathcal{L}_{2\times 2^n}$.

- *Step 2:* Multiply all equations in (34) together to build a system as

$$Mx=c, \tag{35}$$

where $x=\ltimes_{i=1}^n x_i$, $c=\ltimes_{i=1}^s c_i$, and $M\in\mathcal{L}_{2^s\times 2^n}$ is constructed as

$$\mathrm{Col}_i(M)=\ltimes_{j=1}^s\mathrm{Col}_i(M_j),\quad i=1,\cdots,2^n. \tag{36}$$

- *Step 3:* Find all the solutions $\delta_{2^n}^j$, which satisfies $\mathrm{Col}_j(M)=c$.

The fault detection of combinational circuits [17], [19] is a typical example of this problem. Let $f(x_1,\cdots,x_n)$ be a Boolean function describing a combinational circuit. the test vector set for double stuck-at faults $x_i(s{-}a{-}\alpha)$, $x_j(s{-}a{-}\beta)$ is the set of solutions of

$$\bar{x}_i^\alpha x_j^\beta\frac{\partial f}{\partial x_i}\oplus x_i^\alpha\bar{x}_j^\beta\frac{\partial f}{\partial x_j}\oplus\bar{x}_i^\alpha\bar{x}_j^\beta\frac{\partial^2 f}{\partial x_i\partial x_j}=1, \tag{37}$$

where $\alpha,\beta\in\mathcal{D}$, and $x^1:=x$, $x^0:=\bar{x}$.

We give an example to depict it.

Example 4.2: Assume a combinational circuit is described as [19]

$$\begin{aligned} & f(x_1,\cdots,x_5) \\ =&\neg\left\{\neg[x_2\vee(\neg x_1\wedge\neg x_3)]\vee\neg(x_1\vee x_5)\right. \\ &\left.\vee\neg(x_4\vee x_5)\vee\neg[\neg x_3\vee(\neg x_2\wedge\neg x_4)]\right\}. \end{aligned} \tag{38}$$

We look for the test vector set for the double stuck at $x_3(s-a-1)$, and $x_4(s-a-0)$.

That is, to solve the equation

$$\bar{x}_3\bar{x}_4\frac{\partial f}{\partial x_3}\oplus x_3x_4\frac{\partial f}{\partial x_4}\oplus\bar{x}_3x_4\frac{\partial^2 f}{\partial x_3\partial x_4}=1, \tag{39}$$

The structure matrix of f is

$$\begin{aligned} M_f=\delta_2[&1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ &1\ 2\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 2]. \end{aligned}$$

Then, using Corollary 3.6, it is easy to obtain

$$\begin{aligned} M_{\partial_{[3]}f}&=\delta_2[1\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 1\ 2\ 1\ 2\ 2\ 2\ 1\ 2] \\ M_{\partial_{[4]}f}&=\delta_2[2\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1\ 2] \\ M_{\partial_{[3,4]}f}&=\delta_2[2\ 1\ 2\ 2\ 2\ 2\ 1\ 2]. \end{aligned}$$

By direct computation, the matrix form of (39) is

$$Mx=1,$$

where $x=\ltimes_{i=1}^5 x_i$, and

$$\begin{aligned} M=\delta_2[&2\ 1\ 2\ 2\ 2\ 1\ 1\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2 \\ &2\ 2\ 2\ 2\ 2\ 2\ 1\ 2\ 2\ 2\ 2\ 2\ 1\ 2\ 1\ 2]. \end{aligned}$$

Thus, the solution is

$$\left\{x=\delta_{32}^i|i=2,6,7,23,29,31\right\},$$

or, in scalar form

$$\begin{aligned} &\{(1,1,1,1,0),(1,1,0,0,1),(1,1,0,0,0), \\ &(0,1,0,0,1),(0,0,0,1,1),(0,0,0,0,1)\}. \end{aligned}$$

Next, we consider the case that in equations (33), the Boolean function $f(x_1,\cdots,x_n)$ is unknown and with a set of the boundary conditions $f(\mathbf{0})$ and some Boolean derivatives of f at $\mathbf{0}$. Then we call this kind of equations the Boolean differential equations (BDE). If a Boolean function $g(x_1,\cdots,x_n)$ satisfied (33) and the boundary conditions, it is called a solution of the BDE with boundary conditions.

Example 4.3: Consider the following Boolean differential equation with boundary condition $F(\mathbf{0})=0$

$$\begin{cases} \frac{\partial F}{\partial x_3}=\neg x_1\wedge\neg x_4 \\ \frac{\partial^2 F}{\partial x_1\partial x_4}=\neg(x_2\vee x_3)\vee(x_2\wedge x_3) \\ \frac{\partial^2 F}{\partial x_2\partial x_4}=\neg x_1 \\ \frac{\partial^2 F}{\partial x_1\partial x_3}\vee\frac{\partial^2 F}{\partial x_1\partial x_2}=1. \end{cases} \tag{40}$$

In vector form we have

$$\begin{cases} M_{\partial_{[3]}F}=\delta_2[2\ 2\ 2\ 2\ 2\ 1\ 2\ 1] \\ M_{\partial_{[1,4]}F}=\delta_2[1\ 2\ 2\ 1] \\ M_{\partial_{[2,4]}F}=\delta_2[2\ 2\ 1\ 1] \\ \mathrm{Col}(M_\vee M_{\partial_{[1,3]}F}(I_4\otimes M_{\partial_{[1,2]}F})(I_2\otimes W_{[2]})(I_4\otimes M_r^2))=\{\delta_2^1\}. \end{cases}$$

Assume the first row of M_F is $[a_1\ a_2\ \cdots\ a_{16}]$, then

$$\begin{array}{ll} a_1 \oplus a_3 = 0 & a_2 \oplus a_4 = 0 \\ a_5 \oplus a_7 = 0 & a_6 \oplus a_8 = 0 \\ a_9 \oplus a_{11} = 0 & a_{10} \oplus a_{12} = 1 \\ a_{13} \oplus a_{15} = 0 & a_{14} \oplus a_{16} = 1 \\ a_1 \oplus a_2 \oplus a_9 \oplus a_{10} = 1 & a_3 \oplus a_4 \oplus a_{11} \oplus a_{12} = 0 \\ a_5 \oplus a_6 \oplus a_{13} \oplus a_{14} = 0 & a_7 \oplus a_8 \oplus a_{15} \oplus a_{16} = 1 \\ a_1 \oplus a_2 \oplus a_5 \oplus a_6 = 0 & a_3 \oplus a_4 \oplus a_7 \oplus a_8 = 0 \\ a_9 \oplus a_{10} \oplus a_{13} \oplus a_{14} = 1 & a_{11} \oplus a_{12} \oplus a_{15} \oplus a_{16} = 1 \\ a_3 \oplus a_7 \oplus a_{11} \oplus a_{15} = 1 & a_4 \oplus a_8 \oplus a_{12} \oplus a_{16} = 0. \end{array}$$

Since $F(\mathbf{0}) = 0$, we know that $a_{16} = 0$, then the solution is

$$\begin{aligned} m_F^T = \mathrm{Row}_1(M_F) \\ = [a \quad & b \quad a \quad b \\ c \quad & a \oplus \neg b \oplus \neg c \quad c \quad a \oplus \neg b \oplus \neg c \\ \neg b \oplus \neg c \quad & a \oplus \neg c \quad \neg b \oplus \neg c \quad a \oplus c \\ a \oplus \neg b \quad & 1 \quad a \oplus \neg b \quad 0 \quad] \end{aligned}$$

where a, b and c can be arbitrary Boolean numbers

V. Boolean Integral

As we mentioned in the introduction, there is no commonly used definition for Boolean integral. [31] provides a framework for Boolean integral. Unfortunately, the Boolean derivative used in [31] is different from the standard one, and hence the integral is in-consistent with the aforementioned Boolean derivative. Moreover, the computation problem has not been solved yet there.

In the following we define the Boolean integrals in the sense that they are precisely the inverse of the Boolean derivatives.

A. Primitive Function

First, we define primitive function.

Definition 5.1: Given a Boolean function $f(x_1, \cdots, x_n)$. $F(x_1, \cdots, x_{i-1}, z, x_i, \cdots, x_n)$ is called the ith primitive function of $f(x)$ (or the ith partial integral of $f(x)$) , denoted by

$$\int f(x_1, \cdots, x_n) d[i] = F(x_1, \cdots, x_{i-1}, z, x_i, \cdots, x_n), \tag{41}$$

if

$$\frac{\partial F}{\partial z} = f(x_1, \cdots, x_n). \tag{42}$$

In the light of Corollary 3.5, the problem becomes solving the equation

$$\Psi^i_{n+1} m_F = m_f. \tag{43}$$

We give an example to demonstrate this.

Example 5.2: Assume $f(x_1, x_2, x_3) = x_3 \wedge (x_1 \vee (x_2 \leftrightarrow x_3))$. Find

$$\int f(x_1, x_2, x_3) d[2].$$

It is easy to calculate that

$$m_f = [1\ 0\ 1\ 0\ 1\ 0\ 0\ 0]$$

Assume

$$F(x_1, z, x_2, x_3) = \int f(x_1, x_2, x_3) d[2],$$

with its truth table as

$$m_F = [a_1\ a_2\ \cdots\ a_{16}]^T.$$

By (43), we can obtain

$$m_F = [c_1\ c_2\ c_3\ c_4\ \neg c_1\ c_2\ \neg c_3\ c_4\ c_5\ c_6\ c_7\ c_8\ \neg c_5\ c_6\ c_7\ c_8]^T,$$

where $c_i, i = 1, \cdots, 8$ can be arbitrary Boolean numbers.

B. Indefinite Integral

Definition 5.3: Given a logical function $F(x_1, \cdots, x_n)$. Its deferential form, denoted by dF, is defined as

$$dF := \frac{\partial F}{\partial x_1} dx_1 + \cdots + \frac{\partial F}{\partial x_n} dx_n. \tag{44}$$

Note that in (44) the symbol " + " is considered as only an adjacent notation, but not an operator.

Definition 5.4: Given a set of functions

$$f_i(x_1, \cdots, x_{i-1}, \hat{x}_i, x_{i+1} \cdots, x_n), \quad i = 1, \cdots, n.$$

A function $F(x_1, \cdots, x_n)$ is called the indefinite integral of the differential form

$$dh = f_1 dx_1 + f_2 dx_2 + \cdots + f_n dx_n$$

(or simply, integral of $\{f_1, \cdots, f_n\}$), if

$$\frac{\partial F}{\partial x_i} = f_i, \quad i = 1, \cdots, n. \tag{45}$$

Note that according to equation (17) one sees that if F is an indefinite integral of dh, then so is $\bar{F}$.

Next, we consider when the indefinite integral exists.

Theorem 5.5: Consider a differential form

$$dh = f_1 dx_1 + f_2 dx_2 + \cdots + f_n dx_n.$$

There exists at least a pair of complemented indefinite integrals, if and only if

$$\frac{\partial f_i}{\partial x_j} = \frac{\partial f_j}{\partial x_i}, \quad 1 \le i < j \le n. \tag{46}$$

Proof. Necessity is trivial. We prove the sufficiency. Using $\{f_i | i = 1, \cdots, n\}$, we can calculate

$$\frac{\partial f_i}{\partial x_j} = \frac{\partial f_j}{\partial x_i}, \quad 1 \le i < j \le n.$$

Similarly, we have third order cross derivatives as

$$\frac{\partial^2 f_i}{\partial x_j \partial x_k} = \frac{\partial^2 f_j}{\partial x_i \partial x_k} = \frac{\partial^2 f_k}{\partial x_i \partial x_j},$$

and even higher order cross derivatives. Using the obtained partial derivatives and following the form of MacLaurin expansion we can construct

$$\begin{aligned} F(x_1, \cdots, x_n) = & c \oplus \bigoplus_{i=1}^{n} f_i|_{\mathbf{0}} \wedge x_i \\ & \oplus \bigoplus_{1 \le i_1 < i_2 \le n} \left.\frac{\partial f_{i_1}}{\partial x_{i_2}}\right|_{\mathbf{0}} \wedge x_{i_1} \wedge x_{i_2} \oplus \cdots \\ & \oplus \left.\frac{\partial^{n-1} f_1}{\partial x_2 \cdots \partial x_n}\right|_{\mathbf{0}} \wedge x_1 \wedge x_2 \wedge \cdots \wedge x_n. \end{aligned}$$

Then it is ready to verify that

$$F(x) = \int dh.$$

□

The following result comes from the constructive proof of Theorem 5.5.

Corollary 5.6: If $\int dh$ exists, then it is unique (up to a complement equivalence).

In the following when we consider the integral of a differential form, we assume

A1 integrable condition (46) holds.

Hence as long as dh is integrable, we can write an indefinite integral as

$$\int dh = F(x) + C,$$

where $C \in \mathcal{D}$.

In later use, we would like to specify F. So we also use the following notation:

$$\int dh = F(x), \quad F(\mathbf{0}) = 0;$$

$$\int \bar{d}h = \bar{F}(x), \quad \bar{F}(\mathbf{0}) = 1.$$

Next, we consider how to calculate the indefinite integral. In fact, the constructive proof already provides a method to find the integral. We are looking another simple proof.

The following result is an immediate consequence of Corollary 3.5.

Theorem 5.7: Each indefinite integral of a differential form $dh = f_1 dx_1 + \cdots + f_n dx_n$ has a solution z of the following linear Galois algebraic system as its truth table.

$$\Psi_n \ltimes_{\mathcal{B}} z = b, \tag{47}$$

where

$$\Psi_n = \begin{bmatrix} \Psi_n^1 \\ \Psi_n^2 \\ \vdots \\ \Psi_n^n \end{bmatrix} \in \mathcal{B}_{n \cdot 2^{n-1} \times 2^n}; \quad \text{and} \quad b = \begin{bmatrix} m_{f_{[1]}}^T \\ m_{f_{[2]}}^T \\ \vdots \\ m_{f_{[n]}}^T \end{bmatrix} \in \mathcal{B}_{n \cdot 2^{n-1}}.$$

It is worth noting that to get the default solution, we need to set $z_{2^n} = 0$.

Example 5.8: Assume $n = 2$. Then we have

$$\Psi_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Case 1: Assume $f_1 = x_2$, $f_2 = \neg x_1$. Then we have $m_{\partial_{[1]} f} = [1\ 0]^T$, $m_{\partial_{[2]} f} = [0\ 1]^T$. The equation (47) becomes

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \ltimes_B \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Equivalently, we have

$$\begin{cases} z_1 \oplus z_3 = 1 \\ z_2 \oplus z_4 = 0 \\ z_1 \oplus z_2 = 0 \\ z_3 \oplus z_4 = 1. \end{cases}$$

Setting $z_4 = 0$, which corresponds to $F(\mathbf{0}) = 0$, we have

$$\begin{cases} z_1 = 0 \\ z_2 = 0 \\ z_3 = 1 \\ z_4 = 0. \end{cases}$$

That is, $m_F = [0\ 0\ 1\ 0]^T$. Hence, $F = (\neg x_1 \wedge x_2)$. Writing it into integral form, we have

$$\int x_2 dx_1 + \neg x_1 dx_2 = (\neg x_1) \wedge x_2. \tag{48}$$

We also have

$$\int x_2 \bar{d} x_1 + \neg x_1 \bar{d} x_2 = (\neg x_1) \wedge x_2 \oplus 1. \tag{49}$$

Case 2: Assume $f_1 = x_2$, $f_2 = 1$. Then we have $m_{\partial_{[1]} f} = [1\ 0]^T$ and $m_{\partial_{[2]} f} = [1\ 1]^T$. It is easy to check that there is no solution. Hence the integral does not exist.

C. Definite Integral

When the indefinite integral of $\int dh$ exists, it is a pair $(F, \bar{F})$. Then we have the following definition.

Definition 5.9: Assume there is a differential form dh as

$$dh = f_1(\hat{x}_1, x_2, \cdots, x_n) dx_1 + \cdots + f_n(x_1, \cdots, x_{n-1}, \hat{x}_n) dx_n,$$

a subset $S \subset \mathcal{D}^n$, and a logical function $g(x)$.

Assume $\int dh = F(x)$ (with $F(0) = 0$). Then we define

$$\int_S g(x) dh = \sum_{x \in S} g(x) \wedge F(x). \tag{50}$$

We also denote the integral with respect to $\bar{F}$ and define

$$\int_S g(x) \bar{d}h = \sum_{x \in S} g(x) \wedge \bar{F}(x). \tag{51}$$

S is called the integral domain and $g(x)$ the integrand.

It is easy to show that the definite integral, defined in Definition (5.9), satisfies some basic properties of the definite integral. For instance, we have

1) If $f(x) \leq g(x)$,

$$\int_S f(x) dh \leq \int_S g(x) dh. \tag{52}$$

2) If $S_1 \subseteq S_2$,

$$\int_{S_1} g(x) dh \leq \int_{S_2} g(x) dh. \tag{53}$$

3)

$$\int_{S_1 \cup S_2} g(x) dh = \int_{S_1} g(x) dh + \int_{S_2} g(x) dh - \int_{S_1 \cap S_2} g(x) dh. \tag{54}$$

Define

$$\text{supp}(f) = \{x | f(x) \neq 0\}.$$

Let $F = \int dh$. Then we have

$$\int_S g(x)dh = |\text{supp}(F) \cap \text{supp}(g) \cap S|.$$

Example 5.10: Recall Example 5.8. We consider the definite integrals using the corresponding indefinite integrals. Case 1: Assume $f_1 = x_2$, $f_2 = \neg x_1$. Then the default indefinite integral is $F = x_1 \wedge (\neg x_2)$. Assume $S = \{(x_1, x_2) | x_1 \to x_2 = 1\}$. Then

$$S = \{(1,1), (0,1), (0,0)\}.$$

Let the integrand be $g(x_1, x_2) = x_1 \leftrightarrow x_2$. Then it is easy to calculate that

$$\int_S (x_1 \leftrightarrow x_2) x_2 dx_1 + \neg x_1 dx_2 = 0;$$

and

$$\int_S (x_1 \leftrightarrow x_2) x_2 \bar{d}x_1 + \neg x_1 \bar{d}x_2 = 2.$$

Case 2: Assume $f_1 = x_2$, $f_2 = 1$. Then we have that

$$\int_S x_2 dx_1 + dx_2 = \emptyset.$$

VI. Conclusion

Using semi-tensor product of matrices and the matrix expression of logic, the calculation of Boolean derivative was investigated. A very simple formula was obtained, which converts the calculation of derivative into a modulo-2 matrix-vector product. Using it, three kinds of Boolean integrals were proposed. First, find f from $\frac{\partial f}{\partial x_i}$ is called its ith primitive function. Second, find f from a differential form $dh = h_1(\hat{x}_1, x_2, \cdots, x_n)dx_1 + \cdots + h_n(x_1, \cdots, x_{n-1}, \hat{x}_n)dx_n$ is called the indefinite integral of dh. A necessary and sufficient condition was obtained for the existence of indefinite integral. Using indefinite integral, the definite integral was also properly defined. Simple formulas were also obtained for the calculation of each integrals.

References

[1] S.S Agaian, J. Astola, and K. Egiazarian, *Binary Polynomial Transforms and Nonlinear Digital Filters*. New York: Marcel Dekker, 1995.

[2] S.S. Agaian, K.A. Panetta, S.C. Nercessian, E.E. Danahy, Boolean derivatives with application to edge detection for imaging systems, *IEEE Trans. Syst., Man., Man, Cybern. B, Cubern.*, Vol. 40, No. 2, 371-382, 2010.

[3] S.B. Akers Jr., On a theory of Boolean functions, *SIAM J.*, Vol. 7, 487-498, 1959.

[4] D. Bochmann, *Boolean Differential Calculus*, Larl Marx Stadt, German Democratic Pepublic, 1978.

[5] F.M. Brown, *Boolean Reasoning: The Logic of Boolean Equations*, second Ed., Dover Pub., New York 2003.

[6] D. Cheng, Input-state approach to Boolean networks, *IEEE Tans. Neural Networks*, Vol. 20, No. 3, 512-521, 2009.

[7] D. Cheng, H. Qi, Controllability and Observability of Boolean Control Networks, *Automatica*, Vol. 45, No. 7, 1659-1667, 2009.

[8] D. Cheng, Z. Li, H. Qi, Realization of Boolean control networks, Automatica, Vol. 46, No. 1, 62-69, 2010.

[9] D. Cheng, H. Qi, A linear representation of dynamics of Boolean networks, *IEEE Trans. Aut. Contr.*, Vol. 55, No. 10, 2251-2258, 2010.

[10] D. Cheng, H. Qi, State-space analysis of Boolean networks, *IEEE Trans. Neural Networks*, Vol. 21, No. 4, 584-594, 2010.

[11] D. Cheng, H. Qi, Z.Q. Li, J.B. Liu, Stability and stabilization of Boolean networks, *Int. J. Robust and Nonlinear Control*, Vol. 21, No. 2, 134-156, 2011.

[12] D. Cheng, Disturbance Decoupling of Boolean Control Networks, *IEEE Trans. Aut. Contr.*, Vol. 56, No. 1, pp. 2-10, 2011.

[13] D. Cheng, H. Qi, Z. Li, *Analysis and Control of Boolean Networks, A semi-tensor Product Approach*, Springer-Verlag, London, 2011.

[14] P.J. Daniell, The modular difference of classes, *Bull. Amer. Math. Soc*, Vol. 23, 446-450, 1917.

[15] M. Davio, J.P. Deschamps, A. Thayse, *Discrete and Switching Functions*, McGraw Hill, New York, 1978.

[16] K. Egiazarian, P. Kuosmanen, J. Astola, Boolean derivatives, weighted chow parpmeters, and selection probabilities of stack filters, *IEEE Trans. Signal Proc.*, Vol. 44, No. 7, 1341-1634, 1996.

[17] O. Keren, Reduction of average path length in inary decision diagrams by spectral methods, *IEEE Computers*, Vol. 57, No. 4, 520-531, 2008.

[18] V.G. Levashenko, V.P. Shmerko, S.N. Yanushkevich, Parallel algorithms for evaluation of directional Boolean derivatives of multivalued logic functions, *Cybern. Sys. Anal.*, Vol. 32, No. 6, 777-793, 1996.

[19] H. Li, Y. Wang, Boolean derivative calculation with application to fault detection of combinational circuits via the semi-tensor product method, preprinter.

[20] B. Luque, R.V. Sole, Lyapunov exponents in random Boolean networks, *Physica A*, Vol. 284, 33-45, 2000.

[21] P.C. Pabitra, S. Sahoo, M. Chakraborty, S.K. Bhandari, A. Pal, Investigation of the globak dynamics of cellular automata using Boolean derivatives, *Comput. Math. Appl.*, Vol. 57, No. 8, 1337-1351, 2009.

[22] C. Posthoff, B. Steinback, *Logic Foundations and Equations, Binary Models for Computer Sciece*, Springer-Verlag, Dordrecht, 2004.

[23] I.S. Reed, A class of multiple error-correctiong and the decoding schene, *IRE Trans. Inform. Theory*, IT-4, 38-49, 1954.

[24] K.A. Ross, C.R.B. Wright, *Discrete Mathematics*, 5th Ed., Prentice Hall, 2003.

[25] S. Rudeanu, What can be expected from a Boolean derivative? *An. St. Univ. Ovidius Constanţa*, Vol. 7, No. 1, 177-186, 2009.

[26] R. Scheuring, H. Wehlan, On the design of discrete event dynamic systems bu means of the Boolean differential calculus, D. Franke(ed.) F. Kraus (ed.), *Design Methods of Control Systems*, Vol. 2, Pergamon, 723-728, 1991.

[27] W.G. Schneeweiss, *Boolean Functions, withEngineering Applications and Computer Programs*, Springer-Verlag, Berlin, 1989.

[28] J.R. Smith, Jr., C.H. Roth, Jr., Analysis and synthesis of asynchronous sequential networks using edge-sensitive flip-flops, *IEEE Trans. Comput.*, Vol. 20, 847-855, 1971.

[29] A. Thayse, *Boolean Calculus of Differentces*, Springer-Verlag, Berlin, 1981.

[30] A. Thayse, *P-Functions and Boolean Matrix Factorization*, LNCS 175, Springer-Verlag,Berlin, 1984.

[31] J.H. Tucker, M.A. Tapia, A.W. Bennett, Boolean integral calculus, *Appl. Math. and Comput.*, Vol. 26, 201-236, 1988.

[32] E. Vandermeulen, H.A. Donegan, M. Larnac, J. Magnier, The Temporal Boolean derivative applied to verification of extended finite state machines, *Computers Math. Applic.*, Vol. 30, No. 2, 27-36, 1995.

[33] G.Y. Vichniac, Boolean derivatives on cellular automata, *Physica D*, Vol. 45, 63-74, 1990.

[34] S. Yanushkevic, *Logic Differential Calculus in Multi-Valued Logic Design*, Tech. Univ. Szczecin, Szczecin, 1998.

[35] B. Yildiz, M. Subasi, An interpreter for the Boolean derivative, *Applied Math. and Comput.*, Vol. 129, 43-54, 2002.