国 防 科 技 大 学 学 报
第 20 卷 第 6 期　JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY　Vol. 20 No. 6 1998

布尔函数不交化的立方体算法*

武小悦　沙基昌

(国防科技大学系统工程与数学系　长沙　410073)

摘　要　布尔函数的不交化是进行系统可靠度计算中的重要步骤。本文提出了一种进行布尔函数不交化的立方体算法。基于布尔函数的立方体表示法及不交代数, 文中定义了立方体的有关运算法则, 建立了立方体矩阵不交化算法, 并用算例进行了说明。本算法的实现效率高, 操作简便。

关键词　布尔函数, 不交化, 可靠性

分类号　O213.2

Cubical Algorithm for Disjoint of Boolean Function

Wu Xiaoyue Sha Jichang

(Department of Systems Engineering and Mathematics, NUDT, Changsha, 410073)

Abstract　Disjoint of Boolean function is an important apporach to the calculation of system reliability. In this paper, based on cubical form of Boolean function and disjoint algebra, an algorithm is presented for disjoint of Boolean function, several relative laws for cubes operation are defined, and cubical matrix disjoint algorithm is given with a numerical example. The algorithm is effective and simple.

Key words　boolean function, disjoint, reliability

系统可靠度的计算是进行系统可靠性设计与分析的必要组成部分。进行系统可靠度计算的一般步骤是: (1) 依实际系统的物理和功能组成求出结构函数, 一般可以通过建立系统可靠性框图模型或故障树模型得出系统的结构函数, 此结构函数反映了部件失效与系统失效之间的逻辑关系, 通常可归结为一个布尔逻辑表达式。(2) 通过结构函数求出系统的可靠度, 其主要方法是采用概率求和公式。这一步的关键问题是在概率计算中把相交事件的概率和化解为独立事件的概率和形式。常用的算法有: 真值表法、卡诺图法、容斥定理、不交化布尔代数[1] 等。真值表法和卡诺图法具有直观易懂的优点, 但仅适用于部件数目较少的情形。当项数较多时采用容斥定理计算的复杂性会剧增。由我国学者廖炯生提出的不交化代数算法[1]是一种较好的方法, 可以较快地求出系统的可靠度, 但该法在实现时仍会有一些多余操作, 而且目前也未见有高效的计算机实现算法。本文在布尔函数的立方体表示法[2] 及不交化代数法思路的基础上, 定义了布尔函数的几种相关算子, 提出了采用立方体矩阵不交化算法实现布尔逻辑函数的不交化。该算法能高效、简便地在计算机上实现, 适用于大型系统结构函数的可靠度计算。

1　布尔立方体的运算法则

一个布尔逻辑函数除了可用真值表、布尔代数式、卡诺图等方法表示之外, 还可以采用布尔立方体矩阵进行表示。首先将布尔逻辑函数表示为积之和的形式, 据此可以直接写出其布尔立方体矩阵表示, 有关布尔函数及立方表示法的详细说明可参阅文献[2]。

如布尔逻辑函数 $F=\bar{A}+B\bar{C}+A\bar{B}C$ 可表示为: $F=\begin{bmatrix}0 & x & x\\ x & 1 & 0\\ 1 & 0 & 1\end{bmatrix}$

* 1998 年 4 月 28 日收稿
第一作者: 武小悦, 男, 1963 年生, 副教授

其中:A, B, C 为逻辑变量,$\bar{A}$, $\bar{B}$, $\bar{C}$ 分别表示对应的逻辑变量的非,F 的各列分别依次对应于 A, B, C。0 表示补变量,1 表示原变量,x 表示对应的布尔变量可取 0,1 任意值。上述 F 包含三个布尔立方体:二维立方体$(0\ x\ x)$,一维立方体$(x\ 1\ 0)$,零维立方体$(1\ 0\ 1)$。

为了便于不交化运算,本文定义布尔立方体的下述算子及运算法则:

(1) 蕴含运算($\subseteq$)[2]

两个立方体的$\subseteq$运算规则是依立方体元的$\subseteq$运算表(表 1)得出的:

若两立方体 a, b 的对应元素对中,至少存在一个元素对 a_i, b_i 满足 $a_i \subseteq b_i = y$(y 表示 a_i 不蕴含于 b_i),则有 a 不蕴含于 b,否则有 $a \subseteq b$($a \subseteq b$ 表示 a 蕴含于 b,表中 $a_i \subseteq b_i$ 用 ϵ 表示)。

例如:若 $a=(1\ x\ 1)$, $b=(1\ 0\ x)$,则 a 不蕴含于 b。当 $a=(1\ x\ 1)$, $b=(1\ x\ x)$,则有 $a \subseteq b$。

表 1 $\subseteq$ 运算表

$a_i \subseteq b_i$		b_i = 0	b_i = 1	b_i = x
a_i	0	ϵ	y	ϵ
	1	y	ϵ	ϵ
	x	y	y	ϵ

利用$\subseteq$运算,可以在立方体矩阵中,删除那些已蕴含于另一些立方体中的多余立方体,从而简化函数,减少矩阵维数。

(2) 归并运算($\cup$)

若两个立方体 a, b 所有的对应元素对中,至少存在一对元素 a_i, b_i 满足 $a_i \cup b_i = \varnothing$(表示无并),则 a 与 b 不能归并。否则,当仅当 a, b 对应的元素对中只存在一个互补元素对 a_i, b_i(即 $a_i=1$, $b_i=0$ 或 $b_i=1$, $a_i=0$)时,依$\cup$运算表(表 2)进行归并得到一个新的高维立方体 $a \cup b$。

例如:$a=(1\ x\ 0)$, $b=(0\ x\ 0)$,则 $a \cup b=(x\ x\ 0)$

表 2 $\cup$运算表

$a_i \cup b_i$		b_i = 0	b_i = 1	b_i = x
a_i	0	0	x	$\varnothing$
	1	x	1	$\varnothing$
	x	$\varnothing$	$\varnothing$	x

利用$\cup$运算可以将两个低维立方体化简为一个高维立方体,从而减少立方体矩阵中的立方体数。

(3) 交运算($\cap$)[2]

a, b 的$\cap$运算法则依表 3 进行,若立方体 a, b 中对应元素对中,至少存在一对元素 a_i, b_i 满足 $a_i \cap b_i = \varnothing$(表示无交),则 $a \cap b = \varnothing$(无交)。

表 3 $\cap$运算表

$a_i \cap b_i$		b_i = 0	b_i = 1	b_i = x
a_i	0	0	$\varnothing$	0
	1	$\varnothing$	1	1
	x	0	1	x

(4) 不交化运算($\oplus$)

立方体 a 与 b 的不交化运算$\oplus$的规则如下:

①令 x 数目最多者为 a(否则重新标识即可);

②若 $a \cap b = \varnothing$，转⑦;

③将 a 写在第1行;

④令 $i = 1$;

⑤若 $a_i \neq x$，且 $b_i = x$，则第 $i+1$ 行写为

$$b_i^1, \cdots, b_{i-1}^1, \bar{a}_i, b_{i+1}, \cdots, b_n$$

其中 $b_k^i = a_k (k = 1, 2, \cdots, i-1$，且 $b_k = x)$;

⑥若 $i \neq n$，则令 $i = i+1$，返⑤;

⑦停机，得不交化立方体阵 $a \oplus b$。

例如，若 $F = A\bar{B}C + CD$，则令 $a = CD$，$b = A\bar{B}C$，依 $\oplus$ 运算过程为

$$(x\ \ x\ \ 1\ 1) \rightarrow \begin{pmatrix} x & x & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

故　$F = CD + A\bar{B}C\bar{D}$

2　布尔立方体矩阵不交化算法

该算法步骤如下:

(1) 将布尔逻辑函数 F 写成积之和形式: $F = y_1 + y_2 + \cdots + y_m$

(2) 依(1)写出该布尔逻辑函数的立方体矩阵

$$F_{m \times n} = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}$$

其中 $f_1, \cdots, f_m$ 为 F 的项 $y_1, y_2, \cdots, y_m$ 对应的布尔立方表示，依各立方体中 x 的数目大小从上到下排列。

(3) 令 $F^1_{m^1 \times n} = F_{m \times n}$，$i = 1$。

(4) 对 $F^1_{m^1 \times n}$ 进行蕴含与归并运算，得 $\bar{F}_{\bar{m} \times n}$，操作规则如下:

①若 $f_j^1 \subseteq f_i^1 (j > 1)$，则删除 f_j;

②若 $f_j^1 \cup f_k^1 = f_l^1 (j > k > i)$，则删除 f_j^1 及 f_k^1，用 f_l^1 取代 f_k^1 的位置;

③若 $f_j^1 \subseteq f_i^1 (j, k > i)$，则删除 f_j^1;

④反复进行①～③操作，直到无删除操作为止;

⑤将 $\bar{F}_{\bar{m} \times n}$ 中所有 $j > i$ 的 $\bar{f}_j$ 依其含 x 的数目大小从上到下排列，得新的 $\bar{F}_{\bar{m} \times n}$

(5) 对 $\bar{F}_{\bar{m} \times n}$ 进行不交化运算，得 $\bar{F}^*_{m^* \times n}$，规则如下:

$$\bar{F}^*_{m^* \times n} = [\bar{f}_1\ \ \bar{f}_2 \cdots \bar{f}_i\ \ \ \bar{f}_i \oplus \bar{f}_{i+1}\ \ \ \bar{f}_i \oplus \bar{f}_{i+2} \cdots \bar{f}_i \oplus \bar{f}_{\bar{m} \times n}]^{\mathrm{T}}$$

(6) 若 $i \neq m^*$，令 $i = i+1$，$F \doteq F^*$，返回(4)。

(7) 得 F 的不交化立方体矩阵 $\bar{F}^*_{m^* \times n}$。

在上述算法中，将各立方体依含 x 的多少在立方体矩阵中从上到下排列及步骤(4)，主要是为了加快不交化速度，减少操作次数。

3　算例

为了说明本文提出的算法的有效性，我们针对下述布尔逻辑函数进行操作:

$$F = x_1 x_4 + x_2 x_4 + x_2 x_5 + x_3 x_5$$

其不交化操作过程如下:

$$F_{m\times n}=\begin{bmatrix}1&x&x&1&x\\x&1&x&1&x\\x&1&x&x&1\\x&x&1&x&1\end{bmatrix}\xrightarrow{f_1\text{与}f_2,f_3,f_4\text{不交}}\begin{bmatrix}1&x&x&1&x\\0&1&x&1&x\\0&1&x&x&1\\1&1&x&0&1\\0&x&1&x&1\\1&x&1&0&1\end{bmatrix}$$

$$\xrightarrow{\text{依含}x\text{数排列}}\begin{bmatrix}1&x&x&1&x\\0&1&x&1&x\\0&1&x&x&1\\0&x&1&x&1\\1&1&x&0&1\\1&x&1&0&1\end{bmatrix}\xrightarrow{f_2\text{与}f_3,f_4,f_5,f_6\text{不交}}\begin{bmatrix}1&x&x&1&x\\0&1&x&1&x\\0&1&x&0&1\\0&0&1&x&1\\0&1&1&0&1\\1&1&x&0&1\\1&x&1&0&1\end{bmatrix}$$

$$\xrightarrow{\text{蕴含}f_3\supseteq f_5,\text{归并}f_3\cup f_6}\begin{bmatrix}1&x&x&1&x\\0&1&x&1&x\\x&1&x&0&1\\0&0&1&x&1\\1&x&1&0&1\end{bmatrix}\xrightarrow{f_3\text{与}f_4,f_5\text{不交}}\begin{bmatrix}1&x&x&1&x\\0&1&x&1&x\\x&1&x&0&1\\0&0&1&x&1\\1&0&1&0&1\end{bmatrix}$$

由此不交化结果为

$$F=x_1x_4+\bar{x}_1x_2x_4+x_2\bar{x}_4x_5+\bar{x}_1\bar{x}_2x_3x_5+x_1\bar{x}_2x_3\bar{x}_4x_5$$

经作者采用卡诺图验证, 上述结果正确。显然本文提出的算法运算的效率及操作简便性较不交代数法有明显改进, 而且易于在计算机上编程实现。

4 结束语

为了便于系统可靠度计算, 本文基于不交代数法的思想及布尔逻辑函数的立方体表示法, 定义了立方体的几个算子, 并提出了布尔逻辑函数的立方体矩阵不交化算法。该算法操作简便, 易于计算机化, 计算效率高, 是一种较好的不交化方法, 可为大型系统的可靠性分析提供有效支持。

参考文献

1 梅启智, 廖炯生, 孙惠中. 系统可靠性工程基础. 北京: 科学出版社, 1987: 76～110

2 周南良. 数学逻辑. 长沙: 国防科技大学出版社, 1992: 186～193