布尔序集相邻逻辑对称的实现*

林柏钢
(福州大学，福州)

摘要　本文提出一种用逻辑对称轴的关系，解决N维布尔序集唯一相邻的逻辑路径问题。同时还给出一种限维内任一逻辑相邻子集的确定方法。其结果简单直观，适合于计算机实现。这种思想，对于二分树快速搜寻和二分树排序决策等，也具有一定意义。

关键词　布尔序集；唯一相邻路径；逻辑对称轴

一、引言

n维布尔函数 $X_1X_2\cdots X_n$，其中 $X_i(i=1, 2,\cdots, n)\in\{0, 1\}$，当其对应的每组中相邻布尔元素仅一个变元相异的情形，称为二元布尔函数逻辑相邻. 布尔函数这种相邻逻辑关系，在数字控制技术，通讯编码，计算机系统等方面有着广泛应用. 比如，为减少计算机系统内部电子电路的操作，克服因内部时序和速度引起的出错问题，常采用这种编码方式. 但随着维数增加，给识别带来很大困难，因而影响了它的应用。本文借助离散数学工具，通过逻辑空间集合的概念，着重研究N维布尔序集相邻逻辑的实现问题.

二、布尔序集相邻逻辑对称定理

定义1　一类N维布尔逻辑空间所对应的顶点最小项集合称为布尔序集，如果其顶点最小项依次用自然数 $0, 1, 2,\cdots, 2^N-1$ 来表示的话.

定理1　任一由布尔序集张成的逻辑空间Q，其顶点集合 $U=\{v_0, v_1,\cdots, v_{2^n-1}\}$ 依次相邻排列，必构成一条唯一相邻闭环链路R.

如果从每一链e_i始于v_{i-1}，终于v_i的角度出发，证明定理1的存在是显然的.（证明略).

定理2　闭环链路R按2^N规则沿中心依次对折划分，各顶点子集映射成镜像相邻对称.

证明　为讨论方便，假设闭环链路R呈开环链路分布(当然可认为开环的首尾依然相接).

若R_l, R_r为按2^N规则对折划分后的左右两单元链路，$\{R_l, R_r\}\in R$. 与其相对应的R_l, R_r上所有连续存在的顶点子集 $\{V_l\}(l=0, 1, i-1, i)$ 和 $\{V_r\}$ $(r=j, j+$

* 1988年6月7日收到，同年10月定稿.

$1, \cdots, 2^n-2, 2^n-1)$，(i, j 为对折处分离相邻点)．由开环链路使得 $\{V_l\} \subset R_l \subset R$，$\{V_r\} \subset R_r \subset R$，$\{V_l\}$ 与 $\{V_r\}$ 中的各顶点子集一一对折映射，构成 $R=\{(V_l, V_r): R_l(V_l)=R_r(V_r)\}$ 镜像对等．显然，闭环链路 R 满足自反，对称，传递的关系．

对折映射的两两顶点子集中，必有一对布尔元素 x_i 与 x_i' 互为相异．当且仅当两两顶点子集异或结果只存在一个 1 元素（即 $x_i \oplus x_i'=1$），其余为 0 元素时（即除 x_i, x_i' 外，$\overline{x_i \oplus x_i'}=0$），则表明二者存在相邻关系，否则，不存在．从而可以推论，任意对折划分后的各顶点子集，若满足上述关系，必一一映射成镜像相邻对称．定理 2 证毕．

三、相邻逻辑对称轴构成

定义 2　R 以开环水平方向分布的链路，称布尔序集相邻逻辑对称轴．

从 N 维布尔序集张成的逻辑空间图，不难发现，实际上是一个正则图．各顶点对应的链数 e_k 相同，而由正则图上存在着一条唯一相邻路径，其对应最小项顶点集合为 2^N 个．

轴的构成　首先我们约定，0 是独立集，$\{0\} \in R$，连接不可约，并定为 R 的左界．显然，R 的右界为 2^N-1．这样构成的相邻逻辑对称轴必满足定理 1、2．

然后引入顶层，第一层，第二层，…第 n 层概念，根据二分树搜寻思想，以顶层为中心，按二分树生枝规则，寻找出各层情况，见图 1．再通过各层顶点子集的集合，研究逻辑对称轴形成的一般规律．

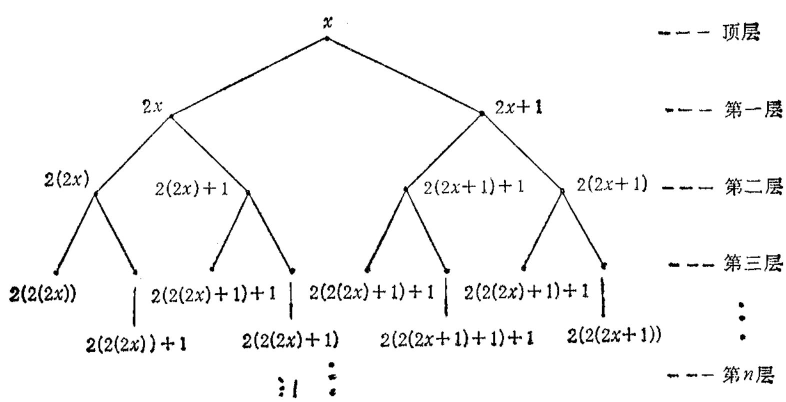

图 1　各层二分树生枝示意图

作法 1　以独立集 {0} 为顶层，沿二分树生枝，即得第一层顶点子集．照此继续分枝，得各层顶点子集，联结各层顶点子集，就是对应的 2^N 层相邻布尔序集逻辑对称轴．

具体作法是：以独立集 {0} 为中点划分，得左右两单元链．左单元链 e_0 约定为偶链，右单元链 e_1 约定为奇链．沿二分树继续分枝，链的划分也相应继续往右界延伸。分别从左界开始，e_0 约定为偶链，e_1 约定为奇链，e_2 为偶链，e_3 为奇链，…依此类推，偶奇交替．分别按"**偶 × 2**"与"**奇 × 2 + 1**"规则，生成下一层的两个顶点子集，所联结的两个新子集增加相同的倍数生成．例如一个六维的情形，如图 2 所示．

从所找到的布尔序集逻辑对称轴三角阵中可以发现：(1) 每一层第一个子集是 0，

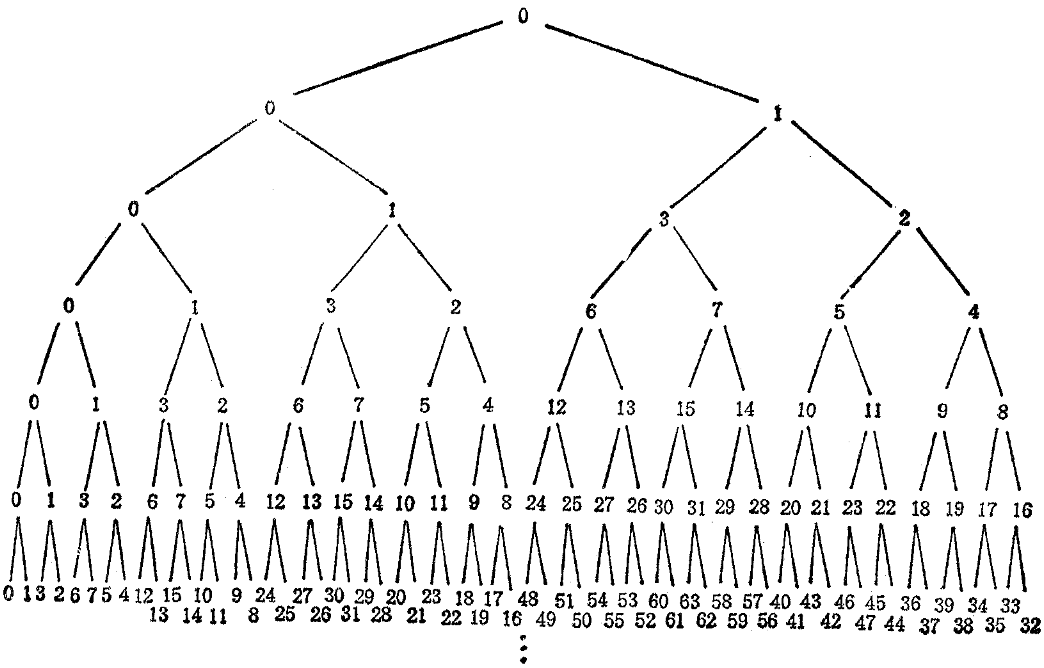

图 2 六维情形的逻辑对称轴构成

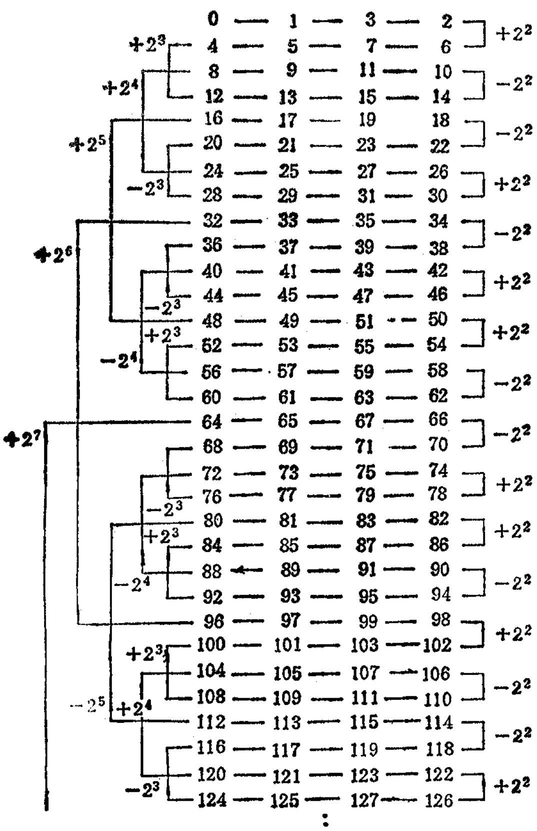

图 3 以基本单元子集链构成的新链簇

最后一个子集是 2^N-1. (2) 同一层中的前子集与后子集，分别按先"偶链 $\times 2$"或后"奇链 $\times 2+1$"规则增加倍数，产生下一层新子集. (3) 高维数情形水平分布，是一个紧收敛过程.

作法 2 找一基本单元子集链

$$R'=\{e_1, v_0, e_2, v_1, \cdots v_k, e_{k+1}\}$$

再以 2^n 倍同样构出一条条新链. 这种方式，同样也能构造一个满足定理 1、2 的新链族. 基本单元越大，所构造的新链簇越快. 例如：当 $n=2$ 时，$R'_2=\{e_1, v_0, e_2, v_1, e_3, v_2, e_4, v_3, e_5\}$，然后再以 2^2 倍同样构出各链族如下，见图 3.

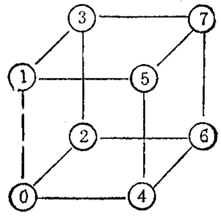

图 4 三维布尔逻辑空间图

四、限维内任一逻辑相邻子集的确定

设 Q 是一个具有顶点子集 $U'=(v_1, v_2, \cdots v_n)$ 和链 $E'=(e_1, e_2, \cdots e_m)$，且 $m>n$ 的逻辑空间正则图．由三维布尔序集张成的逻辑空间图如图 4 所示．

$$R^*=\begin{array}{c} \\ V_0 \\ V_1 \\ V_2 \\ V_3 \\ \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{array}\begin{array}{c} \begin{array}{cccccccc} V_0 & V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \end{array} \\ \left[\begin{array}{cccc:cccc} \phi & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & \phi & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & \phi & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & \phi & 0 & 0 & 0 & 1 \\ \hdashline 1 & 0 & 0 & 0 & \phi & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & \phi & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & \phi & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & \phi \end{array}\right] \end{array}$$

任一顶点子集所联结的相邻点，通过相邻矩阵 R^* 来表征．假定逻辑空间 $Q=(V, E)$ 的顶点集合 $U=\{v_0, v_1, \cdots v_{2^n-1}\}$，则 $R^*=(a_{ij})_{2^n\times 2^n}$ 是 Q 的布尔序集相邻矩阵．其中，

$$a_{ij}=\begin{cases}1, & (v_i, v_j)\in E, \text{ 即 } v_i \text{ 和 } v_j \text{ 相邻接} \\ 0, & \text{不邻接}\end{cases}$$

链路关联集矩阵 $P=(b_{ij})_{2^m\times 2^m}$，其中，

$$b_{ij}=\begin{cases}1, & \text{从顶点子集 } v_i \text{ 存在一条链路到达 } v_j \\ 0, & \text{不存在}\end{cases}$$

而矩阵 R^* 和 P 的所有元素均为 0 或 1，称为布尔序集矩阵．链路关联集矩阵 P 通过相邻矩阵 R^* 可得到：

$$P=\bigvee_{k=1}^{n} R^{(k)}$$

式中 $\vee$ 是布尔和．考虑到每个顶点至少有一条链相邻接，故对于任一顶点子集 v_i 来说，对应有子集 $P_{i1}\vee P_{i2}\vee\cdots\vee P_{im}$，$(i=1, 2, \cdots n)$．一般有相邻顶点 (r, s)，分别对应链路关联子集 $P_{rj}\vee P_{sj}$，$(1\leqslant j\leqslant m)$．

实现办法　把任一顶点布尔序集，先按二进制数**位权**分解成：

$$S=(K_{n-1}K_{n-2}\cdots K_0)_2=\sum_{i=n-1}^{0} K_i(2)^i, \quad (K_i=1/0)$$

然后借助相邻矩阵关系，通过布尔和"$\vee$"运算，依次可全部求出各顶点矩阵 P_{ij} 的全部相邻数，即找到一组相邻顶点集合类．并有以下关系：所分解的位权项数＝关联子集组数．

例 1　$31=2^4+2^3+2^2+2^1+2^0=16+8+4+2+1$

分别求布尔和如下:

$$U_1 = (2^4 \vee 2^3 \vee 2^2 \vee 2^1) = 30$$
$$U_2 = (2^4 \vee 2^3 \vee 2^2 \vee 2^0) = 29$$
$$U_3 = (2^4 \vee 2^3 \vee 2^1 \vee 2^0) = 27$$
$$U_4 = (2^4 \vee 2^2 \vee 2^1 \vee 2^0) = 23$$
$$U_5 = (2^3 \vee 2^2 \vee 2^1 \vee 2^0) = 15$$

因此,$\{31\} = (30, 29, 27, 23, 15)$ 就是所要找的全相邻的集合类. 也可以写成

$$\{(31)_{10} = (11111)_2\} = \left\{\begin{array}{l}(30)_{10} = (11110)_2 \\ (29)_{10} = (11101)_2 \\ (27)_{10} = (11011)_2 \\ (23)_{10} = (10111)_2 \\ (15)_{10} = (01111)_2\end{array}\right\}$$

例 2 $125 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^0 = 64 + 32 + 16 + 8 + 4 + 1$

分别求布尔和如下:

$$U_1 = (2^6 \vee 2^5 \vee 2^4 \vee 2^3 \vee 2^2) = 124$$
$$U_2 = (2^6 \vee 2^5 \vee 2^4 \vee 2^3 \vee 2^0) = 121$$
$$U_3 = (2^6 \vee 2^5 \vee 2^4 \vee 2^2 \vee 2^0) = 117$$
$$U_4 = (2^6 \vee 2^5 \vee 2^3 \vee 2^2 \vee 2^0) = 109$$
$$U_5 = (2^6 \vee 2^4 \vee 2^3 \vee 2^2 \vee 2^0) = 93$$
$$U_6 = (2^5 \vee 2^4 \vee 2^3 \vee 2^2 \vee 2^0) = 61$$

因此,$\{125\} = (124, 121, 117, 109, 93, 61)$ 就是所要找的全相邻的集合类. 同样也可以写成

$$\{(125)_{10} = (1111101)_2\} = \left\{\begin{array}{r}(124)_{10} = (1111100)_2 \\ (121)_{10} = (1111001)_2 \\ (117)_{10} = (1110101)_2 \\ (109)_{10} = (1101101)_2 \\ (93)_{10} = (1011101)_2 \\ (61)_{10} = (0111101)_2\end{array}\right\}$$

综上所述,布尔函数的相邻逻辑关系问题,完全可以由布尔序集相邻逻辑对称轴的关系来实现. 而且能够全部找出相邻集合类. 这种方法简单,容易找到一条唯一布尔逻辑相邻路径. 当 n 变大时. 用手工实现冗繁,可以借助计算机来实现.

对于维数已定的任一顶点来说,利用本文介绍的相邻矩阵和链路关联子集的办法,来解决布尔逻辑相邻顶点子集问题,也是相当实用,同样准确快速.

参考文献

[1] S. 季普舒茨著,杜玮编译,离散数学,宇航出版社,北京,1985,2.
[2] [罗]. I. Tomescu 著,清华大学应用数学系离散数学教研室译,组合学引论,高教出版社,北京 1985. 7.
[3] [美] J. L. 凯莱著,吴从炘等译,一般拓扑学,科学出版社,北京, 1982.5.

REALIZATION OF THE NEIGHBOURING LOGIC SYMMETRY FOR BOOLEAN ORDERED SET

Lin Bogang
(Fuzhou University, Fuzhou)

Abstract By means of logic symmetric relation, single neighbouring logic path for N dimension Boolean ordered set is solved. A new method of determining any logic neighbouring subset in limited dimensions is given. Its results are intuitional and realizable for computer.

Key words Boolean ordered set; Single neighbouring path; Logic symmetry