Xiao-hua Li, Ji-zhong Shen, 2014. An algorithm for identifying symmetric variables in the cannonical OR-coincidence algebra system. *Journal of Zhejiang University-SCIENCE C (Computers & Electronics),* **15**(12):1174-1182. [10.1631/jzus.C1400093]

An algorithm for identifying symmetric variables in the cannonical OR-coincidence algebra system

Key words: Symmetric variable, d_j-map, Canonical OR-coincidence algebra system, Boolean function

Corresponding author: Ji-zhong Shen
E-mail: jzshen@zju.edu.cn

Motivation

- A new symmetry detection algorithm based on OR-NXOR expansion is proposed. Experimental results show that the proposed algorithm is convenient and efficient

- Disadvantages of existing methods:
  - Applicability of the number of logical variables usually $n \leq 6$
  - The identification processes are complicated
  - Only six types can been identified

Features of our method

- Avoid the transformation process from OR-NXOR expansion to AND-OR-NOT expansion, or to AND-XOR expansion
- Solve the problem of completeness in the d_j-map method
- The new algorithm is an optimal detection method in terms of applicability of the number of logical variables, detection type, and complexity of the identification process

Framework of our method (I)

The algorithms can be divided into three parts:

the constraint conditions of the order coefficient subset matrices are revealed for 12 types of symmetric variables

$$[\boldsymbol{C}_0\ \boldsymbol{C}_1\ \boldsymbol{C}_2\ \boldsymbol{C}_3] = [\boldsymbol{C}^0\ \boldsymbol{C}^1\ \boldsymbol{C}^2\ \boldsymbol{C}^3]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Framework of our method (II)

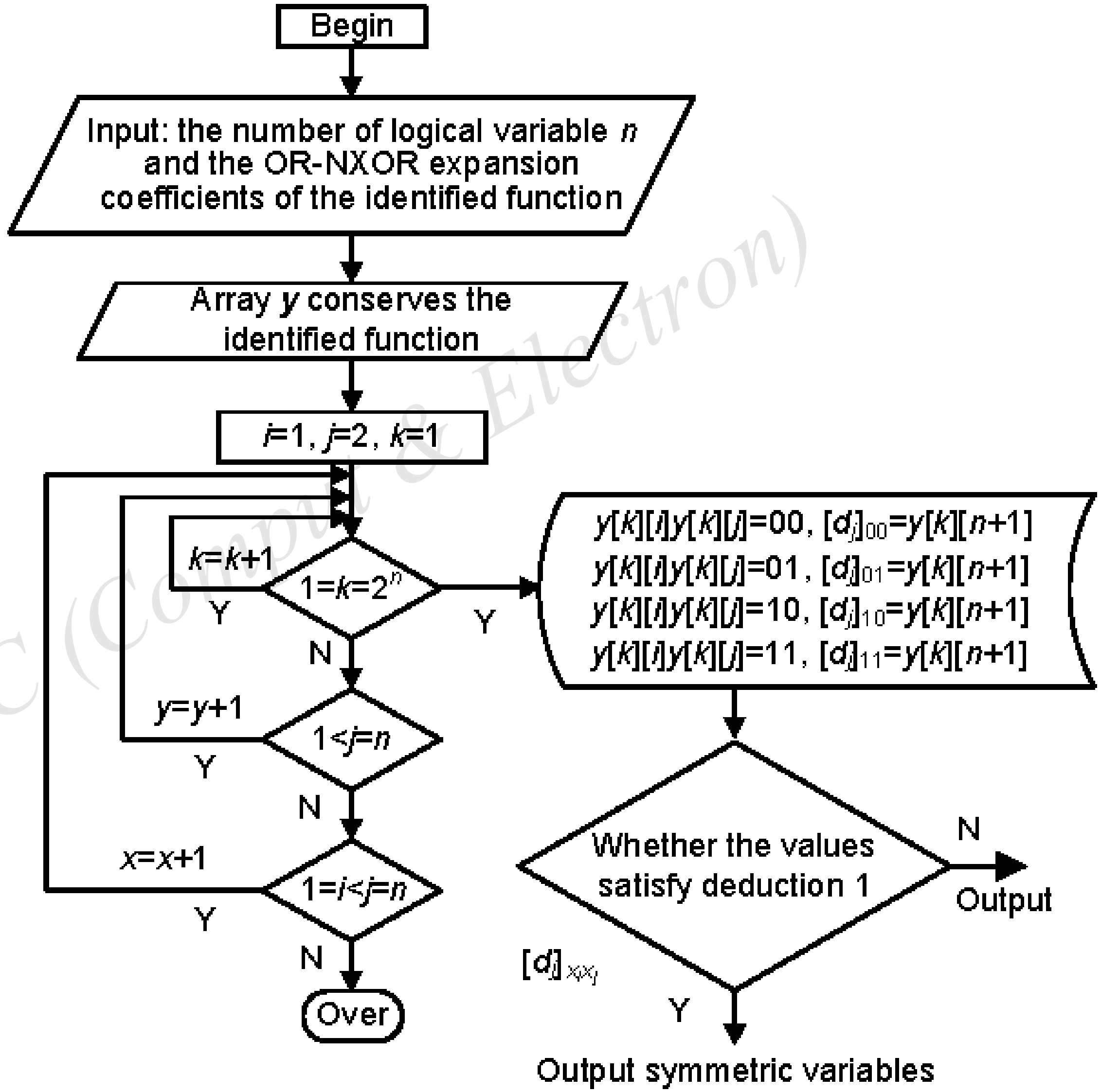
The satisfied conditions of $[d_j]_{x_i x_j}$

Symmetric conditions of logical variables

The satisfied conditions of $[d_j]_{x_i x_j}$	Existing symmetry type
$[d_j]_{00} \odot [d_j]_{01} \odot [d_j]_{10} = [1\,1\,...\,1]^T$	$E(x_i \vert x_j)$
$[d_j]_{01} \odot [d_j]_{10} = [1\,1\,...\,1]^T$	$N(x_i \vert x_j)$
$[d_j]_{01} = [1\,1\,...\,1]^T$	$S(x_i \vert x_j)$
$[d_j]_{00} \odot [d_j]_{01} = [1\,1\,...\,1]^T$	$S(x_i \vert \overline{x}_j)$
$[d_j]_{10} = [1\,1\,...\,1]^T$	$S(x_j \vert x_i)$
$[d_j]_{00} \odot [d_j]_{10} = [1\,1\,...\,1]^T$	$S(x_j \vert \overline{x}_i)$
$[d_j]_{00} \odot [d_j]_{01} \odot [d_j]_{10} = [1\,0\,...\,0]^T$	$CE(x_i \vert x_j)$
$[d_j]_{01} \odot [d_j]_{10} = [1\,0\,...\,0]^T$	$CN(x_i \vert x_j)$
$[d_j]_{01} = [1\,0\,...\,0]^T$	$CS(x_i \vert x_j)$
$[d_j]_{00} \odot [d_j]_{01} = [1\,0\,...\,0]^T$	$CS(x_i \vert \overline{x}_j)$
$[d_j]_{10} = [1\,0\,...\,0]^T$	$CS(x_j \vert x_i)$
$[d_j]_{00} \odot [d_j]_{10} = [1\,0\,...\,0]^T$	$CS(x_j \vert \overline{x}_i)$

Framework of our method (III)

The flow chart of the program implemented in C language



Flow chart of the program

Major results

The new algorithm is an optimal detection method in terms of applicability of the number of logical variables, detection type, and complexity of the identification process

Six methods for comparison

Detection method	Applicability of the number of logical variables	Number of detection types	Foundation work	Time for condition judgment
Graphic method	Usually $n \le 6$	12	Draw the decomposition charts of n-variable Boolean function	$6n(n-1)$
Spectral coefficients method	Usually $n \le 6$	12	Calculate the spectral coefficients, then separate the spectral coefficients into $n(n-1)/2$ groups	$6n(n-1)$
Tabular method	Usually $n \le 10$	2	Draw the 1-valued mintern table	$9n(n-1)/2$
CRM expansion coefficients method	Usually $n \le 6$	12	Separate the AND-XOR expansion coefficients into $n(n-1)/2$ groups	$6n(n-1)$
d_j-map method	Usually $n \le 6$	1	Draw the d_j-map	$n(n-1)/2$
Proposed method	No limit	12	Input the number of logical variable in the test function and the OR-NXOR expansion coefficients	$3n(n-1)$

n: the number of logical variables in the test function