

基于变粒度的大规模真值表快速知识约简

宋 波 闫继雄 陈泽华

(太原理工大学信息工程学院 太原 030024)

摘 要 在大规模逻辑电路的分析与设计,直接由大规模真值表得到最简逻辑函数表达式的过程往往比较复杂。针对此问题,提出了一种基于变粒度的大规模真值表快速知识约简算法。随着真值表的输入逻辑变量的粒度变化,通过引入标记矩阵和启发式算子,对大规模真值表进行知识约简,从而得到最简逻辑函数表达式。最后,通过实例分析并详述算法过程,且通过数据集进行对比实验,验证了该算法的快速性与有效性。

关键词 大规模真值表,逻辑函数,变粒度,知识约简

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.07.029

Rapid Knowledge Reduction of Large-scale Truth Table Based on Variable Granularity

SONG Bo YAN Ji-xiong CHEN Ze-hua

(College of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China)

Abstract In the analysis and design of large-scale logic circuits, the process of obtaining the simplest logic function expression from the large-scale truth table is often complicated. Aiming at this problem, a rapid knowledge reduction algorithm based on variable granularity for large-scale truth table was proposed in this paper. With the change of the granularity of the input logical variables, the simplest logical function expression is quickly acquired from the large-scale truth table by introducing the marker matrix and the heuristic operator. Then, the algorithm is described in detail through an example, and its correctness is proved mathematically. At last, the comparative experiments of data sets are carried out to further prove the rapidness and effectiveness of the proposed algorithm.

Keywords Large-scale truth table, Logic function, Variable granularity, Knowledge reduction

1 引言

在逻辑电路的分析与设计,大规模真值表的化简具有重要意义。传统的对真值表的化简方法主要包括公式法、卡诺图法^[1]、Q-M 算法^[4]及其改进算法^[5-9]、立方体算法^[2]等。其中,公式法化简需要灵活运用逻辑代数的公式与定理,且难以编程实现;卡诺图法化简的逻辑函数具有方便、直观、容易掌握等优点,但是带有试凑性,尤其当输入逻辑变量的个数大于 6 时,画卡诺图以及对圈的识别都会相当复杂;Q-M 算法要求原始描述必须是最小项形式,该算法虽易编程实现,但是其计算步骤繁琐且复杂度高^[3];立方体算法中的星积运算求质立方体的实质就是对 Q-M 算法的改进,尽管星积运算的效率要高于 Q-M 算法,但是其依然不尽如人意^[2]。Espresso 算法^[10]是一种启发式的二级逻辑最小化方法,被广泛应用于逻辑电路的优化。然而,传统的约简算法在处理带有无关项的真值表时,复杂性急剧增加。

形式概念分析是一种基于形式背景进行数据分析和知识获取的强有力工具^[11]。对于基于决策形式背景^[12]进行规则提取,李金海等^[13-14]已经取得了较好的研究成果。从形式上

看,真值表可以看成特殊的决策形式背景,但又有所不同。真值表中的 0 和 1 分别表示低电平和高电平,有明确的物理意义;而决策形式背景中的 0 和 1 分别表示对象具有该属性和对象不具有该属性。利用形式概念分析也可对真值表进行规则获取,但必须首先将真值表转化为决策形式背景,文献^[15]对此进行了初步探讨。

粗糙集理论是一种处理不确定性问题的数学工具,目前已被广泛应用于科学工程领域^[16-18]中。从本质上看,真值表是一个决策信息系统,因此真值表的化简问题可以转化为决策信息系统的最简决策规则的获取过程。对于决策信息系统,众多学者在基于粗糙集理论的决策规则提取上取得了较好的研究成果。代建华等^[19]提出了一种基于分类一致率的规则提取算法,相比于传统的 LEM2 算法,该算法提高了分类精度;张清华等^[20]通过属性粒度由粗到细的变化,直接从正面进行属性值约简,提高了规则提取的效率;陈德刚等^[21]和李天瑞等^[22]针对数据集不断更新的特点,分别提出了增量属性约简的方法,随后进行规则获取;梅长林等^[23]提出了一种基于置信度的规则提取算法,尽管该算法提高了计算效率,但其计算复杂度依然是指数级的。钱宇华等^[24]提出了一种

到稿日期:2017-05-30 返修日期:2017-08-04 本文受国家自然科学基金(61402319,61403273),山西省自然科学基金项目(2014021022-4)资助。

宋 波(1991-),男,硕士,CCF 会员,主要研究方向为粗糙集、形式概念分析;闫继雄(1992-),男,硕士,主要研究方向为粗糙集、形式概念分析;陈泽华(1974-),女,博士,教授,CCF 高级会员,主要研究方向为粒计算、工业大数据、智能信息处理及应用,E-mail: zehuachen@163.com (通信作者)。

基于决策分析的悲观多粒度粗糙集模型,并讨论了悲观多粒度与乐观多粒度之间的关系,极大地丰富了粗糙集的研究范围及使用领域。传统的决策粗糙集模型是基于单个成本矩阵提出的,没有考虑成本的多样性。杨习贝等^[25]针对上述模型的缺点,提出了一种 θ 决策理论粗糙集模型。近年来,针对多输入真值表化简的课题有一些新的方法被提出。文献^[26-27]分别针对多输入单输出、多输入多输出真值表提出并改进了基于粒矩阵的真值表快速约简算法,提高了真值表约简的效率,为真值表的约简提供了一种新的思路。

针对大规模真值表,为了减少文献^[26-27]中的矩阵存储规模及算法的搜索时间,本文通过定义标记矩阵以及启发式算子,提出了一种基于变粒度的大规模真值表快速知识约简算法,该算法进一步优化了真值表的化简过程,避免了大规模矩阵的相乘运算,极大地提高了算法的执行率。

2 预备知识

定义 1^[26-27] 真值表是用来表示逻辑信号输入状态与输出状态之间全部可能的表示形式,是输入与输出之间因果关系的数学表达形式,将这种形式用表格表示,简称真值表。当一个真值表中有多个输出时,称为多输入多输出真值表。

定义 2^[26-27] 若多输入真值表中每一个输入变量都以原变量或反变量的形式作为一个因子在乘积项中出现且仅出现一次,则称这样的乘积项为最小项;真值表中所有逻辑输出为 1 的最小项之和,称为最小项表达式;真值表化简后所有输出为 1 的输入变量乘积的和,称为逻辑函数的最简表达式。

定义 3^[26-27] 用一个四元组 $(U, A \cup Y, V, f)$ 定义真值表,其中 U 称为论域; A 表示真值表中的所有逻辑输入变量,称为条件属性,用 $|A|$ 表示逻辑输入变量的个数; Y 表示真值表中所有的输出变量,称为决策属性;映射 $f: U \rightarrow V_a$ 是信息函数,用来确定论域中每一个元素在属性 a 中的取值, $V_a = \{0, 1\}$ 表示输入变量 $a \in A$ 的值域。

定义 4 对于任意输入逻辑变量 $A' \subseteq A$,定义论域 U 上的不可分辨关系:

$$IND(A') = \{(x, y) \in U \times U \mid \forall a \in A', \text{ s. t. } f_a(x) = f_a(y)\} \quad (1)$$

其中, $IND(A')$ 表示论域 U 上的不可分辨关系; $U/IND(A')$ 表示论域 U 在不可分辨关系 $IND(A')$ 下的划分; $U/IND(Y)$ 表示论域 U 关于决策属性 Y 的划分。

性质 1 真值表中,必有 $U/IND(A) \subseteq U/IND(Y)$,真值表是一个一致决策信息系统;真值表中的每一行都可以看成一条逻辑规则。

定义 5 对于真值表 $(U, A \cup Y, V, f)$,不妨设 $U = \{u_1, u_2, \dots, u_l\}$, $1 \leq l \leq 2^{|A|}$,条件属性 $A' \subseteq A$ 对于论域 U 产生划分:

$$U/IND(A') = \{X_1, X_2, \dots, X_m\}, 1 \leq m \leq 2^{|A|} \quad (2)$$

其中,对于任意 $X_i \subseteq U/IND(A')$,定义 X_i 为信息粒,即:

$$X_i = \{x_{i1}, x_{i2}, \dots, x_{ik}\}, 1 \leq k \leq 2^{|A|} \quad (3)$$

对于任意 $x_{ij} \subseteq X_i$,有:

$$x_{ij} = \begin{cases} 1, & u_k \in x_i \\ 0, & u_k \notin x_i \end{cases}, 1 \leq k \leq l \quad (4)$$

定义 6 在单输出真值表 $(U, A \cup Y, V, f)$ 中,定义逻辑

输出为零的集合为 Y_0 ,条件属性粒矩阵为 $X_{m \times l}$,标记矩阵为 $X'_{m \times l}$:

$$Y_0 = \{x \mid x \in U, y \in Y, f(x, y) = 0\} \quad (5)$$

$$X_{m \times l} \triangleq X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1l} \\ x_{21} & x_{22} & \cdots & x_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{ml} \end{bmatrix} \quad (6)$$

$$X'_{m \times l} = \begin{cases} X'_i = 0_i, & \text{if } \exists j \in Y_0, \text{ s. t. } x_{ij} = 1 \\ X_i, & \text{otherwise} \end{cases} \quad (7)$$

集合 Y_0 表示真值表中逻辑输出为零对应的论域元素集合;条件属性粒矩阵 $X_{m \times l}$ 表示输入逻辑变量的不可分辨关系矩阵;标记矩阵 $X'_{m \times l}$ 表示对 $X_{m \times l}$ 的标记,令 $SZ(i) = 0$ 表示 $X'_{m \times l}$ 中第 i 行向量的和为零, $SZ(j) \neq 0$ 表示 $X'_{m \times l}$ 中第 j 行向量的和非零。

性质 2 若 $SZ(i) = 0$,则信息粒 X_i 不能够区分逻辑输出 Y 中的部分 1;若 $SZ(j) \neq 0$,则信息粒 X_j 能够区分逻辑输出 Y 中的部分 1。

证明:不妨设在标记矩阵中, $SZ(i) = 0$ 与 $SZ(j) \neq 0$ 同时存在,由定义 6 可知, $SZ(i) = 0$ 表示 $X'_{m \times l}$ 中行向量为 0 的行,即 $X_{m \times l}$ 中输入逻辑变量的部分条件等价类不能够正确区分逻辑输出中的 1; $SZ(j) \neq 0$ 对应于 $X_{m \times l}$ 中输入逻辑变量的部分等价类能够正确区分逻辑输出中的 1,即该部分等价类不会识别逻辑输出中的 0。

证毕。

定义 7 在标记矩阵 $X'_{m \times l}$ 中,定义标记矩阵的区分值为:

$$He = \sum_{k=1}^m \{k \mid SZ(k) \neq 0\} \quad (8)$$

其中,区分值 He 表示能够正确识别逻辑输出中的 1 的个数。

3 基于变粒度的大规模真值表快速约简算法

区别于文献^[27]中的算法(记作算法 2),本文中的变粒度算法(记作算法 1)是通过定义标记矩阵以及启发式算子,直接从原始的真值表中获取逻辑规则,从而避免大规模矩阵相乘,缩小算法的搜索空间。真值表的规模随输入逻辑变量个数的增加呈指数增加,由于不存在矩阵相乘运算,本文算法在处理大规模真值表时具有独特的优势。

3.1 算法描述

本节将详述本文算法的过程,如算法 1 所示。

算法 1 基于变粒度的大规模真值表快速约简算法

输入:多输入真值表 $(U, A \cup Y, V, f)$

输出:最简逻辑规则

Step 1 计算集合 Y_0 ,初始粒度 $w = 1$;

Step 2 计算标记矩阵 $X'_{m \times l}$;

Step 3 找到所有标记矩阵 $X'_{m \times l}$ 中的 $SZ(k) \neq 0$,并计算 He ;

Step 4 对 He 由大到小排序,记录所有新识别的论域元素并置于规则表中;

Step 5 判断已识别的论域元素是否覆盖所有的逻辑输出为 1 所对应的论域元素;

Step 6 若已覆盖,转 Step 8;

Step 7 若未覆盖,则 $w = w + 1 \leq |A|$,转 Step 2;

Step 8 输出所有最简逻辑规则,算法结束。

在执行算法 1 的过程中,若粗粒度(不妨设输入变量 A_0)存在逻辑规则,则在计算细粒度下的逻辑规则时,无需再计算

所有与 A_0 组成的多属性粒矩阵 $X_{m \times l}$ 与标记矩阵 $X'_{m \times l}$ (根据逻辑代数中的吸收律 $A_0 + A_0 B_0 = A_0$)。

3.2 实例分析

本节将通过例 1 来具体说明算法 1 的执行过程。

例 1 发光二极管一般用七段 (a, b, c, d, e, f, g) 字形显示,如图 1 所示。表 1 给出某种七段译码器的功能表^[8],输入 8421BCD 码,当输出值为 1 时,对应的字段点亮;当输出为 0 时,对应的字段熄灭,对应的字形显示如图 2 所示。

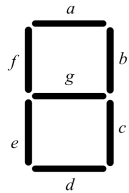


图 1 七段字形

Fig. 1 Seven-segment designation

表 1 七段译码器的真值表

Table 1 Truth table of seven-segment decoder

U	Logic input A				Logic output Y						
	A ₃	A ₂	A ₁	A ₀	a	b	c	d	e	f	g
1	0	0	0	0	1	1	1	1	1	1	0
2	0	0	0	1	0	1	1	0	0	0	0
3	0	0	1	0	1	1	0	1	1	0	1
4	0	0	1	1	1	1	1	1	0	0	1
5	0	1	0	0	0	1	1	0	0	1	1
6	0	1	0	1	1	0	1	1	0	1	1
7	0	1	1	0	1	0	1	1	1	1	1
8	0	1	1	1	1	1	1	0	0	0	0
9	1	0	0	0	1	1	1	1	1	1	1
10	1	0	0	1	1	1	1	1	0	1	1

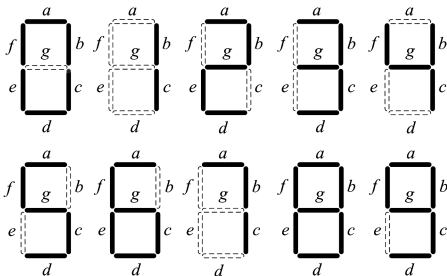


图 2 十进制数字

Fig. 2 Decimal number

本例仅以逻辑输出 a 字段为例(即 $Y = \{a\}$),来详细说明算法 1 的计算过程。如表 1 所列,论域 $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$,条件属性 $A = \{A_0, A_1, A_2, A_3\}, V = \{0, 1\}$ 。

1) 根据定义 6 中的式(5),求出 $Y = \{a\}$ 中输出为 0 的论域元素集合 $Y_0 = \{2, 5\}$,置初始粒度 $\omega = 1, G_1 = \{A_0, A_1, A_2, A_3\}$,设对应的 $X_{m \times l}$ 分别为 $X_0, X_1, X_2, X_3, X'_{m \times l}$ 为对应的标记矩阵,则有:

$$\begin{aligned}
 X_0 &= \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \Rightarrow X'_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 X_1 &= \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \Rightarrow X'_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \\
 X_2 &= \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \Rightarrow X'_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
 X_3 &= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \Rightarrow X'_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}
 \end{aligned}$$

在上述求取 X'_1 的过程中,首先根据式(5)求出 X_1 ,由于在矩阵 X_1 中,第 1 行中的 Y_0 列($\{Y_0 = 2, 5\}$,即第 2 列或者第 5 列)中存在 1,再根据式(7)将标记矩阵 X'_1 中的第 1 行全部置 0 ($X_{m \times l}$ 与 $X'_{m \times l}$ 的维数是相同的);由于在 X_1 矩阵中,第 2 行的 Y_0 列都不是 1,因此 X'_1 中的第 2 行与 X_1 中的第 2 行相等。

当 $\omega = 1$ 时,逻辑输出 a 的规则提取过程如表 2 所列。

表 2 $\omega = 1$ 的规则提取过程

Table 2 Rule extraction process when ω equals 1

$X'_{\omega=1}$	He	Universe elements	Rule
X'_1	4	{3, 4, 7, 8}	$r_1: A_1 = 1 \rightarrow a = 1$
X'_3	2	{9, 10}	$r_2: A_3 = 1 \rightarrow a = 1$
X'_2	0	\emptyset	\emptyset
X'_0	0	\emptyset	\emptyset

由表 2 可知,在单粒度时有规则输出,覆盖论域元素 {3, 4, 7, 8, 9, 10},但未覆盖所有的逻辑输出为 1 的论域元素 {1, 3, 4, 6, 7, 8, 9, 10},故需要计算粒度为 $\omega = 2$ 的情况。

2) 当 $\omega = 2$ 时, $G_2 = \{A_0 A_1, A_0 A_2, A_0 A_3, A_1 A_2, A_1 A_3, A_2 A_3\}$,易知在单粒度 A_1, A_3 下存在逻辑规则,故无需计算 $A_0 A_1, A_1 A_3, A_1 A_2, A_2 A_3$ 的属性组合(根据算法说明),因此只需计算 $G'_2 = \{A_0 A_2\}$ 的情况:

$$X_{02} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \Rightarrow X'_{02} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

当 $\omega = 2$ 时,逻辑输出 a 的规则提取过程如表 3 所列。

表 3 $\omega = 2$ 的规则提取过程

Table 3 Rule extraction process when ω equals 2

$X'_{\omega=2}$	He	Universe elements	Rule
X'_{02}	5	{1, 3, 6, 8, 9}	$r_3: A_0 A_2 = 00 \rightarrow a = 1$ $r_4: A_0 A_2 = 11 \rightarrow a = 1$

至此,逻辑输出 $Y = \{a\}$ 中的 1 已经被全部识别出,算法终止。 $Y = \{a\}$ 的逻辑函数表达式为: $Y_a = A_1 + A_3 + A_0 A_2 + \overline{A_0 A_2}$ 。

同理,可分别计算出其余的输出逻辑函数表达式: $Y_b = A_2 + A_0 A_1 + \overline{A_0 A_1}; Y_c = A_0 + \overline{A_1} + A_2 + A_3; Y_d = A_3 + \overline{A_0} A_1 + \overline{A_0 A_2} + A_1 \overline{A_2} + A_0 \overline{A_1} A_2; Y_e = \overline{A_0} A_1 + \overline{A_0 A_2}; Y_f = A_3 + \overline{A_0} A_1 + \overline{A_0 A_2} + \overline{A_1} A_2; Y_g = A_3 + \overline{A_0} A_2 + A_1 \overline{A_2} + \overline{A_1} A_2$ 。 $Y_a \sim Y_g$ 的逻辑表达式与文献[27]中的计算结果是一致的。

4 算法分析

本节首先通过正确性分析从理论上证明本文算法的正确性,然后通过复杂性分析来说明本文算法的复杂度,最后通过数据集测试来进一步说明本文算法相较于文献[27]的有效性与快速性。

4.1 正确性分析

文献[27]已证明算法 2 与卡诺图法等价,故下面只需证明本文算法(算法 1)与算法 2 等价即可。

算法 1 在执行过程中首先找出 Y_0 (即逻辑输出为 0 的论域元素),然后在粒度变化过程中根据条件属性粒矩阵 $X_{m \times l}$ 得到标记矩阵 $X'_{m \times l}$,因此标记矩阵中不存在使得逻辑输出为 0 的论域元素。最后通过性质 2 找到覆盖逻辑输出为 1 的论

域元素,其中定义启发式因子和逻辑代数的吸收律来加快算法的收敛速度。算法 2 中定义了条件粒矩阵 X 与决策粒矩阵 Y ,并寻找粒关系矩阵 $C=XY^T$ 中 $NE(i)=1$ (C 中第 i 行非零元素的个数),即寻找所有使得逻辑输出为 1 的论域元素,直至覆盖所有的逻辑输出为 1 的论域元素。因此说明了算法 1 与算法 2 等价。因为算法 1 与卡诺图法是等价的,所以证明了算法 1 的正确性。

4.2 复杂性分析

在多输入真值表($U, A \cup Y, V, f$)中,算法 2 计算了粒度为 w 的条件粒矩阵 X_w 与粒关系矩阵 C_w 的时间复杂度分别为 $O(|U| \cdot C_{|A|}^w)$ 和 $O(2 \cdot |U| \cdot C_{|U|}^w)$,决策粒矩阵 Y 的复杂度为 $O(|U|)$,规则提取的时间复杂度为 $O(|U| \cdot |A|)$,故整个算法 2 在最坏情况下的时间复杂度为 $O(|U|^2 \cdot |A| \cdot 2^{|A|})$;Q-M 算法^[4](记作算法 3)的时间复杂度为 $O(3^{|A|} \cdot |U|)$;Espresso 算法^[4](记作算法 4)的时间复杂度为 $O(|U|^2 \cdot |A|^2 \cdot 2^{|A|})$ 。算法 1 中计算粒度为 w 的条件粒矩阵 X_w 的时间复杂度为 $O(|U| \cdot C_{|A|}^w)$,标记矩阵 $X'_{m \times l}$ 的时间复杂度为 $O(C_{|A|}^w)$,集合 Y_0 的时间复杂度为 $O(|U|)$,规则提取的时

间复杂度为 $O(|U| \cdot |A|)$,故整个算法 1 在最坏情况下的时间复杂度为 $O(|U| \cdot |A| \cdot 2^{|A|})$ 。

通过算法 1、算法 2、算法 3 和算法 4 的时间复杂度分析易知,算法 1、算法 2 和算法 4 的时间复杂度是以 2 为底的指数级,而算法 3 的时间复杂度是以 3 为底的指数级,因此算法 1、算法 2 和算法 4 相对于算法 3 更具优势。但是本文算法(算法 1)的复杂度比算法 2、算法 4 的时间复杂度更低,且本文算法中定义了集合 Y_0 和标记矩阵,避免了大规模矩阵相乘,同时定义了启发式因子,进一步减少了算法的搜索时间。

4.3 实验测试

本节通过实验来进一步说明本文算法相对于算法 2、算法 3 和算法 4 的优越性。本次实验环境为:CPU 为 Intel(R) Core(TM) i5-7300HQ,主频为 2.5GHz,内存为 8GB,操作系统为 Windows 10,编程环境为 Matlab 2014a。

数据集采用随机生成的方法获取,由于真值表的知识约简在工业上要求正确率为 100%,本文已经证明了其正确性,因此本次实验仅对比真值表的约简时间,实验结果如表 4 所示,规则提取时间与输入逻辑变量的个数的关系如图 3 所示。

表 4 实验结果
Table 4 Experimental results

Data set	U	A	Y ₀	Running time/s			
				算法 1	算法 2	算法 3	算法 4
Data set1	426	9	43	0.389	0.812	2.606	0.729
Data set2	780	10	78	0.673	1.078	24.628	1.879
Data set3	2000	11	200	2.312	3.369	86.754	16.487
Data set4	2700	12	270	6.752	13.313	270.084	58.359
Data set5	3500	13	350	14.103	21.796	606.446	150.315
Data set6	5400	14	540	51.425	93.993	2314.926	378.639
Data set7	6700	15	670	108.366	179.712	8626.109	1446.828
Data set8	7800	16	780	206.701	310.537	10052.898	5649.658
Data set9	8400	17	840	566.044	949.138	29205.617	...
Data set10	9500	18	950	709.722	1054.793	32381.057	...
Data set11	12000	19	1200	2323.028	3667.742	107460.697	...
Data set12	15000	20	1500	4702.116	7366.171	-	...
Data set13	18000	21	1800	15646.442	27990.873	-	...
Data set14	21000	22	2100	18109.523	49185.634	-	...

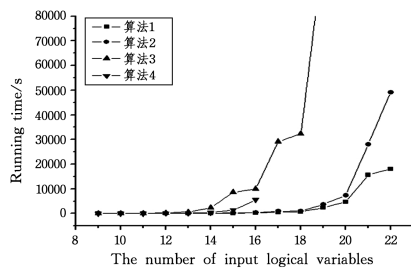


图 3 规则提取时间与输入逻辑变量个数的关系

Fig. 3 Relationship between rule extraction time and number of input logical variables

表 4 中,Data set1—Data set14 表示人工生成的数据集,且数据集中存在无关项;|U| 表示数据集的论域个数,是从 |A| 个逻辑变量的输入变量的全部可能输入($2^{|A|}$ 种可能输入)中随机选择的个数;|A| 表示输入逻辑变量个数;|Y₀| 表示逻辑输出中为 0 的个数;“-”表示内存溢出;“...”表示无法处理。

算法 4 在 Logic Friday^[28] 软件上实现,该逻辑优化软件支持的最大输入逻辑变量的个数为 16,采用 Logic Friday 中

的“Exact”获取最简逻辑规则。

由图 3 的实验对比结果可知,在保证算法正确性的同时,随着数据集的论域个数和逻辑输入变量个数的增加,算法 1—算法 4 的运行时间都有所增加,而且当输入逻辑变量的个数达到 19 时,传统的 Q-M 算法相对于算法 1、算法 2、算法 4 而言,规则获取的时间开销非常大,甚至存在内存溢出的情况。算法 4 尽管在一定程度上优于 Q-M 算法,但是时间开销依然很大,而算法 1 相对于其他算法在规则获取时间上更具优势。因此,对于大规模真值表,本文算法(算法 1)相对于现有算法是有效的且可行的。

结束语 针对数字逻辑电路中大规模真值表的知识约简问题,提出了一种基于变粒度的多变量真值表知识约简算法。该算法首先从真值表中求得逻辑输出为 0 的论域元素集合,然后根据粒度变化求出条件粒矩阵,并且通过定义标记矩阵与启发式算法减少算法的搜索时间,同时避免了大规模矩阵相乘,提取了所有的最简逻辑规则。本文从理论证明、实例分析和复杂性分析方面说明了本文算法的正确性和有效性,同时通过对比实验,验证了本文算法相对于文献[26-27]中算法的快速性与有效性。但本文算法也存在不足之处:随着输入

逻辑变量的增加,算法的复杂度会呈指数级上升。在下一步的研究工作中,可以考虑多变量的逻辑输出,将并行计算引入算法中,通过增加节点数来并行处理多变量输出,以进一步提高算法的执行效率。

参 考 文 献

- [1] 阎石. 数字电子技术基础(第五版)[M]. 北京:高等教育出版社, 2006:29-54.
- [2] 边计年,薛宏熙,苏明. 数字系统设计自动化(第二版)[M]. 北京:清华大学出版社,2005:221-226.
- [3] 刘宝琴,罗嵘,王德生. 数字电路与系统(第2版)[M]. 北京:清华大学出版社,2007:39-56.
- [4] MCCLUSKEY E J. Minimization of Boolean Functions[J]. Bell Labs Technical Journal,1958,35(6):1417-1444.
- [5] JAINT K, KUSHWAHA D S, MISRA A K. Optimization of the Quine-McCluskey Method for the Minimization of the Boolean Expressions[C]// International Conference on Autonomic and Autonomous Systems(ICAS 2008). DBLP,2008:165-168.
- [6] HUANG J B. Programing implementation of the Quine-McCluskey method for minimization of boolean expression[OL]. (2017-06-26) [2017-08-11]. <http://pdfs.semanticscholar.org/6cb2/c79bac6bf7ef341276490707b579760361d9.pdf>.
- [7] BANERJI S. Computer Simulation Codes for the Quine-McCluskey Method of Logic Minimization[OL]. (2017-06-26) [2017-08-11]. <http://arxiv.org/abs/1404.3349>.
- [8] MAJUMDER A, CHOWDHURY B, MONDAI A J, et al. Investigation on Quine McCluskey method: A decimal manipulation based novel approach for the minimization of Boolean function [C]// International Conference on Electronic Design, Computer Networks & Automated Verification. IEEE,2015:18-22.
- [9] HUANG X, WU N, ZHANG X. Quine-McCluskey Repair Technique for Evolutionary Design of Combinational Logic Circuits [J]. Lecture Notes in Engineering & Computer Science,2015, 2(1):674-678.
- [10] ASHMOUNIE F, RAMADAN R A, RASHED A A. Espresso for Rule Mining [J]. Procedia Computer Science,2014,32:596-603.
- [11] WILLER. Restructuring lattice theory:an approach based on hierarchies of concepts [M]//Ordered sets. Springer Netherlands,1982:445-470.
- [12] WEI L, QI J J, ZHANG W X. Attribute reduction theory of concept lattice based on decision formal contexts[J]. Science in China (Series E: Information Sciences),2008,38(2):195-208. (in Chinese)
魏玲,祁建军,张文修. 决策形式背景的概念格属性约简[J]. 中国科学(E辑:信息科学),2008,38(2):195-208.
- [13] LI J H, MEI C L, KUMAR C A, et al. On rule acquisition in decision formal contexts [J]. International Journal of Machine Learning & Cybernetics,2013,4(6):721-731.
- [14] LI J H, MEI C, LV Y. Knowledge reduction in decision formal contexts[J]. Knowledge-Based Systems,2011,24(5):709-715.
- [15] CHEN Z H, YAN J X, CAI J. Formal Concept Analysis Based Parallel Reduction Algorithm for MIMO Truth Table[J]. Journal of Electronics and Information Technology,2017,39(9):2259-2265. (in Chinese)
- 陈泽华,闫继雄,柴晶. 基于形式概念分析的多输入多输出真值表并行约简算法[J]. 电子与信息学报,2017,39(9):2259-2265.
- [16] LIANG J, WANG F, DANG C, et al. A Group Incremental Approach to Feature Selection Applying Rough Set Technique[J]. IEEE Transactions on Knowledge & Data Engineering,2014, 26(2):294-308.
- [17] ZHOUT, YANG X, FANG X, et al. Influencing factors research on CHF at low pressure, low flow conditions in narrow channel based on Rough Set Decision Model[J]. Annals of Nuclear Energy,2015,77:161-164.
- [18] SHIEHM D, YEH Y E, HUANG C L. Eliciting design knowledge from affective responses using rough sets and Kansei engineering system[J]. Journal of Ambient Intelligence and Humanized Computing,2016,7(1):107-120.
- [19] DAI J H, TIAN H W, WANG W T, et al. Decision rule mining using classification consistency rate[J]. Knowledge-Based Systems,2013,43(2):95-102.
- [20] HU S P, ZHANG Q H, YAO L Y. An Algorithm for Rule Extraction based on Changing Granularity [J]. Journal of Chongqing University of Posts and Tele-communications(Natural Science Edition),2016,28(6):856-862. (in Chinese)
胡帅鹏,张清华,姚龙洋. 一种变粒度的规则提取算法[J]. 重庆邮电大学学报(自然科学版),2016,28(6):856-862.
- [21] CHEN D G, YANG Y, DONG Z. An incremental algorithm for attribute reduction with variable precision rough sets[J]. Applied Soft Computing,2016,45:129-149.
- [22] JING Y G, LI T R, HUANG J F, et al. An incremental attribute reduction approach based on knowledge granularity under the attribute generalization [J]. International Journal of Approximate Reasoning,2016,76:80-95.
- [23] ZHANG X, MEI C L, CHEN D G, et al. Multi-confidence rule acquisition and confidence-preserved attribute reduction in interval-valued decision systems[J]. International Journal of Approximate Reasoning,2014,55(8):1787-1804.
- [24] QIAN Y H, LI S Y, LIANG J Y, et al. Pessimistic rough set based decisions: A multi-granulation fusion strategy[J]. Information Sciences,2014,264(6):196-210.
- [25] DOU H L, YANG X B, SONG X N, et al. Decision-theoretic rough set: A multi-cost strategy[J]. Knowledge-Based Systems,2016,91:71-83.
- [26] CHEN Z H, CAO C Q, XIE G. Granular Matrix Based Rapid Reduction Algorithm for Multivariable Truth Table[J]. Pattern Recognition and Artificial Intelligence,2013,26(8):745-750. (in Chinese)
陈泽华,曹长青,谢刚. 基于粒矩阵的多变量真值表快速约简算法[J]. 模式识别与人工智能,2013,26(8):745-750.
- [27] CHEN Z H, MA H. Granular Matrix Based Rapid Parallel Reduction Algorithm for MIMO Truth Table[J]. Journal of Electronics and Information Technology,2015,37(5):1260-1265. (in Chinese)
陈泽华,马贺. 基于粒矩阵的多输入多输出真值表快速并行约简算法[J]. 电子与信息学报,2015,37(5):1260-1265.
- [28] University of California. Free software for boolean logic optimization, analysis, and synthesis (Version 1. 1. 4) [EB/OL]. (2012-09-09) [2017-04-19]. <http://www.sontrak.com>.