国 防 科 技 大 学 学 报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第19卷第5期 1997年10月 Vol. 19 No. 5

多状态系统故障树质蕴含集的求取算法*

曾 亮
(国防科技大学自动控制系 长沙 410073)

摘 要 本章首先给出多状态系统状态空间的立方体表示方法，然后建立相应的运算规则。在这些规则的基础上，提出多状态系统故障树质蕴含集的求取算法。通过实例分析，说明了这种方法的可行性。

关键词 多状态系统，故障树，质蕴含集，立方体
分类号 TP202.1

The Algorithm To Find Prime Implication Sets in Multistate System Fault Tree

Zeng Liang
(Department of Automatic Control,NUDT,Changsha,410073)

Abstract In this paper,we study the algorithm to find prime implication sets in multistate system fault tree. The paper first introduces the cube expression method of sets,and sets up operation rules on it,then the algorithm to find prime implication sets in multistate system fault tree is given,finally the validation of the proposed method is proved by example.

Key words multistate system,fault tree,prime implication sets,cube

通常可靠性分析只考虑元件和系统的状态处于二状态(故障或成功)的情况，然而在实际生活中，元件和系统的状态往往表现为多种故障模式和不同故障程度，因此用二态系统理论不能解释多状态系统的可靠性问题，发展多状态系统可靠性理论成为必然。故障树是可靠性分析的一种重要分析方法，对于多状态系统分析也不例外。多状态系统故障树定性分析往往需要求取质蕴含集，这是由于故障树中存在互斥顶事件，引起顶事件发生的最

* 1997年3月11日收稿

小事件集合不再是最小割集，而是质蕴含集。通常，通过上行法、下行法或其它改进算法可以得到故障树的蕴含集，在蕴含集的基础上可以采取双取补法，合取法等方法求取质蕴含集，然而这些算法实现起来都比较复杂，对于多状态系统更是如此。本文结合合取法的基本思想和多值逻辑的有关原理，提出一种适合多状态系统故障树分析的质蕴含集求取算法，这种算法比较简单，而且容易实现，是一种比较可行的算法。

1　立方体表示与运算

对于多状态系统故障树来说，求取质蕴含集的关键在于集合运算。在介绍多状态系统故障树的质蕴含集求取算法以前，我们先介绍一种集合的立方体表示方法，并建立相应的运算规则。然后在此基础上提出多状态系统故障树的质蕴含集求取算法。

假定 n 个元件 $X_i, i=1,\cdots,n$，它们的状态空间为 $P_i=\{0,1,\cdots,p_i-1\}$，p_i 为它们的状态数。$P=\{p_i, i=1,\cdots,n\}$，$B=\{0,1\}$，故障树顶事件结构函数 $F: P_1\times P_2\times\cdots\times P_n\rightarrow B$ 为多输入单输出函数，$F=1$ 表示顶事件发生。

F 的每个定义域元素称为最小项，对于所有最小项的取值可用判定表表示。

设 S_i 为 P_i 的子集，定义

$$X_i^{S_i}=\begin{cases}1 & X_i\in S_i\\ 0 & X_i\notin S_i\end{cases}$$

对于 F 的定义域的子集（乘积项）

$$X_1^{S_1}X_2^{S_2}\cdots X_n^{S_n}$$

我们用立方体（二态向量）形式

$$C_1^0C_1^1\cdots C_1^{P_1-1}-C_2^0C_2^1\cdots C_2^{P_2-1}-\cdots-C_n^0C_n^1\cdots C_n^{P_n-1}\text{表示，}$$

其中
$$C_i^j=\begin{cases}1 & j\in S_i\\ 0 & j\notin S_i\end{cases}.$$

一个立方体有 n 个坐标，其中 $C_i^0C_i^1\cdots C_i^{P_i-1}$ 为第 i 个坐标。一个最小项在每个坐标都包含一个 1，且只包含一个 1. 一个坐标中只含 0 称为零坐标，相应的立方体称为空的，因为它不包含最小项。若某个坐标 i 包含 P_i 个 1（P_i 为相应元件的状态数），则称其为满坐标（它包含了 X_i 的所有状态）。一个立方体中满坐标的个数，称为立方体的维数。

例 1　设 $P=\{3,4,4\}$，乘积项

$$X_1^{0,1}X_2^1X_3^{0,2,3}$$

可用立方体 110－0100－1011 表示，它包含的最小项有{(0,1,0),(0,1,2),(0,1,3),(1,1,0),(1,1,2),(1,1,3)}.

定义 1　设 P,Q 为函数 F 的定义域的两个子集的立方体表示，$S=\mathrm{AND}(P,Q)$ 定义 P,Q 的立方体表示的同位二态与。P,Q 之间的距离定义为 S 的零坐标个数，用 Distance(P,Q) 表示。

例 2　以下为几个立方体 AND 运算与距离。

P	Q	AND(P,Q)	Distance
011—111—100	110—110—110	010—110—100	0
011—111—100	100—101—101	000—101—100	1
011—111—100	100—111—011	000—111—000	2

引理 1　P,Q 为函数 F 定义域的两个子集的立方体表示，当且仅当 P,Q 的距离为零时，存在最小项包含在 P 和 Q 中。

定义 2　一个立方体 P 覆盖立方体 Q，是指任意 Q 中的最小项都包含在 P 中。此时 Q 称为 P 的子集。

定义 3　设 $P=P_1-P_2-\cdots-P_n$,P_i 为 P 的第 i 个坐标,$Q=Q_1-Q_2-\cdots-Q_n$,P,Q 来自同一个函数 F 定义域的两个子集的立方体表示，若 P,Q 的距离为 1，且 j 为 AND(P,Q)零坐标位置，则 P,Q 的合取(Consensus)运算定义为

$$W = W_1 - W_2 - \cdots - W_n = \mathrm{Consensus}(P,Q)$$

其中

$$W_i = \begin{cases} \mathrm{AND}(P_i,Q_i) & i \neq j \\ \mathrm{OR}(P_i,Q_i) & i = j \end{cases}$$

2　质蕴含集的求取算法

在给出多状态系统故障树质蕴含集求取算法以前，我们先对多状态系统质蕴含集进行定义。为了讨论方便，我们认为质蕴含集是一些立方体的集合，它由满足下面条件的质蕴含项(立方体形式)构成：

(1)每个质蕴含项都是顶事件的一个蕴含，即该质蕴含项发生，必然引起顶事件的发生。

(2)每个质蕴含项都不能被质蕴含集中的其它质蕴含项所包含。

对于多状态系统故障树，我们可以通过上行法、下行法或其它改进算法得到故障树的蕴含集，在蕴含集的基础上，可以利用双取补法、合取法等方法求取质蕴含集。其中合取法由奎恩最早提出，它的基本思想是：若两个蕴含集 P 和 Q 仅含有一个相反的字母 X、X'，则消去 X、X' 后的布尔集 $P \cdot Q$ 也是一个质蕴含集，例如：

$$AB \cup B'C = AB \cup B'C \cup AC$$

最右端三项都是质蕴含集。又如

$$ABC \cup B'C = ABC \cup B'C \cup AC = B'C \cup AC$$

最右端二项是质蕴含集，ABC 不是。

下面利用前面定义和运算规则，结合合取法的原理，我们给出多状态系统故障树的质蕴含集求取算法。这个算法的基本思想是当两个蕴含集的立方体表示距离为 1 时，它们就能产生新的立方体，这时进行合取运算；距离为 0 时，它们含有共同的最小项，则将被蕴含集进行删减运算。下面是这个算法的具体步骤：

质蕴含集求取算法：

(1)将故障树的蕴含集用立方体表示(若蕴含集的某一项不出现，则用满坐标表示)，

并加上相应的序号，放置在一个列表 F 中，同时建立一个空的列表 $F1$. 选定 F 中的一个立方体 $C1$.

(2)选定 $C1$ 后面的没有连接标记 $C1$ 的立方体 $C2$，这里的连接标志 $C1$ 是指 $C1$ 的序号，给 $C2$ 加上连接标记是将 $C1$ 的序号放置在 $C2$ 的连接表中。

①判断 $C1$ 和 $C2$ 的距离。

②若 $C1$ 与 $C2$ 的距离为 0，且 $C3=\text{AND}(C1,C2)$

(a)若 $C3=C1$，将 $C1$ 删除。

(b)若 $C3=C2$，将 $C2$ 删除。

(c)若 $C1$ 和 $C2$ 没有被删除，给 $C1$,$C2$ 加上连接标记，即将各自序号放置在对方的连接表中。

(d)转(3)。

③若 $C1$ 和 $C2$ 的距离为 1，对 $C1$ 和 $C2$ 作合取运算。设 $C3=\text{Consensus}(C1,C2)$

(a)若 $C3$ 为 F 中其它立方体的子集，将 $C3$ 删除，转(3)。

(b)否则将 $C3$ 放置在 F 的最前面，并加上序号，将 F 中 $C3$ 的所有子集全部删除。

(c)若 $C1$ 没有被删除，将 $C1$,$C3$ 加上连接标记。

(d)若 $C2$ 没有被删除，将 $C2$,$C3$ 加上连接标记。

(e)将 $C3$ 作为新的 $C1$，转(2)。

④若 $C1$ 和 $C2$ 的距离大于 1，则转(3)。

(3)若 $C1$ 没有被删除，$C2$ 后面存在没有连接标记 $C1$ 的立方体，则选择一个作为新的 $C2$，转(2)中①。

(4)将 $C1$ 放入 $F1$ 中，并删除 F 中的 $C1$.

(5)选择 $C1$ 后面的立方体，作为新的 $C1$，转(2)。

(6)将最后得到的列表 $F1$ 转化为正事件的集合形式，并进行化简。

在上面的算法中首先通过距离判断来决定是否产生新的合取项。然后通过加上连接标记，来判断两个立方体是否被比较。

3 实例分析

我们通过实际的例子来说明文章中提出的质蕴含集求取算法。假设一个系统包含三个元件 $E1,E2,E3$，它们都有三个状态，用 $E11,E12,E13,E21,E22,E23,E31,E32,E33$ 表示相应的状态事件。我们已经得到系统某个故障树的蕴含集为$\{E11,E22,E32,\}$，$\{E12,E23\}$，$\{E21,E32\}$，$\{E12,E32\}$，$\{E13,E23,332\}$，分别用立方体 001－010－010，001－100－111，111－001－010，010－111－010，100－010－010 表示。下面用质蕴含集求取算法对质蕴含集求解，其基本过程如下：

(1)001－010－010
(2)001－100－111
(3)111－001－010
(4)010－111－010
(5)100－010－010

→

(6)001－110－010　(1,2)
(1)001－010－010　(6)delete
(2)001－100－111　(6)
(3)111－001－010
(4)010－111－010
(5)100－010－010

→

(7)001－111－010　(6,3)
(6)001－110－010　(2,7)delete
(2)001－100－111　(6)
(3)111－001－010　(7)
(4)010－111－010
(5)100－010－010

→

(8)011－111－010　(7,4)
(7)001－111－010　(3,2,8)delete
(2)001－100－111　(7)
(3)111－001－010　(7)
(4)010－111－010　(8)　delete
(5)100－010－010

→

(8)011－111－010　(2,3)
(2)001－100－111　(8)
(3)111－001－010　(8)
(5)100－010－010

→

(9)111－010－010　(8,5)
(8)011－111－010　(2,3,9)
(2)001－100－111　(8)
(3)111－001－010　(8)
(5)100－010－010　(9)　delete

→

(10)111－011－010　(9,3)
(9)111－010－010　(8,5,10) delete
(8)011－111－010　(2,3,9)
(2)001－100－111　(8)
(3)111－001－010　(8,10)　delete

→

(10)111－011－010　(8)　*F*1
(8)011－111－010　(10)
(2)001－100－111　()

→

(8)011－111－010　(2)　*F*1
(2)001－100－111　(8)　*F*1

上面是前 5 步的求解过程，最后得到列表 *F*1 中的立方体为 111－011－010，011－111－010，001－100－111，它们即为所要求取的质蕴含集。第 6 步，将立方体转化为不含补集的事件集合形式为{$E21$，$E32$}，{$E22$，$E32$}，{$E11$，$E32$}，{$E12$，$E23$}，{$E11$，$E23$}，这是为了和实际要求的表达形式相符合。

对于上述例子，如果采用双取补法，则第一次取补就需要产生 $3\times2\times2\times2\times3=72$ 个乘积项，其中还不包括元件状态事件补集带来的复杂度，实现起来比较麻烦。当蕴含集数目增加时，双取补法需要产生的乘积项成指数增加，求质蕴含集甚至无法实现。从上面的例子我们可以看到，本文提供的方法需要产生的项数并不很多，而且在每一步都要对乘积项进行删减，因此，最多需要产生的项数为元件状态的最大数和元件个数的乘积，当然这种情况还是不可能出现的。与双取补法比较，本文方法大大节约了空间；与一般的合取法比较，本文引入多值逻辑立方体的概念，使集合运算能够很好进行；而且本文利用连接标记，避免了算法进入循环。因此本文提供的方法在处理故障树的规模上和计算速度上都要比一般的质蕴含集求取算法优越，是一种比较简单可行的质蕴含集求取算法。

4　结束语

本文提出了多状态系统故障树的质蕴含集求取算法，这种算法不仅适合多状态系统定性分析，而且对于故障树的简化也是很有帮助的。对于多状态系统故障树分析来说，还存在其它许多问题。一些在理论上可行的方法，往往在实际过程中很难实现，这需要我们在实践过程中结合实际工程背景，提出适合工程应用的理论方法，这里我们不再详细论述。

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

2 廖炯生·多状态故障树分析·中国空间科学技术,1986,(2):29~37

3 Rudell R,Sangiovanni－Vincetelli A L. Multiple－Valued minimization for PLA optimization. Proceedings of the 17th International Symposium on Multiple－Valued Logic,May 1987,198~208

4 Dueck G W,Miller D M. RCM－MVL:A Recursive Consensus MVL Minimization Algorithm. IEEE Trans. On Computer 1990,136~143

5 Brayton R K,Hachtel G D,McMullen C T,Sangiovanni－Vincentelli,A L. Logic Minimization Algorithms for VLS I Synthesis. Kluwer Academic Publishers,Boston,1984

(责任编辑 张 静)