

# 基于图的 MC/DC 最小测试用例集快速生成算法<sup>①</sup>

黄孝伦

(重庆信息技术职业学院 软件学院, 重庆 404001)

**摘要:** 依针对 MC / DC 最小测试用例集生成效率的问题, 依据 MC / DC 准则, 以条件判定组合的复杂逻辑关系为研究对象, 采用将原始布尔表达式转化为语法树, 再将语法树转化为图, 分析图中节点对判定结果的影响的方法, 研究了基于图快速生成最小测试用例集的算法. 最后结合实例对该算法进行了验证, 并与其他算法进行了比较, 结果表明该算法在手动和自动生成最小测试用例集时具有实用性和优越性.

**关键词:** MC/DC; 测试用例; 语法树; 图

## Rapidly Generating Algorithm for Minimum Test Case Set on MC/DC Based on Graph

HUANG Xiao-Lun

(Chongqing Information Technology College, Chongqing 404001, China)

**Abstract:** For the problem of efficiency for Minimum Test Case Set on MC/DC, based on MC / DC criteria and the complex relationships of combination of conditions and decision, the paper turned a original boolean expression into a syntax tree, then turned the syntax tree into a graph. According to the effect on nodes of the graph to the decision, the paper proposed the graph-based fast algorithm to generate the smallest set of test cases. Finally, the paper verified and compared the algorithm with others, the results show that the algorithm has practicality and superiority.

**Key words:** MC/DC; test case; syntax tree; graph

MC/DC(即修正条件判定覆盖)标准是由欧美的航空/航天制造厂商和使用单位联合制定的“航空运输和装备系统软件认证标准”. 与完全节点式、完全边式、定义-应用对等测试标准相比较, MC/DC 标准是一种非常严格的测试标准, 已在国外的国防、航空航天领域得到广泛应用. 在国内, 针对 MC/DC 的测试用例集生成算法研究方面, 主要有最小真值表法和快速生成法. 最小真值表法将原始的布尔表达式转换为逆波兰式, 然后归约得到真值表, 最后化解真值表到最小测试集合<sup>[1]</sup>. 快速生成法算法不对布尔表达式进行任何转化, 而是根据布尔表达式与语法树对应特征, 绕过语法树直对布尔表达式进行归约得出最小化测试用例集<sup>[2]</sup>. 目前这些方法只适合处理由标准运算符 and、or 构成的非平凡布尔表达式, 而且难以避免手工基于 MC / DC 准则分析复杂的逻辑关系而造成的考虑不全问题.

本文结合快速生成法与布尔表达式的树型结构的思想, 设计了一种基于图结构快速生成测试用例集的算法, 该算法利用二叉树对布尔表达式进行转化, 然后将二叉树依据规则转换为图结构, 最后直接从图中快速得出最小化测试用例集.

### 1 MC/DC相关概念

MC/DC 准则要求: ① 判定中的每一个条件的所有可能结果至少出现一次; ② 每一个判定本身的所有结果也至少出现一次; ③ 每一个条件都能单独地影响判定的结果, 即在其他条件不变的情况下改变这个条件的值, 使得判定结果改变; ④ 每一个入口与出口点至少被执行一次. MC/DC 准则要求中④是现在编程规则一般要求, 关键的是前三项要求<sup>[3]</sup>.

条件表示不含有布尔操作符号的布尔表达式. 判定表由条件和零或者多个布尔操作符号组成的布尔表

<sup>①</sup> 收稿时间:2012-03-18;收到修改稿时间:2012-04-23

达式. 如果一个判定中的某一条件多次出现, 那么其每次出现均为不同的条件<sup>[4]</sup>.

定义 1. MC/DC 对是由两个不同的布尔取值项构成, 分别使得判断结果为“T”和“F”, 同时构成这两个布尔取值项中的条件有且只有一个条件的布尔取值发生变化, 而其他条件的布尔取值不变<sup>[4]</sup>. MC/DC 准则要求测试用例覆盖所有条件, 即任一条件存在测试用例对应其至少一个 MC/DC 对, 否则该测试用例集就是不完整的.

Chilenski 原则: 对于使用 MC/DC 方法的任何表达式当具有 N 个条件时, 一个测试集合最少含有 N+1 组元素<sup>[5]</sup>.

定义 2. 设 A 是一布尔表达式, p1, p2, ..., pn 为 A 中 n 个分量记为 pi(1 ≤ i ≤ n)称为 A 的条件, 当 pi 每个只出现一次则称 A 为非平凡布尔表达式<sup>[6]</sup>.

### 2 算法设计与验证

任何一个布尔表达式都可以归结为一个语法树, 布尔表达式的条件由树的叶节点表示, 布尔表达式中的 and、or、xor、not 等布尔运算符由非叶节点表示. 对于布尔表达式((p1 or p2) and p3 or p4) and (p5 or p6), 其语法树如图 1 所示.

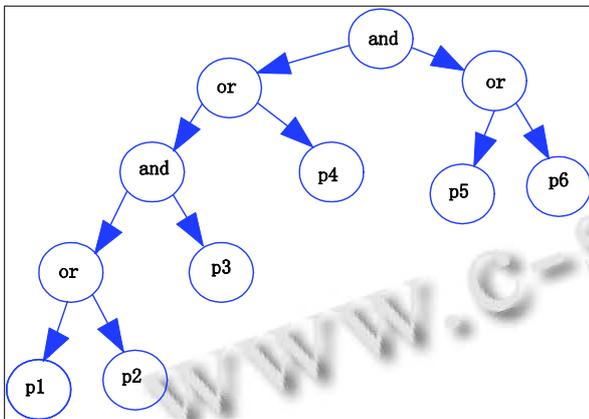


图 1 语法树

按照一定的转换规则, 我们可以将语法树转换为不带布尔运算符的图, 其转换规则如下: and 节点包含的子节点, 子节点之间由串联关系表示; or 节点包含的子节点, 子节点之间由并联关系表示; xor 节点同 or 节点处理规则; 对于 not 节点, 则将 not 运算符按布尔代数的基本定律和规则沿子树下移, 如遇 and 子节点, 则改为 or 子节点, 反之亦然, 直到叶子节点, 与叶子

节点结合, 视为一个特殊条件, 每次取值时要做一个非运算. 图 1 中的语法树经转换赋值后得到的图如图 2 所示, 其中顶点 vs、vi、ve 是为便于描述添加的虚顶点. 在图中, 以同一个顶点为弧头的邻接点称之为并关节点(如 p1, p2, p4), 否则为串关节点(如 p3). 每个顶点的默认值赋值规则为: 串关节点赋默认值 1, 在构成并联关系的节点中任选一个并关节点赋默认值 1, 其余赋默认值 0.

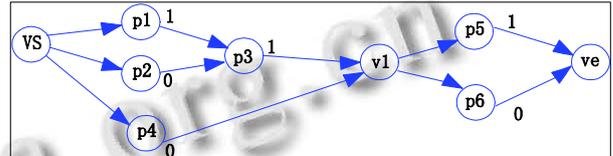


图 2 不带布尔运算符的图

为了说明算法原理, 我们可进一步, 将图 2 简化为一个电路图. 图 2 的简化形式如图 3 所示.

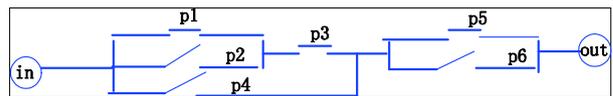


图 3 简化图

在图 3 中假设 out 端有输出表示为 1, 无输出表示为 0; 每个开关 pi 闭合表示为 1, 打开表示为 0. 按照上述默认值赋值规则赋值之后, 可以获得唯一一条通路 p1-p3-p5, 使得 out 端有输出, 我们把这条路径称之为图的关键路径, 下述生成测试用例集的方法称之为关键路径法. 由于隶属于关键路径的每个开关 pi 可以独立影响 out 端的结果, 其特征符合 MC/DC 准则要求, 因此, 根据关键路径法可以直接生成最小测试用例集. 由图 3 的当前状态, 我们可直接得到表 1 中第 1 组测试用例. 同时, 根据定义 1, 我们可以由关键路径获得与第 1 组测试用例构成 MC/DC 对的其他三组测试用例. 由于隶属于关键路径的节点取值的变化可以独立影响 out 端的结果, 因此直接把在第 1 组测试用例中隶属于关键路径的节点的值进行翻转, 就可得到表 1 中的 2、3、4 三组测试用例, 它们和第 1 组形成 MC/DC 对. 接下来, 再考虑其他的节点, 为了满足 MC/DC 准则要求, 我们只需改变关键路径, 纳入其他节点, 然后按照上述方式生成测试用例. 改变关键路径的规则: 以当前关键路径为对象(p1-p3-p5), 考虑每个节点是否有并关节点与之形成并联关系(如 p1 顶点存在并关节

点 p2), 如果有, 就进行替换(即 p1 打开, p2 关闭), 每次仅替换一个节点, 这样形成了新的关键路径(p2-p3-p5). 与之前的关键路径相比, 只改变了一个节点, 我们可获得这个节点独立影响判定结果的 MC/DC 对. 如此循环, 直到最后一个节点被替换. 根据 Chilenski 原则, 为了直接获得最小测试用例集, 在以新的关键路径生成测试用例时, 我们以第 1 组测试用例为对象, 进行转换. 如 p2 替换 p1 后, 我们直接将第 1 组测试用例相应位置的值(即 p1 和 p2 位置)进行翻转即可得到新的测试用例---表 1 中第 5 组测试用例. 由于只是对并联关系的两个节点进行了转换, 其他的节点没变, 在本文设计的算法中, p1、p2 的组合只有 00、10、01 三种取值情况, 在前面考虑 p1 的独立影响时, 已经把 00 这种组合考虑了, 因此在考虑 p2 的独立影响时, 就不需再考虑 00 这种组合情况, 也就是说第 4 组与第 2 组已形成了 MC/DC 对. 这样的话, 除了第一条关键路径外, 每次替换一个节点得到的新关键路径只需生成一组新的测试用例, 即可满足要求. 在本例中, 由第一条关键路径可获得表 1 中的 1、2、3、4 四组测试用例, 然后以 p2,p6,p4 进行替换生成新的关键路径时, 每替换一个节点, 得到一组测试用例. 最终得到的测试用例集如表 1 所示, 显然满足 Chilenski 原则. 在该用例集中, MC/DC 对(1, 2)表明条件 p1 的独立性影响, MC/DC 对(1, 3)表明条件 p3 的独立性影响, MC/DC 对(1, 4)表明条件 p5 的独立性影响, MC/DC 对(2, 5)表明条件 p2 的独立性影响, MC/DC 对(4, 6)表明条件 p6 的独立性影响, MC/DC 对(2, 7)表明条件 p4 的独立性影响.

表 1 最小测试用例集

组数	P1	P2	P3	P4	P5	P6	结果
1	1	0	1	0	1	0	1
2	0	0	1	0	1	0	0
3	1	0	0	0	1	0	0
4	1	0	1	0	0	0	0
5	0	1	1	0	1	0	1
6	1	0	1	0	0	1	1
7	0	0	1	1	1	0	1

基于上述的定义和规则, 该算法的具体流程如下:

Step1. 将布尔表达式归结为语法树.

Step2. 将语法树按规则转换为不带布尔运算符的图.

Step3. 生成最小测试用例集: 遍历图, 给图中每个顶点按规则赋默认值, 同时根据关键路径法直接生成最小测试用例集.

### 3 算法分析

根据上面算法的步骤和规则便可快速实现自动和手动生成 MC / DC 最小测试用例集的具体方法. 与最小真值表法和快速生成法相比, 本文提出的算法存在以下优势.

按照定义 1 的描述, 判定式(p1andp2)or(p1andp3)不是一个非平凡布尔表达式, 而根据判定的定义, 这个布尔表达式由 p1、p2、p1 和 p3 四个条件组成, 前后出现的两个 p1 条件是两个不同条件(即条件耦合), 这与 MC/DC 准则要求③中条件的独立影响相矛盾, 解决办法是把前后出现的两个 p1 条件作为一个条件<sup>[7]</sup>. 最小真值表法和快速生成法仅考虑了非平凡布尔表达式的情况, 不适合处理有条件耦合的布尔表达式. 本文提出的算法不仅能处理非平凡布尔表达式, 而且可以处理有一个条件耦合的布尔表达式. 采用本算法将语法树转换为不带布尔运算符的图时, 能合并的耦合条件尽量合并, 同时, 需要修改上述默认值赋值规则: 在构成并联关系的顶点中选择一个并联节点时, 优先选择是耦合条件的并联节点, 这样, 后面出现的是耦合条件的并联节点就已经赋有默认值, 然后再考虑该默认值是否为 1, 如果不为 1, 就选择其他的并联节点并赋默认值 1. 然后使用关键路径法生成最小测试用例集. 以布尔表达式(notp1orp2)and(p1andp3)为例, 其中 p1 为耦合条件, 简化电路图如图 4 所示(p1 前"|"符合表示 not 运算符), 最小测试用例集如表 2 所示.

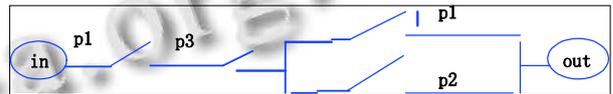


图 4 (notp1orp2)and(p1andp3)的电路图

表 2 最小测试用例集

组数	P1	P2	P3	结果
1	1	1	1	1
2	0	1	1	0
3	1	1	0	0
4	1	0	1	0

最小真值表法和快速生成法生成测试用例时其过程较复杂且不直观, 特别是复杂的布尔表达式, 难以避免手工基于 MC / DC 准则分析复杂的逻辑关系而造成的考虑不全问题, 不适合手动生成. 本文提出的

算法在手动生成最小测试用例集时非常简洁直观. 本算法只需将布尔表达式转换为图, 无论布尔表达式是否复杂, 图中包含了所有逻辑关系, 不存在考虑不全问题, 根据图就可直观快速得到最小测试用例集, 而且从布尔表达式转换为图的过程也适合手动生成且原理简单.

最小真值表和快速生成法只适合处理由标准运算符 and、or 构成的布尔表达式. 本文提出的算法能直接处理包含 xor 运算符的布尔表达式, 无需额外的工作. 如  $p5 \text{ xor } p6$  可用图 5 左边部分表示, 在本算法中两个并联节点的组合只需考虑 00、10、01 三种取值情况, 因此, xor 节点用 or 节点处理规则处理即可, 如图 5 所示.

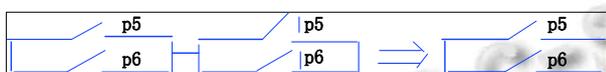


图 5 xor 等价图

在自动生成测试用例所需时间方面, 本算法与最小真值表法一样, 需要经过一次中间转换, 不同的是本算法可以从图的遍历中直接得到最小测试用例集, 而最小真值表法还需对遍历后得到的测试用例集, 进行化解, 所以, 本算法所花费的时间少于最小真值表法, 特别是在条件数目繁多、组合复杂的情况下. 与快速生成法相比, 本算法在进行图的遍历之前要经历一次语法树到图的转换, 而快速算法只用遍历条件判定组合一遍即可, 所以, 本算法所花费的时间多于快速算法. 三个算法比较的仿真实验结果如图 6 所示, 其中, 横坐标表示布尔表达式条件的个数, 纵坐标表示时间(单位: ms). 实验选取了 6 组由标准运算符构成的非平凡布尔表达式进行比较, 布尔表达式分别为  $p1 \text{ and } p2$ 、 $p1 \text{ and } p2 \text{ or } p3$ 、 $(p1 \text{ and } p2) \text{ or } (p3 \text{ and } p4)$ 、 $(p1 \text{ or } p2) \text{ and } ((p3 \text{ or } p4) \text{ and } p5)$ 、 $(p1 \text{ or } p2) \text{ and } ((p3 \text{ or } p4) \text{ and } (p5 \text{ or } p6))$ 、 $(p1 \text{ and } (p2 \text{ or } p3)) \text{ or } (p4 \text{ and } ((p5 \text{ or } p6) \text{ and } p7))$ .

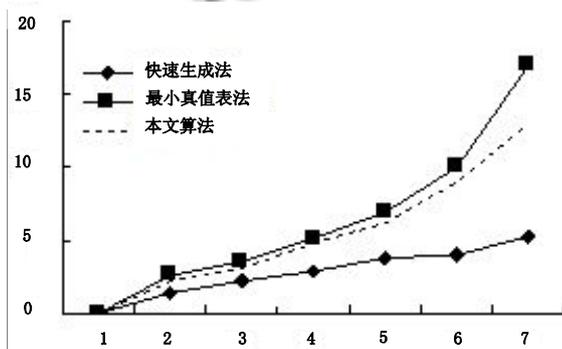


图 6 比较结果

#### 4 结语

MC / DC 测试错误的准确率非常高, 特别适合大型的并且测试要求非常精确的软件测试所用. 本文提出的算法通过对条件判定组合抽象语法树复杂关系的分析, 按照 MC / DC 准则要求针对性地生成最小测试用例集. 应用该算法生成测试用例的过程简洁直观, 可以避免手工基于 MC / DC 准则分析复杂的逻辑关系而造成的考虑不全问题, 适合手动和自动生成测试用例. 应用该算法不仅可以直接处理由运算符 and、or、not 和 xor 构成的非平凡布尔表达式, 而且可以处理有一个条件耦合的布尔表达式. 对于不是非平凡布尔表达式的其他情况还有待进一步深入研究.

#### 参考文献

- 1 朱晓波, 杨伟民, 叶芯. 更改条件/判定覆盖最小真值表生成算法及其应用. 上海理工大学学报, 2007, 29(1): 84-89.
- 2 段飞臂, 吴晓, 张凡, 等. MC / DC 最小测试用例集快速生成算法. 计算机工程, 2009, 35(17): 40-45.
- 3 曾爽, 张育平. 基于 MC / DC 生成测试用例方法研. 工业控制计算机, 2010, 23(2): 67-69.
- 4 潘丽丽. 软件测试用例集简化及其构建方法研究. 长沙: 湖南大学, 2008.
- 5 Chilenski JJ. An Investigation of three forms of the Modified Condition Decision Coverage (MCDC) Criterion. Computer Programming and Software, 2001, 18(4): 214-219.
- 6 Pressman RS. 软件工程实践者的研究方法. 北京: 机械工业出版社, 2002. 330.
- 7 张卫民, 陈宏敏. 几种特定判定形式的 MC/DC 评估分析. 飞行器测控学报, 2008, 27(4): 56-61.