

基于有穷自动机的路径覆盖测试方法

A Method for Path Coverage Testing Based on Deterministic Finite Automata Machine

毛红梅 (南昌航空大学 计算机学院 江西 南昌 330063)

摘要: 软件测试是当前软件生成重要的研究课题。从源程序流程图开始,提出了基于自动机的路径覆盖算法,该算法将源程序通过转换规则构造成与之等价的有穷自动机,应用自动机的特性处理有穷自动机中回边,并通过有限自动机的覆盖路径算法实现路径覆盖测试路径的选择。此方法实现路径覆盖的目标,与其它路径覆盖算法相比,覆盖路径生成数少。

关键词: 覆盖测试 回边 有穷自动机

路径覆盖方法很多,其中具有通过图的方法解决的有:(1)以林华辉老师等编写的“基于有向图的最小完全覆盖互操作测试序列生成算法”。此算法生成的测试集是一个能完全覆盖所有互操作转换的一个最小集^[1],并从理论上证明了用图解决路径覆盖的可能性;(2)以丁雪梅,伦立军编写的“基于 DDGRAPH 图的路径覆盖研究”。此算法通过由 DDGRAPH 图生成支配树和蕴含树,再形成非限制弧集,最后采用近似最小谓词覆盖策略生成路径子集合^[2]。

前两位专家提出了用图解决路径覆盖测试,实现了路径覆盖的目标,对路径覆盖的研究做出了很大的贡献,但是两位专家的算法中要求掌握的概念很多。本文在借鉴前面专家的基础上,用一个常用的自动机概念实现路径覆盖,此算法理解起来更直观,实现也容易。

1 相关概念

语句块,有两类结构称之为语句块,一类由简单的语句组成,比如一个变量声明,一个赋值语句,一个常量说明,以及一个跳转语句等等,特别地,把函数调用也看成是简单语句;另一类由复合语句组成,比如 if-else 结构的条件语句,while、repeat、do-while, for 结构的循环语句,switch-case 结构

的分支语句等。

由于复合语句往往是由大括号“{”和“}”括起来的一系列语句组成的,因此内在地,语句块必然也是一种可以嵌套的结构。即复合语句组成的语句块是由一系列简单语句组成的语句块组合而成的。这里有必要说明一下程序的层次,以标准 C 为例,正确的程序正文包含了嵌套的很多“{”和“}”,每一组对应的“{”和“}”括定程序的一个层次。

定义 1. 有向图 $G = \langle V, R \rangle$ 。其中 V 是结点集, $R = \{(x, y) | x, y \in V\}$ 是弧集。弧 $e = (T(e), H(e))$ 连接 G 中相邻的两个结点 $T(e)$ (称为弧头)和 $H(e)$ (称为弧尾)。对 V 中任意结点 v , $\text{indegree}(v)$ 是结点 v 的入度, $\text{outdegree}(v)$ 是结点 v 的出度。

定义 2. $G = \langle V, R, e_0, e_k \rangle$, $R = \{(x, y) | x, y \in V\}$ 是弧集。弧 $e = (T(e), H(e))$ 连接 G 中相邻的两个结点称 $T(e)$ 为 $H(e)$ 的前驱结点,称 $H(e)$ 为 $T(e)$ 的后继结点。

定义 3. $G = \langle V, R, e_0, e_k \rangle$, 其中 $e_0, e_k \in R$, e_0 是唯一的入口弧,它对 R 中其它任意弧均可达, e_k 是唯一的出口弧, R 中其它任意弧对 e_k 均可达,且对任意的 $v \in V, v \neq T(e_0), v \neq H(e_k)$, 满足 $\text{indegree}(v) + \text{outdegree}(v) = 2$, 而 $\text{indegree}(T(e_0)) = 0, \text{outdegree}(T(e_0)) = 1, \text{indegree}(H(e_k)) = 1, \text{outdegree}(H(e_k)) = 0$ 。

$egree(H(ek))=0$ 。

定义 4. 一个有穷自动机 M 是一个五元组： $M=(K, \Sigma, f, S, Z)$ ，其中， K 是一个有穷集，它的每个元素称为一个集合； Σ 是一个有穷字母表，它的每个元素称为一个输入符号； f 是转换函数，是 $K \times \Sigma$ 到 K 上的映像； $S \subseteq K$ 是唯一的一个初态； $Z \subseteq K$ 是一个终态集^[3]。

定义 5. 在自动机中，对任意两个结点 $m, n (m, n \in \{e_0, \dots, e_k\})$ ，如果从首结点 e_0 出发，到达任意 n 的任一通路都要经过 m ，则称 m 是 n 的必经结点。假设 n 到 m 有一条有向边，则称 n 到 m 是回边。

2 有限自动机的路径覆盖测试方法

根据测试的目标，选择一个有一定效果且开销较小的覆盖准则，或者是根据一定的标准，选择所有完整逻辑路径中的一个有限子集来进行测试，用最小的测试用例，发现程序中最多的错误。采用自动机的思想和理论改进，可以生成少的测试用例，而达到所有路径的覆盖。

2.1 控制流程图到自动机的转换规则(如表 1)

首先在程序的控制流图上，每条路径给一个标识，如 $e_1, e_2, e_3, \dots, e_k$ 等，其中 e_0 为唯一的入口弧， e_k 为最后一个唯一的出口弧^[5]。转换规则如表一所示，通过转化可以将控制流程图转换成只含有路径的自动机，根据自动机可以求出从 e_0 到 e_k 的 $n(n-1)$ 条路径。对于表中的规则 3，要使得等价需要引入一个新的中间节点。

2.2 基于有限自动机的覆盖路径算法

- (1) 输入：程序源代码程序。
- (2) 输出：含有从 e_0 到 e_k 的 $n(n-1)$ 条路径。
- (3) 根据程序块的概念将程序源代码转换成相应的有向图 $G=\langle V, R \rangle$ 。
- (4) 在每条弧上标注 $e_1, e_2, \dots, e_m (m$ 为弧的总条数)。

(5) 为了便于算法的实现，增加了两个空结点，其中一个 S 结点作为整个有向图的起始结点，其入度为 0，出度为 1，并与 e_0 进行相连； e_{n+1} 结点作为整个有向图的终止结点，其入度为 1，出度为 0，并

与 e_k 进行相连。如果，一个有向图的结点数为 n ，则经过变换总的结点数为 $n+2$ 。

(6) 根据表一的转换规则将有向图转换成以弧为结点的自动机 M 。

表 1 有向图到自动机的转换规则

	源程序有向图	有穷自动机
规则 1 (顺序)		
规则 2 (选择)		
规则 3 (选择)		
规则 4 (循环 1)		
规则 5 (循环 2)		

(7) 覆盖路径的寻找，寻找 e_0 到 e_k 的 $n(n-1)$ 条路径。

(a) 寻找回边。并将回边构成的结点抽象成一个子结点，重复做(a)，直到所有的回边均抽象成子结点。将自动机 M 转换成等价的 M' 。

(b) 在自动机 M' 上得到从 e_0 到 e_k 的 $n(n-1)$ 条独立路径。

2.3 自动机 M' 上得到从 e_0 到 e_k 的 $n(n-1)$ 条独立路径算法

- (1) 求图 G 的邻接矩阵。
- (2) 求结点的前趋结点，约定起始结点不求其前趋结点。

```

For ( i = 邻接矩阵的第 1 列 to 第 n 列)
For ( j = 邻接矩阵的第 1 行 to 第 n 行)
  If ( G[j][i] 不为无穷 and 第 j 行没有被删除)
  Then
    
```

```

    结点的入度不为 0，退出循环
  If (结点 i 为入度为 0 的结点)
  Then
    在前趋表的当前位置插入记录结点；
    For (k = 邻接矩阵的第 1 行 to 第 n 行)
      If (G[k][i]不为无穷 )
        Then 将结点 k 的相关信息记入当前
        记录结点的前趋集中
      删除第 i 行、第 i 列
      i = 第 1 个未被删除的列
      前趋表当前位置 = 前趋表当前位置+1
    Else
      i = i + 1
  (3) 计算起点(Souvex)到各结点的所有路径
  初始化路径表
  For(i=前趋表中的第 2 个结点 to 第 n 个结点)
  While ( i 的前趋结点尚未处理完毕)
    复制前趋结点到起点的路径到当前结点
    将前趋结点加入复制的路径尾端。
  
```

3 实例分析

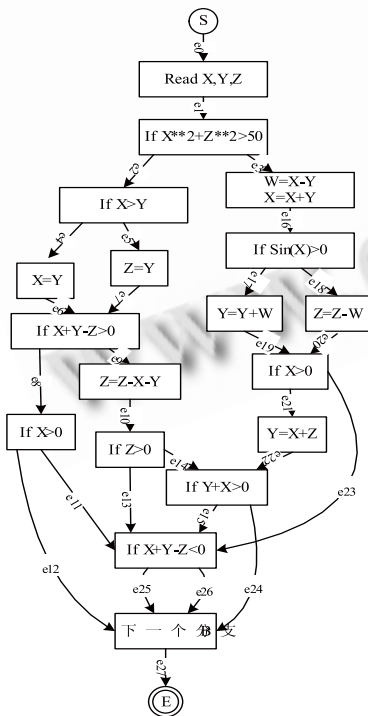


图 1 一个程序的流程图 G'

图 1^[4]是一个程序的流程图 G ,是经过程序块的转换后的有向图，S，E 是新增加的有向图结点，使得有向图只有唯一的入口结点和唯一的出口结点；e0，e27 是新增加的弧，e0 与 S 连接，e27 与 E 相连接。

将图 1 用表 1 的转换规则得到图 2，即以弧为结点的自动机。根据定义 5 可知图 2 中没有回边，因此，无须做回边的处理，但是必须根据表 1 的规则 3，引入 4 个中间节点，即根据标号的顺序，我们将其分别标识为，e7',e20',e22',e25'。

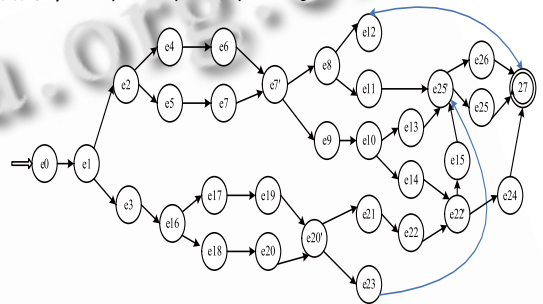


图 2 程序控制流图转换成相应的自动机

图 2 根据基于有限自动机的覆盖路径算法，通过实验可以得到 10 个独立路径，可以通过图 3 描述。覆盖路径 6 和覆盖路径 7 未标出，按照数字顺序进行描述。

- e7→e6→e4→e2→e1→e0 路径覆盖 1
- e5→e2→e1→e0 路径覆盖 2
- e11→e8→e7' →e7→e2→e1→e0 路径覆盖 3
- e13→e10→e9→e7→e2→e1→e0 路径覆盖 4
- e15→e22' {
 - e14→e10→e9→e7→e2→e1 路径覆盖 5
 - e19→e17→e16→e3→e1→e0 路径覆盖 6
 - e22→e21→e20' {
 - e20→e18→e16→e3→e1→e0 路径覆盖 7
- e23→e20' →e19→e17→e16→e3→e1→e0 路径覆盖 8
- e25→e25' →e11→e8→e7' →e7→e2→e1→e0 路径覆盖 9
- e24→e22' →e14→e10→e9→e7→e2→e1→e0 路径覆盖 10

图 3 流程图 G 的独立路径

路径覆盖 1 新增的前趋节点为{e27,e12,e8, e7',e6,e4,e2,e1,e0}，路径覆盖 2 新增的前趋节点为 {e7,e5}，路径覆盖 3 新增的前趋节点为{e26, e25', e11}，路径覆盖 4 新增的前趋节点为{e13,e10,e9}，

(下转第 102 页)

(上接第 105 页)

路径覆盖 5 新增的前趋节点为{e15,e22',e14}, 路径

覆盖 6 新增的前趋节点为{e22,e21,e20',e19,e17,

e16,e3}, 路径覆盖 7 新增的前趋节点为{e20,e18},

路径覆盖 8 新增的前趋节点为{e23}, 路径覆盖 9 新

增的前趋节点为{e25}, 路径覆盖 10 新增的前趋节点

为{e24}。

4 结语

测试序列的覆盖率是衡量测试序列自动生成算法优劣一个重要指标^[5]。该文基于有穷自动机的路径覆盖测试算法覆盖了所有的节点,并实现了每一条路径覆盖测试都有尽量多的新增路径节点,实现了路径覆盖数最小。该方法已成功的应用到作者开发的软件中作为软件测试的重要工具,如果一个程序中所有的路径的子集都已经测试过了,那么就可以认为程序中的每个语句都已经检验过了^[6]。

参考文献

- 1 林华辉,赵保华,屈玉贵.基于有向图的最小完全覆盖互操作测试序列生成算法.中国科学技术大学学报,2006,2:225 - 229.
- 2 丁雪梅,伦立军.基于 DDGRAPH 图的路径覆盖研究.微机发展,2004,3:29 - 31.
- 3 张素琴,吕映芝,蒋维杜,等.编译原理(第 3 版).北京:清华大学出版社,2005.
- 4 候芸,顾刚,高海昌,郭斌.一种路径覆盖自动生成的改进方法.计算机工程,2007,2:67 - 69.
- 5 LV Xin-yan, ZHOU Hao, ZHAO Bao-hua. A dynamic protocol conformance testing method. Journal of University of Science and Technology of China. 2005,35(3):338 - 404.
- 6 Bertolino A, Marre M. How many paths are needed for branch testing. The Journal of Systems and Software, 1996,35(2):95 - 106.