第 37 卷 第 2 期
Vol.37 No.2
计 算 机 工 程
Computer Engineering
2011 年 1 月
January 2011

• 软件技术与数据库 •　　文章编号：1000—3428(2011)02—0025—02　　文献标识码：A　　中图分类号：TP311

基于状态图的测试路径自动生成

李 鹏[1]，彭祥伟[1]，周 喜[1]，董 军[1]，代四广[2]

(1. 中国科学院新疆理化技术研究所，乌鲁木齐 830011；2. 武汉理工大学华夏学院，武汉 430223)

摘 要：提出一种基于状态图的测试路径自动生成算法，该算法从状态图的初始状态到终止状态进行遍历，可以得到一组路径，由该组路径生成的测试用例集可满足状态图的转移覆盖测试准则。根据循环复杂度，对路径集的总长度进行优化，从而减少测试用例数量。实验结果表明，该算法有效可行，在实际应用中能够取得良好的效果。

关键词：状态图；测试准则；转移路径

Automated Test Path Generation Based on State Diagram

LI Peng[1], PENG Xiang-wei[1], ZHOU Xi[1], DONG Jun[1], DAI Si-guang[2]

(1. Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China;
2. Huaxia College, Wuhan University of Technology, Wuhan 430223, China)

【Abstract】 This paper presents an algorithm, which travels the state diagram from the initial state to the ending state and finds all the transition paths. Test cases generation is based on test paths, which satisfies transition test criterion. It can optimize the length of the path, consequently reducing the number of test case based on cycle complexity. Experimental results indicate this algorithm is effective and feasible, and good effects are obtained in practical applications.

【Key words】 state diagram; test criterion; transition path

DOI: 10.3969/j.issn.1000-3428.2011.02.009

1 概述

软件测试是提高软件质量的重要手段[1]。1975 年，相关学者在研究软件测试能否保证软件的正确性时，提出了软件测试充分性这一概念，并指出软件测试的中心问题是度量软件测试充分性的测试准则。此后在不同的软件测试方法中，研究人员提出各种各样的测试准则[2-4]。

以状态图为模型生成测试用例是软件测试领域的研究热点之一，状态覆盖准则和转移覆盖准则是基于状态图软件测试的基本测试准则。此外，文献[5]提出转移对覆盖准则、全路径覆盖准则；文献[6]提出 ZOT 循环覆盖准则和全 ZOT 路径覆盖准则；文献[7]提出 N-转移覆盖准则和循环分类覆盖准则，并对上述准则进行公理化，为基于状态图的软件测试奠定了理论基础。不同的测试准则，发现缺陷的能力不同，其中，状态覆盖准则能检测状态是否可达，转移覆盖准则能检测状态图中的转移是否被正确实现，N-转移覆盖准则能有效测试转移序列。一般来说随着测试准则发现缺陷能力的增强，需要的测试用例也不断增加，对于全路径覆盖，即使一个简单的状态图也会产生大量的测试用例。考虑到测试成本及其效果，转移覆盖准则是众多状态图测试准则中应用最广泛的准则。

针对同一个覆盖准则，不同的测试用例生成算法产生的测试用例及数量有所不同，有效算法可减少生成测试用例的数量。本文提出一种算法，该算法从状态图的初始状态到终止状态进行遍历，可以得到一组路径，由该组路径生成的测试用例集可满足状态图的转移覆盖测试准则。根据循环复杂度，对路径集的总长度进行了优化，从而减少测试用例数量。

2 基于状态图的测试

状态图描述一个特定对象的所有可能状态以及由于各种事件的发生而引起的状态之间的转移。大多数面向对象技术都使用状态图来描述一个对象在其生命周期中的行为。状态图由状态和转移构成。初始状态表示生命周期的开始，它是转移的初始源，每个状态图有且必须有唯一的一个初始状态。终止状态表示对象生命周期的结束，一个状态图可能会有多个终止状态。转移使系统从一个源状态到达一个目标状态。

定义 1 状态图(State Diagram)的形式化定义

令 $SD=(S,\Sigma,\delta,s_0,F)$，其中，$S$ 为非空有限状态集合；Σ 为有限输入集，δ：$S\times\Sigma\to S$ 确定下一个状态，$s_0\in S$ 表示初始状态，$F\subset S$ 是终止状态的集合。

给定 $s,s'\in S,a\in\Sigma$，若 $\delta(s,a)=s'$，a 表示从 s 到 s' 的弧，简记为 $s\xrightarrow{a}s'$；$L\in\Sigma$ 代表输入，L^*是 L 上的序列集。如果 $\exists s_1,s_2,\cdots,s_k,s_{k+!}\in S,k\geq 0,x=u_1,u_2,\cdots,u_k\in L^*$，则 $s_1\xrightarrow{u_1}s_2,s_2\xrightarrow{u_2}s_3,\cdots,s_k\xrightarrow{u_k}s_{k+1}$ 简写为 $s_1\xrightarrow{x}s_{k+1}$，，称 x 为从 s_1 到 s_{k+1} 的路径。在路径 x 中 $s_i,s_j\in S$，若 $s_i\neq s_j$，则称 x 为基本路径；若 $s_i=s_j$，则称 x 为包含循环的路径，其中，若 $s_i\xrightarrow{x_i}s_j$，称 x_i 为从 s_i 到 s_j 的循环路径。

定义 2(回退转移) 对每条循环路径的每条转移 t 的前后节点 s_i、s_j，搜索所有的基本路径，如果 s_i、s_j 出现在某条

基金项目：新疆维吾尔自治区重大专项基金资助项目(200732143-1)

作者简介：李 鹏(1981－)，男，硕士研究生，主研方向：软件测试；彭祥伟，硕士研究生；周 喜，副教授；董 军，助理研究员；代四广，助教

收稿日期：2010-06-10　**E-mail：**xuehubake@163.com

基本路径中，并且 $j < i$，则称转移 t 为回退转移。

基于状态图的测试，基本原理是：给定一个状态图 A 和一个 A 的被测实现 P，检测 P 是否是 A 的一个正确实现。要根据一定的输入序列来判断 A 和 P 的行为是否相同。

3 基于状态图的测试路径自动生成

3.1 测试覆盖准则

测试准则不仅可以用于选择测试数据，生成测试集，还可以用来作为测试停止标准。当软件测试达到预定的测试需求时，测试即停止。实际构造测试用例时，都要以一定的测试准则为基础。基于状态图的测试主要有以下 5 种覆盖准则：

(1)状态覆盖：测试用例集应该使状态图中的每一个状态至少被访问一次。

(2)转移覆盖：测试用例集应该使状态图中的每一个转移至少被激活一次。

(3)转移对覆盖：测试用例集应包含状态图中的每一对邻近转移 $S_i : S_j$ 和 $S_j : S_k$ 的测试。

(4)ZOT 循环覆盖准则：测试用例集应该包含状态图中的每一个循环分别执行 0 次、1 次、2 次的测试。

(5)全 ZOT 路径覆盖准则：当状态图中的每一个循环分别执行 0 次、1 次、2 次，而循环外的独立路径要求至少被访问一次时，测试用例集应该包含状态图中从初始状态开始到终止状态结束的全部路径的测试。

3.2 测试路径的产生

在基于状态图的测试中，每条测试路径把被测实现从初始状态态转化到终止状态。生成测试路径要考虑 2 个问题：

(1)生成的测试路径要满足预定的覆盖准则；

(2)生成的测试路径的长度要尽可能短，以提高测试效率，降低测试成本。本文中测试路径生成采用转移路径覆盖准则。

定义 3 转移路径覆盖：覆盖从初始状态到终止状态的每一条独立转移路径。

转移路径覆盖准则满足了状态覆盖准则和转移覆盖准则，并且使状态图中的循环分别执行了 0 次和 1 次，保证了测试的相对完备性，并且最后生成的测试路径要保证每条路径中至少有一条转移是其他路径未使用的。

3.3 测试路径自动生成算法

为达到测试的有效性并减少冗余，在寻找状态图的转移路径时，要求在遍历过程之中必须保证每一个状态和每一条转移被覆盖，而且循环路径最多执行一次。算法总体流程如图 1 所示。

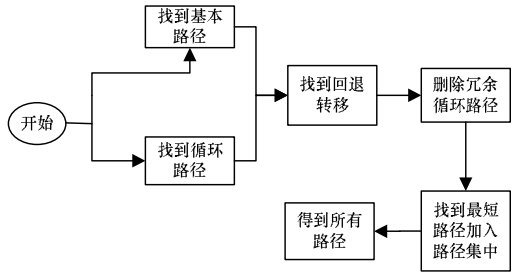

图 1 基于状态图的测试路径自动生成算法流程

首先找到基本路径节点序列和循环路径节点序列；然后根据节点序列中节点的次序和名称，从转移的集合中找到相邻两节点对应的转移，组成基本路径集和循环路径集；然后找到回退转移，删除冗余的循环路径；对每条循环路径查找已生成的路径，从中寻找一条经过该循环路径首节点的最短路径 P，用该循环路径替代路径 P 中的对应节点形成一条新的路径添加到路径集中；生成所有满足覆盖准则的路径集。

算法 1

输入 状态图

输出 转移路径

Step1 从根节点开始搜索，逐个搜索其子节点。

Step2 如果当前节点 S 出现在已搜索路径中，得到一个循环路径节点序列，该路径搜索结束；若当前节点 S 没有子节点，则 S 为终点，得到基本路径节点序列，该路径搜索结束；若当前节点 S 有子节点，则逐个搜索其子节点，重复 Step2。

Step3 若处理完基本路径节点集中的所有节点序列，则跳转到 Step6。

Step4 从基本路径节点序列集中取出一个节点序列。

Step5 根据该序列中各节点的次序和名称，从转移的集合中找到相邻两节点对应的转移，组成基本路径放入基本路径集中，跳转到 Step3。

Step6 若处理完循环路径节点序列集中的所有节点序列，则跳转到 Step9。

Step7 从循环路径节点序列集中取出一个循环路径节点序列。

Step8 根据该序列中各节点的次序和名称，从转移的集合中找到相邻两节点对应的转移，组成循环路径放入循环路径集中，跳转到 Step6。

Step9 找到回退转移。

Step10 检查所有的循环路径，删除冗余的循环路径，使经过同一回退转移的循环路径只保留一条，同时保留没有经过回退转移的循环路径。

Step11 对每条循环路径查找已生成的路径，从中寻找一条经过该循环路径首节点的最短路径 P，用该循环路径替代路径 P 中的对应节点形成一条新的路径添加到路径集中。

Step12 生成所有路径。

3.4 实验分析

为了验证算法的有效性，本文以图 2 为例对算法进行验证。

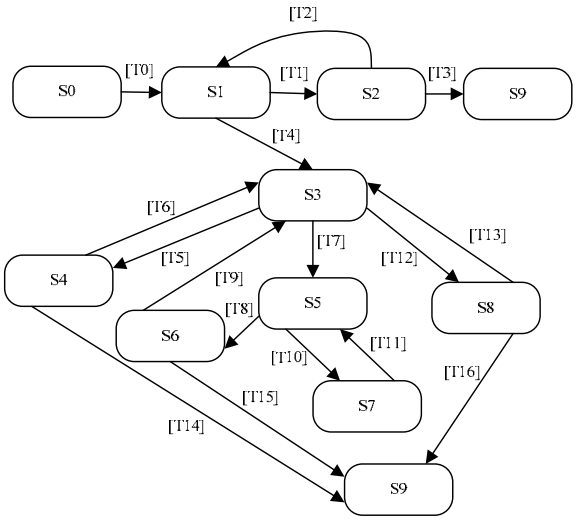

图 2 状态图

使用算法 1 可以得到以下测试路径：

1:T0->T1->T3

2:T0->T4->T5->T14

3:T0->T4->T7->T8->T15

(下转第 29 页)