

一种基于蜕变关系的测试与失效测试用例定位模型

惠战伟¹ 黄松^{1,2} 张婷婷¹ 刘剑豪³

(解放军理工大学指挥信息系统学院软件测评中心 南京 210007)¹

(解放军军事训练软件测评中心 南京 210007)² (空军预警学院 武汉 430019)³

摘要 针对传统的蜕变测试模型 MTM 存在的局限性,提出了一种基于蜕变关系的测试模型 MRTM。首先通过对比分析,指出了 MRTM 的适用范围等特点;其次,针对 MTM 和 MRTM 都面临的失效测试用例难以确定的难题,提出了一种基于可疑度计算的蜕变测试失效测试用例定位方法 FTCL-MT。FTCL-MT 作为对已有测试模型的补充,能够在蜕变关系不满足的情况下实现精确定位失效测试用例,从而能够为现有的故障定位技术提供支持。最后,通过实验验证了 FTCL-MT 方法的有效性。

关键词 蜕变关系,测试模型,蜕变测试,失效测试用例

中图分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.10.010

Testing and Invalid Testing Case Localization Model Based on Metamorphic Relation

HUI Zhan-wei¹ HUANG Song^{1,2} ZHANG Ting-ting¹ LIU Jian-hao³

(Software Testing and Evaluation Centre, College of Command Information System, PLA University
of Science and Technology, Nanjing 210007, China)¹

(PLA Military Training Software Testing and Evaluation Center, Nanjing 210007, China)²

(Air Force Early Warning Academy, Wuhan 430019, China)³

Abstract Aiming at the limitations of traditional metamorphic testing model MTM, this paper proposed a testing model MRTM based on metamorphic relation. Through comparative analysis, firstly, characteristics such as scope of its application are pointed out. Secondly, in view of the problem that the invalid testing case is hard to determine that both MTM and MRTM are facing, a failed test case localization method of metamorphic testing (FTCL-MT) based on dubiety computation was proposed. As a supplement to the existing testing model, FTCL-MT is helpful to achieve precise positioning of invalid test cases in the case that CMR is not satisfied, so that it can provide support for existing fault location technologies. Finally, experiments show the effectiveness of FTCL-MT.

Keywords Metamorphic relation, Test model, Metamorphic testing, Invalid testing case

1 引言

软件测试一直都是软件正确性验证中应用最广泛、操作性最强的方法,它能够建立人们对程序正确性的信心。据统计,在软件的整个开发过程中,测试部分的耗费能占到 50% 以上^[1]。然而在对地形量算功能进行测试时,测试人员能够采用的方法往往是传统的特殊值检验或人工估算等,用这类方法验证软件正确性时,由于难以涉及到程序核心且结果不精确,经常导致测试效果不理想。这种问题在软件测试中又称为测试判定(Test Oracle)难题^[2]。理想的测试判定能够对测试结果是否正确给出正确的 pass/fail 判定^[3]。然而不幸的是,理想的 oracle 并不能直接得到,或者获取的成本远远超出了软件本身的价值。这种测试判定不明确的软件通常被称

为“不可测试程序(Non-testable Programs)”^[4]。

在其它测试技术构造的测试用例(在蜕变测试中称为原始测试用例:Original Test Case)存在测试判定难题时,蜕变测试(Metamorphic Testing, MT)技术^[5,6]就可以通过蜕变关系提供测试判定。首先,需要测试人员针对被测件构造蜕变关系(Metamorphic Relation, MR),即当程序输入满足一定的关系时,其相应的输出也就必须满足某种关系。然后,根据蜕变关系和原始测试用例,生成其它的测试用例(又称为附加测试用例,Follow-up Test Case)。最后,分别运行这两种测试用例并判定对应的输出是否满足蜕变关系。传统上,蜕变测试还是被定位为一种辅助测试技术,并不能像其它测试技术那样独立开展。这种局限性严重制约了蜕变测试的应用领域。本文将现有的这种蜕变测试对应的测试模型称为传统蜕

到稿日期:2015-08-08 返修日期:2015-12-13 本文受国家高技术研究发展计划“863”项目(2009AA01Z402),江苏省自然科学基金项目(BK20141072, BK2012059, BK2012060)资助。

惠战伟(1983-),男,博士生,工程师,主要研究方向为蜕变测试技术、软件安全性测试, E-mail: hzw_1983821@163.com; 黄松(1970-),男,教授,博士生导师,主要研究方向为军用软件测试与质量评估技术; 张婷婷(1981-),女,博士生,讲师,主要研究方向为软件建模; 刘剑豪(1978-),男,博士,讲师,主要研究方向为软件测试技术。

变测试模型。针对传统蜕变测试模型 MTM 存在的局限性,提出了一种基于蜕变关系的测试模型 MRTM。

2 传统蜕变测试模型局限性分析

当前蜕变测试普遍采用的测试模型^[6]如图 1 所示。按照测试的步骤可以将其划分为如图 1 所示的 5 个阶段。由图可见,蜕变测试主要包括以下几步:1)基于其它的软件测试技术生成原始测试用例;2)当这类测试用例面临着测试判定难题时,测试人员针对被测软件,设计蜕变关系集;3)根据该蜕变关系的输入关系生成原始测试用例对应的附加测试用例;4)执行程序并验证测试输出是否满足蜕变关系的输出关系;5)选择下一组测试用例并返回步骤 3)。

根据该模型可以看到,这种测试模型主要存在两个不确定性问题。

(1)蜕变测试依赖于其它的软件测试技术生成原始测试用例,例如结构覆盖测试用例生成技术^[8-10]、随机测试用例生成技术^[7]、基于规格的测试用例生成技术^[8,9]等;

(2)蜕变关系的生成需要测试人员根据对被测件的了解程度来人工构造。

可以说,这两个问题直接限制了蜕变测试的长远发展,特别是第一个不确定性,即蜕变测试对其它测试技术的依赖性直接影响了蜕变测试的应用范围,使得蜕变测试一直只能作为一种辅助测试方法;另一个问题则制约了蜕变测试的自动化程度,众所周知,自动化程度是一种测试方法成熟的重要标志^[10]。

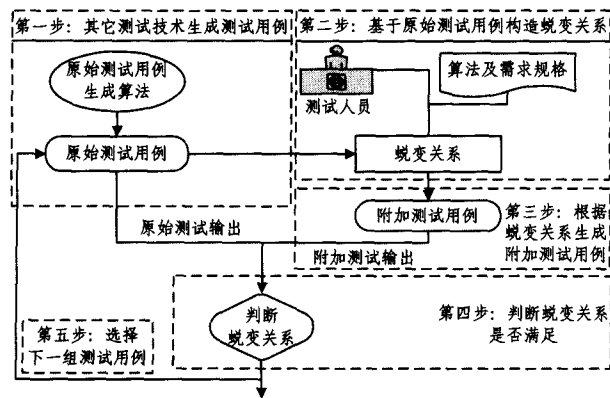


图 1 传统蜕变测试模型 MTM

3 基于蜕变关系的测试模型 MRTM

3.1 测试模型的引入

对于第一个不确定性问题,其实质就是因为缺少适用于蜕变测试自身的测试用例生成技术。虽然蜕变测试的初衷只是通过蜕变关系提供测试判定,基于蜕变关系可以为蜕变测试提供一种有效的测试用例生成方法,这样,测试人员在测试过程中就可以集中精力构造蜕变关系,而不需要再选择其它测试方法生成原始测试用例。

对于第二个不确定性问题,主要有两方面的原因:一方面,蜕变关系的描述缺少规范一致的描述模型;另一方面,蜕变关系的生成大多依赖于被测件。基于该模型给出了现有的两种蜕变关系构造方面的精确描述。这样,针对被测件,测试

人员只需要构造少数蜕变关系,然后基于复合蜕变关系和组合蜕变关系的定义构造更多的蜕变关系。

3.2 MRTM

通过对这两个问题的研究,显然图 1 所示的测试模型已经不能适用于当前的蜕变测试的发展。基于此提出新的测试模型,如图 2 所示。为了后续比较两种模型的不同,称图 1 为蜕变测试模型(Metamorphic Testing Model, MTM),图 2 为基于蜕变关系的测试模型(Metamorphic Relation based Testing Model, MRTM)。

由图 2 可见,MRTM 主要包括以下几步:1)针对被测软件,构造蜕变关系 $MR = \{MR_1, MR_2, \dots, MR_m\}$, m 表示蜕变关系的个数;2)选择一个蜕变关系,并基于此生成蜕变测试用例集(OTC, FTC);3)执行蜕变测试用例并判定是否满足蜕变关系;4)选择下一组蜕变测试用例继续验证蜕变关系,遍历所有测试用例;5)返回 1),选择下一个蜕变关系。

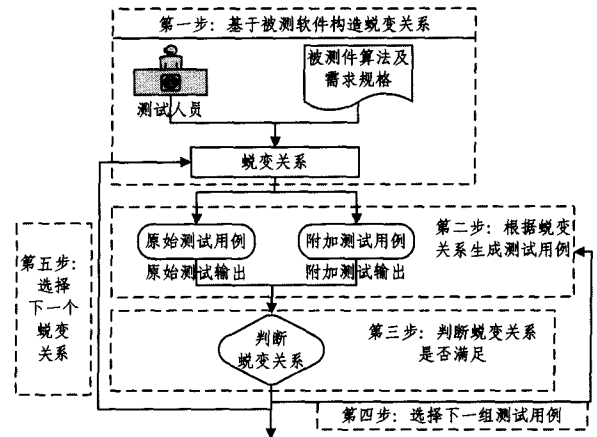


图 2 基于蜕变关系的测试模型 MRTM

3.3 MRTM 与 MTM 对比分析

根据 MTM 与 MRTM 的基本过程可以看到,MRTM 与 MTM 的差异主要体现在表 1 所列的 4 个方面。

(1)目的不同:MTM 的目的是通过蜕变关系为已有的测试用例提供测试判定;而 MRTM 并不局限于此,它的主要目的是为被测件提供测试判定已知的测试用例,存在测试判定只是 MRTM 生成测试用例需要满足的条件。

(2)适用场合不同:通常只有在已经存在的测试用例测试判定未知的条件下,才需要 MTM,如果不存在测试用例或者测试用例的测试判定已知,通常都不需要 MTM。而 MRTM 的适用场合更广泛,它对这些条件都没有要求。

(3)蜕变关系的用途不同:对于 MTM,蜕变关系 MR 通常用来生成附加测试用例,从而为原始测试用例提供测试判定。而对于 MRTM,MR 就有两种用途:一方面,MR 可以指导原始测试用例的生成;另一方面,它也可以为原始测试用例提供测试判定。因此,在 MRTM 中,蜕变关系具有双重作用。

(4)原始测试用例生成方法不同:在 MTM 中,通常依赖于其它类型的测试方法生成原始测试用例,而 MRTM 则可以采用本文提供的 MT-ART 和 MD-ART 生成原始测试用例和附加测试用例。因此,在原始测试用例生成方面,MRTM 并不依赖于其它测试方法。

综上所述, MRTM最大的优点就是克服了 MTM 的第一个不确定性问题, 实现了基于 MT-ART 和 MD-ART 的蜕变测试用例生成方法, 避免了对其它测试技术的依赖。这种改变使得蜕变测试已经超越其提出的初衷, 对于扩展蜕变的应用具有重要的意义。综上所述, 可以说 MRTM 完全克服了 MTM 自动化实施面临的第一个难题, 也部分缓解了第二个难题。

表 1 MTM 与 MRTM 对比

| | MTM | MRTM |
|------------|---------------------------|--------------------------|
| 目的 | 为原始测试用例提供测试判定 | 提供测试判定已知的测试用例 |
| 适用场合 | 1. 已存在原始测试用例 2. 缺少测试判定 | 一般软件测试 |
| 蜕变关系的用途 | 提供测试判定 | 1. 生成蜕变测试用例 2. 提供测试判定 |
| 原始测试用例生成方法 | 其它测试技术 | 1. MT-ART 2. MD-ART |

4 FTCL-MT 与蜕变测试的结合

4.1 蜕变测试中失效测试用例定位必要性分析

4.1.1 SBFL 的必要条件

通常软件测试的主要目的之一就是通过执行测试用例来检测被测软件中是否存在故障。现有的基于测试信息的故障定位方法中, 无论是基于程序特征谱的定位技术 (Spectra-Based Fault Localization, SBFL)^[11], 还是基于模型的故障定位技术 (Model-Based Fault Localization, MBFL)^[12], 其前提条件都是已知测试用例执行结果失效还是正确。例如, 图 3 给出了一个典型的 SBFL 实例, 图中每一列表示一个测试用例, 每一行表示程序代码, 黑点表示测试用例覆盖了当前行所示的语句, P 表示测试用例成功, F 表示测试用例执行失败 (本文中又称为失效测试用例), 最后一列的数字表示根据测试用例的覆盖情况和执行情况计算的可疑度。可见, 这种计算语句可疑度的前提条件就是确定每一个测试用例失效 (F) 还是正确 (P)。

然而, 实际测试中这个前提条件并非总是成立。许多软件的测试判定都难以获取, 即存在测试判定难题^[13], 例如复

$$MR: \frac{\forall X(x_1, x_2, x_3) \in T, P(X) = mid(X)}{\forall (X_1(x_{11}, x_{12}, x_{13}), X_2(x_{21}, x_{22}, x_{23})) \in T, (x_{21} = x_{12}) \wedge (x_{22} = x_{13}) \wedge (x_{23} = x_{11}) \rightarrow P(X_1) = P(X_2)}$$

这时, 蜕变测试用例为: $\{X_1, X_2\}$ 。当该 MR 不满足时, 容易得到 $\{X_1, X_2\}$ 中一定存在失效的测试用例。这时, 根据现有方法只能确定 $\{X_1, X_2\}$ 中一定至少存在一个失效测试用例, 但是并不能确定哪个测试用例失效。因此, 蜕变测试虽然能够提供测试判定, 但是并不能提供故障定位所需的测试用例执行信息。基于此问题, 本文提出一种蜕变测试中失效测试用例的定位方法。

4.2 蜕变测试失效测试用例定位方法 FTCL-MT

蜕变测试中失效测试用例的定位方法采用软件故障定位技术中的基本思路: 通过定位模型计算参与失效蜕变关系的测试用例的可疑度, 然后基于可疑度确定失效测试用例。

4.2.1 基本假设

计算测试用例的失效可疑度, 存在以下基本假设。

覆盖信息假设: 对于目标测试用例, 如果其参与的失效蜕变关系越多, 则该测试用例为失效测试用例的概率越大; 否

杂计算程序^[14,15]、生物计算程序^[16,17]、机器学习软件^[18,19]等。这时, SBFL 可疑度的计算因缺少测试结果的支持而无法进行。例如, 对于故障定位中应用最为广泛的 SBFL 模型 Tarantula^[20], 语句 s 的可疑度 $S_T(s)$ 计算公式如下式所示:

$$S_T(s) = \frac{\frac{a_{ef}}{a_{ef} + a_{nf}}}{\frac{a_{ef}}{a_{ef} + a_{nf}} + \frac{a_{ep}}{a_{ep} + a_{np}}}$$

其中, a_{ef} 表示覆盖到目标语句的执行失败的用例个数; a_{nf} 表示未覆盖到目标语句的执行失败的用例个数; a_{ep} 表示覆盖到目标语句的执行通过的用例个数; a_{np} 表示未覆盖到目标语句的执行通过的用例个数。显然, 这些参数的取值都与测试用例执行结果相关。因此, 测试判定难题严重制约着 SBFL 技术的应用。

| mid() { int x,y,z,m; 1: read("Enter 3 numbers:", x,y,z); 2: m = z; 3: if (y<z) 4: if (x<y) 5: m = y; 6: else if (x<z) 7: m = y; // *** bug *** 8: else 9: if (x>y) 10: m = y; 11: else if (x>z) 12: m = x; 13: print("Middle number is:", m); } | Test Cases | | | | | | | | suspiciousness ^r |
|--|------------|-------|-------|-------|-------|-------|-------|----|-----------------------------|
| | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | |
| | 3,3,5 | 1,2,3 | 3,2,1 | 5,3,4 | 7,5,4 | 2,1,3 | 4,3,5 | | |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.5 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.5 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.67 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.0 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.75 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.86 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.0 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.0 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.0 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.0 |
| | ● | ● | ● | ● | ● | ● | ● | ● | 0.5 |
| | P | P | P | P | P | P | P | P | |

图 3 SBFL 引例

4.1.2 MRTM 的补充

虽然, 蜕变测试可以通过 MRTM 或者 MTM 判断一组测试用例是否满足蜕变关系, 即是否存在失效测试用例, 当存在失效蜕变关系时, 可以判定该失效蜕变测试用例集中存在失效测试用例, 但是并不能确定哪个测试用例失效, 即单个测试用例具体输出并不清楚。例如, 对于图 3 所示的程序 P, 容易得到以下蜕变关系:

则, 如果该测试用例参与的成功蜕变关系越多, 则它为失效测试用例的概率就越小。

蜕变关系输入域假设: 对于蜕变关系, 目标测试用例都在蜕变关系的输入域内, 即确保目标测试用例能够验证所选择的蜕变关系。

4.2.2 FTCL-MT

基于以上假设, 针对失效蜕变测试用例集, 采用基于蜕变关系的测试模型 MRTM 分别计算测试用例失效的可疑度, 形成了一种蜕变测试失效测试用例定位方法 (Failed Test Case Localization of Metamorphic Testing, FTCL-MT)。图 4 示出了该方法的实施步骤。

(1) 当测试用例集 (t_1, t_2, \dots, t_n) 不满足蜕变关系 MR 时, 在其中选择一个未验证的测试用例 t_i 作为被验证对象。

(2) 在被测软件的蜕变关系集合中选择另一个蜕变关系作为被验证蜕变关系 MR_j ; 蜕变关系集合需要人工事先构

造。通常对于同一被测软件,可以得到其多个蜕变关系,从而形成蜕变关系集合 $MR = \{MR_1, MR_2, \dots, MR_m\}$, m 为自然数。其中,蜕变关系的选择顺序并不会影响其后续可疑度的计算,因此可以任意选择。

(3)以测试用例 t_i 作为原始测试用例,根据被验证蜕变关系 MR_j 的输入关系,生成附加测试用例,并验证 MR_j 的输出关系是否满足;当包含多个蜕变关系 MR_j 的原始测试用例时,其它原始测试用例随机生成,然后再根据输入关系 IR_j 生成附加测试用例,最后验证输出关系 OR_j 是否满足。

(4)如果 MR_j 输出关系 OR_j 不满足,记录当前蜕变关系为 1;否则,转到步骤(5);为了后续计算测试用例在满足的蜕变关系和不满足的蜕变关系中参与的次数,当测试用例参与的蜕变关系失效时,用 1 表示。默认情况为 0,表示该测试用例参与的蜕变关系满足。

(5)如果还有蜕变关系没有被验证,转到步骤(2);否则,转到步骤(6);按照以上步骤对测试用例进行验证,直到验证完所有的蜕变关系。

(6)当验证完所有蜕变关系后,计算测试用例可疑度 *Suspiciousness*;为了定位失效测试用例,采用测试用例可疑度算子 *Suspiciousness* 计算每个测试用例的可疑度,其计算方法如下:

$$Suspiciousness(t_i) = \frac{\Psi_{failed}(t_i)}{\Psi_{failed}(t_i) + \Psi_{passed}(t_i)}$$

其中, $\Psi_{failed}(t_i)$ 表示测试用例 t_i 参与的蜕变关系发生失效的次数; $\Psi_{passed}(t_i)$ 表示测试用例 t_i 参与的蜕变关系通过的次数。

(7)如果不是最后一个测试用例,转到步骤(1);否则,转到步骤(8);根据以上步骤,可以得到测试用例 t_i 的可疑度 $Suspiciousness(t_i)$ 。然后,采用同样的步骤计算其它测试用例的可疑度,直到获取到所有测试用例可疑度。

(8)计算 $\text{Max}(Suspiciousness(t_1), Suspiciousness(t_2), \dots, Suspiciousness(t_n))$, 其对应的测试用例即为失效测试用例。

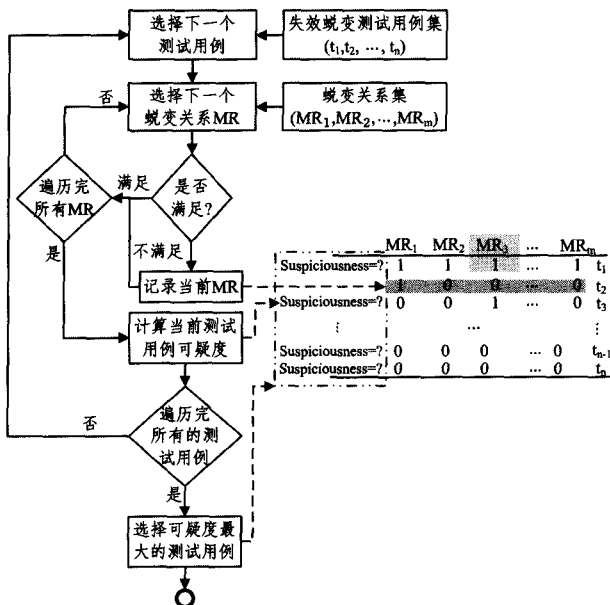


图4 蜕变测试失效测试用例定位方法

依照上述上述步骤通过 FTCL-MT 就可以确定失效蜕变

测试用例集中最有可能的一个失效测试用例。如果需要继续确认其它测试用例是否失效,则需要去除当前已经确定的失效测试用例,然后再对其它测试用例采用 FTCL-MT 方法确定失效测试用例。

5 FTCL-MT 与蜕变测试的结合

FTCL-MT 与蜕变测试之间的关系如图 5 所示。由图可知,FTCL-MT 是对两种蜕变测试模型 MTM 和 MRTM 的有效补充,而蜕变测试又是开展 FTCL-MT 的前提。由表 2 可知,MTM 与 FTCL-MT 的结合不但可以缓解传统测试方法的测试判定难题,而且也能弥补蜕变测试不能确定具体失效测试用例的不足,能够给出生失效的测试用例。MRTM 与 FTCL-MT 的结合则只需要人工构造一定的蜕变关系,就能够得到发生失效的测试用例。

综上所述,FTCL-MT 与蜕变测试的结合为存在测试判定难题的软件提供了一套传统意义上完整的测试方法。



图5 基于蜕变关系的测试与失效测试用例定位之间的关系

表2 FTCL-MT 与 MTM 和 MRTM 的结合

| | MTM+(FTCL-MT) | MRTM+(FTCL-MT) |
|----|-------------------------------|----------------|
| 输入 | 1. 其它测试技术生成的原始测试用例 2. 蜕变关系 | 蜕变关系 |
| 输出 | 已知失效的测试用例 | 已知失效的测试用例 |

6 实例验证

6.1 实验设置

本节实验的主要目的是验证通过蜕变测试后,出现失效蜕变关系时采用 FTCL-MT 方法是否能够及时定位失效测试用例。选择了表 3 所列的 8 个被测件作为实验对象。其中前 5 个为已有的蜕变测试文献中都已经公开使用的实验验证程序,在表的第一列中也给出了被测件相应的来源。第 6 个被测件为软件测试领域标准被测件。最后两个被测件是我们承担的一个测试项目中地理信息系统地彤量算程序中的两个子程序。

针对这 8 个被测件,采用变异测试方法来构造存在故障的被测程序。针对 C 语言程序,采用 Jia 和 Harman^[21] 提供的 Milu 工具随机生成变异;而对于 Java 语言程序,则采用 mu-Java 工具^[22] 构造变异程序。每个被测件都构造 100 个不同的单故障被测件,因此对于一个被测件就存在 101 个实验对象。对于每一个被测件的蜕变关系,由表 3 的第 4-6 行可以看到,这 3 个程序直接采用文献中所提供的蜕变关系,其它 5 个被测件则采用本文提供的蜕变关系。蜕变关系的数量如表 3 的第 4 列所列,最后一列给出了被测件的主要功能。在测

试用例生成方面,我们采用随机测试策略生成 100 组原始测试用例,然后根据相应的蜕变关系生成附加测试用例。

表 3 实验对象

| 被测件 | 编程语言 | 变异 MR | MR 来源 | 主要功能 |
|---------------------------|------|-------|--------|---------------------|
| Sin ^[23] | C | 100 | 附录 D | 计算 sin(x) 函数 |
| Trisquare ^[24] | C | 100 | 附录 C | 计算三角形面积 |
| p ^[25] | C | 100 | 文献[25] | 计算 a 到 b 定积分 |
| NBC ^[19] | Java | 100 | 文献[19] | 基于 kNN 的经典贝叶斯分类程序 |
| BinSearch ^[23] | C | 100 | 文献[23] | 有序数据结构 T 中搜索目标整数 |
| Tcas ^[26] | C | 100 | 附录 E | 飞机空中碰撞检测 |
| Area | C | 100 | 附录 A | 数字地图中表面积计算 |
| Grade | C | 100 | 附录 B | 数字地图中梯度计算 |

6.2 实验结果

对于一个被测件,存在 100 个故障版本以及 1 个原程序。因此对于一个被测件就存在 101 个不同版本。针对这些程序,分别采用相应的蜕变关系对其进行验证,得到相应的不满足蜕变关系集,然后根据 FTCL-MT 方法确定其失效 TC,实验结果如图 6 所示。横轴表示不同的被测件,纵轴表示每一个被测件的失效 TC 定位的平均准确率。由图可知,对于第一可疑测试用例(TC)的确定,准确度全部是 100%,由此可见本文提出的 FTCL-MT 方法在确定最可疑的测试用例时极为有效。同时,在实验中将可疑度排第二的测试用例确定为第二可疑的测试用例。由图 6 可见,根据第二可疑度定位失效测试用例的准确度显然较低。

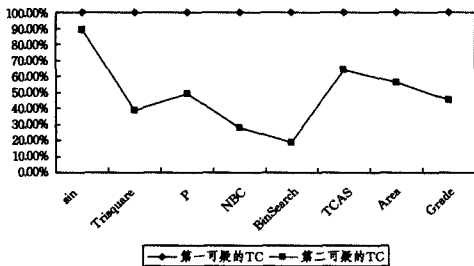


图 6 失效测试用例定位统计结果

然而,需要注意的是,寻找第二可疑的 TC 时,必须考虑到蜕变关系中实际的失效测试用例数量,如果失效测试用例集中只存在一个失效测试用例,即失效的蜕变关系就是由一个失效测试用例引起,则通过 FTCL-MT 方法确定的第二个可疑 TC 肯定不准确,这也会对第二个可疑 TC 的定位结果产生影响。

为此,对 100 组失效测试用例集进行了筛选,选择其中存在两个以上失效测试用例的蜕变关系以及相应的蜕变测试用例集,重新进行实验,结果如图 7 所示。

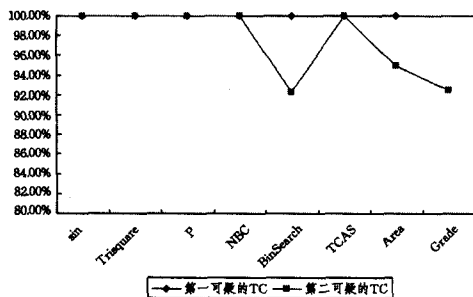


图 7 失效测试用例定位统计结果

由图 7 可知,第一可疑 TC 的定位依旧非常准确,定位准确率全部都为 100%。而第二可疑 TC 的定位准确度也有极大的提升,其中对 sin 程序、P 程序、det 程序、NBC 程序以及 TCAS 程序都能 100%地确定第二个失效测试用例。

然而,令人遗憾的是,在实际的蜕变测试中我们并不能先验确定失效的蜕变关系对应的蜕变测试用例集中至少存在两个失效测试用例。根据当前的蜕变测试方法,只能确定至少存在一个失效测试用例。因此,FTCL-MT 目前并不能用于确定第二个可疑测试用例。

结束语 针对传统蜕变测试模型 MTM 存在的不足,本文首先提出了一种基于蜕变关系的测试模型 MRTM,克服了 MTM 对其它测试技术的依赖,扩展了蜕变测试应用范围;其次,针对失效蜕变关系,提出了一种失效测试用例定位方法 FTCL-MT。当蜕变关系不满足时,为确定失效测试用例提供了一种有效的手段;最后,通过实例验证了 FTCL-MT 的有效性。

参考文献

- [1] Feng Jian-qiang . On the Effectiveness of Metamorphic Testing for Numerical Programs[D]. The University of Hong Kong, 2003
- [2] Weyuker E J. On testing non-testable programs[J]. The Computer Journal,1982,25(4):465-470
- [3] Baresi L,Young M. Test oracles [R]. Technical Report CIS-TR01-02,Department of Computer and Information Science University of Oregon,Eugene,OR,2001:9
- [4] Chen T Y,Kuo Fei-ching ,Tam W K, et al. Testing a software-based PID controller using metamorphic testing[C] // Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS 2011). Vilamoura,Algarve,Portugal,2011:387-396
- [5] Zhan Wei-hui, Huang Song. Achievements and Challenges of Metamorphic Testing[C] // 2013 Fourth World Congress on Software Engineering. Hong Kong,China,2013:73-77
- [6] Chen T Y,Cheung S C, Yiu S M. Metamorphic testing: a new approach for generating next test cases [R]. Department of Computer Science,Hong Kong University of Science and Technology,Tech. Rep. HKUST-CS98-01,1998
- [7] Chen T Y, Yu Y T. On the relationship between partition and random testing[J]. IEEE Transactions on Software Engineering,1994,20(12):977-980
- [8] DeMillio R A, Offutt A J. Constraint-based automatic test data generation[J]. IEEE Transactions on Software Engineering, 1991,17(9):900-910
- [9] Offutt J,Liu Shao-ying,Abdurazik A, et al. Generating test data from state-based specifications[J]. Software Testing, Verification and Reliability,2003,13(1):25-53
- [10] Jia Yue,Harman M. An analysis and survey of the development of mutation testing[J]. IEEE Transactions on Software Engineering,2011,37(5):649-678
- [11] Jones J A, Harrold M J. Empirical Evaluation of the Tarantula Automatic Fault-Localization Technique[C]// ASE 2005. 2005: 273-282
- [12] Wong W E, Qi Yu,Zhao Lei, et al. Effective Fault Localization using Code Coverage[C]//International Computer Software and Applications Conference. 2007:449-456
- [13] Weyuker E J. The Oracle Assumption of Program Testing[C]//

Proceedings of the 13th International Conference on System Sciences (ICSS). Honolulu, HI, 1980

- [14] Baker J, Thornton J. Software engineering challenges in bioinformatics[C]// International Conference on Software Engineering. 2004;12-15
- [15] Xie Xiao-yuan, Ho J, Murphy C, et al. Improving the quality of computational science software by using metamorphic relations to test machine learning applications[J]. Department of Computer Science Columbia University, 2009, 12(1):1-80
- [16] Ho J W K, Lin M W, Adelstein S, et al. Erratum: customising an antibody leukocyte capture microarray for systemic lupus erythematosus; beyond biomarker discovery[J]. Proteomics-Clinical Applications, 2010, 4(6/7): 679-679
- [17] Ho J W K, Stefani M, Remedios C G, et al. Differential variability analysis of gene expression and its application to human diseases[J]. Bioinformatics, 2008, 24:390-398
- [18] Xie Xiao-yuan, Ho J W K, Murphy C, et al. Testing and validating machine learning classifiers by metamorphic testing[J]. Journal of Systems and Software, 2011, 84(4): 544-558
- [19] Xie Xiao-yuan, Ho J, Murphy C, et al. Application of metamorphic testing to supervised classifiers[C]// International Conference on Quality Software. IEEE, 2010;135-144
- [20] Jones J A, Harrold M J, Stasko J. Visualization of test information to assist fault localization[C]// ICSE 2002. 2002;467-477
- [21] Jia Y, Harman M. A Customizable, Runtime-Optimized Higher Order Mutation Testing Tool for the Full C Language[C]// Practice and Research Techniques. 2008;94-98
- [22] Ma Y S, Offutt J, Kwon Y R. MuJava: An Automated Class Mutation System[J]. Journal of Software Testing, Verification and Reliability, 2005, 15(2): 97-133
- [23] Chen T Y, Kuo F C, Liu Ying, et al. Metamorphic Testing and Testing with Special Values[C]// Acis International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. 2004;128-134
- [24] Dong Guo-wei. Metamorphic testing techniques for error detection efficiency[D]. Southeast University, 2009
- [25] Chen T Y, Tse T H, Zhou Zhi-quan. Semi-proving: an integrated method based on global symbolic evaluation and metamorphic testing[C]// ACM Sigsoft Software Engineering Notes, 2002, 27(4):191-195
- [26] Hutchins M, Foster H, Goradia T, et al. Experiments of the effectiveness of dataflow-and controlflow-based test adequacy criteria[C]// International Conference on Software Engineering. IEEE Computer Society Press, 1994;191-200

(上接第 56 页)

到多业务交叉传输实现的情况,这方面将是我们后续研究的一个深入点。

参 考 文 献

- [1] Charara H, Scharbag J, Ermont J, et al. Methods for bounding end-to-end delays on an AFDX network[C]// 18th Euromicro Conference on Real-Time Systems. IEEE, 2006;193-202
- [2] Mifdaoui A, Frances F, Fraboul C. Real-time guarantees on Full-Duplex Switched Ethernet for military applications[OL]. <http://oataa.univ-toulouse.fr/2160>
- [3] Zeng X, Song D. The Research on End-to-End Delay Calculation Method for Real-Time Network AFDX[C]// International Conference on Computational Intelligence and Software Engineering. 2010;1-4
- [4] Zhao Yong-ku, Wang Hong-chun, Tang Lai-sheng, et al. Analysis Method of End-to-End Delays on an AFDX Avionic Network[J]. Electronics Optics & Control. 2013, 20(4): 81-83 (in Chinese)
- 赵永库, 王红春, 唐来胜. AFDX 网络端到端时延分析方法[J]. 电光与控制, 2013, 20(4): 81-83
- [5] Wu Z T, Huang N, Wang X W, et al. Analysis of end-to-end delay on AFDX based on stochastic network calculus[J]. Systems Engineering and Electronics, 2013, 35(1): 168-172 (in Chinese)
- 伍志韬, 黄宁, 王学望, 等. 基于随机型网络演算的 AFDX 端端时延分析方法[J]. 系统工程与电子技术, 2013, 35(1): 168-172
- [6] Scharbag J, Ridouard F, Fraboul C. A Probabilistic Analysis of End-To-End Delays on an AFDX Avionic Network[J]. IEEE Transactions on Industrial Informatics. 2009, 5(1):38-49
- [7] Georges J P, Rondeau E, Divoux T. Evaluation of switched Ethernet in an industrial context by using the Network Calculus[C]// IEEE International Workshop on Factory Communication Systems. 2002;19-26
- [8] Zhang J, Li D, Wu Y. Modelling and performance analysis of AFDX based on Petri Net[C]// International Conference on Future Computer and Communication. IEEE, 2010; V2-566-V2-570
- [9] Wang C, Li J, Hu F. Fault tree synthesis for an avionic network[C]// International Conference on Transportation, Mechanical, and Electrical Engineering. IEEE, 2011;155-159
- [10] Wang Chen-hu. Reliability Model and Performancy Analysis of AFDX Avionics Networks[D]. Shanghai: Shanghai Jiao Tong University. 2012 (in Chinese)
- 王臣虎. AFDX 航空网络的可靠性建模与性能分析[D]. 上海: 上海交通大学, 2012
- [11] Huang N, Wu Z T. Survey of network reliability evaluation models and algorithms[J]. Systems Engineering and Electronics, 2013, 35(12): 2651-2660 (in Chinese)
- 黄宁, 伍志韬. 网络可靠性评估模型与算法综述[J]. 系统工程与电子技术, 2013, 35(12): 2651-2660
- [12] Li S, Wang X W, Kang R. Investigation on Reliability Parameters of Avionics Full Duplex Switched Ethernet (AFDX) for Integrity Requirements[J]. Journal of Xi' an Jiaotong University, 2013, 47(3): 126-131 (in Chinese)
- 李硕, 王学望, 康锐. 面向完整性要求的航空电子全双工交换式以太网可靠性评价参数研究[J]. 西安交通大学学报, 2013, 47(3): 126-131
- [13] Dugan J B, Bavuso S J, Boyd M A. Dynamic fault-tree models for fault-tolerant computer systems[J]. IEEE Transactions on Reliability. 1992, 41(3): 363-377
- [14] Du Y J. Research on avionics full duplex switched Ethernet[J]. Computer Engineering, 2009, 35(11): 77-79 (in Chinese)
- 杜亚娟. 航空全双工交换式以太网探究[J]. 计算机工程, 2009, 35(11): 77-79
- [15] Zhu Zheng-fu, Li Chang-fu, He En-shan. The dynamic fault tree analysis method based on Markov chain[J]. Acta Armamentarii, 2008, 29(9): 1104-1107 (in Chinese)
- 朱正福, 李长福, 何恩山, 等. 基于马尔可夫链的动态故障树分析方法[J]. 兵工学报, 2008, 29(9): 1104-1107