



北京大学

硕士研究生学位论文

题目: 测试文档

姓 名: 某某

学 号: 0123456789

院 系: 某某学院

专 业: 某某专业

研究方向: 某某方向

导 师: 某某教授

某年某月

版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以任何方式传播。否则一旦引起有碍作者著作权之问题，将可能承担法律责任。

摘要

关键词：其一，其二

Test Document

Test (Some Major)

Directed by Prof. Somebody

ABSTRACT

Test of the English abstract.

KEYWORDS: First, Second

目录

序言	1
第一章 研究背景	3
1.1 缺陷定位技术	3
1.1.1 自动缺陷定位技术概述	3
1.1.2 基于频谱的缺陷定位	4
1.1.3 基于状态覆盖的缺陷定位	4
结论	5
参考文献	7
附录 A 附件	9
致谢	11
北京大学学位论文原创性声明和使用授权说明	13

序言

第一章 研究背景

1.1 缺陷定位技术

随着软件的发展，生活中越来越多的方面都与软件有着紧密的关系。小到人们的日常出行、购物、餐饮等，大到航空航天、医药等领域，软件在人们的生活中扮演着重要的角色。随着软件的应用领域的扩大，软件的复杂性上升，提升了软件缺陷的可能性。软件缺陷可能会导致巨大的损失。一个著名的被广泛引用的例子，是在海湾战争时，一颗导弹由于导航软件的精度缺陷而偏离了目标，导致28人死亡和100人受伤。**{TODO:cite}**。美国国家标准与技术研究院(NIST)2002年发表的一篇报告(**{TODO:cite}**)显示，软件缺陷每年会导致约595亿美元的经济损失。**{TODO:每年这么多国内公司漏洞事件}**发现并修复软件缺陷，保障软件的高质量成为一项重要的任务。

在发现软件缺陷之后，开发人员为了解决这个缺陷往往需要三步[15]。第一步，缺陷定位，需要找到程序中和这个缺陷有关的语句。第二步，理解缺陷，明白为什么会发生缺陷。第三步，修复缺陷，修改代码以让缺陷消失。这三个步骤合起来就是调试的过程。缺陷定位作为调试的第一步，其完成速度和准确性对后面的步骤有着很大的影响。在传统的开发环境当中，人们可以手动调试来定位缺陷，比如插入断点、打印日志信息等等。但是随着软件的复杂性的上升，手动地定位软件缺陷非常耗费开发者的时间和精力。为了提高定位缺陷的速度，研究人员对自动化的缺陷定位展开了研究，并取得了巨大的进展。**{TODO:cite}**然而在2011年，Partin和Osro的一篇调查[15]通过研究缺陷定位技术在实际应用场景下的效果，发现以往的评价指标并不能准确的反映缺陷定位技术在实际应用中的效果。以往的缺陷定位技术是基于一系列关于开发人员会如何调试的假设，而这些假设在实际场景的某些情况下会失效。自动化缺陷定位技术还有很大的发展空间。

1.1.1 自动缺陷定位技术概述

Weiser在1981年提出的程序切片[17, 18]是自动调试（特别是缺陷定位）最早的技术之一。给定一个程序 P 和一个在 P 的语句 s 中使用的变量 v ，程序切片会找到 P 中所有可能会影响 s 中 v 的值的语句。如果 s 中 v 的值是错误的，那么导致这个错误的错误语句一定在这个切片当中。也就是说，不在这个切片当中语句可以在调试过程中被忽略。尽管程序切片已经减少了可能出错的语句的数量，但是切片中的语句的数量仍然比较大。为了解决这个问题，Korel和Laski在1988年提出了动态程序切片[10]。动态程序切

片计算某一个特定执行的切片。后来又有很多的动态程序切片的变种被提出[4, 6, 21, 22], 用于解决调试问题, 并且产生了大量研究工作[1, 2, 8, 9, 13, 14, 19]。

为了解决程序切片调试方法的短板, 一种通过观察错误程序的执行特征和正确程序的执行特征的调试技术被提出。这些技术通过收集程序执行信息, 观察不同的某种特征, 来定位缺陷。比如使用路径概要[16], 反例[3, 5], 语句覆盖[7]和谓词值[11, 12]等等。

本文根据北京大学熊英飞研究员对缺陷定位的分类[20], 将缺陷定位分为以下几类。

- 基于切片的缺陷定位
- 基于频谱的缺陷定位
- 基于状态覆盖的缺陷定位
- 基于变异的缺陷定位
- 基于构造正确执行状态的缺陷定位
- 基于算法式调试的缺陷定位
- 基于差异化调试的缺陷定位

本文的研究内容主要根据基于频谱的缺陷定位和基于状态覆盖的缺陷定位。

1.1.2 基于频谱的缺陷定位

基于频谱的缺陷定位是使用最广泛的自动化缺陷定位方法[20]。Harrold等人在

1.1.3 基于状态覆盖的缺陷定位

结论

参考文献

- [1] Hiralal Agrawal, Richard A. Demillo and Eugene H. Spafford. *Debugging with dynamic slicing and backtracking*, **1993**: 589–616.
- [2] Elton Alves, Milos Gligoric, Vilas Jagannath *et al.* “Fault-localization using dynamic slicing and change impact analysis”. In: *Ieee/acm International Conference on Automated Software Engineering*, **2011**: 520–523.
- [3] Thomas Ball, Mayur Naik and Sriram K Rajamani. “From symptom to cause: localizing errors in counterexample traces”. *Acm Sigplan Notices*, **2003**, 38(1): 97–105.
- [4] Richard A. Demillo, Hsin Pan and Eugene H. Spafford. “Critical slicing for software fault localization”. In: **1996**: 121–134.
- [5] Alex Groce, Daniel Kroening and Flavio Lerda. “Understanding Counterexamples with explain”. In *Computer-Aided Verification*, **2004**, 3114: 453–456.
- [6] Gyim, Tibor Thy, Besz *et al.* “An efficient relevant slicing method for debugging”. *Acm Sigsoft Software Engineering Notes*, **1999**, 24(6): 303–321.
- [7] Jones, A James, Harrold *et al.* “Visualization of test information to assist fault localization”. *Biochemical Engineering Journal*, **2002**, 24(2): 115–123.
- [8] Xiaolin Ju, Shujuan Jiang, Xiang Chen *et al.* “HSFal: Effective fault localization using hybrid spectrum of full slices and execution slices”. *Journal of Systems & Software*, **2014**, 90(1): 3–17.
- [9] Z. A. Al-Khanjari, M. R. Woodward, Haider Ali Ramadhan *et al.* “The Efficiency of Critical Slicing in Fault Localization”. *Software Quality Journal*, **2005**, 13(2): 129–153.
- [10] Bogdan Korel and Janusz Laski. “Dynamic program slicing ☆”. *Information Processing Letters*, **1988**, 29(3): 155–163.
- [11] Ben Liblit, Mayur Naik, Alice X. Zheng *et al.* “Scalable statistical bug isolation”. In: **2005**: 15–26.
- [12] Chao Liu, Xifeng Yan, Long Fei *et al.* “SOBER: statistical model-based bug localization”. In: *European Software Engineering Conference Held Jointly with ACM Sigsoft International Symposium on Foundations of Software Engineering*, **2005**: 286–295.
- [13] Chao Liu, Xiangyu Zhang, Jiawei Han *et al.* “Indexing Noncrashing Failures: A Dynamic Program Slicing-Based Approach”. In: *IEEE International Conference on Software Maintenance*, **2007**: 455–464.
- [14] Xiaoguang Mao, Yan Lei, Ziyang Dai *et al.* “Slice-based statistical fault localization ☆”. *Journal of Systems & Software*, **2014**, 89(1): 51–62.
- [15] Chris Parnin and Alessandro Orso. “Are automated debugging techniques actually helping programmers?” In: *International Symposium on Software Testing and Analysis*, **2011**: 199–209.

- [16] Thomas Reps, Thomas Ball, Manuvir Das *et al.* “*The use of program profiling for software maintenance with applications to the year 2000 problem*”. *Acm Sigsoft Software Engineering Notes*, **1997**, 22(6): 432–449.
- [17] Mark Weiser. “*Program Slicing*”. *IEEE Transactions on Software Engineering*, **1984**, SE-10(4): 352–357.
- [18] Mark Weiser. “*Program slicing*”. In: *International Conference on Software Engineering*, **1981**: 439–449.
- [19] Franz Wotawa. “*Fault Localization Based on Dynamic Slicing and Hitting-Set Computation.*” In: *International Conference on Quality Software*, **2010**: 161–170.
- [20] Yingfei Xiong. *Fault Localization*, **2018**. http://sei.pku.edu.cn/~xiongyf04/SA/2017/18_fault_localization.pdf, retrieved on 2018-04-07.
- [21] Xiangyu Zhang, Neelam Gupta and Rajiv Gupta. “*Pruning dynamic slices with confidence*”. *Acm Sigplan Notices*, **2006**, 41(6): 169–180.
- [22] Xiangyu Zhang, R Gupta and Youtao Zhang. “*Precise dynamic slicing algorithms*”. In: *International Conference on Software Engineering, 2003. Proceedings*, **2003**: 319–329.

附录 A 附件

致谢

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名： 日期： 年 月 日

学位论文使用授权说明

（必须装订在提交学校图书馆的印刷本）

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文；
- 因某种特殊原因需要延迟发布学位论文电子版，授权学校在 ☐ 一年 / ☐ 两年 / ☐ 三年以后在校园网上全文发布。

（保密论文在解密后遵守此规定）

论文作者签名： 导师签名： 日期： 年 月 日