组合测试

COMBINATORIAL TESTING

聂长海 著

科学出版社

组合测试

聂长海 著

科学出版社

北京

内 容 简 介

组合测试是一种重要的软件测试方法. 该方法系统地检测软件系统中各种组合的相互作用, 例如, 软件中不同功能、不同输入、不同配置或环境等因素的相互作用可能引发的故障. 由于应用场景的广泛性, 组合测试越来越受到工业界和学术界的研究和重视. 本书系统介绍组合测试的概念、特点、应用场景、具体的应用步骤, 以及应用和发展过程中所面临的各种科学问题和解决方案等. 通过丰富具体的示范性应用实例介绍各种抽象的概念和过程, 力图简洁清楚、通俗易懂, 尽可能降低对读者专业基础要求. 此外, 本书还提供了组合测试领域的300多篇文献列表下载(http://gist.nju.edu.cn), 以供读者学习研究.

本书是目前软件测试领域第一本专门系统介绍组合测试方法的中文著作, 是计算机相关领域的工作人员, 特别是计算机软件开发和测试人员、软件测试领域的教学、科研以及相关从业人员的重要参考书.

图书在版编目(CIP)数据

组合测试/聂长海著. —北京: 科学出版社, 2015.3
ISBN 978-7-03-043727-3

I. ①组… II. ①聂… III. ①软件-测试-研究 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2015) 第 049821 号

责任编辑: 惠 雪 王 苏 / 责任校对: 胡小洁
责任印制: 赵 博 / 封面设计: 许 瑞

科 学 出 版 社 出版
北京东黄城根北街 16 号
邮政编码: 100717
http://www.sciencep.com

双 青 印 刷 厂 印刷
科学出版社发行 各地新华书店经销
*
2015 年 3 月第 一 版 开本: 720×1000 1/16
2015 年 3 月第一次印刷 印张: 19 1/4
字数: 386 000

定价: 78.00 元

(如有印装质量问题, 我社负责调换)

序

在过去 30 年中, 以计算机互联网为代表的信息技术的迅速发展和广泛应用, 极大地激发和释放了人们的创造力. 在计算机领域, 各种新概念、新理论和新技术层出不穷, 百花齐放, 呈现出一片繁荣的景象. 信息技术快速发展在给广大科技工作者带来很多挑战的同时, 也带来了很多新的机遇. 一方面, 为了适应新需求, 解决新问题, 人们需要不断开拓新的方向; 另一方面, 爆炸式增长的新知识, 需要专业人士进行有序的整理, 不断总结其中成功的经验、失败的教训和存在的问题. 开拓是创新, 而总结则可以服务于创新, 为进一步创新奠定夯实的基础.

如何提高软件生产率与保障软件产品质量一直是软件领域的基本问题. 软件测试理论、方法与技术是保障软件质量的重要途径之一. 作为一种新兴的软件测试技术, 组合测试已经发展成为软件测试领域的一个重要分支, 受到人们普遍的认可和关注. 该书作者及其合作者一直坚持组合测试领域的研究, 既不断开拓也不断总结, 取得了丰硕成果. 自 2002 年以来, 发表相关学术论文 40 多篇, 获得多项软件著作权和发明专利. 其中最有代表性的研究成果是提出了关于组合测试的最小故障模式概念, 并给出了一种基于最小故障模式的组合测试方法学, 该项成果发表在软件工程领域著名期刊*ACM Transactions On Software Engineering and Methodology* 上; 另一项代表性工作是他在充分分析已有文献的基础上, 结合自己在组合领域已有研究实践, 总结出其中的关键问题、方法、重要应用和未来发展方向, 论文发表在计算机领域著名学术期刊*ACM Computing Survey* 上, 得到了国际同行的关注.

该书在全面总结组合测试领域 30 多年来已经发表的 300 多篇研究论文及相关文献的基础上, 系统地介绍组合测试从低级到高级, 从简单到复杂的发展过程、科学问题及解决方案. 该书具有以下特点: ①从软件测试角度, 详细介绍组合测试起源和发展过程, 将组合测试作为一种重要的软件测试方法进行系统细致地分析. ②以组合测试过程为主线, 介绍组合测试理论、方法及其应用. ③在博采众长的基础上, 精心组织该书内容, 语言简洁清楚, 深入浅出. 相信该书的出版必将进一步推动组合测试在学术界和产业界的研究和应用.

该书作者所从事的组合测试研究在南京大学软件新技术国家重点实验室一直得到高度的重视, 并得到了实验室的持续资助, 同时, 他的这项研究也是国家自然科学基金重大研究计划集成项目支持的一个重要子课题. 所以, 该书的出版既是我

们软件新技术国家重点实验室的一项成果, 也是国家自然科学基金重大研究计划项目的成果之一. 特此向广大读者推荐这本书.

中国科学院院士
南京大学计算机软件新技术国家重点实验室主任
吕　建

2015 年 2 月 18 日于南京大学

前　言

软件测试的关键问题是寻找尽可能小的测试用例集, 以最小的成本发现待测试软件中潜在的问题. 人们从不同角度, 基于不同理论, 针对软件开发的不同阶段和不同类型的软件提出了很多软件测试的方法. 这些方法的综合使用为成功解决软件测试的关键问题提供了可能. 组合测试便是这些方法中的一种, 它旨在系统测试软件中存在的各种可能的 "组合", 检查各种 "组合" 可能促发的软件故障.

在计算机科学领域中, 组合测试是一个相对比较小的研究领域, 但由于它广泛的应用价值使其越来越受到人们的关注. 近年来, 关于组合测试的研究得到了快速发展. 2001 年开始关注这个领域时, 可以查到的参考文献只有二十几篇, 到 2014 年已有 300 多篇.

到目前为止, 作者从事组合测试的研究已经有十多年了, 一直关注这个领域的发展, 也一直通过应用、研究、思考和交流参与这个领域的发展, 越来越觉得应该写一本专门的著作, 完整系统地介绍这个领域, 以填补国内相关专著在这个领域的空白.

为了写好这本书, 作者对该领域到目前为止可以收集到的 300 多篇研究论文进行系统研究和梳理, 以期巩固和完善组合测试领域的研究基础, 将该领域工作做到更好. 本书努力将组合测试领域的理论体系、学术研究、实证实践和产业应用等结合起来, 以期全方位发展.

全书共 8 章. 第 1 章为组合测试概论, 介绍组合测试的由来和背景, 具体包括组合测试在计算机科学体系中的地位和意义、应用场景、概念、历史演变和度量等. 第 2 章为组合测试的测试用例生成, 介绍利用各种贪心算法、数学方法和演化方法等生成满足各种不同组合覆盖要求的测试用例集. 第 3 章为组合测试的优化与约简, 介绍各种测试用例集排序方法和最小化方法, 以追求组合测试的高效率和低成本. 第 4 章为组合测试的故障定位, 介绍各种基于组合测试结果定位出触发软件故障的因素组合的方法. 第 5 章为自适应组合测试, 介绍组合测试的自适应机制和应用. 第 6 章为组合测试的应用, 介绍组合测试方法在浏览器功能测试、手机测试、机顶盒测试、软件配置测试和兼容性测试等方面的应用. 第 7 章为组合测试的比较研究及工具, 将组合测试与随机测试和自适应随机测试进行系统比较. 第 8 章为组合测试的研究现状与未来, 总结组合测试的当前研究现状, 并给出未来的研究方向.

本书在写作过程中得到很多同仁和朋友的帮助和支持, 在此一并表示感谢. 首先感谢南京大学的徐宝文, 香港理工大学的 Hareton Leung 多年来给予的指导和帮

助. 感谢作者指导的博士和硕士研究生: 吴化尧、钮鑫涛、陈思洋、梁亚澜、李杰、蒋静、黄海波、孙文雯、刘洋华等多年来的跟随并致力于组合测试领域的研究, 他们辛勤的工作为本书的编写提供了非常有力的支持. 感谢中国科学院数学研究所的董昭、刘克和李晓花在组合测试用例约简方面提供的帮助. 感谢北京航空航天大学蔡开元在组合测试的自适应方面的大力支持. 感谢南京晓庄学院的徐家喜和王燕在组合测试应用方面提供的帮助. 感谢美国亚利桑那州立大学的 Charlie Colbourn 在多次访问时提供的有益的合作与交流; 感谢澳大利亚斯威本科技大学的 Diana Kuo 在关于组合测试与自适应随机测试和随机测试比较方面提供的支持和帮助; 感谢中国科学院软件研究所的张健, 美国微软公司软件开发测试工程师史亮, 南京邮电大学的王子元多年来的交流、合作和支持.

同时, 本书的出版得到了南京大学软件新技术国家重点实验室、南京大学计算机科学与技术系的大力支持, 并得到了国家自然科学基金项目 (项目编号 61272079、60773104)、高等学校博士学科点专项科研基金 (博导类)(项目编号 20130091110032)、国家自然科学基金重大研究计划集成项目 (项目编号 91318301)、国家 "863" 高科技研究发展计划专题项目 (项目编号 2008AA01Z143)、南京大学创新群体项目 (项目编号 61321491) 以及江苏省自然科学基金项目 (项目编号 BK2010372) 等的资助.

由于作者水平有限, 书中不足之处在所难免, 望广大读者批评指正.

聂长海

2015 年 2 月

南京大学仙林校区和园

目　录

第 1 章　组合测试概论

组合测试是一种重要的软件测试方法. 软件测试是软件工程中必不可少的保证软件质量的重要环节. 软件工程是计算机领域一个重要的专业方向. 故组合测试理论与方法的研究和发展对于软件工程, 乃至整个计算机领域的发展都具有重要的意义. 本章从组合测试的研究背景入手, 介绍组合测试的基本概念、发展历程、基本理论和方法.

1.1　软件测试在软件工程中的位置

计算机软硬件新技术的不断发展, 已经促使计算机领域出现了很多新的专业方向. 例如, 除了传统的计算机科学、计算机工程和计算机应用专业等方向, 出现了网络工程专业 (目前已演变为物联网专业)、信息技术专业和软件工程专业 (图 1-1), 特别是软件工程专业受到空前重视. 在继我国 30 多所重点高等学校兴办软件学院培养专门人才之后, 软件工程已从原来的二级学科上升为一级学科.

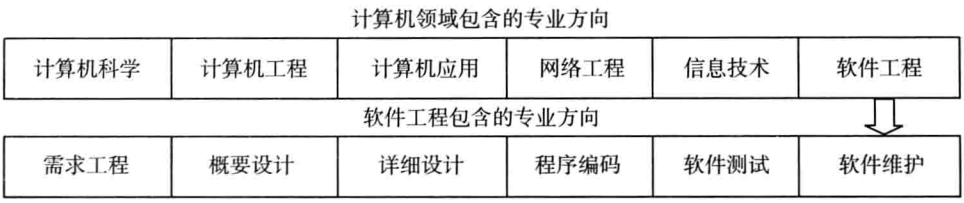

图 1-1　软件工程及其在计算机领域的位置

软件工程是一门研究如何以最经济的方式, 在最短的时间内开发用户最满意的软件产品的学科. 一般情况下, 软件工程根据软件过程可以分为需求分析、概要设计与详细设计、软件编码、软件测试和软件维护等. 可以看出, 软件测试是软件工程中一个重要的、必不可少的组成部分, 它其实是贯穿整个软件工程过程的[1].

整个软件工程可以说是一个不断与软件错误和缺陷斗争的过程. 为了生产高质量的软件, 软件工程为软件质量把好图 1-2 所示的 “四道关”. 第一道关是利用形式化方法、高度集成的软件开发环境和各种支持工具等在软件需求分析、设计和编码阶段预防可能出现的各种问题和错误. 这一步虽然非常重要和有效, 但一般不可能消灭所有错误, 仍有很多错误防不胜防地隐藏在软件中. 第二道关就是通过软件测试方法来发现和纠正这些没有防得住而隐藏在软件中的错误. 即使软件测试技术

已非常成熟, 也很难保证通过软件测试之后, 软件中就不再有错误, 因此, 需要第三道关. 通过容错计算技术, 在软件系统中植入容错能力, 使得即使在前两关中没有防得住或没有检查出来的错误存在于软件系统, 也不至于给系统造成重要损失. 第四道关是在前面三道关的基础上, 对软件系统仍然存在错误风险的预测, 即软件可靠性的研究[2].

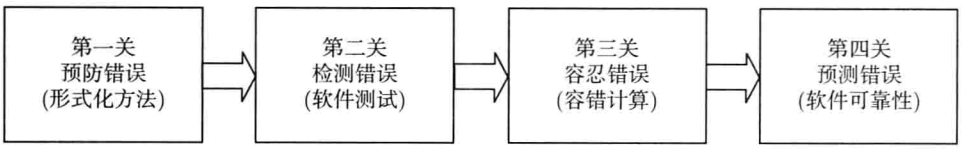

图 1-2　软件质量的 “四道关”

1.2　组合测试在软件测试中的位置

软件测试是指以发现错误、度量和提高软件质量为目的而理解、分析和运行程序的过程. 软件测试的目标是以最小的成本, 及时准确地发现软件中隐藏的错误, 从而提高软件的质量, 降低风险. 人们根据软件开发的不同阶段、不同的测试对象、基于不同的理论和角度提出了一系列软件测试方法, 形成了图 1-3 所示的软件测试方法体系, 具体可以有如下几类[3,4].

(1) 软件测试根据是否运行程序可分为静态测试和动态测试. 静态测试包括桌面检查、代码审查和代码走查等方法. 动态测试根据测试用例设计是否依据程序内部结构可以分为黑盒测试和白盒测试. 白盒测试包括语句覆盖、判定覆盖、条件覆

图 1-3　软件测试的方法体系

盖、判定/条件覆盖、条件组合覆盖、路径覆盖、线性代码序列和跳转测试等. 黑盒测试包括等价类划分、边际值分析、因果图分析、错误猜测和状态转换测试等.

(2) 根据软件开发的不同阶段, 可以将软件测试划分为单元测试、集成测试、系统测试、验收测试、回归测试、验证测试、确认测试、Alpha 测试、Beta 测试和 Gamma 测试等.

(3) 根据被测试软件的开发方法和应用环境的不同, 可以分为面向对象软件测试、面向方面软件测试、面向服务软件测试、基于构件软件测试、嵌入式软件测试、Web 应用软件测试、网构软件测试等, 后面还要出现普适计算环境软件测试、云计算和物联网环境软件测试等.

(4) 根据软件不同特性和方面的测试, 可以分为负载测试、压力测试、性能测试、安全性测试、安装测试、可用性测试、稳定性测试、授权测试、用户接受性测试、一致性测试、配置测试、文档测试、兼容性测试和 Playtest 等.

(5) 根据不同特殊的测试技术, 可以分为组合测试、蜕变测试、变异测试、演化测试、FUZZ 测试、基于性质的测试、基于故障的测试、基于模型的测试、基于操作剖面的测试、基于用例和/或用户陈述开发的测试、基于规格说明的测试、统计测试、逻辑测试、随机测试、自适应随机测试、GUI 测试、冒烟测试和探索测试等.

以上各种软件测试方法在文献 [4] 中有较为详细的介绍. 组合测试属于一种特殊的软件测试技术 (图 1-3 中已标出). 该方法旨在检测软件系统中各种因素及其相互作用引发的故障, 其中, 所谓的各种因素可以是软件系统的各种输入、功能选项、配置等. 将在 1.3 节具体介绍组合测试的应用场景.

1.3 组合测试的应用场景

软件系统是一个复杂的逻辑体, 它的正常运行受到很多因素的影响. 例如, 软件系统一般有很多输入, 这些输入以及它们之间的相互作用可能会影响系统的正常运行; 复杂的软件系统一般有很多功能选项, 这些功能选项以及相互之间的作用可能引发软件故障; 构件软件或者面向对象软件都是由多个构件或对象组成的, 这些构件或对象及它们之间的相互作用都可能促发软件错误. 为了确保软件在这些因素及其相互作用下正常工作, 需要设计测试用例对影响软件系统正常运行的因素及其相互作用进行系统检测. 以下列出一些典型的组合测试应用场景[5].

场景 1 Word 字体效果测试

图 1-4 中的 Word 字体效果部分有 11 个选项 (可以认为 11 个参数), 每个选项可以选择或不选择 (即每个参数有两个取值, 可以设为 1(选择) 和 0(不选择)). 如果要系统测试这 11 个效果选项及其相互作用的效果, 完全测试需要进行 $2\times2\times2\times2\times2\times2\times2\times2\times2\times2\times2=2^{11}=2048$ 次, 这是一个庞大的测试用例集合. 然而, 这里的

11 个选项的效果测试只是 Word 中很小的一个测试场景, 类似的测试场景还有很多, 将在后面章节中继续研究.

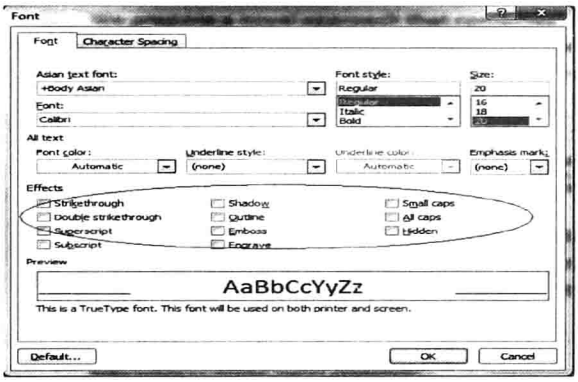

图 1-4　Word 字体效果测试测试场景

对图 1-4 中 11 个效果选项进行 2048 次完全测试不仅成本巨大, 而且也根本没有必要. 只要从这 2048 次测试中抽取少数几次就可以了. 组合测试方法就可以用来解决这个问题. 例如, 组合测试选择如图 1-5 所示的 18 条测试用例 (每一行是一条测试用例). 可以检查这 18 条测试用例具有一个非常重要的性质: 任意 3 个效果选项的 8 个组合都在这 18 条测试用例中至少出现过一次, 例如, 选项 1、2、3 的 8 种组合 (线圈标出), 选项 4、7、11 的 8 种组合都可以在这 18 条测试用例中找到 (线圈标出). 还有选项 5、6、7 的 8 种组合, 选项 8、9、10 的 8 种组合等任何 3 个选项的 8 种不同组合都在该 18 条测试用例中至少出现了一次.

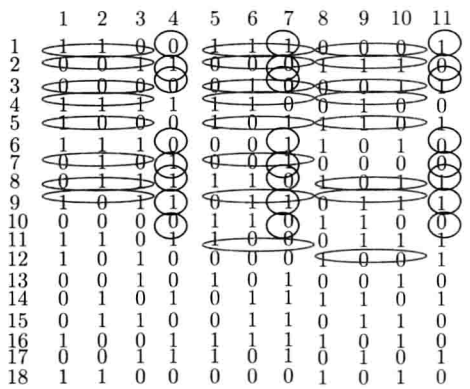

图 1-5　18 条测试用例覆盖任意 3 个选项间的 8 种组合

场景 2　控制系统的开关测试

图 1-6 是一个具有 34 个开关的控制系统, 完全测试需要 2^{34}(大约 1.7×10^{10}) 次

测试, 这几乎是不可能的. 在这样一个庞大的测试空间中, 选择少量的测试用例进行科学有效的测试, 显得尤为重要.

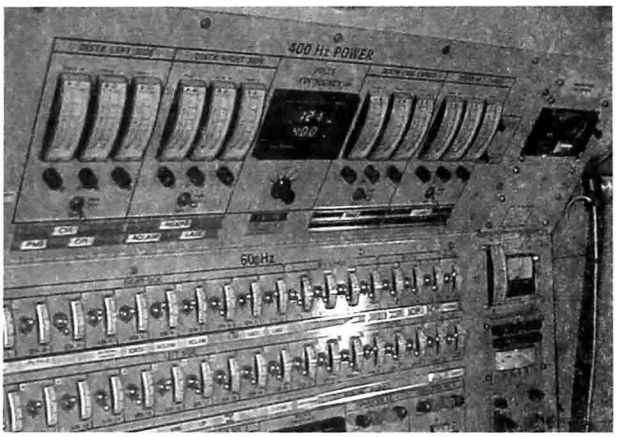

图 1-6 具有 34 个开关的控制系统

场景 3 机票预订

在网上预订机票时, 一般涉及以下选项: 航程类型 (可选单程、往返和联程 3 个值)、出发城市 (可以有上百个选择, 这里设为 10)、到达城市 (可以有上百个选择, 这里设为 10)、出发日期 (可以有上百个选择, 这里设为 10)、返回日期 (可以有上百个选择, 这里设为 10)、出行人数 (1~9 或更多, 这里设为 5)、送票城市 (可以有很多选择, 这里设为 5)、航空公司 (可以有很多选择, 这里设为 5)、舱位等级 (经济舱和商务舱 2 个选择)(图 1-7). 该机票查询系统的完全测试至少需要 $3\times10\times10\times10\times10\times5\times5\times5\times2=7.5\times10^6$ 次. 类似的场景还有酒店预订.

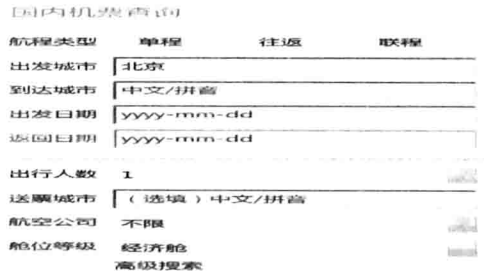

图 1-7 某网站机票预定界面测试

场景 4 软件系统配置测试

假设对某个待测软件进行配置测试时, 经过分析和处理, 最后确定需要测试 m 类硬件设备, 每类设备分别有 $a_1, a_2, \cdots, a_m$ 种硬件 (通过商标、型号和驱动程序