

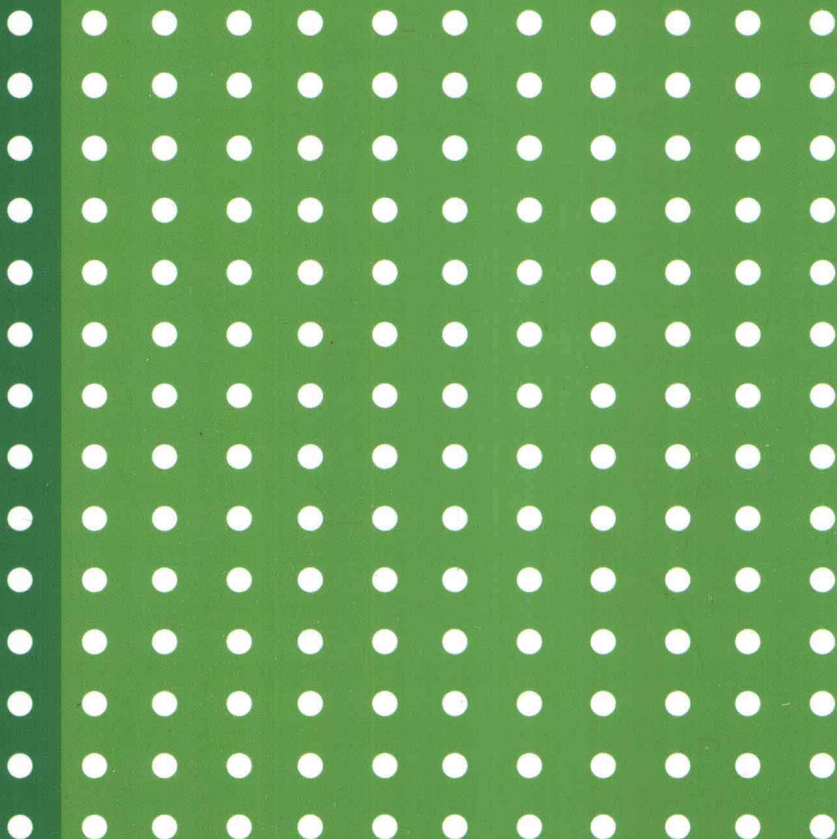


普通高等教育“十一五”国家级规划教材

重点大学软件工程规划系列教材

高级软件测试技术

杜庆峰 编著



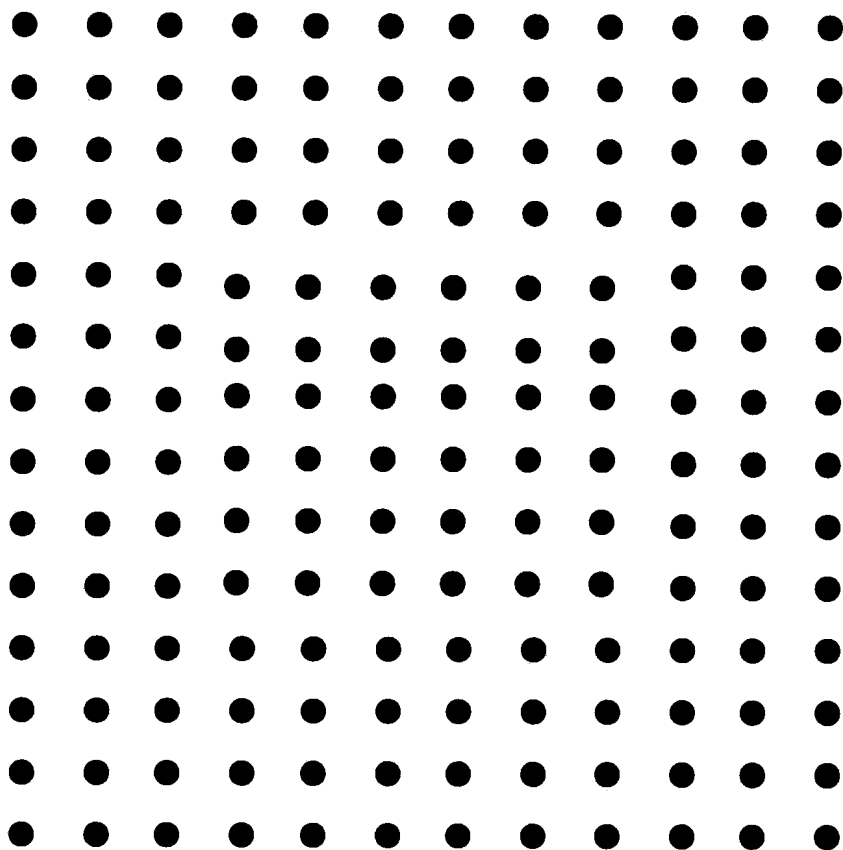
清华大学出版社

通高等教育“十一五”国家级规划教材

重点大学软件工程规划系列教材

高级软件测试技术

杜庆峰 编著



清华大学出版社

北京

内 容 简 介

本书详细地阐述了软件测试领域的基本理论、基本技术及专门测试领域的测试技术。首先,从软件测试的数学基础入手,介绍了和软件测试相关的基础知识;然后全面深入地介绍了静态测试技术和动态测试技术;在了解这些基础技术的基础上,再从单元测试、集成测试、系统测试及验收测试4个层面分析了如何进行测试的计划、用例分析和设计等过程;由于面向对象的测试技术是随着面向对象技术的发展在原有测试技术基础上发展起来的,本书专列一章重点介绍面向对象的测试技术;本书还用较长的篇幅阐述了专用应用系统测试,主要包括GUI测试、Web系统测试、数据库测试、嵌入式系统测试、负载压力测试及游戏测试;最后,讨论了软件测试管理和软件测试自动化技术并介绍了两个开源测试工具的使用方法。附录中收录了正交试验法的部分正交试验表和软件测试中英文术语对照表。

本书作为软件测试的理论、技术及应用参考书,不但突出基础知识和方法,而且阐述了一些高级的测试技术和方法,同时也注重测试技术的应用。作者在分析知识点的同时,均用实际的案例进行深入浅出的分析,使读者能更好地理解和掌握软件测试理论知识,并能很好地运用到实际测试工作中去。

本书可以作为不同层次高等院校的计算机、软件工程等相关专业的本科生、研究生的教学用书,也可作为软件测试技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

高级软件测试技术/杜庆峰编著. —北京:清华大学出版社,2011.6

(重点大学软件工程规划系列教材)

ISBN 978-7-302-26051-6

I. ①高… II. ①杜… III. ①软件—测试 IV. ①TP311.5

中国版本图书馆CIP数据核字(2011)第131815号

责任编辑:魏江江 顾冰

责任校对:梁毅

责任印制:何芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京季蜂印刷有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:28.5 字 数:715千字

版 次:2011年6月第1版 印 次:2011年6月第1次印刷

印 数:1~3000

定 价:49.00元

出版说明

随着信息时代的来临,软件已被广泛应用到工业、农业、商业、金融、科教卫生、国防、航空等各个领域,成为国民经济和社会信息化的一个基础性、战略性产业。因此,与之相关联的软件工程专业也越来越受到社会的关注。

从国际范围来看,1996年,美国Rochester技术大学(RIT)率先设立软件工程专业,其后美国、加拿大、英国和澳大利亚的许多大学相继跟进。1998年,ACM和IEEE-CS两大计算机学会联合设立软件工程教育项目(SWEEP),研究软件工程课程设置。2001年,IEEE和ACM发布CC2001教程,将计算(computing)学科划分为计算机科学、计算机工程、软件工程、信息系统和信息技术五个二级学科。2003年6月,《计算机课程——软件工程》(CCSE)大纲第一稿发表,后正式更名为《软件工程2004教程》(SE2004)。

在我国,教育部十分重视软件工程专业的发展。2001年,教育部和原国家计委联合下文,成立了35所示范性软件学院(全部下设于重点大学);2005年5月,教育部和清华大学出版社联合立项支持的研究课题组发布《中国软件工程学科教程》;同年,教育部组织编写了《软件工程专业规范》;2006年3月,在教育部高等学校教学指导委员会成立大会上,宣布成立软件工程专业教学指导分委员会。截至2007年初,全国有139所高等院校设立了软件工程专业。显然,软件工程已经成为一门迅速兴起的独立学科。

从我国的国民经济和社会发展来看,软件人才的需求非常迫切。随着国家信息化步伐的加快和我国高等教育规模的扩大,软件人才的培养不仅在数量的增加上也在质量的提高上对目前的软件工程专业教育提出更为迫切的要求,社会需要软件工程专业教学内容的更新周期越来越短,相应地,我国的软件工程专业教育在不断地发展和改革,而改革的目标和重点在于培养适应社会经济发展需要的、兼具研究能力和工程能力的高质量专业软件人才。

截至2007年,我国共有72个国家一级重点学科,绝大部分设置在教育部直属重点大学。重点大学的软件工程学科水平与科研氛围是培养一流软件人才的基础,而一流的学科专业教材的建设已成为目前重点大学学科建设的重要组成部分,一批具有学科方向特色优势的软件工程教材作为院校的重点建设项目成果得到肯定。清华大学出版社一向秉承清华的“中西兼容、古今贯通的治学主张,自强不息、厚德载物的人文

精神,严谨勤奋、求实创新的优良学风”。在教育部相关教学指导委员会专家的指导和帮助下,在国内许多重点大学的院系领导的大力支持下,清华大学出版社规划并出版本系列教材,以满足软件工程学科专业课程教学的需要,配合全国重点大学的软件工程学科建设,旨在将这些专业教育的优势得以充分的发扬,强调知识、能力与素质的系统体现,通过这套教材达到“汇聚学科精英、引领学科建设、培育专业英才”的目的。

本系列教材是在软件工程专业学科课程体系建设基本成熟的基础上总结、完善而成,力求充分体现科学性、先进性、工程性。根据几年来软件工程学科的发展与专业教育水平的稳步提高,经过认真的市场调研并参考教育部立项课题组的研究报告《中国软件工程学科教程》,我们初步确定了系列教材的总体框架,原则是突出专业核心课程的教材,兼顾具有专业教学特点的相关基础课程教材,探索具有发展潜力的新的专业课程教材。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

一、体现软件工程学科的发展和专业教育的改革,适应社会对现代软件工程人才的培养需求,教材内容坚持基本理论的扎实和清晰,反映基本理论和原理的综合应用,在其基础上强调工程实践环节,并及时反映教学体系的调整和教学内容的更新。

二、反映教学需要,促进教学发展。教材规划以新的专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

三、实施精品战略,突出重点。规划教材建设仍然把重点放在专业核心(基础)课程的教材建设;特别注意选择并安排了一部分原来基础较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型的专业教学内容和课程体系改革成果的教材。

四、支持一纲多本,合理配套。专业核心课和相关基础课的教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源的配套。

五、依靠专家,择优落实。在制订教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。

六、严格把关,质量为重。实行主编责任制,参与编写人员在编写工作实施前经过认真研讨确定大纲和编写体例,以保证本系列教材在整体上的技术领先与科学、规范。书稿完成后认真实行审稿程序,确保出书质量。

繁荣教材出版事业、提高教材质量的关键是教师。建立一支高水平的、以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材的教师能够加入到我们的编写队伍中来。

“重点大学软件工程规划系列教材”丛书编委会

联系人: 付弘宇 fuh@tup.tsinghua.edu.cn

前 言

软件已经应用到人们生活的各个领域,是生活中不可缺少的一部分。一方面,随着软件的普及,人们对软件质量的要求越来越高;另一方面,由于软件系统变得越来越庞大而复杂,如何提高软件质量是广大的技术人员所关注的,这使得软件开发人员和软件测试人员面临着巨大挑战。所以,保证软件质量是软件工程领域一直在深入研究的课题之一。

软件测试技术本身是不断发展的。目前,提高软件质量的方法就是在提高软件测试人员技术水平的同时规范并优化软件开发过程的管理。但软件测试在国内仍然处于发展的初期,在测试标准、测试计划的制定、测试方法的使用和推广、测试的组织和管理等方面处于不断发展完善阶段,处在一个“百家争鸣”的时期。在软件测试行业表面“蒸蒸日上”的现象背后,同时也存在着软件质量危机。

基于这种情况,国内许多高校的计算机、软件工程及其相关专业纷纷开设软件测试课程以培养更多的软件测试人才。目前,市场上的软件测试教材良莠不齐,精品教材少,尤其对软件测试技术介绍全面、深入的教材更少。为了适应当前教学和软件测试技术人员的需要,编者查阅了大量国内外有关软件测试方面的著作和文献,并结合自己多年的从业和教学经验编写了这本教材。

本书的特点是测试技术介绍全面,不但阐述了所有基本的软件测试技术,而且介绍了许多高级主题和专门应用系统的测试技术,并附有许多测试案例。

本书共分 12 章,第 1 章从讨论软件测试的数学基础入手,阐述了软件测试的发展史、软件测试的定义及基本原则等方面的基础知识;第 2 章介绍了静态测试技术;第 3 章全面地分析了动态测试技术,包括黑盒测试技术和白盒测试技术等;第 4~7 章分别介绍了单元测试、集成测试、系统测试及验收测试技术;第 8 章介绍了面向对象测试技术,讨论了面向对象测试的基本方法和策略等;第 9 章阐述了 GUI 测试、Web 系统测试、数据库测试、嵌入式系统测试、负载测试和游戏测试,这些均属于专门应用系统的测试;第 10 章全面分析了软件测试管理的过程;第 11 章介绍了软件测试自动化,分析了自动化测试和手工测试的优点与缺点等内容;第 12 章介绍了两个开源的测试工具 Junit 和 Jmeter 的使用。

本书由杜庆峰编著,在编写的过程中韩梅、李娜、董笑、龚柳青、王

怡、孙艺、于丹等为本书做了案例的程序调试和相关校对工作,在此一并致谢。

本书在编写过程中参阅了大量国内外同行的著作及文献,汲取了软件测试领域的最新知识。在此,对这些作者表示深深的感谢。

由于编者的水平有限、时间仓促,书中难免存在错误和不足之处,希望大家批评指正。

编 者

2010年12月

目 录

第 1 章 软件测试基础知识	1
1.1 软件测试的数学理论基础	1
1.1.1 集合论	1
1.1.2 函数	3
1.1.3 关系	4
1.1.4 命题逻辑	6
1.1.5 概率论	8
1.1.6 图论	9
1.2 软件测试的现状与发展	13
1.2.1 软件测试的现状	13
1.2.2 软件测试的发展趋势	15
1.3 软件测试和软件质量	16
1.3.1 什么是软件测试	16
1.3.2 什么是软件质量	17
1.3.3 软件测试和软件质量保证的区别	17
1.3.4 进行多少测试才终止	18
1.4 为什么需要软件测试	19
1.4.1 导致软件缺陷的原因	19
1.4.2 软件测试在软件开发、维护和使用中的角色	19
1.4.3 软件测试心理学	20
1.5 软件测试的目的	20
1.6 软件测试的原则	22
1.6.1 所有的软件测试都应追溯到用户需求	22
1.6.2 尽早地和不断地进行软件测试	22
1.6.3 不可能完全地测试	23
1.6.4 增量测试,由小到大	25
1.6.5 避免测试自己的程序	25
1.6.6 设计完善的测试用例	26
1.6.7 注意测试中的群集现象	27
1.6.8 确认缺陷的有效性	27
1.6.9 合理安排测试计划	27
1.6.10 进行回归测试	28

1.6.11	测试结果的统计和分析	28
1.6.12	及时更新测试	29
1.7	软件测试级别和模型	29
1.7.1	软件测试级别	29
1.7.2	软件测试生命周期	32
1.7.3	开发和测试模型	33
1.8	软件测试的类型	38
1.8.1	功能性测试	39
1.8.2	非功能性测试	39
1.8.3	软件结构性测试	40
1.8.4	变更相关的测试(再测试和回归测试)	41
1.9	基本的软件测试过程	42
练习		43
第2章	静态测试技术	44
2.1	静态测试技术概要	44
2.2	评审技术概述	44
2.2.1	评审的定义和类型	44
2.2.2	评审过程	45
2.2.3	责任和角色	45
2.3	代码检查	45
2.3.1	代码检查类型	45
2.3.2	代码检查内容	47
2.3.3	编码规范	47
2.3.4	代码缺陷检查表	55
2.4	正规技术评审	57
2.4.1	定义	57
2.4.2	技术评审的目的	57
2.4.3	评审小组成员	57
2.4.4	技术评审活动过程	58
2.4.5	技术评审注意事项	58
练习		59
第3章	动态测试技术	60
3.1	黑盒测试技术	60
3.1.1	边界值分析法	60
3.1.2	等价类测试法	63
3.1.3	错误推测法	69
3.1.4	因果图法	72
3.1.5	决策表测试法	76

3.1.6	Use Case 法(场景法)	80
3.1.7	正交实验法	84
3.1.8	黑盒测试方法选择的策略	97
3.2	白盒测试技术	97
3.2.1	白盒测试概念	97
3.2.2	程序结构分析	99
3.2.3	逻辑覆盖测试法	104
3.2.4	程序插装	115
3.2.5	其他白盒测试方法简介	120
3.2.6	白盒测试方法选择的策略	122
	练习	122
第 4 章	单元测试	124
4.1	单元测试概述	124
4.2	单元测试环境及过程	126
4.3	单元测试策略	135
4.4	单元测试的分析和用例设计	138
4.5	单元测试实例	142
	练习	158
第 5 章	集成测试	159
5.1	集成测试概述	159
5.2	集成测试环境及过程	161
5.3	集成测试方法	169
5.4	集成测试的分析和用例设计	180
5.5	集成测试实例	185
	练习	197
第 6 章	系统测试	198
6.1	系统测试概述	198
6.2	系统测试环境及过程	198
6.3	系统测试类型	201
6.4	系统测试的分析和用例设计	208
6.5	系统测试实例	212
	练习	221
第 7 章	验收测试	222
7.1	验收测试概述	222
7.2	验收测试过程	223
7.3	验收测试实例	225

练习	228
第 8 章 面向对象软件测试	229
8.1 面向对象技术简介	229
8.1.1 面向对象的概念	229
8.1.2 面向对象的开发模式	230
8.1.3 统一建模语言	230
8.2 面向对象测试模型	232
8.3 面向对象的用例设计方法	232
8.3.1 基于缺陷的测试	232
8.3.2 OO类的随机测试	233
8.3.3 类层次的划分测试	234
8.3.4 类行为模型测试	234
8.4 面向对象测试的策略	235
8.4.1 面向对象分析的测试	235
8.4.2 面向对象设计的测试	237
8.4.3 面向对象编程的测试	239
8.4.4 面向对象的单元测试	240
8.4.5 面向对象的集成测试	245
8.4.6 面向对象的系统测试	247
练习	249
第 9 章 专用应用系统测试	250
9.1 GUI 测试	250
9.1.1 GUI 测试概述	250
9.1.2 GUI 测试原则	250
9.1.3 GUI 测试内容	256
9.1.4 GUI 测试的设计及自动化	256
9.2 Web 应用系统测试	260
9.2.1 Web 系统基本组成	260
9.2.2 Web 应用系统测试综述	261
9.2.3 Web 应用系统测试的实施	262
9.3 数据库测试	278
9.3.1 数据库测试概述	278
9.3.2 数据库功能性测试	279
9.3.3 数据库性能测试与原因分析	281
9.3.4 数据库可靠及安全性测试	288
9.4 嵌入式系统测试	291
9.4.1 嵌入式系统介绍及测试基础	291
9.4.2 嵌入式测试策略	297

9.4.3	嵌入式测试环境的创建与实施	301
9.4.4	嵌入式测试的工具	306
9.4.5	嵌入式测试应用——手机测试	307
9.5	负载压力测试	312
9.5.1	负载压力测试基础	312
9.5.2	负载压力测试的策略	314
9.5.3	负载压力测试的解决方案和实施	315
9.5.4	负载压力测试结果分析	337
9.5.5	系统性能调优	341
9.6	游戏测试	345
9.6.1	游戏测试的基本概念	345
9.6.2	游戏测试的主要内容	347
9.6.3	游戏测试的实施	348
	练习	357
第 10 章	软件测试管理	358
10.1	测试计划的制定与估算	358
10.2	测试的组织	361
10.3	测试过程监控	364
10.4	测试文档	369
10.5	软件配置管理	369
10.6	测试与风险	373
10.7	缺陷管理	375
	练习	380
第 11 章	软件测试工具	381
11.1	测试工具的类型	381
11.1.1	测试管理工具	381
11.1.2	静态测试工具	382
11.1.3	功能测试执行工具	384
11.1.4	覆盖工具(结构性测试工具)	385
11.1.5	性能和压力测试工具	385
11.1.6	其他工具	386
11.1.7	测试工具的选择	387
11.2	测试自动化与手工测试	388
11.2.1	手动测试的不足	388
11.2.2	自动测试的优点	389
11.2.3	自动测试的局限性	389
	练习	390

第 12 章 开源测试工具应用	391
12.1 单元开源测试工具 JUnit	391
12.1.1 JUnit 简介	391
12.1.2 安装	393
12.1.3 使用	393
12.2 开源负载压力测试工具 Jmeter	403
12.2.1 Jmeter 的基本框架	403
12.2.2 常用测试范围	404
12.2.3 Web 服务器测试	404
附录 A 部分正交实验表	408
附录 B 软件测试中英文术语对照表	415
参考文献	442

软件测试基础知识

1.1 软件测试的数学理论基础

正如软件开发过程是一个需要逻辑思维的过程一样,软件测试也同样需要掌握和使用一些数学理论和数学分析方法。数学是一切科学的基础,软件测试技术中运用的这些数学知识将使软件测试本身更严谨、精确和高效。本节介绍用于软件测试技术中的数学基础知识。

从软件测试设计方法的角度,软件测试一般分为功能性测试(也称黑盒测试,Black-Box Testing)和结构性测试(也称白盒测试,White-Box Testing)。一般而言,在功能性测试领域中,通常要运用到离散数学知识;而在结构性测试领域中,则要用到一些有关图论的知识。

1.1.1 集合论

集合论是数学的一个基本分支学科,研究对象是一般集合。集合论在数学中占有一个独特的地位,它的基本概念已渗透到数学的所有领域。这里的集合是指由一些抽象的数学对象构成的整体。集合、元素和成员关系是数学中最基本的概念。集和论(加上逻辑和谓词演算)是数学的公理化基础之一,通过集合及成员关系来形式化地表示其他数学对象。

例如,小于 10 的所有奇数,应用集合论的表示法可以写成 $M_1 = \{1, 3, 5, 7, 9\}$,读作“ M_1 是元素为 1、3、5、7、9 的集合”; $M_2 = \{\text{所有参加 2008 年北京奥运会的中国代表团成员}\}$ 。

集合中包含元素或成员。

对于集合必须注意以下方面:

(1) 集合的元素是确定的,对于集合,任一个元素要么属于该集合,要么不属于该集合,两者必居其一。若一个元素是集合成员,用 \in 表示;若不是集合成员,则用 \notin 表示。如 $3 \in M_1, 9 \notin M_2$ 。

(2) 集合中的每个元素均相同,即集合 $\{1, 3, 5, 7, 9\}$ 与集合 $\{1, 3, 5, 7, 9, 9\}$ 是一样的。

集合 M_1 由元素 1, 3, 5, 7, 9 组成,可写成 $M_1 = \{1, 3, 5, 7, 9\}$,这种

将集合的所有元素一一列出的表示法叫做“枚举法”。但有时也可以只列出一部分元素,可以通过决策规则推出其余的部分,如 $M_1 = \{1, 3, 5, 7, 9, \dots\}$ 表示奇数的集合。列出所有元素的方法只适合少量元素的集合或元素符合某种明显模式的集合。集合还可以有另外一种方法表示,就是用一个集合所具有的共同性质来描述这个集合,如 $N = \{t: t \text{ 是所有获得奖牌的中国运动员}\}$ 。这种使用决策规则的方法的主要缺点是逻辑上有些复杂,尤其是当采用谓词演算时,而且当决策规则用于自身的时候会出现循环问题,如著名的理发师问题: $\{\text{为他人理发而不为自己理发的人}\}$,但是测试人员很少使用自引用。

对于集合元素的个数不作任何限制,它可以是有限个或无限个。一个集合若是由有限个元素组成,则叫做有限集合;一个集合由无限个元素组成,则叫做无限集合。特别情况下,元素个数为 0 的集合叫做空集。空集在集合论中有重要的位置,空集不包含元素。空集是唯一的,如 $\phi, \{\phi\}, \{\{\phi\}\}$ 都是不同集合。如果集合基于某种决策规则的定义永远是不成立的,那么该集合就是空集,如 $\phi = \{\text{年: } 2012 \leq \text{年} \leq 1812\}$ 。

集合论的运算主要表现在集合的基本操作上:并、交、差和补。其定义如下:

- 由集合 A, B 中所有元素合并组成的集合叫做集合 A 与 B 的并集,记作 $A \cup B$ 。
- 由集合 A, B 中所有的公共元素组成的集合叫做 A 与 B 的交集,记作 $A \cap B$ 。
- 假设有两个集合 A, B ,差集就是属于 A 但是不属于 B 的元素的集合,记作 $A - B$ 。
- 对于一个集合 A 和全集 B 来说, A 的补集是在全集 B 中所有不属于 A 的元素。

还可以由差集定义对称差。集合 A, B 的对称差(或叫布尔和) $A + B$ 可定义为 $A + B = (A - B) \cup (B - A)$ 。

使用集合操作即通过现有集合操作构建新集合,通过定义的新集合来确定新集合和旧集合之间是如何关联的。给定两个集合 A, B ,定义三种基本集合关系:

- A 是 B 的子集,记做 $A \subseteq B$,当且仅当 $a \in A \Rightarrow a \in B$ 。
- A 是 B 的真子集,记做 $A \subset B$,如果 $A \subseteq B$ 且存在 b ,使得 $b \in B$ 但 $b \notin A$,则称 A 是 B 的真子集。
- A 和 B 是相等集合,记做 $A = B$,当且仅当 $A \subseteq B$ 且 $B \subseteq A$ 。

集合的划分是一种非常特殊的情况,对于测试人员非常重要。“划分”的含义就是将一个整体分成小块,使得符合某种性质的所有事物都在某个小块中,不会遗漏。在日常生活中,将一套房子划分成不同的独立房间、将一个行政区域划分成不同的独立县市、将一个外部接口格式划分成不同规格的独立含义的数据表示区域等均是划分的例子。其定义如下:

给定集合 A ,以及 A 的一组子集 A_1, A_2, \dots, A_n ,这些子集是 A 的一种划分,当且仅当 $A_1 \cup A_2 \cup \dots \cup A_n = A$,且 $i \neq j \Rightarrow A_i \cap A_j = \phi$ 。图 1-1 就是集合与其子集的关系图。

集合的一种划分形成了一组子集,因此常常把单个子集看作是划分的元素。“划分”的定义对于软件测试人员非常有用,一方面保证 A 的所有元素都在某个子集中,另一方面保证任意一个元素都不在两个子集中。例如,对学生集合的一种划分,可以包括小学生、中学生、大学生、研究生 4 个子集元素。

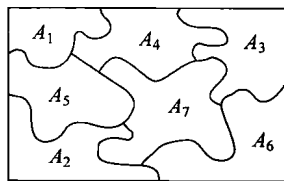


图 1-1 集合与子集的例子

集合的“划分”对软件测试会产生重要保证:完备性和无冗余性。因此“划分”的概念对于测试人员要灵活掌握。对于功能性测试而言,其固有弱点是对所测试的功能点可能存在漏洞和冗余性,如学生成绩管理系统中教师更新学生成绩这个功能可能包括更新成绩成功;学生

信息不存在,更新成绩失败;学生信息存在但成绩不存在,更新成绩失败三个功能点。而另一方面,一些功能可能被重复测试,所以,如何对被测试软件进行功能的合理“划分”很重要。再例如,对于银行系统的测试,可以把银行系统的账户看成一个集合,而不同的账户类型可以看成该集合的不同子集,对不同账户的子集的划分是对不同账户进行功能性测试的完备性和无冗余性的体现。

1.1.2 函数

函数表示每个输入值对应唯一输出值的一种对应关系。函数 f 中对应输入值 x 的输出值的标准符号为 $f(x)$ 。包含某个函数所有输入值的集合被称作这个函数的定义域,包含所有输出值的集合被称作值域。若先定义映射的概念,可以简单定义函数为:定义在非空数集之间的映射称为函数。函数是软件开发和测试的重要核心概念,所有功能性测试的基础都是函数。

可以将函数看成“黑盒”,它将有效的输入值转换为唯一的输出值。通常将输入值称作函数的参数,将输出值称作函数的值。例如,可以把网上订票系统看成函数,其中的订票时间、起止地点、乘机人等信息是函数的输入,函数的输出可能是相关显示信息和机票。

软件或程序都可以看做是将其输入与输出关联起来的函数。用数学公式表示函数,输入是函数的定义域,输出是函数的值域。其定义如下:

给定的集合 A 和 B ,函数 f 是 $A \times B$ (A 与 B 的笛卡儿积)的一个子集,使得对于 $a_i, a_j \in A, b_i, b_j \in B$,对于 $f(a_i) = b_i, f(a_j) = b_j$,若 $b_i \neq b_j$,则推出 $a_i \neq a_j$ 。

在上面的定义中,输入值的集合 A 被称为 f 的定义域;可能的输出值的集合 B 被称为 f 的对映域。函数的值域是指定义域中全部元素通过映射 f 得到的实际输出值的集合。注意,把对映域称作值域是不正确的,函数的值域是函数的对映域的子集。

在计算机科学中,参数和返回值的数据类型分别确定了程序、子程序、模块的定义域和对映域。因此,定义域和对映域是函数一开始就确定的强制约束。另一方面,值域和程序、子程序和模块的具体实现有关。

函数可分为:

(1) 单射函数:将不同的变量映射到不同的值。即若 x 和 y 属于定义域,则仅当 $x = y$ 时有 $f(x) = f(y)$ 。

(2) 满射函数:其值域即为其对映域。即对映射 f 的对映域中的任意 y ,都存在至少一个 x 满足 $f(x) = y$ 。

(3) 双射函数:既是单射的又是满射的,也叫一一对应。双射函数经常被用于表明集合 x 和 y 是等势的,即有一样的基数。如果在两个集合之间可以建立一个一一对应关系,则说这两个集合等势。

在实际的测试中,软件、程序、子程序、模块和单射函数、满射函数及双射函数具有相似的对对应关系,对于测试有很重要的启发。

复合函数在软件开发中也有着很多实践意义。假设有下列集合和函数,使得一个函数的值域是另一个函数的定义域:

$$F: A \rightarrow B$$

$$g: B \rightarrow C$$

$$h: C \rightarrow D$$

设 $a \in A, b \in B, c \in C, d \in D$, 并且 $f(a)=b, g(b)=c, h(c)=d$, 则函数 h, g 和 f 的复合为:

$$h \circ g \circ f(a) = h(g(f(a))) = h(g(b)) = h(c) = d$$

例如, 一个软件系统, 通过销售人员的销售额、请假次数、级别和工作时间长短作为标准进行定量考评, 根据考评结果来计算年终奖, 这就是一个复合函数问题:

$$F_1(\text{销售额, 请假次数, 级别, 工作时间长短}) = \text{考评结果}$$

$$F_2(\text{考评结果}) = \text{年终奖}$$

由于复合函数会出现定义域/值域兼容性的问题, 即上面复合函数 $g(b)$ 中 b 可能取多个值的问题, 这可能会对测试人员造成困扰。对于测试人员有帮助的一面是, 对于给定函数, 其逆函数能充当某种“交叉检查”的角色, 而这常常有助于功能性测试用例的设计。

1.1.3 关系

从哲学的角度, 关系是反映事物及其特性之间的相互联系, 是不同事物、特性的一种统一形式。世界上的任何事物都同它周围的事物相互联系着, 这种联系表明它们彼此存在着一致性、共同性, 在此基础上形成不同的事物、特性的统一形式, 即表现为一定的关系。诸如家庭关系、社会关系、人际关系、组织关系和社团关系等。就人而言, 可以说没有人不处在各种关系交织的网络之中。

从数学的角度, 对于函数, 规定定义域元素不能与多个值域元素关联, “函数”意味着事物之间存在着某种确定的关系表示。并不是所有关系都是严格意义上的函数, 如果将零件种类和产品的关系看作是零件种类集合与产品集合之间的映射, 一种零件可以装配多种产品, 一种产品有多种类型的零件装配, 这是一种多对多的映射。而函数与关系都是以集合论为基础的。

函数是关系的一种特例, 两者都是笛卡儿积的子集。

集合之间的关系定义如下:

给定两个集合 A 和 B , 关系 R 是笛卡儿积 $A \times B$ 的一个子集。

如果希望描述整个关系, 则通常只写 $R \subseteq A \times B$ 。对于特定元素 $a_i \in A, b_i \in B$, 记作 $a_i R b_i$ 。有关关系的详细论述这里从略, 我们更有兴趣的是在计算机领域关系是数据建模和面向对象分析的基础, 也是软件测试用例设计的依据。

下面解释一个术语: 势。势在用于集合时, 是指集合中元素的个数。由于关系也是集合, 因此可以期望关系的势是指有多少有序对偶在集合 $R \subseteq A \times B$ 中。但是, 实际上并非如此。关系的势的定义为:

给定两个集合 A 和 B , 一个关系 $R \subseteq A \times B$, 关系 R 的势是:

- 一对一势: 当且仅当 R 是 A 到 B 的一对一函数。
- 多对一势: 当且仅当 R 是 A 到 B 的多对一函数。
- 一对多势: 当且仅当至少有一个元素 $a \in A$ 在 R 中的两个有序对偶中, 即 $\langle a, b_i \rangle \in R$ 和 $\langle a, b_j \rangle \in R$ 。
- 多对多势: 当且仅当至少有一个元素 $a \in A$ 在 R 中的两个有序对偶中, 即 $\langle a, b_i \rangle \in R$ 和 $\langle a, b_j \rangle \in R$ 。并且至少有一个元素 $b \in B$ 在 R 中的两个有序对偶中, 即 $\langle a_i, b \rangle \in R$ 和 $\langle a_j, b \rangle \in R$ 。

函数映射到值域上或值域之间的差别可以与关系类比, 这就是参与的概念, 其定义为: 给定两个集合 A 和 B , 一个关系 $R \subseteq A \times B$, 关系 R 的参与是:

- 全参与: 当且仅当 A 中的所有元素都在 R 的某个有序对偶中。