

# 几种范畴划分测试充分性准则的比较<sup>\*</sup>

朱彬 刘玲 缪淮扣

(上海大学计算机工程与科学学院 上海200072)

**摘要** 选择测试充分性准则对软件的测试是至关重要的。本文对范畴划分(Category-Partition)测试中的 All-Combination、Each-Choice-Used 和 base-Choice-Coverage 这三个充分性准则在揭错能力、软件可靠性、测试开销这几个方面进行分析和比较,并给出一个量化的比较结果,帮助测试员选择测试充分性准则。最后对基于票据计算的形式规格说明的例子进行了分析。

**关键词** 范畴划分测试,测试充分性准则,揭错能力,软件可靠性,测试开销

## A Comparison of Testing Adequacy Criteria in Category-Partition Testing

ZHU Bin LIU Ling MIAO Huai-Kou

(School of Computer Engineering and Science, Shanghai University, Shanghai, 200072, China)

**Abstract** It is very important choose the testing adequacy criteria in software testing. There exist three testing adequacy criteria in category-partition testing: All-Combination, Each-Choice-Used and base-Choice-Coverage. This paper gives an analysis and comparison among these three criteria. Moreover, a quantitative result of comparison on fault-detecting ability, software reliability and test cost is produced to assist the test engineer in choosing testing adequacy criteria. Finally, this paper gives an analysis by an example of a formal specification based on invoice computing.

**Keywords** Category-partition testing, Testing adequacy criteria, Fault-detecting ability, Software reliability, Test cost

### 1 引言

软件测试在整个软件开发过程中占有举足轻重的地位,它是保证软件质量的关键技术。所谓软件测试,就是选出有代表性的输入数据来驱动被测程序,观测程序的执行结果,比较该结果与期望的结果是否一致,然后做出相应的纠错、调整和评估。

软件测试的传统过程是基于代码的。Goodenough 和 Gerhart 已经认识到基于代码实现的测试从根本上来讲是有缺陷的<sup>[9]</sup>,因为它无法确认程序是否符合规格说明和用户的需求,而这一点正是测试的目的和最终的目标。形式化规格说明精确描述了“做什么”而不是“怎么做”,因此,测试者可以从中获得重要的信息而无需从繁琐以及不重要的细节中提取。基于规格说明的测试与传统的基于代码的测试相比,它提供了一种更严格的方法,并且简化了回归测试。基于规格说明测试的其他优点包括可以从测试数据中得出预期值,以及测试的开发和程序的设计与运行是同步的。后者更是有利于打破软件工程中的“先编码后测试”的实际做法,并且有利于对软件的生命周期中所有阶段并行地进行测试活动<sup>[3,10]</sup>。

Z 语言是目前人们非常感兴趣的一种规格说明方法。它以一阶谓词逻辑和集合论作为形式语言基础将函数、映射、关系等数学方法用于规格说明中,具有精确简洁、无二义性的优点,有利于保证程序的正确性,尤其适合于无法进行现场调试的高安全性系统的开发。

本文所讨论的范畴划分测试是一种基于规格说明的测试方法。用基于规格说明的测试方法进行测试时,完全不考虑程

序的内部特性和内部结构,测试者只知道输入和输出之间的关系,或是程序的功能。他必须依靠能够反映这种关系和程序功能的需求规格说明来确定测试数据集,并推断测试结果。可以说,基于规格说明的测试是从用户的观点出发的测试。

软件测试充分性这一概念是 Goodenough 和 Gerhart 最早提出的。测试充分性准则是一组可用于判断是否进行了充分测试的规则,也可指导测试数据的选择,明确地说明如何选择测试数据。使用不同的充分性准则进行程序的测试,所付出的测试开销和得到的软件的质量是不同的。所以对测试充分性准则进行分析和比较是非常有意义的,可以自动产生测试数据集,提高软件的质量和可靠性,节约开发软件的成本。

本文对范畴划分测试中的 All-Combination、Each-Choice-Used 和 Base-Choice-Coverage 这三个充分性准则进行了分析和比较。其主要目的是,帮助软件测试员按照自身的需要和条件选择产生测试数据集的准则,比如,要求软件可靠性要高,而测试开销没有特别要求的,根据比较,选择的准则应该是通过该准则测试的软件,可靠性较高,反之就可选用测试开销低的准则。

目前已有对测试充分性准则和自动产生测试数据集的研究。Amla 和 Ammann<sup>[11]</sup>描述了一种在 Z 规格说明中抽取信息来进行范畴划分测试的方法。Paul Ammann and Jeff Offutt<sup>[6]</sup>也提出了用形式化方法在范畴划分测试中获得测试框架得一种方法。HONG ZHU<sup>[1,2]</sup>对各种测试充分准则用公理系统进行了比较和评估。B. Haworth and M. Roper<sup>[12]</sup>讲到了面向对象测试中测试性准则。

第2节将回顾 Z 语言与基于规格说明的测试。范畴划分

<sup>\*</sup> 本文受国家自然科学基金(编号60173030)和上海市教委青年基金(编号02AQ85)支持。

测试将在第3节介绍。第4节中将讨论测试充分性准则。第5节将给出测试数据充分性准则的比较。第6节对范畴划分测试的三个准则进行比较,并给出实例进行分析。最后将总结我们的结论和发现。

## 2 Z语言与一个Z规格说明的例子

Z语言主要结构是模式,每个模式由变量说明和谓词两部分组成。模式可分为状态模式和操作模式。状态模式是对系统的状态空间及其约束特性的描述,是系统最基本的模式,而操作模式描述了系统某部分的行为特征,它通过描述操作前的状态值和操作后状态值之间的关系,来定义系统的一种操作。

下面描述一个本文要分析的票据计算模型。假定某公司生产两种产品,X和Y,每件X的价格为5元,每件Y的价格为10元。一个需求的定单包括一定数目的X和Y。采购的花费按以下说明打折,如果总价大于200元,折扣为5%,如果总价大于1000元,折扣为20%,公司为了促销X,如果订购X的数目大于30则再折扣10%。采购产品最终应支付的金额可根据两种折扣来计算,设产品X的数量为x,产品Y的数量为y,则计算方法概括如下:

- 1)当 $5x + 10y \leq 200$ 时,第一种折扣率d1为0
- 2)当 $200 < 5x + 10y \leq 1000$ 时,第一种折扣率d1为5%
- 3)当 $5x + 10y > 1000$ 时,第一种折扣率d1为20%
- 4)当 $x \leq 30$ 时,第二种折扣率d2为0
- 5)当 $x > 30$ 时,第二种折扣率d2为10%

将1),2),3)与4),5)相组合可有六种情况,用Z语言描述的规格说明如下:

DisSys

discount1: N  
discount2: N  
sum: N

Discount

ΔDisSys  
y?: N  
x?: N  
total!: N

sum' = 5\*x? + 10\*y?  
sum' ≤ 200 ⇒ discount1' = 100  
(sum' > 200) ∧ (sum' ≤ 1000) ⇒ discount1' = 95  
(sum' > 1000) ⇒ discount1' = 80  
x? ≤ 30 ⇒ discount2' = 100  
x? > 30 ⇒ discount2' = 90  
total! = sum' \* discount1' \* discount2' / 10000

模式上方为声明部分,模式下方为谓词部分,状态模式的谓词给出了状态不变式,它决定了这个系统的特征。第一个模式是状态模式,该模式中状态不变式是缺省的,第二个模式是操作模式。x?, y?为输入变量, total!为输出变量,谓词部分的前一个式子是表示输入变量的前置条件。即进行一个操作时输入变量要满足的条件。后一个式子反映了输入、输出之间的约束关系。关于Z语言的细节可参阅文[3]。

## 3 范畴划分的方法

范畴划分方法<sup>[6,8]</sup>是一种基于规格说明的测试方法,它根

据对输入/输出的分析来产生测试数据集。范畴划分方法为测试员提供了产生测试数据集的基本过程。测试员的主要工作是确定范畴(category)。这些范畴是根据被测试功能的输入域的主要特征定义的,范畴不相交,且必须导出不同的输出。一个范畴是一个参数值的子集,它确定了一个特殊的行为或输出。一个范畴又被划分为若干个选择(choise)。根据上述定义,一个选择是一个对范畴的特定测试值。任何合理的标准都可用于产生选择,一个范畴中的选择是不相交的,并且每个范畴中所有的选择的并集必须覆盖整个输入域。用范畴划分测试方法产生测试数据集的步骤如下:

- a. 分析规格说明找出可以单独测试的单元
- b. 确定输入域和影响测试单元功能的参数和环境变量
- c. 确定范畴,即参数和环境变量的重要特征
- d. 把每个范畴划分成若干个选择
- e. 根据所选的测试充分性准则产生选择的组合(测试数据集)并用具体的值实例化测试数据集

本文所关注的是各个测试充分性准则的比较,使测试员可以根据比较的结果来选择充分性准则,从而产生测试数据集。

不同范畴选择之间的每一种组合都将产生一个测试框架(test frame)。例如,根据规格说明划分为两个范畴X和Y,并且X有两个选择,Y有三个选择:

范畴	X	Y
	* P <sub>1</sub>	* Q <sub>1</sub>
	* P <sub>2</sub>	* Q <sub>2</sub>
	* P <sub>3</sub>	

这里任何一个选择P<sub>i</sub>和选择Q<sub>j</sub>的组合[P<sub>i</sub>, Q<sub>j</sub>]都是一个可能的测试框架。称之为可能的测试框架是因为这些选择的组合可能产生的交集是空的。像上面的例子产生的可能测试框架有六个。在实际中,测试员通常要在可能的测试框架中排除某些选择的组合。排除某些选择的组合出于两个方面的考虑。首先,测试员所选用的产生测试数据集的测试充分性准则不同。其次,某些选择的组合为空。排除选择的组合是一个较难解决的问题,定理证明系统可以帮助测试员解决部分此类问题<sup>[6]</sup>。

## 4 测试充分性准则

### 4.1 什么是测试的充分性准则

在本文的引言部分已经提到测试充分性准则有三种不同的形式定义<sup>[1,2]</sup>。在介绍三种测试充分性准则之前先介绍一下各个符号所代表的含义。设D为一个可数的集合,它代表软件的输入、输出数据的集合;P为D上一个可计算函数集合的子集,它代表以D中的数据为输入和输出的程序集合;S为D上的二元关系的一个子集,它代表软件的规格说明的集合;T = 2<sup>P</sup>,它是测试数据集的集合。

•测试充分性判断准则 也谓谓词形式的充分性准则,该准则可以作为停机规则。

定义1(充分性准则谓词判定) 一个测试充分性准则C定义为

$$C: P \times S \times T \rightarrow \{true, false\}.$$

$C(p, s, t) = true$ 表示用测试数据集t来测试对应于规格说明s的程序p是充分的。否则,该测试是不充分的。

•测试充分性度量准则 也称度量函数形式的充分性准

则。

**定义2(度量函数形式的充分性准则)** 一个软件测试的充分性标准  $C$  是一个从测试数据集合、被测程序及其功能规格说明到实数区间  $[0,1]$  的函数。

$$C: P \times S \times T \rightarrow [0,1].$$

$C(p,s,t)=r$  (其中  $r$  是属于  $[0,1]$  的实数) 表示程序  $p$  相对于规格说明  $s$  在测试数据集合  $t$  上的测试充分度为  $r$ ,  $r$  越大, 充分度就越高。

从上可以看出谓词形式的充分性准则是度量形式的充分性准则的特例, 因为可以把谓词形式的充分性准则的值看作是集合  $\{0,1\}$ , 关于细节可参阅文[1]。

#### •测试数据选择准则

**定义3(测试数据选择准则)** 一个测试数据充分性准则  $C$  是一个从被测程序  $p$  所对应的规格说明  $s$  到一组测试数据集合的映射

$$P \times S \rightarrow 2^T.$$

$t \in C(p,s)$  表示测试数据集合  $t$  对于测试程序  $p$  (相对于规格说明  $s$ ) 来说是充分的, 否则就是不充分的。

#### 4.2 范畴划分测试的三个充分性准则

先假定  $x_1, x_2, \dots, x_n$  是被测单元中根据参数和环境变量中的特征划分成的范畴。假定范畴  $x_i$  所对应的选择是  $A_{i,1}, A_{i,2}, \dots, A_{i,k_i}, k_i > 0, i=1, 2, 3, \dots, n$ 。设

$$C = \{A_{1,u_1} \times A_{2,u_2} \times \dots \times A_{n,u_n} \mid 1 \leq u_i \leq k_i \text{ and } 1 \leq i \leq n\}$$

其中  $k_i$  表示第  $i$  个范畴的选择个数,  $n$  表示总的范畴个数。

那么  $C$  就是各范畴之间所有选择的组合, 三个充分性准则的定义如下:

#### •All-Combination 准则

**定义4** 如果对于所有的  $c \in C$ , 至少存在一个  $t \in T$  使得  $t \in c$ , 那么测试数据集合  $T$  满足 All-Combination 准则。

也就是说测试数据集合  $T$  包含了各范畴之间的所有选择的组合。

#### •Each-Choice-Used 准则

**定义5** 如果  $E = \{e \mid e \in C \text{ and } \exists t \in T, (t \in E)\}$ , 它满足下列条件:

$$\forall i, (1 \leq i \leq n \Rightarrow (E_i = \{A_{i,1}, A_{i,2}, \dots, A_{i,k_i}\}))$$

其中,  $E_i = \{e \mid \exists X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n, (X_1 \times \dots \times X_{i-1} \times e \times X_{i+1} \times \dots \times X_n \in E)\}$ 。

那么测试数据集  $T$  满足 Each-Choice-Used 准则。也就是范畴中的每一个选择要被选中一次。

#### •base-Choice-Coverage 准则

基本选择 (base choice) 是参数和环境变量的组合, 表示被测测试单元的正常操作, 所以基本选择的测试数据集对比较正常的操作是有用的。我们假设  $A_{1,1} \times A_{2,1} \times \dots \times A_{n,1}$  是基本选择。

**定义6** 如果  $E = \{e \mid e \in C \text{ and } \exists t \in T, (t \in E)\}$  满足下列条件:

$$E \supseteq \bigcup_{j=1}^K B_j$$

其中,  $B_j = \{A_{1,1} \times \dots \times A_{i-1,1} \times A_{i,j} \times A_{i+1,1} \times \dots \times A_{n,1} \mid j=1, 2, \dots, K_i\}$ 。

那么称测试数据集集合  $T$  满足 Base-Choice-Coverage 准则。也就是说对于范畴中的每一个选择用其它范畴中基本选择和它进行组合, 这样使得每一个非基本的选择至少一次, 而基本的选择将使用几次。

## 5 测试数据充分性准则的比较

测试充分性准则之间的比较是非常有意义的。它可以帮助测试人员更好地选择测试充分性准则, 从而提高软件的质量, 降低测试开销。测试充分性准则在软件测试中主要在揭错能力、软件可靠性和测试开销这三个方面进行分析和比较。

### 5.1 测试开销

测试是软件开发中一项昂贵的活动。测试开销和完成某种充分性测试而选择的测试充分性准则密切相关。测试开销的比较涉及很多因素。一种简单的对测试开销的比较就是一种充分性测试集合的大小。测试集合大的测试开销大, 测试集合小的测试开销小。Weyuker<sup>[13]</sup> 研究的对数据流充分准则复杂度分析属于这种类型。

### 5.2 软件的可靠性

通过某种充分性测试准则的软件可靠性也是对该充分性规则的一个直接度量。但是在软件可靠性方面对充分性准则进行比较是非常困难的<sup>[2]</sup>。

软件可靠性是衡量软件质量最重要的指标和软件开发的最终目标。1983年美国 IEEE 计算机学会软件工程技术委员会将软件可靠性定义如下:

1. 在规定的条件下、规定的时间内, 软件不引起系统失效的概率。该概率是系统输入和系统使用的函数, 也是软件中存在的错误的函数。

2. 在规定的时期内, 在所述条件下程序执行要求的功能的能力。

软件可靠性模型是软件可靠性定量分析的技术基础, 软件可靠性模型包括, 理论假设 (假设软件故障行为的方式, 亦即软件故障行为的方式, 亦即对故障强度与时间的关系的一种数学描述, 一般多为一些常用的概率分布), 模型体 (数学描述, 用以估计当前软件可靠性, 预测将来某时刻软件可靠性, 软件中剩余错误个数等。目前出现了很多种不同的模型, 在这里要介绍的是 Jelinski-Moranda 模型<sup>[4]</sup>。该模型有三个假设:

A) 软件初始时包含  $M$  个错误

B) 每次软件运行失败时, 一个错误被发现, 更正并且不引入新的错误

C) 时间间隔的失败率  $T_i$  和余下的错误数  $M-i+1$  成比例

这样, 每次被发现的错误和失败的概率以一个固定的常数  $\phi$  递减。现在就可计算出可靠度

$$R(t) = e^{-\phi(M-1)t}$$

这里  $\phi$  和  $M$  根据实验的经验是可以估计的。

### 5.3 揭错能力

揭错能力是测试充分性准则有效性的最直接度量之一。在揭错能力上对充分性准则的比较有三种方法分别是实验统计 (Statistical Experiments)、模拟 (Simulation) 和形式化分析 (Formal Analysis)<sup>[2]</sup>。下面将分别进行介绍:

5.3.1 实验统计 比较测试充分性准则的实验统计方法的基本形式如下:

$C_1, C_2, \dots, C_n$  是要进行比较的测试充分性准则。实验从选择样板程序  $P_1, P_2, \dots, P_m$  开始。每一个程序植入已知的错误。对每个程序  $P_i (i=1, 2, \dots, m)$  和充分性准则  $C_j (j=1, 2, \dots, n)$ ,  $k$  个测试数据集  $T_{i,1}, T_{i,2}, \dots, T_{i,k}$  按某种方式产生, 使得  $T_{i,k}$  根据准则  $C_j$  对于程序  $P_i$  的测试是充分的。测试数据集  $T_{i,k}$  的揭错的比率  $r_{i,k}$  可由测试产生的错误数和已知的植入错误数

得到。用这种方法就可对每一个 i, j, u 计算出比率。

5.3.2 模拟 模拟是基于简化软件测试模型的一种方法。最著名的用模拟的方法对测试充分性进行比较的是 Duran 和 Ntafos 的对划分测试和随机的比较<sup>[7]</sup>。其主要方法是把软件输入空间划分成有限个不相交的子集来模型化划分测试的方法。假定给定软件的输入空间为 D, 并把 D 划分成  $D_1, D_2, \dots, D_k$  这样 k 个不相交的子集, 一个输入随机的选择落在  $D_i$  上的概率为  $P_i$ , 并假设  $\theta_i$  是在  $D_i$  上出错的概率, 那么

$$\theta = \sum_{i=1}^k p_i \theta_i$$

是程序在根据相应的输入分布随机输入引起程序失败的概率。对于划分测试从  $D_i$  域中选出  $n_i$  个测试数据集, 则至少发现一个错误的概率为  $P_p$

$$P_p = 1 - \prod_{i=1}^k (1 - \theta_i)^{n_i}$$

至少发现一个错误的概率是揭错能力的一个重要指标<sup>[1,2]</sup>。

5.3.3 充分性准则之间关系的形式化分析 充分性准则之间比较的基本方法之一是形式化的分析。它通过证明各个准则之间的关系来比较。比如在基于程序的测试中, 满足分支覆盖充分性准则的测试数据集一定满足语句覆盖准则。详细情况见文<sup>[1,2]</sup>。

## 6 范畴划分测试中测试充分性准则的比较与分析

在本节中将结合票据计算的例子, 对范畴划分测试方法的三个充分性准则在测试开销、软件可靠性和揭错能力这三个方面进行分析。

### 6.1 划分范畴和确定选择

首先根据第2节中模式的前置条件和环境变量划分范畴。

范畴:

范畴1

choice1: (base)  $x \leq 30$

choice2:  $x > 30$

范畴2

choice1: (base)  $5 * x + 10 * y \leq 200$

choice2:  $200 < 5 * x + 10 * y \leq 1000$

choice3:  $5 * x + 10 * y > 1000$

根据以上的范畴划分可以总共产生六个测试框架。可以用如图1所示的坐标图画出:

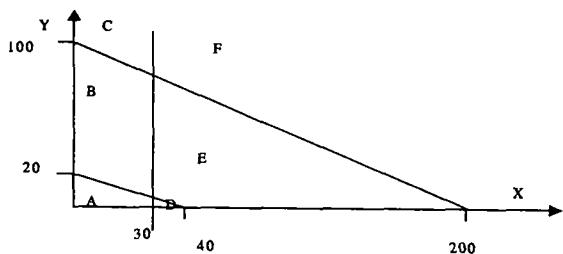


图1 测试框架坐标图1

下面结合例子对范畴划分方法的三个准则进行分析和比较。

首先对三个准则产生测试数据集的大小进行分析。

考虑最简单的情况, 每一个范畴中的选择都是范畴的一个划分, 也就是说各个范畴之间的组合都是对输入空间划分的加细, 因此各个输入的子空间是不相交的。各个选择的组合

可能会出现不可行的情况, 在这里假定所有选择的组合可行。现假设有 N 个范畴, 第 i 个范畴有  $k_i$  个选择, 则一共可以产生

$\prod_{i=1}^N k_i$  个选择的组合, 这是按 All-Combination 准则产生的测试框架的个数。在票据计算中所有选择的组合为6。同样可以计算出按 Each-Choice-Used 准则产生的组合数。显然满足这个准则的组合数是不定的, 但是按照一定算法我们可以找到满足这个准则的最小值为  $\text{MAX}(K_i)$ , 对票据计算, 则为3。最后, 对于 base-Choice-Coverage 准则, 满足该准则的组合数为  $(\sum_{i=1}^N K_i) - N + 1$ , 在票据计算中为4。

### 6.2 虚空间

不是所有的组合都能产生可执行的输入子空间。由于条件组合的矛盾, 产生的输入子空间为空。在这里称这样空间为虚空间。在票据计算的例子中如果把对 X 促销的数目改为40则产生的实际输入空间为5个而不是6个。同样可以画出其坐标图, 如图2所示。

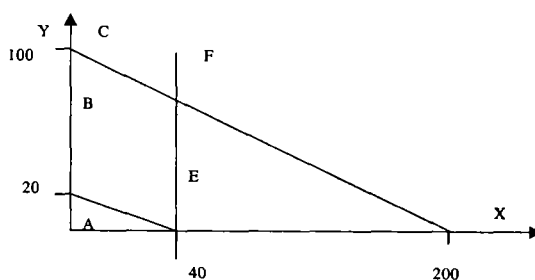


图2 测试框架坐标图2

决定一个组合是否可行等价于决定被涉及的约束是否可满足, 虽然满足性问题是一个非常困难的问题, 但已有工具能解决特定的问题。这里假设所有组合成的子空间所包含虚空间的个数为  $V_a$  个, 按 Each-Choice-Used 准则选择的子空间中包含的虚空间个数为  $V_e$ , 按 base-Choice-Coverage 准则选择的子空间中包含的虚空间个数为  $V_b$ 。显然可以得到以下关系式

$$V_a \geq V_b \text{ 和 } V_a \geq V_e.$$

### 6.3 三个测试充分性准则的比较

有了以上的数据和假设, 就可很方便地对范畴划分的三个充分性准则在测试开销、软件可靠性和揭错能力三个方面进行比较。

·测试开销 在这里我们用测试数据集集合大小来衡量。三个充分性准则产生的测试集合的大小分别为:

All-Combination 准则  $\prod_{i=1}^N k_i - V_a$

Each-Choice-Used 准则  $\text{MAX}(K_i) - V_e$

base-Choice-Coverage 准则  $(\sum_{i=1}^N k_i) - N + 1 - V_b$

在票据计算的例子中我们已给出了各个测试数据集的大小。它们之间的比例为6:3:4。

·揭错能力 在进行三个准则比较和分析之前先对符号  $\theta$  进行说明, 它表示每个子域发生错误的概率, 为了方便计算假设每一个子域出错概率是相同的。下面分别进行分析。

用 All-Combination 准则产生的测试数据集至少发现一个错误的概率

$$P_e = 1 - (1 - \theta)^{\prod_{i=1}^N k_i - V_e}$$

在票据计算实例中为  $1 - (1 - \theta)^6$ 。

用 Each-Choice-Used 准则产生的测试数据集至少发现一个错误的概率

$$P_e = 1 - (1 - \theta)^{\text{MAX}(K_i) - V_e}$$

在票据计算实例中为  $1 - (1 - \theta)^3$ 。

用 base-Choice-Coverage 准则产生的测试数据集至少发现一个错误的概率

$$P_e = 1 - (1 - \theta)^{(\sum_{i=1}^N k_i) - N + 1 - V_e}$$

在票据计算实例中为  $1 - (1 - \theta)^4$ 。

由于  $0 < \theta < 1$ , 那么在票据计算实例中揭错能力的关系为  $P_e < P_b < P_a$ 。

• 软件可靠性 在这里采用 Jelinski-Moranda 模型对通过某种充分性测试的软件可靠性进行分析, 先分别计算出通过某种充分性测试以后发现的错误个数如下:

All-Combination 准则  $(\prod_{i=1}^N k_i - V_e)\theta$

Each-Choice-Used 准则  $(\text{MAX}(K_i) - V_e)\theta$

base-Choice-Coverage 准则  $((\sum_{i=1}^N k_i) - N + 1 - V_e)\theta$

把上面的数据代入 Jelinski-Moranda 模型计算的公式, 就可计算出各通过此充分性准则测试的软件的可靠性度量, 分别计算如下:

用 All-Combination 准则

$$R(t) = e^{-\Phi(M - (\prod_{i=1}^N k_i - V_e)\theta + 1)t}$$

在票据计算例子中

$$\text{可靠度: } R(t) = e^{-\Phi(M - 6\theta + 1)t}$$

用 Each-Choice-Used 准则

$$R(t) = e^{-\Phi(M - (\text{MAX}(K_i) - V_e)\theta + 1)t}$$

在票据计算例子中

$$\text{可靠度: } R(t) = e^{-\Phi(M - 3\theta + 1)t}$$

用 base-Choice-Coverage 准则

$$R(t) = e^{-\Phi(M - ((\sum_{i=1}^N k_i) - N + 1 - V_e)\theta + 1)t}$$

在票据计算例子中

$$\text{可靠度: } R(t) = e^{-\Phi(M - 4\theta + 1)t}$$

从公式中可以看出软件经过测试的次数越多则软件可靠性越高。根据测试的实验和经验得出  $M$ 、 $\Phi$  和  $t$  后就可计算出它们之间的比值。

得到了三个测试充分性准则在揭错能力、软件可靠性和测试开销方面的比值后, 在实际的计算中可以根据需要对揭错能力、软件可靠性和测试开销设定一定的权值, 得到一个测试充分性准则在这三方面综合的比较结果, 测试员可以根据这个综合的结果来选择测试充分性准则。从票据计算例子的比较中可以看出, 根据某个测试充分性准则产生的测试数据

集越大, 则测试开销也越大, 而揭错能力和软件可靠性则越好。这个比较结果和我们的测试经验是一致的, 但这样的比较使得在选择测试充分性准则时有量化的依据。

总结 范畴划分测试是规格说明测试中一种非常重要的测试方法。测试充分性准则在软件的测试过程中起着举足轻重的作用。在范畴划分测试中按照充分性准则的不同产生的测试数据集是不一样的, 用不同的充分性准则测试软件所用的花费和通过测试以后的软件性能是不一样的, 而这两个方面是测试员非常关心的问题。本文提供了对范畴划分测试中充分性准则的比较, 帮助测试员选择用那种测试充分性准则产生测试数据集。本文中我们提出了虚空间的概念, 它的数目对比较是非常重要的, 但是自动地判断它们的数目是困难的。编制工具实现自动化比较过程是我们接下来要做的工作, 该工具帮助测试员判断虚空间。

## 参考文献

- 1 朱鸿, 金陵紫. 软件质量保障与测试. 北京: 科学出版社, 1997
- 2 ZHU HONG. Software Unit Test Coverage and Adequacy ACM. Computing Surveys, 1997, 29(4)
- 3 缪准扣, 李刚, 朱关铭. 软件工程语言-Z. 上海科学技术文献出版社, 1999
- 4 Wiper M P, et al. Bayesian inference for a software reliability model using metrics information, Departamento de Estadística y Econometría Universidad Carlos III de Madrid Calle Madrid, 126 28903 Getafe (Spain) Working Paper01-02 Statistics and Econometrics Series 14 March 2001
- 5 Ostrand T J, Balcer M J. The Category-Partition Method for Specifying and Generating Functional Tests. Communications of the ACM, 1988, 31(6): 676~686
- 6 Ammann P, Offutt J. Using Formal Methods To Derive Test Frames in Category-Partition Testing. Ninth Annual Conference on Computer Assurance (COMPASS 94), IEEE Computer Society Press, Gaithersburg, Maryland, June 1994. 69~80
- 7 Duran J W, Ntafos S. An evaluation of random testing. IEEE Trans. Eng. SE-10, 4(July)438~444
- 8 Binder R V, 华庆一等译. 面向对象系统的测试. 人民邮电出版社, 2001
- 9 Goodenough J B, Gerhart S L. Toward a Theory of Test Data Selection. IEEE Trans. Software Eng., 1975, 1(2): 156~173
- 10 Stocks P, Carrington D. A framework for specification-based testing. IEEE Trans. On Software Engineering, 1996, 22(11)
- 11 Amla N, Ammann P. Using Z Specifications in Category Partition Testing. In: Proc. of the Seventh Annual Conf. on Computer Assurance (COMPASS 92), Gaithersburg MD, IEEE Computer Society Press, 1992
- 12 Haworth B, Roper M. Towards the Development of Adequacy Criteria for Object-Oriented Systems. euroSTAR' 97 - 5th European Conf. on Software Testing Analysis and Review, Edinburgh, Nov. 1997
- 13 Weyuker E J. An empirical study of the complexity of data flow testing. In: Proc. of SIGSOFT Symposium on Software Testing, Analysis, and Verification, 1988. 188~195