

如何设计黑盒测试的测试用例

淡 艳

(成都教育学院 四川成都 610036)

[摘 要] 文章介绍了在黑盒软件测试工作中,如何有效地划分测试种类和编写对应的测试用例,使测试工作更加合理、高效率地运行。

[关键词] 黑盒测试;测试用例;对等区间划分法;边界值分析法;错误推测法;因果图方法

[中国图书分类号] TP311.56 [文献标识码] A [文章编号] 1008 - 9144(2005)11 - 0119 - 02

软件测试是软件质量保证的重要内容,随着软件规模的不断扩大,复杂程度的不断提高,以及面向对象程序设计方法和工具的使用,软件测试的难度也进一步提高,测试质量更加难以度量。

不完全、不彻底是软件测试的致命缺陷,任何程序只能进行少量而有限的测试。测试用例在此情况下产生,同时它也是软件测试系统化、工程化的产物。而测试用例的设计一直是软件测试工作的重点和难点。

测试用例(Test Case)目前没有经典的定义。比较通常的说法是:指对一项特定的软件产品进行测试任务的描述,体现测试方案、方法、技术和策略。内容包括测试目标、测试环境、输入数据、测试步骤、预期结果、测试脚本等,并形成文档。测试用例广义分为两类:

白盒测试:也称结构测试或逻辑驱动测试,已知产品的内部工作过程,可以通过测试来检测软件产品内部动作是否按照规格说明书的规定正常进行。

黑盒测试:也称功能测试,已知产品的功能设计规格,可以进行测试证明每个实现了的功能是否符合要求,不需要了解被测单元的内部结构。相应的,白盒测试用例设计:使用程序设计的控制结构导出测试用例。黑盒测试用例的设计:使用详细设计导出测试用例。

这里我就黑盒测试用例分析其设计方法。

采用黑盒测试的目的主要是:(1)检查功能是否实现或遗漏;(2)检查人机界面是否错误;(3)数据结构或外部数据库访问错误;(4)性能等其他特性要求是否满足;(5)初始化和终止错误。

黑盒测试的设计方法

1. 对等区间划分法:

该方法是测试用例设计的形式化方法。它将被测软件的输入输出划分成一些区间,被测软件对一个特定区间的任何值都是等价的。形成测试区间的数据不只是函数/过程的参数,也可以是软件可以访问的全局变量,系统资源等,这些变量或资源可以是以时间形式存在的数据,或以状态形式存在的输入输出序列。

对等区间划分假定位于单个区间的所有值对测试都是对等的,应该为每个区间的一个值设计一个测试用例。

例:考虑下面计算实数平方根的函数的设计说明:

输入:实数;输出:实数。

处理:当输入 0 或大于 0 时,返回输入数的平方根;当输入小于 0 时,显示:“Square root error - illegal negative input”,并返回 0;库函数 Print_Line 用于显示出错信息。

有 2 个输入区间和 2 个输出区间,表示如下:

输入分区		输出分区	
i	< 0	a	> = 0
ii	> = 0	b	Error

可以用 2 个测试用例测试 4 个区间:

测试用例 1:输入 4,返回 2// 区间 ii 和 a

测试用例 2:输入 - 10,返回 0,输出“Square root error - illegal negative// 区间 i 和 b

上例的对等区间划分是非常简单的。当软件变得更加复杂,对等区间的确定和区间之间的相互依赖就越难,使用对等区间划分设计测试用例技术难度会增加。对等区间划分基本上还是移植正面测试技术,需使用负面测试进行补充。

对等区间划分的原则:

(1)如果输入条件规定了取值范围,或者值的个数,则可以确定一个有效等价类和两个无效等价类;

(2)如果输入条件规定了输入值的集合,或者是规定了“必须如何”的条件,这时可以确立一个有效等价类和一个无效等价类;

(3)如果输入条件是一个布尔量,则可以确立一个有效等价类和一个无效等价类;

(4)如果规定了输入数据的一组值,而且程序要对每一个输入值分别进行处理,这时要对每一个规定的输入值确立一个等价类,而对于这组值之外的所有值确立一个等价类;

(5)如果规定了输入数据必须遵守的规则,则可以确立一个有效等价类(即遵守规则的数据)和若干无效等价类(从不同角度违反规则的数据);

【收稿日期】2005 - 06 - 08

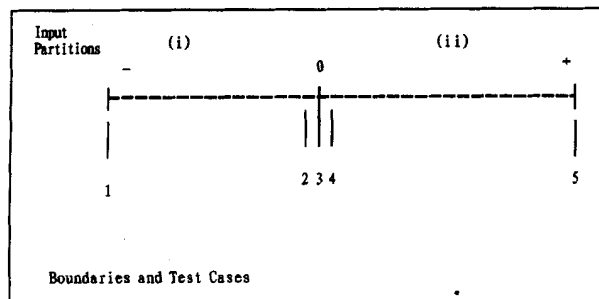
【作者简介】淡 艳(1976 →)女,成都教育学院助教。

(6) 如果确知以划分的等价类中的各元素在程序中的处理方式不同,则应进一步划分成更小的等价类。

2. 界值分析法:

边界值分析方法是等价类划分方法的补充,边界值分析假定错误最有可能出现在区间之间的边界。边界值分析的基本思想是使用在最小值、略高于最小值、正常值、略低于最大值和最大值处取输入变量值。

考虑平方根函数的 2 个输入区间,0 和大于 0 区间的边界是 0 和最大实数,小于 0 区间的边界是 0 和最大负实数。输出区间的边界是 0 和最大正实数。根据边界值分析可以设计 5 个测试用例:



Test Case 1: 输入最大负实数,返回 0,使用 Print_Line 输出“Square root error - illegal negative input”// 区间 (i) 的下边界

Test Case 2: 输入仅比 0 小的数,返回 0,使用 Print_Line 输出“Square root error - illegal negative input”// 区间 (i) 的上边界

Test Case 3: 输入 0,返回 0// 区间 (i) 的上边界外,区间 (ii) 的下边界和区间 (a) 的下边界

Test Case 4: 输入仅比 0 大的数,返回输入的正数平方根// 区间 (ii) 的下边界外

Test Case 5: 最大正实数,返回输入的正平方根// 区间 (ii) 的上边界和区间 (a) 的上边界

对于复杂的软件,使用对等区间划分就不太实际了,对于枚举型等非标量数据也不能使用对等区间划分。如区间 (b) 并没有实际的边界。边界值分析还需了解数的底层表示。一种经验方法是使用任何高于或低于边界的小值和合适的正数和负数。

选择测试用例的原则:

(1) 如果输入条件规定了值的范围,则应该取刚达到这个范围的边界值,以及刚刚超过这个范围边界的值作为测试输入数据;

(2) 如果输入条件规定了值的个数,则用最大个数、最小个数、比最大个数多 1 格、比最小个数少 1 个的数做为测试数据;

(3) 如果程序的规格说明给出的输入域或输出域是有序集合(如有序表、顺序文件等),则应选取集合的第一个和最后一个元素作为测试用例;

(4) 如果程序用了一个内部结构,应该选取这个内部数据结构结构的边界值作为测试用例;

(5) 分析规格说明,找出其他可能的边界条件。

3. 错误推测法

错误猜测大多基于经验,需要从边界值分析等其他技术

获得帮助。这种技术猜测特定软件类型可能发生的错误类型,并且设计测试用例查出这些错误。对有经验的工程师来说,错误猜测有时是惟一最有效发现 BUG 的测试设计方法。

为了最好地利用现成的经验,可以列出一个错误类型的检查列表,帮助猜测错误可能发生在单元中的位置,提高错误猜测的有效性。

4. 因果图方法

前面介绍的等价类划分方法和边界值分析方法,都是着重考虑输入条件,但未考虑输入条件之间的联系,相互组合等。考虑输入条件之间的相互组合,可能会产生一些新的情况,但要检查输入条件的组合不是一件容易的事情,即使把所有输入条件划分成等价类,他们之间的组合情况也相当多,因此必须考虑采用一种适合于描述对于多种条件的组合,相应产生多个动作的形式来考虑设计测试用例,这就需要利用因果图(逻辑模型)。

因果图方法最终生成的就是判定表,它适合于检查程序输入条件的各种组合情况。

利用因果图生成测试用例的基本步骤:

(1) 分析软件规格说明描述中,哪些是原因(即输入条件或输入条件的等价类),哪些是结果(即输出条件),并给每个原因和结果赋予一个标识符。

(2) 分析软件规格说明描述中的语义找出原因与结果之间、原因与原因之间对应的关系,根据这些关系,画出因果图。

(3) 由于语法或环境限制,有些原因与原因之间,原因与结果之间的组合情况不可能出现。为表明这些特殊情况,可在因果图上用一些记号表明约束或限制条件。

(4) 把因果图转换为判定表。

(5) 把判定表的每一列拿出来作为依据,设计测试用例。

结束语

软件测试是计算机发展中出现的一门崭新的学科,虽然目前有很多的测试技术和测试方法,但还没有完全形成一个统一的行业标准,有很多的测试方法很难在实践工程中进行应用,很多测试工具也不能很好的与实际工作结合,所以说,在具体的测试活动中还需要根据工程本身的特点,灵活的选择测试方法和测试工具,实践与理论相结合才能取得时间、资源等的合理投入和最大产品效益的获得。

【参考文献】

[1] Paul C Jorgensen. Software Testing[M]. 机械工业出版社,2003.

[2] 【美】Elfriede Dustin. Addison Software Testing[M]. 清华大学出版社,2003.

[3] 飞思科技产品研发中心. 使用软件测试方法与应用[M]. 北京电子工业出版社,2003.

[4] 王 健,苗 勇. 软件测试员培训教材[M]. 北京电子工业出版社,2003.