

基于输入输出关系的局部组合测试技术

李龙澍,王洪娟

(安徽大学 计算机科学与技术学院,安徽 合肥 230039)

摘要:主要针对软件测试中黑盒测试时测试用例集过大,以及测试效率低的问题,提出了一种有用有效的测试用例集生成设计方法。该方法根据待测系统参数的输入输出关系,对输入参数进行分组生成输入参数的组合关系集,再仅对组合关系集中每组输入变量生成两两组合覆盖测试用例集合,然后进行水平拼接生成最终的用于待测系统测试的测试用例集合。实验结果表明,该方法不仅能有效地减少了测试用例数目,而且还能够保持了原来测试检错能力,从而提高测试效率。

关键词:用例生成算法;组合关系;组合测试;两两组合测试

中图分类号:TP206

文献标识码:A

文章编号:1673-629X(2010)06-0047-04

Partial Combinatorial Design Approach for Testing Based on Input/Output Relationship

LI Long-shu, WANG Hong-juan

(College of Computer Science and Technology, Anhui University, Hefei 230039, China)

Abstract: According to that the number of the set of test case in the method of black box test of software testing is too big, and the low efficiency of the testing, a useful and effectively designing method is presented: according to the input/output relationship of the software under testing, divide the input parameters into combinatorial relation set, only covering each pair combination of every group in combinatorial relation set, then combine them horizontally, which makes the last test case of the software under testing. The experiment proves that method effectively cuts down the set of test case, and keeps the ability of detecting defects of testing, and improves the testing efficiency.

Key words: test case generation algorithm; combinatorial relationship; combinatorial testing; pair-wise test

0 引言

由于电脑以及网络的普及,在现实生活中,软件已经无处不在,由于软件本身特有的性质决定了只要一个很小的错误,就可能带来灾难性的后果,如:千年虫,“冲击波”计算机病毒、火星登陆事故等,而由于软件开发人员思维上的主观局限性,且开发的软件系统都具有相当的复杂性,决定了在开发过程中出现软件错误的必然性。而软件测试就是为了发现这些错误而执行的程序。这也就奠定了软件测试在软件工程中的重要性。

软件测试在程序的每个开发阶段都是必不可少的,它根据各个阶段的用户需求、规格说明和详细设计

文档精心设计一个软件测试计划。按照测试计划生成一组测试用例去运行程序,发现软件错误。

当今的软件测试根据不同划分标准,有不同的分类。从软件内部结构和具体实现的角度可以划分为^[1]:白盒测试(white-box),黑盒测试(black-box)以及灰盒测试;从是否执行程序的角度可以划分为:静态测试和动态测试;从软件开发的过程按阶段可以划分为:单元测试(unit testing),集成测试(integration testing),确认测试、验收测试(acceptance testing, beta test),系统测试(system testing);按照测试的目的分类可以划分为:正确性测试(correctness testing)、性能测试(performance testing)、可靠性测试(reliability testing)、安全性测试(security testing)。而不论按何种分类的测试都离不开测试用例的设计,而在软件测试的过程中,影响软件测试效果的一个重要因素就是测试用例集,测试用例集的大小影响测试完成的速度,测试用例集全面与否影响测试的纠错能力:如能否发现待

收稿日期:2009-10-19;修回日期:2010-01-23

基金项目:安徽省自然科学基金(090412054);安徽省重大科技专项项目(08010201002)

作者简介:李龙澍(1956-),男,安徽亳州人,教授,博士生导师,研究方向为知识工程、软件分析与测试。

测试系统中的 BUG, 能否发现潜在的问题等等, 由此可见, 测试用例在软件测试中的重要性。而文中主要针对用于功能测试的黑盒测试中, 如何根据参数的输入输出关系来设计测试用例, 使得在不降低测试用例集的检错能力的条件下, 降低测试用例集的大小, 提高测试效率。

1 组合测试

组合测试^[2](Combinatorial Testing 或者 Interaction Testing)是基于组合覆盖的软件测试方法的简称。大量的实践表明, 组合测试方法是在保证测试用例集错误检测能力的前提下, 利用待测系统中各个元素以及它们之间的相互作用关系, 有效地减少了测试用例, 降低了测试成本。

目前, 组合测试研究的主要内容是组合测试覆盖测试用例的生成。而从 1985 年, Mandl 在对 Ada 编译器进行测试时首次提出二维组合测试方法(Pair-wise Testing)以来^[3,4], 人们对组合测试进行了大量的研究, 主要包括: N 维组合覆盖和变力度组合覆盖下的组合测试用例集的生成、组合测试的错误检测能力分析、组合错误定位、组合测试用例优先级技术、组合约束条件以及组合测试策略等方面内容^[4]。

而组合测试覆盖测试用例生成方面主要研究的是两两组合测试用例覆盖(即二维组合测试用例覆盖)。当前根据组合覆盖测试用例生成方法主要有 D. M. Cohen 和 S. R. Dalal 提出的 AETG 自动生成系统, Y. Lei 和 K. C. Tai 提出的 IPO 策略, N. Kobayashi 和 A. W. William 提出的代数生成方法以及 CATS、TCG、DDA、PSST、PICT 等方法。

虽然通过提高成对组合覆盖率来改善测试用例的纠错能力是一个很有效的方法, 但是还是不够的, 当输入参数很多而且每个参数的取值范围很大的时候, 两两组合测试用例覆盖生成的测试用例集仍然是一个庞大的数据集, 而且因为不知道参数之间的关系, 很多成对组合是没有意义的。为了解决这个问题, 文中提出了基于参数的输入输出关系的局部参数组合测试用例覆盖算法。

2 局部组合测试

局部组合测试是根据待测系统输入输出关系进行软件测试的方法。该方法主要是将输入输出关系运用到测试用例集合的生成过程中, 用局部组合覆盖替代传统的所有参数之间的组合覆盖, 从而减小测试用例集的大小, 提高测试的效率。

设有一个待测系统 SUT(Software Under Testing)

由 n 个输入参数 X_1, X_2, \dots, X_n 组成^[5,6], 用等价类划分和边界值分析方法将参数的取值处理后, 参数的取值为的一组离散值 $Y_1, Y_2, \dots, Y_n, Y_i (1 \leq i \leq n)$ 表示参数 $X_i (1 \leq i \leq n)$ 可以取值的有限离散点的集合, $|Y_i|$ 表示集合的大小。用 T_1, T_2, \dots, T_m 表示待测系统 SUT 的输出。

定义 1 一个 n 元组 $m = (v_1, v_2, \dots, v_i, \dots, v_n) (1 \leq i \leq n, v_i \in Y_i)$ 表示待测系统 SUT 的一个测试用例。

定义 2 对于一个输出 $Y_i (1 \leq i \leq m)$, 如果存在输入参数 $X_j \dots X_k (1 \leq j < k \leq n)$ 对 Y_i 有影响, 则称之间有组合关系, 记为 $Q_i = \{X_j, \dots, X_k\}$ 。

定义 3 当 $Q_i \cap Q_j \neq \text{null} (i \neq j)$ 时, 则将定义 $Q = Q_i \cup Q_j$ 为新的组合关系的集合。

定义 4 用一个二维数组 $D[m][n]$ 表示待测系统 SUT 的测试用例集合, $D[i][j]$ 表示第 i 个测试用例的第 j 个参数的取值。

根据上述定义, 为一个待测系统生成一组合理高效的测试用例, 主要可以分为三个步骤:

(1) 根据待测系统的输入输出关系, 根据定义 1 和定义 3 将输入参数划分为具有组合关系集合 $\{Q_1, Q_2, \dots, Q_t\}$, $Q_1 \cup Q_2 \cup \dots \cup Q_t = \{X_1, X_2, \dots, X_n\}$ 且 $Q_1 \cap Q_2 \cap \dots \cap Q_t = \text{null}$ 。

(2) 对每个组合集 $Q_i (1 \leq i \leq t)$, 根据 IPO 策略生成局部组合覆盖测试用例集 $S_i (1 \leq i \leq t)$ 。

(3) 对每个局部组合测试用例集 $S_i (1 \leq i \leq t)$ 通过水平拼接生成一个完整的测试用例集 $M, M = \max(|S_i|) (1 \leq i \leq t)$ 。

2.1 生成组合关系集 $\{Q_1, Q_2, \dots, Q_t\}$ 算法

在该算法中根据各个阶段的用户需求、规格说明得出输入输出关系, 再根据定义 1 和定义 3 生成组合关系集 $\{Q_1, Q_2, \dots, Q_t\}$ ^[7]。

输入: 输入输出参数的关系。

输出: 组合关系集 $\{Q_1, Q_2, \dots, Q_t\}$ 。

(1) For($i=1; i < m; i++$)

$Q_i = \text{null}$; //初始化组合关系集都为空

(2) For($i=1; i < m; i++$)

For($j=1; j < n; j++$)

{if(X_j 影响输出 T_i)

$Q_i = Q_i \cup X_j$;

//根据定义 1 生成组合关系集

(3) $Q_i (1 \leq i \leq m)$ 中任意两个相交不为空的合并, 并重新编号为 $\{Q_1, Q_2, \dots, Q_t\}$ 。

//根据定义 3, 对第二步中的组合关系集进行合并, 生成最终的组合关系集

2.2 生成局部覆盖测试用例集算法

对于2.1中生成的每个组合关系,采取IPO策略生成局部组合覆盖测试用例集^[2]。该策略最早是由Lei和Tai等人提出的,并应用于二维组合测试用例的生成中。

该算法运用了贪心算法策略,使满足成对测试的任意两个参数满足两两覆盖的同时,生成较小的测试用例集。IPO策略对于参数的扩展包含两步^[8]:

1)水平增长,通过增加一个新的参数值来扩展每一个现行测试。

2)垂直增长,通过增加测试用例集来增加新的测试。

对于2.1中组合关系集中的一个关系 $Q_i (1 \leq i \leq m)$ 的水平增值算法 IPO-HA(S_i, P_i):

输入:2.1生成组合关系集 Q_i 中的测试用例集以及 Q_i 的第 i 个参数。

输出:新的测试用例集合。

IPO-HA(S_i, P_i)

// $S_i (1 \leq i \leq m)$ 是2.1生成组合关系集 Q_i 中的测试用例集

// $P_i (1 \leq i \leq |Q_i|)$ 是 Q_i 的第 i 个参数

{假设 P_i 的域包含值 v_1, \dots, v_k ;

$\Pi = \{p_i \text{ 与 } p_1 \text{ 到 } p_{|Q_i|} \text{ 的值之间的对}\}$

For($1 \leq i < |S_i| + 1$)

for($0 < j < k$)

{

对于 S_i 中的每个用例,根据贪心算法,增加覆盖 Π 中组合最多的 $v_t (1 \leq t \leq k)$,并从 Π 中删除由已扩展的所覆盖的组合

}

}

如果经过水平增长算法后, Π 集合为空,就不需要使用垂直增长算法进行测试用例集的生成;当 Π 不为空的时候,则需要经过垂直增长算法 IPO-VA(T, Π) 产生新的测试用例集:

输入:测试用例集合 S_i 和经过水平算法处理后的 p_i 与 p_1 到 $p_{|Q_i|}$ 的值之间的对 Π 。

输出:每个局部的最终的测试用例集合。

IPO-VA(S_i, Π)

{设一个空集 T' ;

对于 Π 中的每一对 (α, β) , α 是 p_k 值 ($0 < k < i$), β 是 p_i ($0 < i < |Q_i|$) 的值。

If($\Pi < > \text{null}$)

{

If(T' 包含了一个以“-”作为 p_k 值, β 作为 p_i 的值

测试用例)

修改这一用例,用 α 替换-,并从 Π 中删除 (α, β)

Else

为 T' 增加一个新的用例, α 作为 p_k 的值, β 作为 p_i 的值,其他的参数值都用“-”代替,并从 Π 中删除 (α, β)

}

}

2.3 对局部测试用例集进行水平拼接算法

在2.2的算法对应组合集合关系集 $\{Q_1, Q_2, \dots, Q_t\}$ 产生测试用例集 S_1, S_2, \dots, S_t , 由于 S_1, S_2, \dots, S_t 是 n 元参数划分为 t 个组合关系产生的测试用例,不是一个完整的测试用例集,需要用一个算法,将 S_1, S_2, \dots, S_t 拼接为一个完整的测试用例集^[9,10]。下面就是水平拼接算法 Join-LA(S, t):

输入:每个组合关系的测试用例集合和组合关系的个数。

输出:最终的测试用例集合。

Join-LA(S, t)

{

Int $M = \max(|S_i|)$; // N 记录测试用例集中 $\{S_1, S_2, \dots, S_t\}$ 最大的集合数目

$D[M][n] = \text{null}$; // $D[M][n]$ 表示测试用例集,并初始化

For($i = 0$; $i < t$; $t++$)

{

将 S_i 中所有的所有参数的取值,按行填入到 $D[M][n]$ 中,没有值的用“-”代替

}

}

3 实例分析以及理论验证

这部分主要从实验和理论两方面验证改进后的算法能够达到预期的效果。

3.1 实验分析

实验是根据相同输入条件下(参数输入数目相同,每个参数的取值范围相同),分别运用改进的算法和原始算法产生的测试用例集,并对产生的测试用例集的大小进行比较。

在实验中,先对每个参数进行离散,假设每个参数离散后取值个数相同,在该情况下进行对比分析,表1是不同输入参数个数下的实验结果的对比。

另外,为了方便表述,如果用3个参数,就用 $\{a, b, c\}$ 表示,4个则是 $\{a, b, c, d\}$,5个则是 $\{a, b, c, d, e\}$,依

次类推。

表 1 实验结果对比

| 根据输入输出 关系划分的组 合关系集 | 输入参数 的个数 | 每个参数 的取值数目 | 改进前算法 产生测试用 例集大小 | 改进算法产 生的测试用 例集大小 |
|--------------------------|-------------|---------------|------------------------|------------------------|
| $\{(a,b),(b)\}$ | 3 | 2 | 4 | 4 |
| $\{(a,b),(c,d)\}$ | 4 | 2 | 6 | 4 |
| $\{(a,b,c),(d)\}$ | 4 | 2 | 6 | 4 |
| $\{(a,b),(c,d)\}$ | 4 | 3 | 9 | 9 |
| $\{(a,b,c),(d)\}$ | 4 | 3 | 9 | 9 |

上面的实验结果已经表明了,改进后的算法,有效地降低了测试用例集的大小,下面,用理论证明改进算法能够降低测试用例集合的大小。

3.2 理论证明

证明:假设一个待测系统 SUT 的输入参数集合为 $\{X_1, X_2, \dots, X_n\}$, 根据定义 2 和定义 3 得出的组合关系集合中最大的集合大小为 $M(1 \leq M \leq n)$ 。

根据 IPO 策略,假设加入第 i 个参数时,前 $i-1$ 个参数的测试用例集合 S_i 上进行水平增长和垂直增长,则有 $|S_i| \geq |S_{i-1}|$, 则有在改进前 IPO 策略中,当输入参数数目为 N 时,产生的测试用例集为 S_1 , 当输入参数数目为 M 时,产生的测试用例集为 S_2 , 当有 $N > M$ 时,则有 $|S_1| \geq |S_2|$ 。

在改进后的待测系统 SUT 中,由题设知道产生测试用例集合大小和输入参数个数为 M 时产生的测试用例集合 S_M 相同。

由于 $M \leq n$, 根据上面结论则有 $|S_M| \leq |S_n|$ 。

由此,可以知道改进后的算法通过实验和理论证明有效地降低了测试中测试用例集的大小。

4 结束语

文中主要研究在软件测试中测试用例集的生成,主要是根据待测系统的输入输出关系,并将其应用于 IPO 策略中用于测试用例集的生成。通过实验以及理论证明文中在不降低测试用例集的检错能力情况下,改进后的算法,能够有效地降低测试用例集合的大小。

由于文中改进的算法产生的测试用例集中,有很多参数的取值是“-”代替的,即表示该位置的取值为

该参数取值范围内的任意值,这就没有充分利用该参数取值的价值,就这一点还有很多方面值得研究和挖掘的,例如从该参数每个参数值的重要度即权重来研究,或从边界值来研究。另一个研究方向是因为很多情况产生的测试用例集合中的一个测试用例中有多个参数的取值用“-”代替,对于这些参数,可以从其他而非输入输出关系来研究原来不在同一组合关系的参数之间的关系^[11],或用两两组合覆盖扩张到部分三元组合或者多元组合覆盖,这些方面将是下一步的研究和改进的方向。总之,改进后的算法有一定的优势,也还有很多扩展空间,这方面的研究还有广阔的前景。

参考文献:

- [1] 牛为华,孟建良,张素文.成对测试中测试生成策略研究[J].计算机仿真,2003,20(7):111-115.
- [2] Lei Yu, Tai Kuo-Chung. In-parameter-order: A Test Generation Strategy for Pairwise Testing[R]. USA: Dept. of Computer Science, North Carolina State University, 2001.
- [3] Whittaker J A, Thomason M G. A Markov Chain Model for Statistical Software Testing[J]. IEEE Transactions on Software Engineering, 1994, 2(10): 812-824.
- [4] 王子元,徐宝文,聂长海.组合测试用例生成技术[J].计算机科学与探索,2008,2(6):571-588.
- [5] 潘祖烈,孙乐昌,蔡 铭,等.基于遗传算法的黑箱测试用例自动生成模型[J].计算机工程,2008,34(9):205-210.
- [6] Rothermel G, Untch R H, Chu Chengyun, et al. Prioritizing Test Cases for Regression Testing[J]. ACM Transactions on Software Engineering Methodology, 2001, 27(10): 929-948.
- [7] 英 伟,高仲仪.基于遗传算法的软件结构测试数据生成技术研究[J].北京航空航天大学学报,1997,23(1):36-40.
- [8] 向 渝,张 骏,汪文勇.成对测试中的一种用例生成算法[J].电子科技大学学报,2005,34(6):821-824.
- [9] 徐宝文,聂长海,史 亮,等.一种基于组合测试的软件故障调试方法[J].计算机学报,2006,29(1):132-138.
- [10] 朱 劼,高建华.优先级参数配对组合测试集生成策略[J].计算机工程,2008,34(13):34-39.
- [11] 张海藩.软件工程导论[M].北京:清华大学出版社,2003.

(上接第 46 页)

com. cn.

- [9] 程佩青.数字信号处理[M].第2版.北京:清华大学出版社,2001.
- [10] Zhu Xuefen, Chen Xiyuan, Li Zigang. Architecture design of

GPS software receiver and implementation of its acquisition algorithm with fine frequency estimation[J]. Journal of Southeast University (English Edition), 2008, 24(1): 38-41.