第 33 卷 第 9 期 计 算 机 工 程 2007 年 5 月
Vol.33 No.9 Computer Engineering May 2007
·博士论文· 文章编号：1000—3428(2007)09—0046—03 文献标识码：A 中图分类号：TP31

基于模糊逻辑的测试用例揭错能力分析

杨劲涛[1]，郭荷清[2]
(1. 广东工业大学计算机学院，广州 510090；2. 华南理工大学计算机科学与工程学院，广州 510640)

摘 要：利用模糊逻辑，分析了各测试用例中交元的测试特性，得到如下结果：交元在不同测试用例中所发挥的测试作用是相同的。据此给出了生成充分考虑参数取值组合作用、且个数少的测试用例（测试用例基）的一种方法，建立了以测试用例基表示所有测试用例的表达式。利用模糊逻辑证明了用测试用例基执行测试，不仅能确保软件质量满足用户需求，且提高了测试用例的揭错能力。由于测试用例基具有数量少、测试效率高的特点，该文的研究有利于改善测试工作，降低测试成本。

关键词：模糊逻辑；测试用例；揭错能力

Analysis of Test Case Error Discover Ability Based on Fuzzy Logic

YANG Jintao[1], GUO Heqing[2]
(1. Faculty of Computer, Guangdong Univ. of Tech., Guangzhou 510090;
2. College of Computer Science and Engineering, South China Univ. of Tech., Guangzhou 510640)

【Abstract】 Fuzzy logic is applied to analyze the test characteristic of same-elements in all test cases. It is concluded that there is same effect same-elements in each test case set. Based on the result, a method which is used to generate lesser test cases (TCB) thinking about the action of combination of parameters values is given. The expression of TCB expressing all test cases is established. It is proved that TCB test can ensure that the software's quality can satisfy user requirement and improve the ability of test cases error discover by fuzzy logic. The research is propitious to reform testing process and to reduce test cost since TCB is lesser but more efficient.

【Key words】 Fuzzy logic; Test case; Ability of error discover

在进行软件测试前，必须精心构造出高效的测试用例是测试过程的关键。迄今已有不少选择测试用例的方法。近几年，如Y.Lei和K.C.Tai等更取得了可喜的成绩[1~3]。但是从理论上去研究被选测试用例的揭错能力、如何定量地规定软件测试的要求、避免不必要的测试等方面的研究仍有待深入。

在文献[4]的基础上，分析了各测试用例中参数值的测试作用，从而得到选取测试用例基的一种方法：应用模糊逻辑证明了用这种方法所选取的测试用例，在确保满足测试需求的条件下，减少测试用例的数量，既降低了测试成本，又提高软件测试的效率和质量。

1 测试用例及与系统状态的关系

定义 1 设 m 个接口参数 $c_1, c_2, \ldots, c_m$ 的某种取值组成的有序集 $U = (u_1, u_2, \ldots, u_m)$，则称 U 为一个测试用例，所有测试用例的集合，记为 E。其中参数 c_j 的取值集记为 T_j，可取值的个数为 t_j。

定义 2 软件系统在输入某个测试用例 U 后，所出现的可监测的状态参数的有序集，称为系统的位态(PS)，记为 p。而由业务模型推算出的位态，称为与 U 相应的预期位态(APS)记为 p^0。

用 u 表示 U 中某一个参数值或某一参数值组，与之对应的位态 p 的集为 P，R(u,p)为从参数值集 U 到位态集 P 的模糊关系，即 u 对位态 p 的作用程度。

若在U上的模糊集记为A，P上的模糊集记为B，那么，这个二元关系可用蕴涵关系表示为 $R = A \to B$，亦称A对B的作用关系[5]。u对p的作用程度(或称真值)，记为

$$R(u, p) = (A \to B)(u, p) \tag{1}$$

其算法规定如下：

$$(A \to B)(u, p) = (1 - A(u)) \vee B(p) \tag{2}$$

真值的大小，表示位态 p 与 p^0 的接近程度。

2 交元 u 的测试不变性

因为在系统的测试用例集 E 中，不同的测试用例 U_i 可能含有若干个这样的参数，其取值分别相同，即这些测试用例的相交部分。为了叙述方便，引进如下概念：

定义 3 设 $U_i \in E$，$(i = 1, 2, \ldots, n)$，当 $\bigcap_{i=1}^{n} U_i \neq \phi$ 时，则称这些相同的参数值组合为 U_i 的交元，并用一个字母 u 表示，即 $\bigcap_{i=1}^{n} U_i = u$。

显然，u 可由一个或多个参数的取值组成。

设p是对应u的位态，从式(1)知：若在测试用例集 U_i 上的模糊集记为A_i，相应的位态集P_i上的模糊集记为B_i，那么U_i到P_i上的二元关系R=A_i→B_i，即u对p的作用程度用真值表示为

$$R(u, p) = (A_i \to B_i)(u, p)$$

下面研究这样的交元 u 与 p 之间的作用关系所具有的逻辑特性。

基金项目：广州市重点科技攻关资助项目(B2-109-550)
作者简介：杨劲涛(1971－)，男，博士、讲师，主研方向：软件工程，软件测试技术；郭荷清，博导、教授
收稿日期：2006-06-12 **E-mail：**yothomas@21cn.com

由于 $\forall U \in E$，可以看成是 U 上隶属度为

$$U(u_j)=\begin{cases}1, & u_j \in U\\ 0, & u_j \notin U\end{cases} \quad (j=1,2,\ldots,m)$$

的模糊集，并记为 A，这时也说 A 是 U 上模糊集。

由于交元 u 表示所论各测试用例的相交部分，因此 u 相对属于相应测试用例的模糊集。

定理 1(交元的测试不变性) 设 $\bigcap_{l=1}^{s} U_l = u$，而测试用例 U_l 上的模糊集记为 A_l，相应的位态集 P_l 上的模糊集记为 B_l，则对 $l_0 \in \{1,2,\ldots,s\}$，$\lambda \in [\frac{1}{2},1]$ 及 (u,p)，当 $(A_{l_0} \to B_{l_0})(u,p) < \lambda$ 时，对一切 l=1,2,...,s 均有

$$(A_l \to B_l)(u,p) < \lambda$$

证明 因当仅考虑 $\bigcap_{l=1}^{s} U_l$ 中的交元 u，对位态 p 的作用时，由于 p 至少属于 $P_1,P_2,\ldots,P_s$ 中的某个，即 $p \in \bigcup_{l=1}^{s} P_l$，而已知 $u \in \bigcap_{l=1}^{s} U_l$，按模糊逻辑运算及式(2)，可推得如下等式

$$\left(\bigcap_{l=1}^{s} A_l \to \bigcup_{l=1}^{s} B_l\right)(u,p) = \bigvee_{l=1}^{s}(A_l \to B_l)(u,p) \tag{3}$$

由式(3)可得

$$\left(\bigcap_{l=1}^{s} A_l \to \bigcup_{l=1}^{s} B_l\right)(u,p) < \lambda \Leftrightarrow \bigvee_{l=1}^{s}(A_l \to B_l)(u,p) < \lambda \tag{4}$$

而 $p \in \bigcup_{l=1}^{s} P_l$，即 p 至少属于 $P_1,P_2,\ldots,P_s$ 中的某个位态 P_{l_0}，故由式(3)、式(4)知：

$$(A_{l_0} \to B_{l_0})(u,p) < \lambda \Leftrightarrow \bigvee_{l=1}^{s}(A_l \to B_l)(u,p) < \lambda$$

由此可见：存在 $l_0 \in \{1,2,\ldots,s\}$ 及 (u,p)，当 $(A_{l_0} \to B_{l_0})(u,p) < \lambda$ 时，对一切 l=1,2,...,s，均有

$$(A_l \to B_l)(u,p) < \lambda$$

证毕。

若 λ 为隐患是否可接受的衡量值，定理 1 说明，只要 $\bigcap_{l=1}^{s} A_l \neq \phi$，那么无论在 s 个测试用例的哪一个中，交元 u 对位态 p 的作用程度小于 λ，均同样成立。

u 无论跟哪一部分参数取值组成测试用例，u 对位态 p 的作用程度均小于 λ，均可发现同样程度的隐患。即暴露被测系统隐患的效果均是相同的。因此，称为交元测试不变性。

这就从理论上证明了：如果集 E 中某个测试用例，其各参数所取的值，或任意两个参数取值的组合，均能在所选取的一组测试用例中出现，那么去掉 E 中这个测试用例，而只采用所选取的测试用例组检测系统，也能达到同样的检测效果。这时认为此测试用例被所选取的那组测试用例覆盖。

又由于测试用例中，任何多个元素的组合总是由两两组合构成的，因此在生成测试用例时，只要使所有参数及其两两组合不要重复，则这些参数的多元组也不会重复。故得到如下选取测试用例的方法。

3 测试用例基的选取

在执行测试时，若输入测试用例 U 导致系统发生问题，究其原因：U 中 m 个参数的取值状况所触发的。由于在集 E 中的某些测试用例存在交元。由定理 1 知，交元 u 具有测试不变性的特点，为了提高测试效率，必须考虑各参数值不同组合的测试效果，希望从 E 中挑选出这样的一组测试用例，使组中含测试用例的个数较少，但通过它们能覆盖 E 中所有其它测试用例。下面就把这样的测试用例组，称为“测试用例基”。

定义 4 在 m 个参数的所有取值构成的参数值表中，由同行的 m 个参数取值及不同行不同列的 m 个参数取值按以下方法组成的测试用例的集合：

$$M = \{U_i \mid U_i = (u_{i1},u_{i2},\ldots,u_{im}), i=1,2,\ldots,s\}$$

称为该系统的“测试用例基”(TCB)。

构成 TCB 的算法如下：

第 1 步 制成一个 n 行、m 列的参数取值表(TPV)，其中 $n = \max_{1\le j\le m} t_j$，$m$ 是参数个数($t_1 \ge t_2 \ge \ldots \ge t_m$)。表中第 j 列的元素是第 j 个参数所取的各种值，使这个参数的每个取值在该列中出现且只出现一次，若某个参数取值数<n，则该参数的取值可重复出现。

第 2 步 按参数取值表中第一列展开，即将该列的第 i 个元与第 i 个剩余表(剩余表 i)的每一行组合，就得到一个测试用例 $(i=1,2,\ldots,n)$。

剩余表 i 的产生方法如下：

(1)产生剩余表 1：它就是由 TPV 中第 2,...,m 列组成；

(2)依次产生下一个剩余表：以原剩余表的第 1 列作为新剩余表的第 1 列不变，而新剩余表的第 i 行的其它元素产生方法为：划去原剩余表的第 1 列与第 i 行元素后，在余下的不同行、不同列中各取一个元，就是新剩余表的第 i 行相应列上的元。

剩余表中生成行元素时须注意：

1)新生成行的各元素，不得重复本表前面行相应列上的元素，也不得是原剩余表 1 的同行元素；

2)剩余表 1 中的所有元素无遗漏地出现在以后的剩余表中。

上述方法所构成的测试用例的集合，就是测试用例基 M，显然集 M 包含了 m 个参数的各种取值组合。

4 TCB 的揭错能力分析

揭错能力是指某种测试方法能够发现软件错误的能力。目前评判测试揭错能力的标准主要有 4 种：揭错概率，揭错个数，揭错比率，揭错效率[6]。

下面借助模糊逻辑运算，从满足用户需求或功能规约的角度来分析 TCB 的揭错能力。

定理 2 设系统的测试用例基 $M = \{U_1,U_2,\ldots,U_r\}$，则对系统中任一测试用例 $U_k = (u_{k1},u_{k2},\ldots,u_{km}) \in E$，有

$$U_k = U_k \cap (\bigcup_{i=1}^{r} U_i) \tag{5}$$

证明 因 $U_k = (u_{k1},u_{k2},\ldots,u_{km})$ 中第 j 个元素 u_{kj} 就是参数 c_j 的某个取值(j=1,2,...,m)。由 TCB 的生成方法知，TPV 参数的每个取值或某种组合至少在 M 中出现一次，由定义 4，必存在测试用例 $U_i \in M$，$1 \le i \le r$，使

$$U_k \cap U_i = (\ldots,u_{kj},\ldots)$$

于是 E 中任一测试用例 U_k 可以表示为

$$U_k = (u_{k1},u_{k2},\ldots,u_{km}) = \bigcup_{i=1}^{r}(U_k \cap U_i) = U_k \cap (\bigcup_{i=1}^{r} U_i)$$

证毕。

定理 2 给出的表达式(5)说明：测试用例基 M 在系统的测

试用例集 E 中，对于参数取值的各种组合来说，起到了基底的作用。

按前面的记号，设 A_k, A_i 分别是 U_k, U_i 上的模糊集，B 是位态集上的模糊集，那么有下面的定理：

定理 3 设测试用例基 $M=\{U_1,U_2,\ldots,U_r\}$，常数 $\lambda\in[\frac{1}{2},1]$，则 $\forall U_k\in E$， $\exists U_s\in M$ 使得

当 $(A_k\to B)(u,p)<\lambda$ 时， $(A_s\to B)(u,p)<\lambda$

证明 由于普通集是隶属度取 1 或 0 的模糊集，根据定理 2，相应有

$$A_k = A_k \cap (\bigcup_{i=1}^{r} A_i)$$

根据式(2)的算法， $\forall(u,p)$ ，有

$$\begin{aligned}
&(A_k \to B)(u,p)\\
&=\left\{\left[A_k\cap(\bigcup_{i=1}^{r}A_i)\right]\to B\right\}(u,p)\\
&=\left[1-\left(A_k\cap(\bigcup_{i=1}^{r}A_i)\right)(u)\right]\vee B(p)\\
&=\left\{1-\left[A_k(u)\wedge\left(\bigvee_{i=1}^{r}A_i(u)\right)\right]\right\}\vee B(p)\\
&=\left\{[1-A_k(u)]\vee\left[1-\left(\bigvee_{i=1}^{r}A_i(u)\right)\right]\right\}\vee B(p)\\
&=\left\{[1-A_k(u)]\vee\left[\bigwedge_{i=1}^{r}(1-A_i(u))\right]\right\}\vee B(p)\\
&=\{[1-A_k(u)]\vee B(p)\}\vee\left\{\bigwedge_{i=1}^{r}[(1-A_i(u))\vee B(p)]\right\}\\
&=(A_k\to B)(u,p)\vee\left[\bigwedge_{i=1}^{r}(A_i\to B)(u,p)\right]
\end{aligned}$$

由上式可见对 $\forall(u,p)$ ，

$$(A_k\to B)(u,p)\ge\left[\bigwedge_{i=1}^{r}(A_i\to B)(u,p)\right] \tag{6}$$

由于至少存在 $s\in\{1,2,\ldots,r\}$ ，使

$$\left[\bigwedge_{i=1}^{r}(A_i\to B)(u,p)\right]=(A_s\to B)(u,p)$$

于是式(6)可改写为

$$(A_k\to B)(u,p)\ge(A_s\to B)(u,p) \tag{7}$$

因此，对 $\lambda\in[\frac{1}{2},1]$，当 $\forall U_k\in E$，且 $(A_k\to B)(u,p)<\lambda$ 时，必有 $\exists U_s\in M$ 使

$$(A_s\to B)(u,p)<\lambda$$

证毕。

定理 3 说明：如果系统的任一测试用例 U_k 对位态 p 的作用程度小于 λ，那么在测试用例基 M 中，至少存在 U_s 对位态 p 的作用程度也小于 λ，即若 $\forall U_k\in E$，它能发现某种程度的隐患，则至少有 $U_s\in M$ 也能发现同样程度的隐患。反之，若使用测试用例基 M 执行测试时，未发现软件系统存在隐患，就是 $\forall U_s\in M$，总有

$$(A_s\to B)(u,p)\ge\lambda$$

因此据式(6)，必然 $\forall U_k\in E$ 有

$$(A_k\to B)(u,p)\ge\lambda$$

就是说使用集 E 中的其它测试用例，在模糊 λ 真[4]的条件下，仍然不能发现软件系统的隐患。

上面的讨论从理论上证明了采用 TCB 执行测试，与采用穷举测试或随机选取用例等方法进行测试的不同之处是：用 TCB 执行测试是在确保满足用户需求的前提下，使测试用例的数量大大减少，这就节省了测试资源，从而提高了测试用例的揭错效率。

5 小结

本文在文献[4]的基础上，通过模糊逻辑运算展示了交元具有良好的检测特性，这是构造 TCB 的理论依据；定理 3 用模糊逻辑运算从理论上证明：TCB 提高了测试用例的揭错能力。

此外，据定理 2 与定理 3 知， E 中任意测试用例，能被构成的 TCB 覆盖。即用尽可能少次数的测试结果最大限度地反映被测软件的内在规律、代表被测软件的内部性能。从测试效果来看，是一种高效的测试。因此，本文的研究在降低测试成本与提高测试质量方面是有益的探索。

参考文献

1 Lei Y, Tai K C. In-parameter-oder: A Test Generation Strategy for Pairwise Testing[R]. Raleigh, North Carolina: Dept. of Computer Science, North Carolina State Univ., Technical Report: TR-2001-03. 2001.

2 Tai K C, Lei Y. A Test Generation Strategy for Pairwise Testing[J]. IEEE Trans. on Software Engineering, 2002, 28(1): 109-111.

3 Kobayashi N, Tsuchiya T, Kikuno T. A New Method for Constructing Pair-wise Covering Designs for Software Testing[J]. Information Processing Letters, 2002, 81(2): 85-91.

4 杨劲涛, 郭荷清. 一种精简测试用例方法的研究[J]. 计算机科学, 2005, 32(5): 236-238.

5 杨纶标, 高英仪. 模糊数学原理与应用[M]. 3 版. 广州: 华南理工大学出版社, 2003.

6 朱　鸿, 金凌紫. 软件质量保障与测试[M]. 北京: 科学出版社, 1997.

（上接第 45 页）

2 Harrold M J, Sayre G R K, Wu R, et al. An Empirical Investigation of the Relationship Between Spectra Differences and Regression Faults[J]. Software Testing, Verification and Reliability, 2000, 10(3): 171-194.

3 Renieres M R. Fault Localization with Nearest Neighbor Queries[C]//Proceedings the 18th IEEE International Conference on Automated Software Engineering. 2003-10-06: 30-39.

4 Hirschberg D S. Algorithms for the Longest Common Subsequence Problem[J]. Journal of the Association for Computing Machinery, 1977, 24(4): 664-675.

5 Breimer E A, Lim D T. A Learning Algorithm for the Longest Common Subsequence Problem[J]. Journal of Experimental Algorithmics, 2003, 8.