一种软件测试过程的博弈模型

崔小乐 王轶辰 阮 镰
(北京航空航天大学 工程系统工程系, 北京 100083)

摘 要: 为分析软件测试的动态过程及其中诸影响因素之间的关系,运用博弈理论,提出了一种新的软件测试过程建模分析方法. 建立了一种软件测试过程的博弈模型,并在此基础上对测试过程中的各方决策条件和博弈均衡条件进行了讨论,其结论有助于理论分析和解决"测试策略选择"和"何时停止测试"等问题. 由于考虑了参与软件测试活动的各方利益和可能行为,此方法较之软件测试控制论更能够对软件测试过程中所出现的情况进行解释. 本模型可为软件测试规程设计提供理论依据,增强软件测试过程的可管理性.

关 键 词: 博弈论; 决策; 均衡; 软件测试过程; 随机性测试; 选择性测试
中图分类号: TP 311
文献标识码: A **文 章 编 号**: 1001-5965(2004)08-0797-06

Process model for software testing based on game theory

Cui Xiaole Wang Yichen Ruan Lian
(Dept. of System Engineering of Engineering Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract: To analysis the dynamic process of software testing and the relationship of relevant factors in it, a new method on modeling and analysis of software testing process with the game theory was provided. A game model of software testing process was established and a series of analysis were made on decision and equilibrium conditions of the game model. The conclusions are helpful for analyzing and solving the problems such as "selection on the software testing strategy" and "when to stop testing" theoretically. It can deduce a better explanation on the activities in the software testing process comparing with the theory of software testing cybernetics because it takes all participants' interests and their possible actions into consideration. This model has founded a theoretical basis for designing the rules of the software testing, which could make the process more manageable.

Key words: game theory; decisions; equalization; software testing process; random testing; partition testing

软件工程研究软件开发的方法、工具和过程. 软件测试作为软件工程的一个重要阶段,其研究自然也应包括这 3 个方面的内容. 从当前软件测试的研究情况来看,软件测试的方法和相应工具的研究者较多,而软件测试过程的研究相对比较薄弱.

目前对软件测试过程的研究大致分为 3 大类:第 1 类是从工程的角度给出的软件测试过程描述和过程定义,以工程准则指导软件测试过程,如 CMM(能力成型熟度模型)/TSP(小组软件过程)/PSP(个人软件过程)中测试相关部分. 这一类方法产生的软件测试过程工程可操作性较强,但缺少较精确的定义. 第 2 类是从测试充分性问题出发,主要研究软件测试过程中的测试充分性准则和测试充分性公理系统[1]. 测试充分性问题是目前软件测试领域的两大难点之一,至今尚未

收稿日期: 2003-05-12
作者简介: 崔小乐(1975-),男,陕西西安人,博士生, cuixiaole2004@263.com.

完全解决. 因此在测试充分性准则下的软件测试过程设计和控制尚不能形成完整的方法. 第 3 类研究将控制理论应用于软件测试过程,从而形成对软件测试的过程控制方法. 这类控制方法多是采取一个或多个度量作为反馈变量,采用最优控制理论、自适应控制理论、马尔柯夫过程等理论和工具进行过程控制[2]. 从这个角度对软件测试过程的研究称为软件测试控制论,是一个比较有前景的方向.

本文从博弈理论和方法出发,给出一种新的软件系统测试过程的理论模型. 将博弈理论应用到软件测试过程建模中,不但能够给出较严格的软件测试过程描述,而且可以在测试过程中考虑参与测试过程各方的主动策略选择. 因此通过博弈模型不但可以为软件测试过程决策提供分析方法,也可为软件测试的过程管理提供准则.

1 预备知识

1.1 选择性测试与随机性测试的测试效率

软件测试可以分成选择性测试和随机性测试两大类. 选择性测试中的测试数据是有目的地选取的,一个充分的测试数据集必须覆盖软件输入空间中的一系列子域,而在每个子域中选择一个或多个测试数据的方法是随机的. 随机性测试则在软件的输入空间中随机选取测试数据进行测试[3].

Ntafos 和 Duran[4] 与 Hamlet 和 Taylor[5] 等人对这两类测试的测试效率问题进行过实验研究,但是多次实验的结果并未得出这两种测试孰优孰劣的决定性证据. 对两类测试的效率有以下理论分析:

设 D 为软件输入空间. 一个选择性测试方法从 D 中构造出一组子集 $\{D_1, D_2, \cdots, D_k\}$. 当选取 n 个测试数据构成一次测试时,对于随机性测试而言,n 个测试数据的选取是随机抽取的,而对于分域测试,则需要先确定每个子域中的测试数据选取个数,记为:$n_i, i=1,2,\cdots,k$,且 $n=\sum_{i=1}^{k} n_i$,在每个子域中,采用均匀分布来随机选取分配给它的测试数据. 设测试数据落入子域 D_i 的概率为 p_i,一个测试数据在 D_i 上发现错误的概率为 θ_i,则一个测试数据发现软件中存在错误的概率为 $\theta=\sum_{i=1}^{k} p_i\theta_i$.

随机性测试发现至少一个错误的概率为

$$P_{\mathrm{R}}(n)=1-(1-\theta)^n=1-\left(1-\sum_{i=1}^{k} p_i\theta_i\right)^n$$

选择性测试发现至少一个错误的概率则为

$$P_{\mathrm{S}}(k,n)=1-\prod_{i=1}^{k}(1-\theta_i)^{n_i}$$

取 $\eta=1-\prod_{i=1}^{k}(1-\theta_i)^{n_i/n}$,则 $P_{\mathrm{S}}(k,n)$ 可以表示为

$$P_{\mathrm{S}}(k,n)=1-(1-\eta)^n\equiv P_{\mathrm{R}}(\eta)$$

可以看出,随机性测试和选择性测试在揭错概率比较上的优劣取决于 θ 和 η 的大小. 当 $\eta>\theta$ 时,选择性测试发现至少一个错误的概率大于随机性测试;$\eta<\theta$ 时,随机性测试发现至少一个错误的概率大于选择性测试;$\eta=\theta$ 时,选择性测试与随机性测试发现至少一个错误的概率相等.

一次选择性测试发现错误的个数的期望为

$$E_{\mathrm{S}}(k)=\sum_{i=1}^{k}\theta_i$$

一次随机性测试发现错误的个数的期望为[3]

$$E_{\mathrm{R}}(n,k)=k-\sum_{i=1}^{k}(1-p_i\theta_i)$$

1.2 博弈与均衡

n 人博弈的策略型表示指定 n 个局中人以及他们各自的纯策略空间 $S_1, S_2, \cdots, S_n$ 和这些局中人各自得到的盈利函数 $u_1, u_2, \cdots, u_n$. 将该博弈记为[6]

$$G=\{S_1, S_2, \cdots, S_n; u_1, u_2, \cdots, u_n\}$$

$S_{ij}\in S_i$,表示第 i 个局中人的第 j 个策略. 当局中人只有一个时,博弈问题退化为优化问题.

N 人混合策略:

设局中人 $i(i=1,2,\cdots,n)$ 有 k_i 个可选策略,则局中人 i 的一个混合策略是该局中人的纯策略空间 $S_i=(S_{i1}, S_{i2}, \cdots, S_{ik_i})$ 上的概率分布,记为 δ_i. N 个局中人各自采取的混合策略 $\delta_1, \delta_2, \cdots, \delta_n$ 是统计独立的. 这 n 个混合策略构成的 $\delta=(\delta_1, \delta_2, \cdots, \delta_n)$ 是博弈各方采取的策略组合,由于 δ_i 是概率分布,故称此策略组合为一个策略剖面. 记号 $\delta_i(S_{ij})$ 表示概率分布 δ_i 赋予纯策略 S_{ij} 的概率. 局中人 i 的混合策略的全体记为 $\sum_i=\{\delta_i\}$,则混合策略剖面的全体 $\sum=\{\delta\}$ 为 n 个 $\sum_i(i=1,2,\cdots,n)$ 的乘积空间. 若局中人 i 取定纯策略 S_{ij} 时的盈利为 $u_i(S_{ij})$,则局中人 i 在混合策略剖面 δ 上的期望盈利为[7]

$$u_i(\delta)=\sum_{j=1}^{k_i}\delta_i(S_{ij})U_i(S_{ij})$$

纳什均衡:

在博弈 $G=\{S_1, S_2, \cdots, S_n; u_1, u_2, \cdots, u_n\}$ 中,

如果策略组合($S_1^*,S_2^*,\cdots,S_n^*$)中任一博弈方 i 的策略 S_i^* 都是其余博弈方的策略组合($S_1^*,\cdots,S_{i-1}^*,S_{i+1}^*,\cdots,S_n^*$)的最佳对策,即

$$u_i(S_1^*,\cdots,S_{i-1}^*,S_i^*,S_{i+1}^*,\cdots,S_n^*)\geqslant u_i(S_1^*,\cdots,S_{i-1}^*,S_{ij},S_{i+1}^*,\cdots,S_n^*)$$

对任意 $S_{ij}\in S_i$ 都成立,则称$\{S_1^*,S_2^*,\cdots,S_n^*\}$为 G 的一个纳什均衡[6].

纳什定理:

在博弈 $G=\{S_1,S_2,\cdots,S_n;u_1,u_2,\cdots,u_n\}$中,若 n 是有限的,且 S_i 都是有限集($i=1,2,\cdots,n$),则该博弈至少存在一个纳什均衡,但可能包含混合策略[6].

2　软件测试过程博弈模型

2.1　问题分析

软件测试是由软件测试工作相关各方(至少有软件测试方和被测方)参加的技术活动. 尽管在测试中为了使被测方积极配合测试活动,普遍强调不以测试结果作为程序员水平评定的依据,但是在软件测试活动中的各方具有各自的利益却是不争的事实. 软件测试人员希望用较小的代价检查出尽量多的缺陷;软件被测人员希望软件测试方能够帮助自己寻找软件中的缺陷,更希望己方的软件产品得到功能性能验证,同时由于被测人员认为某些软件缺陷对被测软件功能影响较小、修改成本比较大或某些个人原因,而可能对这些缺陷不予承认或修改,这种种行为又会由于测试性质(如单元测试、系统测试、验收测试等)和测试参与各方的角色的不同(如测试是否是真正的第 3 方测试、测试有无软件使用方参加等)而有所变化.

本文认为测试活动本身就是一场由多方参与的博弈,使用博弈理论建立软件测试过程的模型不但能够给软件测试过程以合理的解释,并且能够对测试活动中各方策略的选择、测试过程的安排、测试停止条件等问题的研究提供解决的新途径.

2.2　软件测试过程博弈模型

本文讨论测试方和被测方两方参加的软件测试活动,约定如下:软件使用方 U 欲购买软件产品,指定软件测试方为认定的软件质量评测机构. 软件使用方并不知道测试方采用何种测试方法. 承接软件测试工作的测试方为独立的第 3 方,仅负责进行软件测试并出具测试结果. 软件使用方不组织和参与测试过程,但是将根据测试方出具的由测试方和被测方共同确认的测试结果决定是否购买该软件产品. 本模型中软件使用方认为在测试中发现的缺陷越多,说明软件开发方软件生产水平越差,软件产品中残留的缺陷可能越多,并使用被确认的软件缺陷数换算成的千行代码平均缺陷数作为判断的指标. 由于代码量为一常值,为了计算方便,在此模型中用被确认的缺陷个数代替千行代码平均缺陷数作为判断的指标. 软件开发方向使用方 U 销售软件产品,需要到软件评测机构进行软件产品测试. 对测试出的缺陷如果确认,则必须修改,如果不确认,则不进行修改. 在软件投入使用后由软件开发方负责维护.

软件测试方派出测试人员 T,软件开发方派出软件开发人员 P 共同进行测试活动,T 和 P 在测试活动中分别对测试方和开发方的利益负责.

{选择性测试,随机性测试}是一种对软件测试类型集合的划分,本模型以选择性测试和随机性测试两类测试作为 T 的可选策略,分别记为 S 和 R. P 由于综合考虑被发现的缺陷类型、修改难度等因素,可能仅确认并修改了其中部分缺陷,可用{高缺陷确认修改率,低缺陷确认修改率}作为其策略集,记为$\{M_h,M_l\}$. U 按照软件验收规定进行决策,其可能的选择为购买或不买.

软件测试活动过程如下:

首先,T 在收取 P 交付的测试费 C_T 后,对被测软件 SW 进行测试需求分析和测试特征分析,预计选择性测试和随机测试对应的参数 η 和 θ. T 对处于测试过程中的被测软件缺陷分布特征不能完全了解. 设 $\eta\geqslant\theta$ 的概率为 p_a,不妨记此时的参数分别为 η_I 和 θ_I;$\eta<\theta$ 的概率为 $1-p_a$,不妨记此时的参数分别为 η_{II} 和 θ_{II}. 这时 T 可能以概率 p_b 选选择性测试作为手段进行测试,以概率$1-p_b$选随机性测试作为测试手段. 设 C_S 为 T 采取选择性测试时的测试花费,C_R 为 T 采用随机性测试时的测试花费. 为了鼓励 T 努力发现软件缺陷,测试方规定 T 每发现一个缺陷将得到奖金 R_{Te}.

接着,P 以概率 p_c 选择高确认修改率方案,以概率 $1-p_c$ 选择低确认修改率方案. 假设修改不会引入新的缺陷. 设 C_M 为 P 每修改一个缺陷需要花费的单位平均修改成本. 设若 P 取较大值 M_h,则确认的缺陷数将大于等于 U 的质量要求指标 E,若 P 取较小值 M_l,则确认的缺陷数将小于 U 的质量要求指标 E. P 进行 p_c 选择时应当考虑到,如果取 M_h,则修改成本较高,而且 U 购买该软件产品的可能性下降,同时降低软件开发方对开发

人员本人软件开发水平的评价. 但如果取 M_l,则软件中可能确实隐藏着较多的缺陷,如果这些缺陷在U购买后形成软件运行的失效,则可能形成较多的维护费用. 设每个被P确认的缺陷对P带来的负面收益为 R_{Pe}. 假定U在P取定方案后软件运行期发现软件失效数的期望值为 N_{Ue},不妨设当P取 M_h 和 M_l 的情况下软件运行期发现软件失效的期望值分别为 N_{Ueh} 和 N_{Uel},每个失效的维护费用为 C_{maint}.

最后,U在P与T双方共同确认的测试缺陷数小于质量要求指标缺陷数 E 时,考虑到抽样测试时测试方可能会将不合格的软件错误的判定为合格的风险,将以较高概率 p_{dh} 购买该软件产品. 而在P与T双方共同确认的测试缺陷数大于等于质量要求指标缺陷数 E 时,考虑到抽样测试时测试方可能会将合格的软件错误的判定为不合格,将以较低概率 p_{dl} 购买该软件产品. 统一记 p_d 为U购买该软件产品的概率. 设软件产品购买价格为 R_{acc}.

由T和P的博弈行为形成的测试过程展开如图1所示. 由于U不直接参与测试活动,故对U的收益和策略在此模型中不予讨论.

R_T 和 $R_{Pi}(i=1,\cdots,8)$ 分别表示不同路径对应的T和P的收益.

T的收益为

$$R_T = C_T + E_{Error}R_{Te} - C_{test}$$

P的收益为

$$R_P = -C_T - E_{Error}R_{Pe} - C_M E_{Error}M + R_{acc}p_d - N_{Ue}C_{maint}$$

其中 E_{Error} 为测试中发现的缺陷数;C_{test} 为T的测试成本;M 为缺陷确认修改率,M 在不同分支中可对应的选 M_h 和 M_l;当T选择R和S时,E_{Error} 分别为 $E_R(n,k)$ 和 $E_S(k)$,C_{test} 分别为 C_R 和 C_S;当P选 M_h 时,N_{Ue} 取 N_{Ueh},p_d 为 p_{dl};而当P选 M_l 时,N_{Ue} 取 N_{Uel},p_d 为 p_{dh}. 据此,可以分别确定 R_{Ti} 和 R_{Pi},$i=1,\cdots,8$.

2.3 决策条件分析

测试过程中存在两次决策. 测试人员T和被测人员P都具理性,希望争取自己的决策收益最大. 下面分析博弈方的决策条件.

2.3.1 被测试方P的决策条件

P进行决策,策略集为:$\{M_h, M_l\}$.

当P选择 M_h 时,P的收益为

$$R_{PM_h} = p'_a p_b p_c R_{P1} + p'_a(1-p_b)p_c R_{P3} + (1-p'_a)p_b p_c R_{P5} + (1-p'_a)(1-p_b)p_c R_{P7}$$

当P选择 M_l 时,P的收益为

$$R_{PM_l} = p'_a p_b(1-p_c)R_{P2} + p'_a(1-p_b)(1-p_c)R_{P4} + (1-p'_a)p_b(1-p_c)R_{P6} + (1-p'_a)(1-p_b)(1-p_c)R_{P8}$$

其中 p'_a 在此处为先验概率,由P长期参加测试活动所得经验决定. 由于P在T之后进行决策,故 p_b 可通过观察T的行为得知. p_c 为当前P可主动选择的参数. 因此,理性的被测方P的策略选择条件为:当 $R_{PM_h} \geqslant R_{PM_l}$ 时,选择策略 M_h;反之,选择策略 M_l.

2.3.2 测试方T的决策条件

T进行决策,策略集为:$\{S,R\}$

当T选择S时,T的收益为

$$R_{TS} = p'_a p_b p'_c R_{T1} + p'_a p_b(1-p'_c)R_{T2} + (1-p'_a)p_b p'_c R_{T5} + (1-p'_a)p_b(1-p'_c)R_{T6}$$

当T选择R时,T的收益为

$$R_{TR} = p'_a(1-p_b)p'_c R_{T3} + p'_a(1-p_b)(1-p'_c)R_{T4} +$$

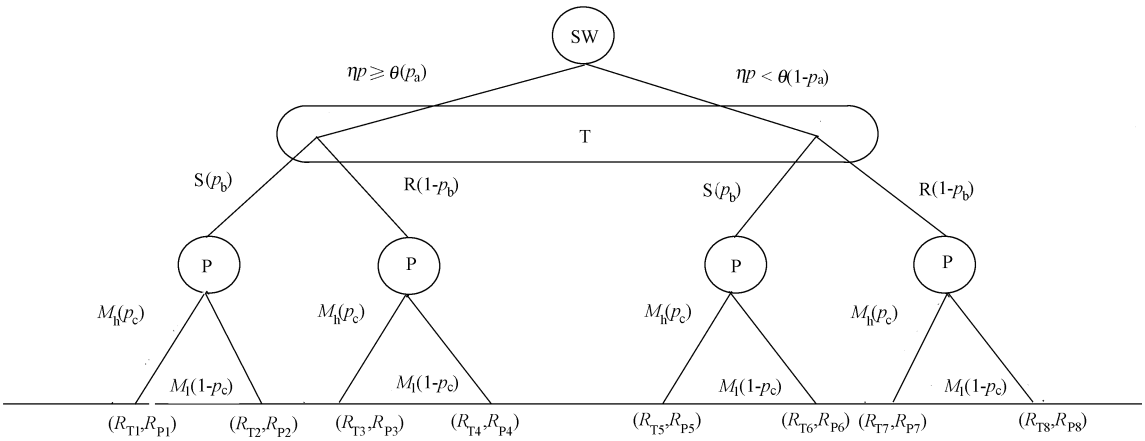


图1 博弈过程展开图

$$(1-p'_a)(1-p_b)p'_cR_{T7}+(1-p'_a)(1-p_b)(1-p'_c)R_{T8}$$

其中 p'_a 和 p'_c 在此处为先验概率,由 T 长期参加测试活动所得经验决定,p_b 为当前 T 可主动选择的参数. 因此,理性的 T 的策略选择条件为:当 $R_{TS}\geqslant R_{TR}$ 时,选择策略 S;反之,选择策略 R.

2.4 均衡条件

根据纳什定理,在允许博弈方进行混合决策时,至少存在一个纳什均衡. 下面讨论本测试过程博弈模型的均衡条件.

对于 T 来说,当 T 选择 S 时,如果 P 选 M_h,则 P 的收益为

$$R_{TSPM_h}=p_ap_bp_cR_{P1}+(1-p_a)p_bp_cR_{P5}$$

当 T 选择 S 时,如果 P 选 M_l,则 P 的收益为

$$R_{TSPM_l}=p_ap_b(1-p_c)R_{P2}+(1-p_a)p_b(1-p_c)R_{P6}$$

在博弈达到均衡状态时,应当满足条件:

$$R_{TSPM_h}=R_{TSPM_l}\tag{1}$$

当 T 选 R 时,如果 P 选 M_h,则 P 的收益为

$$R_{TRPM_h}=p_a(1-p_b)p_cR_{P3}+(1-p_a)(1-p_b)p_cR_{P7}$$

当 T 选 R 时,如果 P 选 M_l,则 P 的收益为

$$R_{TRPM_l}=p_a(1-p_b)(1-p_c)R_{P4}+(1-p_a)(1-p_b)(1-p_c)R_{P8}$$

同理,在博弈达到均衡状态时,应当满足条件:

$$R_{TRPM_h}=R_{TRPM_l}\tag{2}$$

T 为使 P 选两策略时的收益相等,从而无法作出有针对性的策略选择,应争取式(1)、式(2)同时成立.

而对于 P 来说,当 T 选择 S 时,如果 P 选 M_h,则 T 的收益为

$$R_{PM_hTS}=p_ap_bp_cR_{T1}+(1-p_a)p_bp_cR_{T5}$$

当 T 选择 S 时,如果 P 选 M_l,则 T 的收益为

$$R_{PM_lTS}=p_ap_b(1-p_c)R_{T2}+(1-p_a)p_b(1-p_c)R_{T6}$$

在博弈达到均衡状态时,应当满足条件:

$$R_{PM_hTS}=R_{PM_lTS}\tag{3}$$

当 T 选 R 时,如果 P 选 M_h,则 P 的收益为

$$R_{PM_hTR}=p_a(1-p_b)p_cR_{T3}+(1-p_a)(1-p_b)p_cR_{T7}$$

当 T 选 R 时,如果 P 选 M_l,则 P 的收益为

$$R_{PM_lTR}=p_a(1-p_b)(1-p_c)R_{T4}+(1-p_a)(1-p_b)(1-p_c)R_{T8}$$

同理,在博弈达到均衡状态时,应当满足条件:

$$R_{PM_hTR}=R_{PM_lTR}\tag{4}$$

P 为使 T 选择两策略时的收益相等,从而无法作出有针对性的策略选择,应努力使式(3)、式(4)同时成立.

虽然 P 在 T 之后进行决策,但是理性的 P 在考虑到 T 有以上的计算,同时又不甘心接受仅对 T 有利的条件,因此可能会以事先声明等手段说明自己的概率选择准则,以使 T 无法忽视 P 的收益要求.

因此,本博弈的均衡条件为式(1)~式(4)同时成立. 此时,在 T 和 P 进行决策,如果概率的取值满足这组条件,则两博弈方中的任意一方都不能通过单方改变自己的策略而使自己的收益增加,从而达成双方都满意的局面.

假设 T 和 P 双方都是理性的,则双方都会通过上面的计算明确本测试活动的均衡条件,从而预先对测试活动的进行有所把握. 当测试过程中,均衡条件满足时,理性的 T 和 P 都将对局面满意,从而停止测试活动. 因此,此组博弈均衡条件在实际测试活动中可以理解为测试活动的停止条件. 但是不幸的是 p_a 的真值,T 和 P 都无法确知.

若双方都认同先验概率 p'_a,则在式(1)~式(4)中 p_a 应取为 p'_a,此时可以达成共同确认的测试活动停止条件.

若 T 对 p_a 的主观估计为 p_{aT},而 P 对 p_a 的主观估计为 p_{aP},则若 T 和 P 都知道对方对 p_a 的估计值,则式(1)和式(2)中 p_a 应取 p_{aP},而式(3)和式(4)中 p_a 应取为 p_{aT},此时依然能够到成共同确认的测试活动停止条件.

若 T 和 P 双方中仅有一方知道对方对 p_a 的估计值,或 T 和 P 都不知道对方对 p_a 的估计值,则由于信息缺少,形成共同确认的测试活动停止条件将变得困难.

从上面的讨论可以看到,在测试活动中如果双方能够得到对方的有关信息,是可能取得“双赢”局面的. 因此在本文讨论的这种购买方不参加测试的情况下,在进行测试活动的规程细节设计时,应当努力使双方能够良好沟通,这样对 P 和 T 双方的工作开展都较为有利,双方的利益都可以得到均衡条件下的最大实现. 但是应当注意,这样对软件使用方来讲可能会产生不利的影响.

3 结 束 语

软件测试过程的博弈分析方法是不同于工程建模法、测试充分性建模法和测试控制论方法的崭新的软件测试过程建模分析方法,它从博弈决

策的角度对软件测试过程进行了描述和分析,能够揭示软件测试过程中各参与方的理性行为及其所导致的结果,这是前述的3种建模分析方法所无法具备.

本文以{选择性测试,随机性测试}为测试人员策略集,{高缺陷确认修改率,低缺陷确认修改率}为被测人员策略集,在考虑使用方购买软件的方法、维护成本、修改成本等多个收益影响因素的条件下,对只有测试方和被测方参加的测试活动进行了建模与分析,揭示了理性的双方的决策条件,并通过描述均衡局面形成的条件讨论了一组软件测试活动的停止条件. 本模型没有考虑测试方和开发方对自己在软件购买方的声誉影响因素,双方都没有考虑软件购买方的利益.

使用软件测试过程的博弈分析方法,还能够对测试过程中的许多问题和许多种情况进行有意义的建模分析. 本文将在考虑不同测试活动参与方,不同策略集,不同收益函数和不同过程等因素的基础上,进一步利用博弈分析对软件测试问题和软件工程中的过程问题进行讨论. 本文认为博弈分析方法在软件工程领域的应用将会有良好的前景.

参考文献(**References**)

[1] Zhu Hong. Axiomatic assessment of control flow based software test adequacy criteria[J]. Software Engineering Journal, 1995, 10(9):194~204

[2] 蔡开元,李永超,景 涛,等. 软件测试的控制论方法[J]. 航空学报,2002,23(5):448~454
Cai Kaiyuan, Li Yongchao, Jing Tao, *et al*. Software testing in the context of software cybernetics. [J]. Acta Aeronautica et Astronautica Sinica, 2002, 23(5): 448~454(in Chinese)

[3] 朱 鸿,金凌紫. 软件质量保障与测试[M]. 北京:科学出版社, 1997.153~155
Zhu Hong, Jin Lingzi. Software quality assurance and testing [M]. Beijing: Science Press, 1997.153~155(in Chinese)

[4] Duran J W, Ntafos. An evaluation of random testing[J]. IEEE Trans on Software Engineering, 1984,10(4): 438~444

[5] Hamlet R, Taylor R. Partition testing does not inspire confidence [J]. IEEE Trans on Software Engineering,1990, 16(12): 1402~1411

[6] 谢识予. 经济博弈论[M]. 上海:复旦大学出版社, 1997.41~83
Xie Shiyu. Game theory for economics[M]. Shanghai: Fudan University Press, 1997.41~83(in Chinese)

[7] 施锡铨. 博弈论[M]. 上海:上海财经大学出版社, 2000.15~17
Shi Xiquan. Game theory[M]. Shanghai: Shanghai University of Finance & Economic Press, 2000.15~17(in Chinese)