

文章编号:1000-6893(2002)05-0455-04

灰盒测试方法在软件可靠性测试中的应用

李秋英, 刘 斌, 阮 镰

(北京航空航天大学 011 教研室, 北京 100083)

APPLICATION OF GREY-BOX TESTING METHOD IN SOFTWARE RELIABILITY TESTING

LI Qiu-ying, LIU Bin, RUAN Lian

(Faculty 011, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

摘 要: 阐述了软件可靠性测试的概念和灰盒测试方法在软件可靠性测试中生成测试数据方面的应用,给出了方法用于可靠性测试中的工作流程,分析了其与传统的可靠性测试方法相比的优越性以及尚未解决的局限性,展望了未来的研究方向。

关键词: 软件;可靠性;运行;软件可靠性测试;灰盒测试;运行剖面

中图分类号: TP311.53; V215.7 **文献标识码:** A

Abstract: The concept of software reliability testing and the application of the grey-box testing method in generating software reliability test data are presented. A method of how it is used in practice is given. The advantage superior to the traditional software reliability testing method and limitation are also discussed. The research direction in the future is proposed.

Key words: software; reliability; operation; software reliability testing; grey-box testing; operational profile

软件可靠性测试技术作为提高软件质量和软件可靠性的重要手段,正成为国内外软件可靠性工程的主要研究方向。如何通过软件可靠性测试,及早暴露出实际使用过程中高发生概率的软件缺陷,是亟待突破的技术难关。

传统的软件可靠性测试方法耗费的时间长,测试代价高,同时,饱和效应和不精确的运行剖面等因素也严重影响了对软件可靠性的估计。针对上述问题,本文提出了一种灵活的“灰盒测试”方法,既能加速软件可靠性测试的进程,提高发现错误的可能性,又能保证生成的测试满足可靠性测试数据的统计特征,得到更为理想的可靠性估计。

1 软件可靠性测试的概念

软件测试是为了发现软件中存在的缺陷并予以排除,确保软件功能满足用户需求而进行的一系列活动。

软件可靠性测试是为了达到或验证用户对软件的可靠性要求而对软件进行的测试;通过测试发现并纠正软件中的缺陷,提高其可靠性水平,并验证其是否达到了用户的可靠性要求。

软件可靠性测试能够有效地暴露在实际使用过程中影响可靠性要求的软件缺陷,最先暴露的

一般是高发生概率的缺陷,然后是较低发生概率的缺陷。

2 灰盒测试方法的重要性

灰盒测试方法是指结合白盒测试和黑盒测试的测试方法。

白盒测试,又称结构测试,在测试过程中测试者可以看到被测的源程序,通过分析程序的内部结构,根据其内部结构设计测试用例^[1]。

黑盒测试,又称功能测试,在测试过程中被测程序被视为黑盒,测试者在完全不考虑程序内部结构和内部特征(或对于上述信息无从获知)的情况下,根据需求规格说明书设计测试用例和推断测试结果的正确性^[1]。

显然,这两类测试方法是从完全不同的角度出发对软件进行测试。实践证明,两类方法各有侧重,在测试的实践中都是有效和实用的,不能指望一类方法能够完全代替另一类方法。但显然二者又各具缺点,这些缺点不是通过在各自的测试方法内部进行完善就能够解决的。只有通过将二者有效地结合,即进行所谓的“灰盒测试”,才能弥补任何一种方法的不足,使测试方法的机理更完善。

传统意义上的软件可靠性测试的实践方法是根据用户实际使用软件的方式,即运行剖面生成测试数据,利用测试数据驱动软件运行,获得失效

数据,利用现有的软件可靠性模型进行评估,直到可靠性估计以某种形式收敛。软件可靠性测试从概念上讲采用的是黑盒测试方法,因为它是面向需求、面向使用的测试,它不需要了解程序的结构以及如何实现等问题,测试数据的生成和可靠性估计与测试中软件如何运行没有任何明确的关系。这种明确关系的缺乏,是我们反对用单纯的“黑盒测试”方法估计软件可靠性的出发点。

而且,这种传统作法生成的测试数据中,很多是对程序结构的重复执行,使得某些结构测试的密度高,某些结构测试的密度低,带来了许多无用的测试代价。大量的测试数据需要花费大量的人力、物力和财力,在交货期限紧张的情况下,往往不能满足,导致获得的可靠性估计值是不可信的。因此,需要结合“白盒测试”的信息,作为衡量测试质量和控制测试进程的手段。那么结合哪些信息以及如何有效地结合呢?

3 灰盒测试方法

(1) 理论基础 人们在长期的测试实践中,认识到:在对程序进行的可靠性估计、程序结构以及测试量之间存在着一种重要的联系。在这种思想的指导下,人们设计了各种方法来表示测试量。基于代码覆盖的度量是一种公认的简单可行的方法。一些软件可靠性方面的理论家提出:在可靠性分析中必须密切注意软件结构^[2]。

软件可靠性测试是按照用户实际使用软件的方式来测试软件,软件的运行剖面(Operational Profile)是定量描述用户实际使用软件方式的有力工具,也是软件可靠性测试最主要的特征。

本文提出的“灰盒测试”方法将已有的时域方法和测试中获得的覆盖信息相结合,利用软件可靠性测试最主要的特征,结合结构覆盖信息作为生成可靠性测试数据的约束条件和评价测试质量的精确和可测量的标准。

(2) 统计特征 在构造运行剖面的过程中,需要了解用户是如何使用该软件的。要充分了解用户使用软件的各种模式和各种功能及发生的概率,完成这些功能相应的输入变量。构造准确的运行剖面是软件可靠性测试中的一个难题,本文不讨论该问题,文中研究内容的前提假设是所获得的运行剖面足够准确。

可靠性测试数据是基于运行剖面随机生成的,用来驱动可靠性测试运行的测试数据。其统计规律,反映了用户实际使用软件方式的统计规

律,它与运行剖面描述的概率分布应该是一致的。

不妨设运行剖面具有如下形式: $\{OP_i | OP_i = \langle O_i, P_i \rangle, i = 1, 2, \dots, m\}$,其中 m 为软件运行剖面中的运行总数; O_i 为第 i 个运行, P_i 为第 i 个运行发生的概率, $\sum_{i=1}^m P_i = 1$,这里运行相互独立。

由数理统计的知识我们知道软件可靠性测试数据集是依据上面的总体分布进行抽样而产生的一组样本的观察值,因此满足下面的统计特性。

假设测试数据集是依据上面的分布,进行 N 次抽样而产生的一组样本观察值,在 OP_1, OP_2, \dots, OP_m 中的抽样个数分别为: n_1, n_2, \dots, n_m 。其中:

$$\sum_{i=1}^m n_i = N, \quad \sum_{i=1}^m P_i = 1$$

由统计理论可知,有统计量

$$T = \sum_{i=1}^m \frac{(n_i - Np_i)^2}{Np_i}$$

皮尔逊证明了当 N 足够大时,统计量 T 的渐近分布满足自由度为 $m - 1$ 的 χ^2 分布^[3]。因此可以通过检验该统计量的分布判断测试数据集是否满足了可靠性测试数据的统计特性。这是现有的软件可靠性测试方法中常常被忽视的内容。

(3) 覆盖信息的利用和存在的问题 引入文献[4]中定义的“有用测试量”的概念。所谓的“有用测试量”是指当且仅当扩大了某种覆盖的测试量。例如, k 个测试数据达到了 10% 的语句覆盖,若 $k + 1$ 个测试数据使覆盖达到 15%,则说第 $k + 1$ 个测试数据是有用的。忽略掉无用而费力的测试量,将过滤后的数据用于可靠性估计模型。如果用 \bar{R}_0 和 \bar{R}_r 分别表示原始和过滤后的数据从 Musa 模型中得到的可靠性估计,那么分析可知 $\bar{R}_0 > \bar{R}_r$,因此用过滤后的数据可以得到更为理想的可靠性估计。

文献[4]中提出了一个基于时间/结构的软件可靠性估计方法。为了解决功能测试的饱和效应导致对可靠性过高估计的影响,提出了使用覆盖信息过滤失效数据的方法。利用有用的测试量,仅当测试数据增大了某种覆盖才作为有用数据用于可靠性模型,从而忽略掉无用的测试数据。该方法尚存在严重的不足,从以下方面进行剖析:

软件可靠性测试数据是基于运行剖面生成的,满足用户实际使用软件的频率分布。所谓频率分布,是指使用哪些输入和按什么样的顺序^[2]。使用哪些输入体现的是可靠性测试数据

的统计特性,即高发生概率的运行抽样点多,低发生概率的运行抽样点少;按什么样的顺序,体现的是可靠性测试数据在时间上出现的先后次序。

文献[4]中提出的方法没有解决上述两个问题。过滤后的数据滤掉了不增加覆盖的数据,往往滤掉了运行剖面中高发生概率的执行相同程序结构的输入,会导致过滤后的数据的分布特征不符合运行剖面所代表总体的分布特征。顺序上的不满足是显而易见的,这种提高覆盖的抽取方式使得运行剖面中后发生可能的输入提前发生的可能性大大提高了。上述不足使得该方法得到的可靠性测试数据可能不能代表实际的可靠性测试数据,得到的可靠性估计值与实际值的偏差可否忽略是一个不确定的问题。

(4) 灰盒测试方法在软件可靠性测试中的应用 文章提出的基于“灰盒测试”的软件可靠性测试方法,试图解决上述第一个问题。首先通过运行剖面生成一定数量的测试数据,由统计理论可知,这些测试数据与母体具有相同的分布。用这些测试数据驱动软件运行,获得失效数据后,利用可靠性模型估计可靠度。如果可靠度的估计值大于用户提出的可靠度指标,则可以停止测试。否则通过有针对性地为提高覆盖率而生成测试数据,当此时的测试数据集合的分布与母体具有相同的分布时,重复上述测试、估计、比较的过程直至可靠度的估计值达到可靠度指标。否则,给测试数据集合一个增量,该增量按照运行剖面进行抽取,再重复上述步骤,直至可靠度的估计值达到可靠度指标。工作流程图详见图 1 所示。

该方法的优越性: 当运行剖面准确时,该方法生成的测试数据既满足可靠性测试数据的总体分布特征,又具有较高的覆盖率,具有发现更多的缺陷的可能性。当运行剖面不够准确时,该方法通过较高的覆盖率也能够保障高发生概率的错误被发现的可能性大。克服由于饱和效应带来的对可靠性估计过高的问题。

综上所述,该方法能够保证在同样的基础上,所需的测试数据的数量尽可能小,并且满足运行剖面的统计特征,保证覆盖率尽可能高。

(5) 方法的局限性 文章提出的方法解决了文献[4]中提出的方法未能解决的问题,通过总体分布类型的假设检验保证了测试数据满足运行剖面的统计特征。

需要指出的是,该方法同样未能解决上述分析中的第二个问题。利用覆盖信息辅助得到的测

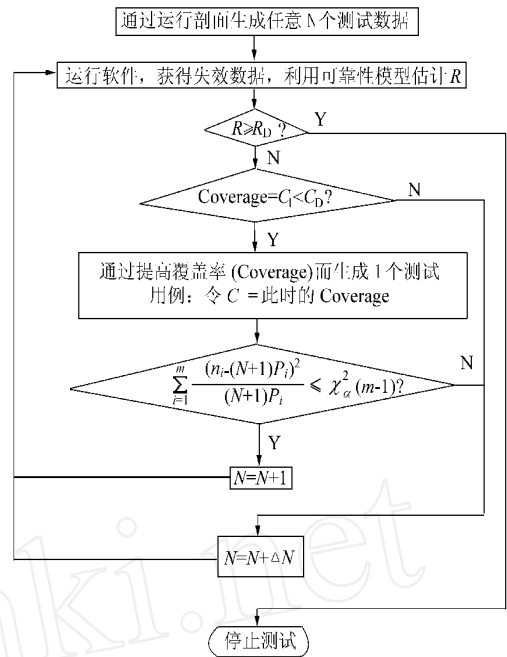


图 1 灰盒测试方法的工作流程图

Fig. 1 Work flowchart of grey-box testing method

试数据使得实际使用中后发生的输入提前发生的可能性大大提高了。这是该方法需要改进之处。

在后续的研究工作中,将通过引入“压缩比因子”的方法,将引入覆盖信息的测试数据“恢复”到未引入覆盖信息时的测试数据,解决输入顺序的问题。具体关于定义“压缩比因子”的方法和“恢复”测试数据的方法,都是需要研究解决的问题。

4 结论

(1) 本文提出的灰盒测试方法,能够将白盒测试和黑盒测试方法有效地结合起来,不仅保证测试数据满足运行剖面的统计特征,而且保证测试数据的覆盖率尽可能高。

(2) 本文提出的方法尚存在不足,不能保证测试数据的时序特征,这将在后续的研究工作中加以改进。

参 考 文 献

- [1] 郑人杰. 计算机软件测试技术[M]. 北京:清华大学出版社,1992, 44 - 45.
(Zheng R J. Computer software testing technique[M]. Beijing: Tsinghua University Press, 1992. 44 - 45.)
- [2] Michael R. Lyu 主编. 软件可靠性工程手册[M]. 刘喜成译. 北京:电子工业出版社,1997. 356 - 375.
(Michael R. Lyu Ed., Handbook of software reliability engineering[M]. Liu X C Tr. Beijing: Publishing House of Electronics Industry, 1997. 356 - 375.)
- [3] 贺国芳. 可靠性数据处理与寿命评估[M]. 北京:北京航空

航天大学出版社,1991,122-123.

(He G F. Reliability data processing and life-span assessment [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 1991, 122-123.)

- [4] Chen M, Horgan J R, Mathur A P, *et al.* A time/structure based model for estimating software reliability[R]. Technical Report SERC-TR-117-P, W Lafayette Ind: Purdue University, 1992.

作者简介:



李秋英(1973-),女,黑龙江省大庆人,北京航空航天大学工程系统工程在读博士研究生,主要研究方向为软件可靠性工程、软件可靠性测试及软件可靠性测试充分性理论,Email: li-qiuying@sohu.com,联系电话:82316442,通信地址:北京航空航天大学2-71信箱,100083。

刘斌(1968-),男,山西太原人,北京航空航天大学博士后,主要研究方向为软件可靠性工程、软件可靠性测试及软件可靠性仿真测试,联系电话:82316440,通信地址:北京航空航天大学011教研室,100083。

阮镰(1938-),男,上海人,北京航空航天大学工程系统工程系教授、博士生导师,主要从事软件可靠性工程方面的研究,现为IEEE member。联系电话:82316570,通信地址:北京航空航天大学011教研室,100083。

(责任编辑:李铁柏)

www.cnki.net