

数据变异测试技术

Data Mutation Testing Technology

海军工程大学 马 乐

开展软件测试用例生成技术的研究并开发出适用的测试用例生成系统,实现软件测试过程自动化有着十分重要的现实意义。

可靠性是计算机系统的重要指标之一,在航天、航海、金融、交通、核工业控制等很多任务或时间关键性领域中,计算机系统一旦在使用中出现故障后果不堪设想,可能造成巨大的经济损失和人员伤亡。因此对这些系统在投入使用之前一定要进行充分的测试,消除系统内可能存在的故障和漏洞,同时也要在使用之前模拟未来使用环境中可能出现的环境扰动,如常见的单粒子翻转故障,以对系统的可靠性等指标进行充分评测,根据测试结果采用必要的设计和防护方法,保证其满足可靠性要求。

目前,软件测试的自动化工具主要集中在测试用例的执行和维护以及测试覆盖情况的度量方面,测试用例的自动生成还远不尽如人意。提高测试用例生成的自动化程度是提高动态测试工具乃至整个测试过程自动化程度的关键。测试用例自动生成这一技术的实现,将大大改变以往靠直觉、经验产生测试用例的传统做法,无疑将使软件测试的效率获得显著提高,同时减轻人们在编写大量测试用例过程中所付出的劳动。因此开展软件测试用例生成技术的研究并开发出适用的测试用例生成系统,实现软件测试过程自动化有着十分重要的现实意义。

自 20 世纪 60 年代以来,国内外许多学者和组织在软件测试方面进行了大量研究工作。目前,测试用例的自动生成技术大体上可以分为两种:基于规范的功能测试和基于程序的结构测试。

Tsai 等^[1]提出用关系代数查询表示的规格说明生成测试数据的方法,该方法针对关系代数查询表示的规格说明设计测试用例,适于数据库和数据处理系统。Weyuker 等^[2]提出基于布尔规格说明的测试数据生成方法,该方法将 Loveson 的基于过程控制的形式化规格说明方法^[3]进行改进。Hall^[4-5]提出利用 Z 规格说明进行软件测试数据生成的方法,该方法要求测试人员具有丰富测试经验和较高专业水平。兰毓华等^[6]在此基础上,提出了基于 Z 规格说明的软件测试用例自动生成方法,该方法在自动生成测试用例之前,必须将软件需求规格说明用 Z 语言进行形式化描述,因此对大型系统实现困难。

Chen 等^[7]提出了一种对关系进行一致性检查和自动演绎的理论框架和方法,在此基础上描述了一个支持分类测试用例生成的选择关系框架,并开发了一个从关系生成测试框架的算法,实现了基于测试框架的测试用例生成。虞凡等^[8]提出了一种

基于时序逻辑语言 XYZB 软件规范说明的测试用例自动生成方法,该方法采用 XYZ/E 描述,首先确保规范与被测软件的语义一致,其次,利用 XYZ/E 规范的逻辑程序属性,实现了对测试用例中输出期望的 Prolog 自动推导求值。

结构测试要求对被测程序的结构特性做到一定程度的覆盖,如语句覆盖、分支覆盖、数据流覆盖、路径覆盖等。其测试数据生成是在输入域中,寻找满足测试准则(如语句覆盖)的输入数据的过程。关于结构测试,有 3 种测试数据生成方法:随机测试数据生成方法、面向目标的测试数据生成方法以及面向路径的测试数据生成方法^[9]。

(1)对给定的语句,随机测试数据生成方法是在输入域内随机选取测试数据,使得给定语句被执行。

(2)面向目标的测试数据生成方法依据程序控制流信息,将程序中所有的分支分成两类:一类是影响目标结点的分支,另一类是不影响目标结点的分支。测试数据生成时,考虑影响目标结点的那些分支,利用分支函数极小化,生成相应的测试数据,使得给定语句被执行^[10-12]。

(3)对面向路径的测试数据生成方法,首先确定一条经过给定语句

的程序路径,然后在输入域中寻找输入数据,使得在此输入之下,程序执行沿该路径进行,从而使得给定语句被执行。面向路径的测试数据生成又分为符号执行^[13]和实际程序执行两类^[14-16]。

符号执行

符号执行允许程序的输入不仅可以是具体数值,也可以是符号值、符号表达式等。符号执行以符号计算代替实际执行的数值计算,产生一个符号输入值的代数表达式——路径约束,它是选定路径的谓词系统,实际上是对输入数据的限制要求,由多个不等式(等式)组成。通过求解不等式,求取满足路径上各限制谓词的测试数据,若不等式无解,则相应的路径为不可行路径。因此符号执行能够判定路径的可行性,但在遇到循环、过程调用、动态数据结构、数组和指针处理时,符号执行实现困难。其优点是一次符号测试的结果代表了一类普通测试的运行结果,因此测试成本较低。

实际程序执行

基于程序实际执行的测试数据生成方法中, Korel 对选定路径上不满足要求的分支利用分支函数极小化,确定新的输入值,直至找到输入数据,使程序执行沿选定路径进行。Gupta 等^[14]利用迭代逼近法,求取满足选定路径上所有谓词的输入值,生成测试数据。Gotlieb 提出用约束求解的方法生成测试数据,利用静态单一指派和控制依赖,将被测程序转化为一个约束系统,然后寻找经过给定语句的路径,生成相应的测试数据。Visvanathan 等^[15]提出一种针对指针输入函数的方法,这种方法允许输入数据结构形式的生成与其数据值的生成相对独立,从而促使函数的控制流沿着给定的路径流动。在实时数据驱动的航天软件测试中,一般常用

模飞或标校数据验证软件数据处理接口和流程正确性,但这种单一的理想数据无法冲击软件使之暴露异常,因此,在此类软件测试活动中测试数据的构造成为测试用例设计的核心。在允许的数据笛卡尔空间中用人工挑选、机器枚举、随机组合寻找软件失效数据基本上等于无的放矢,采用基于模拟退火的多亲遗传算法^[17]帮助搜索可能的最优或局部最优解将大大降低测试数据工作量,提高测试命中率,从而达到软件可靠性测试的目的。

变异测试也是一种常用的软件测试方法,此方法将一个已通过验收测试的程序进行修改,生成一个或多个变异体,再用测试用例集对变异体进行测试。若测试用例的运行结果出错,则说明测试用例集能检测出变异体中的故障。通常,可以通过对生成的多个变异体进行故障检测,来评估测试用例集的质量和效率^[18]。

然而,以上这些方法在应用于复杂结构输入的软件测试时可用性将受到一定程度的限制。通过修改某些测试用例来生成新的测试用例的构想也在其他研究中被提出过。Meek 和 Siu^[19]曾使用随机错误播种的方法来评估语言处理器在源程序中检测和报告错误的能力。另外一种使用测试用例的变化产生新用例的方法是由 Chen 等^[20]提出的变形测试方法,主要目的是通过使用变形关系而不是正式规格说明书来降低测试中预测结果的困难。

将数据变异方法应用于测试用例的自动生成由 Shan 等提出^[21]。生成足够的测试用例是软件测试中最困难也是最昂贵的任务之一,特别是在软件系统的输入具有复杂结构的情况下。一个具有复杂结构的输入可能包含多个元素,其相互关系也会受到一系列约束规则的限制。例如,软件建模工具的输入就是一个受到建模语言语法限制的模型^[22]。建模

工具在建立一个模型的过程中不仅依赖于单个元素的取值,同时还受到元素间相互关系的约束。在这样的背景下,提出了一种称为数据变异测试的测试用例生成技术。此方法通过选择一个小的种子测试用例集,对它进行一系列数据变异算子的设计和应用,从而生成大量的测试用例。

那么在航空航天领域,大量安全关键的实时软件系统等一系列工作环境、过程、任务承担着数据采集、处理和分析任务,如何模拟各种故障数据,对其进行充分测试一直是工程实践中的现实难题,因此在 20 世纪 70 年代提出了一个新技术软件故障注入^[23],然而在当时并没有引起学者们的注意,直到 20 世纪 90 年代才开始真正注意故障注入技术的研究,并迅速使之成为计算机领域的一个热点研究领域。国外科研机构开发出很多可应用于实际的软件故障注入器,包括 Doctor、FST、FERRARI、GOOFI、FTAPE、FINE、Xception、JIFI 等,其中最有代表性的是 NASA 的 JPL 实验室开发并实施的 JIFI 软件故障注入工具。

在国内,对故障注入技术的研究虽然也始于 20 世纪 90 年代,目前的研究主要集中在硬件故障注入工具,而对软件故障注入技术的研究相对较少,软件故障注入技术的应用也相对比较受局限。中科院电子研究开发了基于单片机系统的软件故障注入器;哈尔滨工业大学开发了以软件实现的故障注入工具。国内未来故障注入技术的主要创新点以及将来工作重点是在基于变异测试的故障注入研究中如何完善基于变异测试的故障注入,完善 Tester,增加更多的变异算子,执行更有效变异试验,充分评测测试集的有效性和充分性。

本文共有参考文献 23 篇,因篇幅有限,未能一一列出,如有需要,请向本刊编辑部索取。(责编 良辰)