软件测试技术的发展*

黄 明 李文健 许建潮

（吉林工学院计算机系）

摘 要

软件测试是提高软件可靠性的重要手段，也是软件工程中最活跃的研究领域之一。鉴于今后若干年内软件测试仍将是软件质量保证的主要手段，因而我们在本文中讨论了软件测试的发展历史、测试策略、测试工具、软件测试的现状及其发展趋势。

一、引 言

软件测试是保证软件可靠性的主要方法之一，其目的是发现软件错误，提高软件产品质量。大量统计资料表明，软件测试的工作量往往占软件开发总工作量的40%以上。对软件可靠性来说，尽管测试与程序正确性证明技术相比是一种不完善的技术，但目前程序正确性证明技术还远远没有达到实用阶段。即使有了正确性证明程序，软件测试也仍然需要，因为程序正确性证明只证明程序功能是正确的，并不能证明程序的动态特性是符合要求的，另外，正确性证明过程本身也可能发生错误。所以，到目前为止软件质量主要靠软件测试来保证。要想在短时间内避开成本昂贵的软件测试活动，看来很难做到。

本文首先介绍目前比较流行的软件测试技术，然后以我们正在开发的Prolog软件测试程序从新测试程序文件中取出，然后编译和运行。

由于用户可通过命令和控制信息对测试程序进行灵活的结合和修改，而花费的代价最小。因此，编辑方式优于前两种方式。

*吉林省青年科学基金资助项目

参考文献

[1] 编译程序的测试方法和实现，李友仁等，计算机技术，1985年，第5期。

[2] NBS FORTRAN Test programs, Volume 1—3, F.E.Holberton, E.G.Parker, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, 1974.

[3] The Validation System of Pascal, British Standard Institute, 1982.

[4] The 1985 COBOL Compiler Validation System (CCVS)——User Guide Version 1.6, The National Computing Centre Ltd., Manchester, U. K., January 1988.

[5] Programming Languages CERTIFIED COMPILER LIST, Edited by J. B. Kailey, U.S. DEPARTMENT OF COMMERCE National Institute of Standard and Technology, April, 1989.

[6] COMPILER TESTING PROCEDURE, U. S. DEPARTMENT OF COMMERCE National Institute of Standard and Technology, April, 1989.

[8] The ADA Compiler Validation Capability, J.B.Goodenough, 1980. ACM.

试环境RSTE为背景，简单介绍逻辑程序的测试技术及其支持环境的现状。最后，对软件测试技术的发展趋势作一展望。

二、软件测试技术的发展历史

1975年，Goodenough和Gerhart定义了测试可靠性模型，奠定了软件测试的理论基础。此后，Howden、Geller和Gourlay等人相继探讨了路径测试的可靠性、软件测试在程序正确性证明上的应用以及软件测试的数学模型等问题，促使测试理论深入发展。总的来说，软件测试技术的发展经历了三个阶段：

1．初期阶段（60年代初—70年代初）。这个阶段的主要工作是个别方法的探索和不同方法的收集，这个时期的代表著作是Hetzel的"程序测试方法"。

2．发展阶段（70年代初—70年代末）。这个阶段的主要工作是逐步建立了软件测试的理论基础，代表著作是Myers的"软件测试技巧"。

3．成熟阶段（70年代末—现在）。这个阶段的主要工作是建立了各种测试工具和环境，大大丰富了测试的理论和方法，代表著作是Beizer的"软件测试技术"与"软件系统测试和质量保证"。

三、软件测试的方法

测试是为了发现程序中的错误而执行程序的过程。测试技术只能发现程序中有差错，无论如何也不能断定程序中没有差错。典型的软件测试方法有两种：一种是黑盒测试法，在使用这种方法时，测试者把程序看成一个黑盒。即完全不考虑程序内部结构和内部特性。测试者仅仅关心寻找程序没按规范运行的情况，并且仅按程序的规范导出测试数据；另一种是白盒测试法，这种方法的前提是可以把程序看成装在一个透明的白盒子里，也就是完全了解程序的结构和处理过程。这种方法按照程序内部的逻辑测试程序，检验程序中的每条通路是否都按预定要求正确工作。测试者从检查程序的逻辑着手，得出测试数据。不论用哪种方法进行测试，要作到穷尽测试都是不可能的。设计测试方案是测试阶段的关键技术问题，其实用策略是：用黑盒法（边界值分析、等价划分和错误推测法等）设计基本的测试方案，再用白盒法补充一些必要的测试方案。

四、测试工具和软件开发支持环境

在现有的软件工具中，测试工具所占的比例最多。测试工具可分成静态分析工具、动态测试工具、测试数据生成工具等。静态分析工具不需要执行被测试的程序，它仅仅扫描被测试程序的正文，从中寻找可能导致错误的异常情况；动态测试工具需要运行被测试的程序，利用插入的检查指令来获取关于程序行为的统计数字；测试数据生成工具可以为测试某个系统而自动产生大量输入数据，但是目前最完善的工具也不能生成测试用例，因为它包括测试输入和期望的输出结果。所谓软件开发支持环境包括整体化工具系统与高级语言环境两个方面。1985年8月在伦敦举行的第八届国际软件工程会议上，一个由IEEE和ACM的SIGSOFT支持的国际工作小组提出了对软件开发环境等的定义："软件开发环境是相关的一组软件工具的集合，它支持一定的软件开发方法或按照一定的软件开发模型组织而成。"

近年来由于迫切需要提高软件生产率、降低软件开发成本，这就进一步促进了软件开发环境的发展，已研制了许多成功的软件开发环境。支持整个软件生存周期的软件开发支持环境能产生需求规范和设计规范等中间结果，这些结果使开发的进程易为人们所见，便于管理人员有效地控制他们的课题进度，保证了软件产品的质量。

五、软件测试技术现状及发展趋势

1．现状

G.Myers概括了三条比较重要的测试原则：第一，程序测试是为了发现错误而执行程序的过程；第二，好的测试情况能够有高度的可能性从程序中查出先前未发现的错

误；第三，能查出先前还未发现的错误的测试是成功的测试。

1）目前，大量的测试工作几乎处在盲目进行的状态，没有理论和现成方法的指导。虽然测试理论已有一些，但要把这些理论或方法直接用于实际中还有一段距离。如Boris Beizer提出的求程序流程图的路径表达式方法，它只适用于单入口、单出口程序，只能求出入口到出口的路径表达式。这对于测试实际程序以及程序数据流分析是不够的。另外，R.E.Parther提出的求最小路径集合的算法只能适合于结构化程序，而且用该算法求出的最小路径集中的路径，对多数程序来说都不符合程序测试路径的选择原则。

2）测试工具的使用使软件的可靠性及生产率都有所提高，但目前的测试工具并非完美无缺。有的测试工具本身有缺陷，要求用户提供大量的辅助信息，降低了测试效率。有的测试工具用户接口设计得不好，测试完毕后输出大量信息，而其中只有一少部分是有用的，加重了用户的负担。

3）Prolog程序设计环境的不足。现有的高级程序设计语言都是建立在冯·诺依曼体系结构上的。自从1982年日本的第五代计算机计划把Prolog选为核心语言以后，在世界范围内掀起了一股研究和使用Prolog语言的热潮。但Prolog是一种描述式语言，其执行机制完全不同于传统的程序设计语言。目前在Prolog测试工具方面的研究进展很小：Lawrence提出了描述Prolog程序控制流程的框模型，重庆大学童频等人提出了一种检测Prolog程序中无限循环的算法，Shapiro提出了算法调试方法，Pereira提出了关系调试方法等。对Prolog的测试不同于传统的测试，因为Prolog程序的执行是不确定的；规则的输入输出关系不象传统软件中过程那样精确；激活规则的方式太多、不能使用分支—路径—覆盖工具。

2．发展趋势

人工智能技术将被用于软件测试过程中。由于测试是一门以经验知识为主的学科，AI特别是专家系统方法的引进，可能使软件测试产生一次飞跃。清华大学已提出了一种基于知识的测试模型。光靠AI技术尚不足以解决软件测试方面的问题，因为实际的AI系统所需要的不只是AI技术，还需要许多其它技术（例如数据库技术和用户界面技术等）支持测试活动。

六、结束语

由于Prolog语言越来越多地应用于各行各业，研制Prolog测试工具是当务之急。我们已开始Prolog程序设计环境和工具的研究。准备用两年时间开发一个Turbo-Prolog程序设计环境。

参考文献

1. H. J. Komorowski, S. Omori, A Model and an Implementation of a Logic Programming Environment, ACM 0—89791—165—2/85/006/0191, 191—198
2. J.B.Goodenough, S.L.Gerhart, Toward a theory of test data selection, IEEE Trans. Software Eng. June, 1975
3. R. Balzer et al., Software Technology in the 1990's, Using a New Paradigm, Computer, vol.16, №.11, 1983
4. David Casey, Roddy W. Erickson, Practical Tools for Software Test Certification, COMPCON SPRING, IEEE, 1984
5. 黄明，人工智能研究的现状，计算机工程，1989.6
6. 郑人杰等，软件测试技术现有缺陷分析，计算机研究与发展，1988.9
7. 董士海，计算机软件工程环境和软件工具，科学出版社，1988