

基于软件自动修复评估缺陷定位技术的工具: GenProg-FL

纪 涛 齐玉华 毛晓光

(国防科学技术大学计算机学院 长沙 410073)

摘 要 虽然缺陷定位技术和软件错误自动修复技术已经得到一定的发展,但是软件的修复工作仍然需要程序员投入大量的时间和精力。大多数开发者仍然使用传统调试技术(例如断点)来进行手工的调试,缺陷定位技术的研究成果并没有较好地运用到实际的修复工作中。近来,软件错误自动修复技术得到了快速的发展和广泛的关注。在软件错误自动修复工作中,利用缺陷定位技术自动定位错误代码是必需的,而定位的精度直接影响到补丁的生成,从而对修复的效果产生较大的影响。GenProg-FL 工具可以接受不同的缺陷定位技术去自动修复故障程序。同样,使用 GenProg-FL 可以从软件自动修复的角度评估现有的基于程序谱的缺陷定位技术定位的有效性。

关键词 缺陷定位,软件自动修复,GenProg-FL

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.09.016

Tool for Evaluating Effectiveness of Fault Location Techniques Based on Automated Software Repair: GenProg-FL

Ji Tao Qi Yu-hua MAO Xiao-guang

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

Abstract Although the technique of fault location and automated program repair have been developed in recent years, programmers still need to spend a lot of time and effort on repairing. Most developers still work with traditional debugging techniques such as breakpoints. The research results of fault location have not been applied in practical work of repairing. In recent years, the technique of automated program repair has been concerned and developed. In the work of automated software repair, using the technique of fault location to locate the bugs is a necessary activity and the accuracy of localization affects the generation of patches, which has a great impact on the effect of repair. GenProg-FL tool can use different techniques of fault location to repair programs. Also, GenProg-FL can be used to evaluate the effectiveness of fault location techniques.

Keywords Fault location, Automated program repair, GenProg-FL

1 引言

随着软件规模的日益增长,软件维护成为软件生命周期中非常重要的过程。因为人力有限,程序员几乎不可能开发出没有缺陷的软件。在软件运行的过程中,如果缺陷被激活,将导致软件的失效^[1]。为了尽快地修复软件故障,程序员必须花费大量的时间与精力查找导致软件失效的原因,并发布补丁。

多年来,缺陷定位技术已经得到了很大的发展,它的发展旨在为程序员提供一个精确的缺陷定位结果,以方便程序员能够在最短的时间内修复存在缺陷的故障代码。有关缺陷定位技术的研究相对较早,一些经典的定位技术已经开始应用于实际软件开发中。另外,许多文献对各种缺陷定位技术的定位有效性进行了评估,并且提出了一些定位效果突出的技术。但是,目前的定位技术也具有计算复杂度与精度之间的平衡关系。对于较大规模的软件系统,计算量过大以及精度

不理想等因素仍然影响此技术在工业界的普遍应用。如何将这些缺陷定位技术引入到实际软件开发中,是一个日益重要的问题。

为了将程序员从繁重的修复工作中脱离出来,近来许多软件自动修复技术被提出来,例如 GenProg^[2,3],即一个达到高水准、自动修复 C 程序的基于遗传算法的工具。在最近的实验中,其成功修复了故障程序中真实存在的 105 个 bug 中的 55 个^[3]。但是 GenProg 在修复的过程中,只使用了一种非常简单的缺陷定位策略来计算代码中存在缺陷的可疑值。虽然作者在 GenProg 最近的工作中声明“其它的缺陷定位策略可以被引入到 GenProg 中”^[3],但是他们并没有做更进一步的研究。

2 GenProg-FL 的使用说明

GenProg-FL¹⁾ 基于 GenProg,并在此基础上增加了对其它基于程序谱的缺陷定位技术的支持。通过使用 GenProg-

¹⁾ <http://jaist.dl.sourceforge.net/project/afl/GenProg-FL.tar.bz2>

到稿日期:2013-12-25 返修日期:2014-01-16 本文受国家自然科学基金(61379054)资助。

纪 涛(1991—),男,硕士生,主要研究方向为软件自动修复,E-mail:jitao_91@163.com;齐玉华(1986—),男,博士生,主要研究方向为软件自动修复、软件调试技术等;毛晓光(1970—),男,博士,教授,博士生导师,主要研究方向为高可信软件技术、软件自动修复等。

FL 工具可以研究在软件错误自动修复领域中,各种缺陷定位技术的定位有效性。同时,可以使用高效的缺陷定位技术来提高软件修复的效率。

2.1 GenProg 简介

GenProg 扩展了遗传算法的概念,并将其开创性地应用到了故障修复工程上。GenProg 作为软件错误自动修复技术的一种,主要的修复步骤仍然是 1) 缺陷定位, 2) 生成候选补丁, 3) 验证候选补丁, 如图 1 所示。

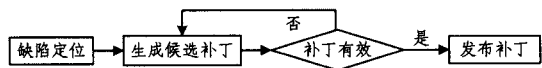


图 1 软件错误自动修复的过程

2.1.1 缺陷定位

通过研究发现, GenProg 所使用的缺陷定位技术是一种非常简单的缺陷定位策略, 每条语句经过处理之后, 运行不同的正反测试用例, 便可以通过计算得到一个可疑值 sp 。如果一条语句只被正测试用例执行过, 那么它的可疑值 sp 为 0; 如果一条语句只被反测试用例执行过, 那么它将拥有高的可疑值 1.0; 如果一条语句既被正测试用例执行过, 又被反测试用例执行过, 那么它将被赋予 0.1 的可疑值。GenProg 的遗传算法是一种随机算法, 在补丁生成的过程中, 将依据语句的可疑值进行随机的操作, 显然这样的可疑值在大规模程序中并不一定能出色地完成缺陷定位的工作。GenProg 的作者 Werimer 也提出: 在生成补丁的过程中, 在限制可修改的代码区域为与软件故障相关的语句的措施下, GenProg 可以大大缩减候选补丁的数量^[2]。显然, 精确的缺陷定位信息可以更好地缩减候选补丁的数量。

2.1.2 补丁的生成与验证

GenProg 的作者 Werimer 提出: 假设软件缺失或者表现异常的特性, 可能出现在软件的其他部分, 因此 GenProg 可以通过复制或替换部分语句, 修复软件故障。GenProg 基于遗传算法, 其在生成候选补丁的过程中主要涉及到 3 个基本的操作: 突变 (mutation)、交叉变异 (crossover) 和选择 (selection)。合适度 (fitness) 的大小即为该补丁能通过正反测试用例的数量。突变主要有插入 (insert)、删除 (delete)、交换 (swap) 的操作。通过这些对源代码的语句进行操作, 便可以产生一个新的补丁。将两个个体 P, Q 分别复制给 C, D , 随机选择一个分割点 (cutoff), 然后对在此之后的语句进行交换, 比如 $[P1, P2, P3, P4]$ 和 $[Q1, Q2, Q3, Q4]$ 有分割点 2, 那么交叉变异之后便得到 $[P1, P2, Q3, Q4]$ 和 $[Q1, Q2, P3, P4]$ ^[3]。

GenProg 首先对故障程序进行突变的操作, 生成一定数量的补丁, 组成第一代种群 (population)。进入到下一代之后, 便开始进行选择的操作, 通过对其合适度计算, 筛选出合适度不为零的补丁, 数目不能超过期望种群大小的一半, 如果种群中所含补丁的数目不足期望的一半, 那么将源程序重新填入到此种群中。此时, 便可以得到数目为期望值一半的种群。这时, 进行交叉变异的, 便可以得到数目为期望值的种群。接着需要进行变异的操作, 在每个补丁进行变异后, 将所得的补丁填入到这一代的种群中, 便得到了数目为期望值两倍的新一代种群。迭代运算直到发现合适度达到要求的补丁。

候选补丁生成之后, 选择一个原来引起软件失效的反测

试用例来确定故障已经被修复, 还需要多个原来运行正常的正测试用例来保证软件的正常行为没有因为代码的修改而破坏。即只有当合适度 (fitness) 达到指定数值, 才能认为此补丁有效。

2.2 功能介绍

通过上述分析发现, GenProg 的缺陷定位策略在软件错误自动修复中略显不足。另外, GenProg 的缺陷定位策略是一种统计的办法, 基于统计的缺陷定位技术主要是采用统计方法进行缺陷定位。最典型的该类缺陷定位技术就是基于程序谱的缺陷定位技术。基于程序谱的缺陷定位技术作为最成熟的缺陷定位方法之一, 通过动态程序分析技术, 通常以代码覆盖率来分析程序位置与失效之间的联系, 最后输出一个所有程序位置可疑值排序表^[4]。基于程序谱的缺陷定位技术的输出结果与 GenProg 原有的缺陷定位策略的输出结果在格式上基本一致。

GenProg-FL 在 GenProg 的基础上增加了对不同基于程序谱的缺陷定位技术的支持。不同的缺陷定位技术有不同的缺陷定位效果, 从而影响到软件的自动修复效率。从软件错误自动修复的角度出发, 选取修复过程中产生的候选补丁数量作为评估标准, 可以使用 GenProg-FL 对现有的基于程序谱的缺陷定位技术进行评估。这对缺陷定位技术的发展提供了一种新的思路并且具有十分重要的意义。

在我们最近的实验中, 对 14 种基于程序谱的缺陷定位技术进行了评估, 发现某种计算模型在软件缺陷自动修复中的定位效果突出, 在找到正确的补丁之前, 所生成的候选补丁数量明显少于其它 13 种缺陷定位技术和 GenProg 原有的定位策略。所以, GenProg-FL 在更先进的缺陷定位技术的指导下, 能够有效地提高其软件错误自动修复的效率。配合使用了先进的缺陷定位技术的 GenProg-FL 工具, 相比于 GenProg, 其软件自动修复的效率得到了很大提高。

2.3 安装使用

GenProg-FL 能够正常运行在 ubuntu10.04 的操作系统上。GenProg-FL 由 OCaml 语言编写, 所以在安装的过程中首先需要先安装 OCaml 的编译运行环境。另外, GenProg-FL 需要使用 CIL 对 C 代码进行插装处理和生成抽象语法树等操作。

基于自动修复的角度评估缺陷定位技术的定位效果, 需要一定数量的故障 C 程序作为 benchmarks, GenProg 在最近的实验中成功修复了 105 个 bug 中的 55 个, 所使用的故障程序¹⁾可以很出色地作为基准程序来评估各个缺陷定位技术在软件自动修复中的定位效果。

在我们最近的实验中, 发现 Jaccard 的缺陷定位效果在自动修复中明显优于其它的缺陷定位技术。在使用 GenProg-FL 对故障程序进行修复的过程中, 使用 Jaccard 可以大大地提高软件错误自动修复的效率。所以, 我们建议使用 Jaccard 来指导 GenProg-FL 进行软件错误自动修复^[5]。

3 可行性分析

3.1 评估标准

一般来说, 在 GenProg-FL 修复过程中, 缺陷定位得到了每条语句对应的可疑值。如果故障语句的可疑值越大, 其它

¹⁾ <http://dijkstra.cs.virginia.edu/genprog/resources/genprog-icse2012-benchmarks>

语句的可疑值越小,修复的效果就越好。从软件错误自动修复的角度出发,缺陷定位技术所得到的可疑值与修复的效果有直接的关系。

而缺陷定位技术的传统评估方法,是通过观察故障语句在产生的可疑值排序表中的排序情况来评估有效性。但是,这种评估测量方法受到其它问题的影响。一方面,精确地定位故障语句是很费时的或者说不可能的,尤其是对现实中复杂的故障;另一方面,在排序表中排序一定,可疑值不同,对软件错误自动修复的影响也会不同。

在软件错误自动修复的过程中,生成候选补丁的效率在很大程度上受到缺陷定位效果的影响。也就是说缺陷定位的效果直接影响到所产生的候选补丁的数量。在有效补丁发现之前,缺陷定位的效果越突出,产生的候选补丁就越少,而验证补丁的过程基本一致,所以缺陷定位的效果决定了软件自动修复的效率。所以,我们提出在一个有效的补丁得到验证之前,所产生的所有候选补丁数量(NCP)可以作为评估 GenProg-FL 所使用的缺陷定位技术的定位。在修复的过程中,NCP 值越低,GenProg-FL 的修复效果越好,更好的修复效果也就意味着被 GenProg-FL 使用的缺陷定位技术有更好的缺陷定位效果。

3.2 实验数据

在我们最近的实验中,以生成的候选补丁数量(NCP)作为评估标准,对 14 种基于程序谱的缺陷定位技术和 GenProg 原有缺陷定位策略从软件错误自动修复角度进行了评估。

在实验中,我们选择最近 GenProg 使用的主要程序作为实验的基准测试程序,所有程序都是用 C 语言编写,来自于不同领域的现实系统,缺陷都是现实存在的。但是完成所有程序的实验评估可能花费太多的时间,鉴于昂贵的实验消耗,我们从 GenProg 的基准测试程序中,随机选择了 6 个程序。我们使用统计分析的方法来衡量比较 14 种缺陷定位技术的有效性,在实验中,我们使用 GenProg-FL 在不同的缺陷定位技术指导下分别对每个程序进行 100 次的修复。同时,为了减轻运算量,我们对 GenProg-FL 的补丁验证过程进行了优化。在对候选补丁的合适度进行计算的过程中,只随机选择了 3 个正用例和 1 个反用例对补丁进行测试。如果补丁能通过,那么将会使用全部的测试用例对其进行测试,如果能全部通过,说明补丁有效,否则,补丁无效^[5]。

表 1 libtiff-bug-0860361d-1ba75257t 的 NCP 数据统计表

缺陷定位技术	均值	中间值	成功率
Jaccard	34	18	100%
Tarantula	39	21	100%
Kulczynski2	45	25	100%
Ample	48	22	97%
Ochiai	48	25	100%
Cohen	49	27	100%
M2	52	23	100%
Russel_rao	56	29	100%
Fleiss	60	27	100%
Ample2	65	24	100%
Optimal_p	72	27	99%
Rogers_tanimoto	74	27	100%
Arithmetic_mean	78	31	98%
Wong3	98	32	97%
Baseline	150	88	97%

以 libtiff-bug-0860361d-1ba75257t 为例,表 1 详细列举了

使用 GenProg-FL 在 14 种缺陷定位技术和原有缺陷定位策略的指导下,对故障程序进行 100 次修复而生成候选补丁的情况^[6]。从表 1 可以看出,不同的缺陷定位技术对软件错误自动修复的效率有不同的影响。

3.3 统计分析方法

因为 GenProg-FL 使用了随机算法(遗传编程),所以我们使用统计分析方法来分析比较不同缺陷定位技术的定位有效性。每个缺陷定位技术的定位有效性的评估标准(指 NCP 值)使用的是均值,而不是中间值,因为当结果数据由两个或是更多的离散值组成,平均值比中间值更有统计意义^[5]。

在评估过程中,可以根据 Mann-Whitney-Wilcoxon 测试^[6]和 A-test^[7]这两种统计分析方法分别定性、定量地对不同的缺陷定位技术进行评估^[5]。这两种统计方法都属于非参数统计测试方法。尽管参数统计测试在一些情况下更精确,但是这类测试需要额外的假设。而这些假设可能导致存在微小的偏差,所以我们使用非参数测试保证分析的有效性。给定两种缺陷定位技术 AFL_1 和 AFL_2 ,在 AFL_1 的平均值高于 AFL_2 的假设下,使用非参数统计测试:Mann-Whitney-Wilcoxon 测试和 A-test,可以分别定性和定量地分析 AFL_1 优于 AFL_2 的程度^[5]。也就是说,通过这两个统计分析方法去分析 AFL_1 是否比 AFL_2 的定位效果突出,如果突出,能达到什么程度。

对于非参数的 Mann-Whitney-Wilcoxon 测试,零假设是说来自 AFL_1 和 AFL_2 的数据同分布;备择假设是两类数据有不同的分布。在本实验中,如果在显著性水平为 0.05 条件下拒绝零假设,又因为 AFL_1 的均值高于 AFL_2 ,那么就认为 AFL_1 对 AFL_2 的优越性是显著的。

为了更深入地评估两种缺陷技术定位效果的差异,我们使用非参数的 Vargha-Delaney A-test。在这次实验中,如果通过 A-test 得到的 A-statistic 偏离 0.5 越远,那么 AFL_1 越优于 AFL_2 。Vargha 和 Delaney 建议 A-statistic 大于 0.64 或小于 0.36 表示优越性达到中等程度,如果大于 0.71 或是小于 0.29 则表示两者之间可能存在着很大的差别^[7]。

4 GenProg-FL 的意义

近来,缺陷定位技术已经得到了很大的发展。现有的研究表明,基于统计方法的故障定位技术中比较典型的就是基于程序谱的缺陷定位技术,几十年中,提出了一定数量的计算模型,例如 Ample、Jaccard、Tarantula 等。许多文献对这些缺陷定位技术的定位效果进行了一些评估。先前经验式的研究已经显示 Ochiai 总是比 Jaccard 和 Tarantula 表现得好^[8]。此外,最新的理论分析证明 Optimal_p (同样被称作 Naish^[9])、Russel_rao 是最佳的缺陷定位技术,也就是说这两个缺陷定位技术的效果根据 EXAM 测量标准^[10,11]要比其它的基于统计的技术突出。但是,从我们的实验数据中分析发现,这些在传统评估标准下表现良好的缺陷定位技术,在软件错误自动修复中,并没有表现得很出色。GenProg-FL 工具从另外一种完全不同的角度出发,开创性地以软件自动修复过程中生成的候选补丁数量(NCP)作为评估标准,对各个基于程序谱的缺陷定位技术进行评估。这对缺陷定位技术的发展具有十分重要的意义并创新性地提出了一种新的思路。在未

(下转第 124 页)

- [7] Tseng Chien-chao, Lin Chia-liang, Shih B-Y, et al. Topology-aware handoff scheme for surveillance patrol robot[J]. Non-linear Dynamics, 2013, 73(3): 2073 - 2081
- [8] Abo-Zahhad M, Ahmed S M, Mourad M. Future location prediction of mobile subscriber over mobile network using Intra Cell Movement pattern algorithm[C]// International Conference on Communications, Signal Processing, and their Applications (ICCSPA). 2013; 1-6
- [9] Jin Zhao-yan, Shi Dian-xi, Wu Quan-yuan, et al. Random Walk Based Location Prediction in Wireless Sensor Networks. [J] International Journal of Distributed Sensor Networks, 2013; 1-9
- [10] Samaan N, Karmouch A. A Mobility Prediction Architecture Based on Contextual Knowledge and Spatial Conceptual Maps [J]. IEEE Transactions on Mobile Computing, 2005, 4(6): 537-551
- [11] 孙巍巍, 苏寒松, 滕友伟, 等. 3GPP LTE 系统中结合位置预测的

切换算法[J]. 计算机应用, 2012, 32(7): 1849-1851

- [12] Yuan Yue-ming, Guan Wei, Qiu Wei. Map matching of mobile probes based on handover location technology[C]// 2010 International Conference on Networking Sensing and Control, IC-NSC, 2010; 587-592
- [13] Wang Yuan-lin, Yu Zhao-yuan, Luo Wen, et al. A 3D GIS spatial data model based on conformal geometric algebra[J]. Science China Earth Sciences, 2011, 54(1): 101-112
- [14] Shafer G. A Mathematical Theory of Evidence [M]. Princeton, NJ: Princeton Univ. Press, 1976; 124-130
- [15] Miao Y Z, Zhang H X, Zhang J W, et al. Improvement of the combination rules of the D-S evidence theory based on dealing with the evidence conflict[C]// 2008 IEEE International Conference on Information and Automation, Shanghai, 2008; 331-336
- [16] Peterfreund N. Robust tracking of position and velocity with Kalman snakes [J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000, 22(6): 564-569

(上接第 90 页)

来有更先进的缺陷定位技术提出时, 可以使用 GenProg-FL 工具从软件错误自动修复的角度对其进行评估, 验证其定位的有效性。

通过使用 GenProg-FL 工具对 Jaccard、Ample 等缺陷定位技术的评估发现, 在 GenProg-FL 中使用更先进的缺陷定位技术, 能明显提高软件错误自动修复的效率, 所产生的候选补丁数量得到很明显的减少, 而验证补丁有效性的过程一致, 所以修复所占用的时间便得到很明显的减少。另外, 通过对 14 种缺陷定位技术的评估, 我们认为在未来有更高效率的缺陷定位技术提出之前, 应该优先考虑使用 Jaccard 指导 GenProg-FL 来对软件故障进行修复^[5], Jaccard 可以有效地提高软件自动修复的效率。

GenProg-FL 工具将基于程序谱的缺陷定位技术与软件自动修复技术结合起来, 对两者的发展都具有十分重要的意义。

结束语 本文系统地对 GenProg-FL 工具进行了介绍。首先, 在对 GenProg 的介绍中, 重点分析研究了 GenProg 的缺陷定位策略, 发现其存在的不足。缺陷定位技术虽然已经得到很大的发展, 传统的评估方法也对现有的缺陷定位技术进行了评估, 但并没有促使其很好地运用于工业实践中。在此基础上, 本文提出了一种创新性的评估方法, 即从软件自动修复的角度出发, 使用 GenProg-FL 工具接受某种缺陷定位技术的指导, 去修复现实存在缺陷的故障程序。同时, 以产生的候选补丁数量(NCP)为评估标准, 对所使用的缺陷定位技术的定位有效性进行评估。本文对此评估方法的可行性进行了系统的阐述。同时, 指出了更有效的缺陷定位技术对软件自动修复的意义。在更有效的缺陷定位技术的指导下, GenProg-FL 在软件错误自动修复上的效率得到明显的提升, 这对软件错误自动修复技术的发展同样具有十分重要的意义。

参 考 文 献

- [1] IEEE Standard Classification for Software Anomalies[S]. IEEE

Std 1044-2009, 2010, (1044-1993); C1-15

- [2] Le Goues C, Dewey-Vogt M, Forrest S, et al. A systematic study of automated program repair[C]// fixing 55 out of 105 bugs for \$ 8 each; International Conference on Software Engineering (ICSE), 2012. 2012; 3-13
- [3] Le Goues C, Nguyen T, Forrest S, et al. Genprog: A generic method for automatic software repair[J]. IEEE Transactions on Software Engineering (TSE), 2012, 38(1): 54 - 72
- [4] Abreu P Z, van Gemund A J C. Spectrum-based Multiple Fault Localization[C]// Proceedings of the ASE'09. 2009
- [5] Qi Yu-hua, Mao Xiao-guang, Lei Yan, et al. Using Automated Program Repair for Evaluating the Effectiveness of Fault Localization Techniques[C]// Proceedings of the 2013 International Symposium on Software Testing and Analysis, 2013. 2013; 191-201
- [6] Wilcoxon F. Individual comparisons by ranking methods[J]. Biometrics Bulletin, 1945, 1(6): 80-83
- [7] Vargha A, Delaney H D. A critique and improvement of the CL common language effect size statistics of mcgraw and wong[J]. Journal of Educational and Behavioral Statistics, 2000, 25(2): 101-132
- [8] Abreu R, Zoetewij P, Golsteijn R, et al. A practical evaluation of spectrum-based fault localization[J]. Journal of Systems and Software (JSS), 2009, 82(11): 1780-1792
- [9] Xie X Y, Chen T Y, Kuo F-C, et al. A Theoretical Analysis of the Risk Evaluation Formulas for Spectrum-Based Fault Localization[J]. ACM Transactions on Software Engineering and Methodology (TOSEM), 2013, 22(4)
- [10] Yu Y, Jones J A, Harrold M J. An empirical study of the effects of test-suite reduction on fault localization[C]// International Conference on Software Engineering (ICSE), 2008. 2008; 201-210
- [11] Ali S, Andrews J H, Dhandapani T, et al. Evaluating the accuracy of fault localization techniques[C]// International Conference on Automated Software Engineering (ASE), 2009. 2009; 76-87