



江 蘇 大 學

JIANGSU UNIVERSITY

FSE 2016论文分析

课程名称: 高级软件工程

学院名称: 计算机科学与通信工程学院

专业班级: 研 1602

学 号: Z1608048

学生姓名: 陈 成

教师姓名: 陈锦富

2017 年 1 月

FSE 2016 共收录了 93 篇论文, 分成 20 类, 分别是 Specification、HCI and Process、Bug Detection and Debugging、Security and Privacy、Adaptation and Change、API Mining and Usage、Verification、Requirements and Models、Android、Static Analysis、Recommendation、Test Coverage、Program Analysis、Build and Configuration、Code Search and Similarity、Program Repair、Development Environments、Concurrency、Open Source、Test Generation。

一、 软件工程相关论文的综述

这部分对 22 篇与软件工程强相关的论文进行综述, 包括论文题目、所属国家、发表单位、作者和摘要。

1、Correctness Witnesses: Exchanging Verification Results between Verifiers

国家: Germany

单位: LMU Munich 、 University of Passau 、 University of Freiburg

主题: Verification

作者: Dirk Beyer, Matthias Dangl, Daniel Dietsch, Matthias Heizmann

摘要

标准验证工具提供了反例以证明违反规范, 并且, 由于几年, 这种证明可以由独立验证者使用可交换证明格式来验证。这样, 关于违规的信息可以跨验证工具共享, 用户可以使用标准工具可视化和探索证明。这种技术尚未建立用于正确性情况, 其中程序满足规范。即使

对于简单的程序，用户通常难以理解给定程序为什么是正确的，并且没有办法独立地检查验证结果。我们通过补充我们早期关于违规证明与正确证明的工作来弥补这一差距。虽然我们使用已确立的共同交换格式作为违反证明的代表来表示正确的证明，但是生产和验证正确证明的技术是不同的。向工程师提供证明的总体目标可能与编程本身一样古老，而且二十年前提出了验证代码 - 我们的目标是使其可行：我们认为证明是一流的可交换对象，独立于源代码存储并独立于生产它们的验证者进行检查，尊重分离关注点的重要原则。在任何时候，来自正确性证据的不变量可以用于重建正确性证据以建立信任。我们扩展了两个最先进的验证器 CPAchecker 和 UltimateAutomizer，以生成和验证证明，并报告该方法在大量的验证任务上是有希望的。

2、 Can Testedness be Effectively Measured?

国家： United States

单位： Oregon State University

主题： Test Coverage

作者： Iftekhar Ahmed 、 Rahul Gopinath 、 Caius Brindescu 、 Alex Groce 、 Carlos Jensen

摘要

实践测试人员面临的主要问题之一是决定在哪里集中额外的测试工作，以及决定何时停止测试。测试最少测试的代码，并且当所有代码都经过良好测试时停止，是一个合理的答案。已提出了许多“测

试”的措施;不幸的是, 我们不知道这些是否真正有效。

在本文中, 我们提出了一个新的评估两个最重要和广泛使用的测试套件质量的措施。第一个措施是语句覆盖, 最简单和最熟悉的代码覆盖率测量。第二个测量是突变分数, 据说是更强大, 但昂贵的测量。

我们使用感兴趣的实际标准来评估这些度量: 如果程序元素(通过这些度量)在给定时间点被良好测试, 则它比未被测试的元素需要更少的未来错误修复。如果不是, 那么似乎很可能我们没有有效地衡量测试。使用大量来自 Github 和 Apache 的开源 Java 程序, 我们显示语句覆盖率和突变分值与错误修复只有很弱的负相关。尽管缺乏强相关性, 但是在用于各种二元标准的程序元素之间存在统计上和实际上显著的差异。任何测试用例覆盖的程序元素(除类外)看到的错误修复大约是那些没有覆盖的错误修复, 并且可以绘制突变分数阈值的类似线。我们的结果对软件工程实践和研究评估有重要的影响。

3、 Anti-patterns in Search-Based Program Repair

国家: 新加坡、United States

单位: 新加坡国立大学、Fujitsu Laboratories of America

主题: Program Repair

作者: Shin Hwei Tan , Hiroaki Yoshida Mukul R. Prasad Abhik Roychoudhury

摘要:

基于搜索的程序修复会自动搜索给定修复空间中的程序修复。这

可以通过改进用于程序修复的通用搜索算法来实现，如 GenProg 工具所证明的，或者通过构建用于 SPR 中的程序修复的定制搜索算法。不幸的是，自动程序修复方法可以产生可能被程序员拒绝的补丁，因为过去的工作已经建议使用人工编写的补丁来产生指导程序修复的模板。在这项工作中，我们采取的立场，我们将不提供模板来指导修复搜索，因为这可能会不当地限制修复空间，并试图过度配合修复到提供的模板之一。相反，我们建议使用一组反模式 - 一组通用的禁用转换，可以在任何基于搜索的修复工具之上强制执行。我们表明，通过强制我们的反模式，我们获得修复，本地化正确的行或函数，涉及较少删除程序功能，并且大多是更有效地获得。由于我们的反模式集是通用的，我们已经将它们集成到现有的基于搜索的修复工具，包括 GenProg 和 SPR，从而允许我们以最小的努力获得更高质量的程序修补程序。

4、What Would Users Change in My App? Summarizing App

Reviews for Recommending Software Changes

国家：Italy 、Switzerland 、Japan

单位：University of Sannio, Department of Engineering、University of Zurich, Department of Informatics Sony Mobile Communications

主题：Recommendation

作者：Andrea Di Sorbo, Sebastiano Panichella, Carol V. Alexandru, Junji Shimagaki, Corrado A. Visaggio, Gerardo Canfora, Harald Gall

摘要

移动应用开发人员不断监控用户评价中的反馈，目的是改进他们的移动应用，更好地满足用户的期望。因此，在文献中已经提出了自动化方法，目的在于通过根据特定主题的自动分类/优先化来减少对包含在用户评论中的反馈所需的努力。在本文中，我们介绍 SURF（用户评论反馈的 Summarizer），一种新颖的方法，压缩大量的信息，开发人员的流行的应用程序必须管理，由于每天收到的用户反馈。

SURF 依赖于捕获对开发人员执行维护和演化任务有用的用户需求的概念模型。然后，它使用复杂的概括技术，总结成千上万的评论，并产生一个交互式，结构化和凝聚的推荐软件更改的议程。我们对 17 个移动应用（其中 5 个由 Sony Mobile 开发）的用户评价进行了 SURF 的端到端评估，共涉及 23 个开发人员和研究人员。结果表明 SURF 在总结评价和推荐的变化的有用性的高准确性。在评估我们的方法时，我们发现 SURF 帮助开发人员更好地了解用户需求，与手动分析用户（更改）请求和规划未来软件更改相比，大幅减少开发人员所需的时间。

5、iGen: Dynamic Interaction Inference for Configurable Software

国家：United States

单位：University of Maryland, College Park, USA

主题：Build and Configuration

作者：ThanhVu Nguyen Ugur Koc Javran Cheng Jeffrey S. Foster Adam

A. Porter

摘要

为了开发, 分析和发展当今高度可配置的软件系统, 开发人员需要深入了解系统的配置选项, 例如, 如何设置选项以到达特定位置, 用于测试的配置等。今天, 信息需要手动努力, 这是困难的, 昂贵的, 并且容易出错。在本文中, 我们提出 iGen, 一种小说, 轻量的动态分析技术, 自动发现程序的交互表达逻辑公式, 给开发人员丰富而详细的信息, 关于系统的配置选项设置如何映射到特定的代码覆盖。

iGen 采用迭代算法, 在一小组配置下运行系统, 捕获覆盖数据; 处理覆盖数据以推断潜在的交互; 然后生成新配置以进一步细化下一次迭代中的交互。我们评估 iGen 在 29 个程序跨越五种语言; 使用先前的交互推理工具将无法实现这项研究的广度。我们的研究表明, iGen 发现基于可能配置数量的非常小的一部分的精确的相互作用。此外, iGen 的结果证实了关于典型的相互作用分布和结构的几个早期假设。

6、 Lightweight Specification and Analysis of Dynamic Systems with Rich Configurations

国家: Portugal

单位: HASLab, INESC TEC and U. Minho

主题: Requirements and Models

作者: Nuno Macedo

摘要

模型检查在软件开发过程的早期阶段越来越流行。为了建立软件设计的正确性，通常必须验证结构和行为（或时间）属性。不幸的是，大多数规范语言和伴随的模型检查器仅仅在分析一种或另一种方面是优异的。这限制了它们验证具有丰富配置的动态系统的能力：其状态空间的特征在于具有丰富的结构特性，但是其演变也被期望满足某些时间特性的系统。

为了解决这个问题，我们首先提出 **Electrum**，利用时间逻辑运算符扩展合金规范语言，其中可以容易地定义丰富的配置和表达的时间属性。然后提出两种替代的模型检查技术，一种是有界的，另一种是无界的，以验证以该语言表达的系统，即验证每个可能的时间属性对于每个可能的配置是成立的。

7、Discovering Bug Patterns in JavaScript

国家：Canada

单位：University of British Columbia Vancouver, BC

主题：Bug Detection and Debugging

作者：Quinn Hanam

摘要

JavaScript 已经成为开发人员用于客户端和服务端编程的最流行的语言。然而，语言仍然缺乏对代码中潜在错误的警告形式的适当支持。目前使用的大多数错误查找工具涵盖通过阅读最佳实践或通过

开发人员的直觉和轶事观察发现的错误模式。因此，仍然不清楚哪些错误在实践中频繁发生，并且对于开发人员是固定的重要。我们提出一种新的半自动技术，称为 BugAID，用于发现最流行和可检测的 bug 模式。 BugAID 是基于无监督的机器学习，使用基于语言结构的更改从 AST 中区分错误修复。我们通过挖掘来自 134 个服务器端 JavaScript 项目的 105K 提交，提出了一个常见 bug 模式的大规模研究。我们发现了 219 个 bug 修改更改类型，并讨论了在多个项目中发生的 13 种普遍的 bug 模式，并且可能通过更好的工具支持来防止。我们的发现有助于改进工具和技术，以防止 JavaScript 中的常见错误，指导 IDE 的工具集成，并使开发人员了解与 JavaScript 编程有关的常见错误。

8、 Understanding and Detecting Wake Lock Misuses for Android

Applications

国家：中国

单位：Dept. of Comp. Science and Engineering, The Hong Kong Univ. of Science and Technology、State Key Lab for Novel Software Tech、 Dept. of Comp. Sci. and Tech., Nanjing University

主题：Android

作者：Yepang Liu, Chang Xu, Shing-Chi Cheung, Valerio Terragni

摘要

唤醒锁在 Android 应用程序中广泛使用，以保护关键计算免受设

备睡眠中断。不当使用唤醒锁常常严重影响用户体验。然而，很少知道如何使用唤醒锁在真实世界的 Android 应用程序和他们的滥用的影响。为弥合差距，我们对 44736 个商业和 31 个开源 Android 应用程序进行了大规模的实证研究。通过自动程序分析和手动调查，我们观察到（1）获取和释放唤醒锁的常见程序点，（2）经常由唤醒锁保护的 13 种类型的关键计算任务，以及（3）八种唤醒锁滥用通常导致功能和非功能问题，其中只有三个已经通过现有工作进行了研究。基于我们的发现，我们设计了静态分析技术 Elite，以检测两种最常见的唤醒锁滥用模式。我们对现实世界主题的实验表明，精英是有效的，可以超越两个最先进的技术。

9、 Detecting Table Clones and Smells in Spreadsheets

国家： 中国

单位：

（1） State Key Lab of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China ,

（2） Dept. of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China,

（3） State Key Lab for Novel Software Technology and Dept. of Computer Sci. and Tech., Nanjing University, Nanjing, China

主题： Development Environments

作者： Wensheng Dou, Shing-Chi Cheung, Chushu Gao, Chang Xu

Liang Xu, Jun Wei

摘要

电子表格被最终用户广泛用于各种业务任务，例如数据分析和财务报告。最终用户可以通过在其电子表格中克隆单元格（表）块来执行类似的任务。这些克隆表中的相应单元应该保持相同或类似的计算语义。然而，当电子表格演变时，由于克隆表可能由于临时修改而变得不一致，并且因此受到种种迹象妨碍。

在本文中，我们建议 TableCheck 检测表克隆和相关的迹象，由于它们之间的不一致。我们观察到在它们相应的单元格处具有相同标题信息的两个表可能是表克隆。受现有的基于指纹的代码克隆检测技术的启发，我们开发了一种检测算法来检测这种表克隆。我们进一步在相应的细胞中检测异常值作为检测的表克隆中的迹象。我们将我们的想法实现到 TableCheck 中，并将其应用于来自 EUSES 语料库的现实电子表格。实验结果表明，表克隆通常存在（21.8%），并且具有表克隆的电极板的 25.6% 由于这些克隆之间的不一致性而受到种种迹象妨碍。 TableCheck 检测到的表克隆及其迹象分别具有 92.2% 和 85.5% 的精确度，而现有技术检测到 TableCheck 可检测到的真实迹象不超过 35.6%。

10、Python Predictive Analysis for Bug Detection

国家：china、United States

单位：

(1) State Key Laboratory of Novel Software Technology, Nanjing
University, China

(2) Department of Computer Science, Purdue University, USA

主题: Bug Detection and Debugging

作者: Zhaogui Xu, Peng Liu, Xiangyu Zhang, and Baowen Xu

摘要:

Python 是一种流行的动态语言, 允许快速软件开发。然而, Python 程序分析引擎在很大程度上缺乏。在本文中, 我们提出一个 Python 预测分析。它首先收集执行的跟踪, 然后将跟踪和未执行的分支编码到符号约束。引入符号变量来表示输入值, 它们的动态类型和属性集, 以推断它们的变化。解决约束识别错误及其触发输入。我们的评估表明, 该技术在分析具有大量动态特性和外部库调用的现实世界复杂程序中非常有效, 这是由于其基于跟踪的复杂编码设计。它从 11 个现实世界的项目中识别 46 个 bug, 有 16 个新 bug。所有报告的错误是确切的。

11、An Extensive Study of Static Regression Test Selection in Modern Software Evolution

国家: United States

单位:

(1) Department of Computer Science, University of Illinois at
Urbana-Champaign

(2) Department of Computer Science, The University of Texas at Dallas

主题: Test Coverage

作者: Owolabi Legunsen, Farah Hariri, August Shi, Yafeng Lu,

Lingming Zhang, and Darko Marinov

摘要

回归测试选择 (RTS) 旨在通过仅重新运行受代码更改影响的测试来减少回归测试时间。之前对 RTS 的研究可以大致分为动态和静态技术。最近开发的称为 Ekstazi 的动态 RTS 技术在实践中获得了一些采用, 它的评估表明, 以更粗略的类级别粒度选择测试比以更精细的方法级粒度选择测试提供更好的结果。随着动态 RTS 正在获得采用, 它是及时评估静态 RTS 技术, 其中一些提出了三十多年前, 但没有广泛评估现代软件项目。

本文介绍了第一次广泛的研究, 评估静态 RTS 技术的性能优势及其安全性; 如果选择运行可能受代码更改影响的所有测试, 则该技术是安全的。我们实现了两个静态 RTS 技术, 一个类级和一个方法级, 并比较这些技术的几个变体。我们还将这些静态 RTS 技术与 Ekstazi (一种最先进的, 类级, 动态 RTS 技术) 进行比较。22 个开源项目的 985 次修订的实验结果表明, 类级静态 RTS 技术与 Ekstazi 可比, 具有类似的性能优势, 但有时有不安全的风险。相比之下, 方法级静态 RTS 技术执行相当差。

12、A Deployable Sampling Strategy for Data Race Detection

国家：China

单位：

(1) State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences, Beijing, China

(2) Institute of Information Engineering, Chinese Academy of Sciences,
Beijing, China

主题：Concurrency

作者：Yan Cai, Jian Zhang, Lingwei Cao, and Jian Liu

摘要

动态数据竞争检测导致大量的运行时开销。最近，已经提出了许多采样技术来检测数据竞争。然而，一些采样技术（例如，Pacer）基于传统的发生先前关系并且招致大的基本开销。其他人使用硬件来减少它们的采样开销（例如，DataCollider），然而，它们仅当比赛真的通过延迟程序执行而发生才检测比赛。在本文中，我们研究现有技术的局限性，并提出一种新的数据竞争定义，称为时钟竞争，用于低开销抽样目的。时钟赛的创新在于它们的检测不依赖于具体的锁，并且还避免了跟踪发生在之前的关系的重的基本开销。我们进一步建议 CRSampler（时钟竞争采样器）通过基于硬件的采样检测时钟竞争，而不直接延迟程序执行，以进一步减少运行时开销。我们在 Dacapo 基准测试 CRSampler。结果表明，CRSampler 平均以 1% 的采样率产生了 5% 的开销。而 Pacer 和 DataCollider 的开销分别大于 25% 和 96%。

此外，在相同的采样率，CRSampler 检测到比 Pacer 和 DataCollider 显着更多的数据竞争。

13、Gray Links in the Use of Requirements Traceability

国家：United States

单位：Department of EECS University of Cincinnati Cincinnati, OH,
USA 45221

主题：Requirements and Models

作者：Nan Niu 、 Wentao Wang 、 Arushi Gupta

摘要

可追溯性的价值在于它的使用。不同的软件工程任务如何影响同一需求的跟踪？在本文中，我们通过一个实证研究来回答这个问题，其中我们明确地将参与者分配到一个需求的 3 个跟踪组中：找到它的实现用于验证和验证目的，在原始软件系统内改变它，并且重用到另一个应用。结果揭示了我们所谓的“灰色链接” - 大约 20% 的总痕迹被投票为只有一个任务，而不是其他任务的真正链接。我们提供一种机制来识别这样的灰色链接，并讨论如何利用它们来推进基于价值的需求追踪的研究和实践。

14、Detecting and Fixing Precision-Specific Operations for Measuring Floating-Point Errors

国家：China

单位: Key Laboratory of High Confidence Software Technologies
(Peking University), MoE Institute of Software, EECS, Peking University,
Beijing, 100871, China

主题: Program Analysis

作者: Ran Wang, Daming Zou, Xinrui He, Yingfei Xiong, Lu Zhang,
Gang Huang

摘要

浮点计算的精度对于许多应用是关键,并且围绕浮点精度提出了不同的方法,诸如检测程序中的错误,验证程序的准确性,以及优化程序以产生更准确的结果。这些方法需要程序的规范以理解由程序执行的理想计算,其通常通过以精确非特定的方式解释程序来进行。然而,在现有代码中编程的许多操作是固有地精确特定的,其不能以精确非特定的方式容易地解释。事实上,在现有方法中使用的语义通常不能正确地解释精确特定的操作。

在本文中,我们提出了一个系统的研究精度特定操作。首先,我们提出一种检测方法来检测精度特异性操作。第二,我们提出一种固定方法,以在精确特定操作的存在下调整精度。第三,我们基于我们的检测和固定方法研究了 GNU C 标准数学库中的精确特定操作。我们的结果表明,(1)在标准 C 数学库中有大量的代码片段是精确特异性操作,并且在现有研究中报道的一些大的不准确性是由于精确特异性操作而导致的假阳性或潜在假阳性;(2)我们的检测方法具有高精度和回忆;(3)我们的固定方法可以导致整体更准确的结果。

15、Hunter: Next-Generation Code Reuse for Java

国家: United States

单位: Arati Kaushik UT Austin, USA 、 Isil Dillig UT Austin, USA 、
Steven P. Reiss Brown University, USA

主题: Code Search and Similarity

作者: Yuepeng Wang 、 Yu Feng 、 Ruben Martins

摘要

在许多常见的情况下,程序员需要实现一些第三方库已经提供的功能。本文提出了一个称为 Hunter 的工具,它通过在大型代码库中查找相关方法并自动合成任何必要的包装代码来促进代码重用。由于 Hunter 内部使用高级程序合成技术,即使在需要代码适配时,它也可以自动重用现有的方法。我们已经实现了 Hunter 作为 Eclipse 插件,并通过(a)将其与 S6,一个最先进的代码重用工具进行比较,以及(b)执行用户研究来评估它。我们的评估表明, Hunter 与 S6 相比有利,并提高了程序员的生产力。

16、Parameter-Free Probabilistic API Mining across GitHub

国家: UK

单位: School of Informatics University of Edinburgh, Edinburgh, EH8
9AB, UK

主题: API Mining and Usage

作者: Jaroslav Fowkes 、 Charles Sutton

摘要

现有的 API 挖掘算法可能难以使用,因为它们需要昂贵的参数调整,并且返回的一组 API 调用可能是大的,高度冗余的并且难以理解。为了解决这个问题,我们提出 PAM (Probabilistic API Miner), 一个近无参数的概率算法挖掘最有趣的 API 调用模式。我们显示 PAM 在从 GitHub 检索相关 API 调用序列时,显著优于 MAPO 和 UPMiner, 实现 69% 的测试集精度。此外,我们专注于开发人员已经明确提供代码示例的库,从数据集中的 967 个客户端项目中产生了超过 300,000 个手写 API 示例代码。这个评估表明,手写的示例实际上具有有限的实际 API 用法的覆盖。

17、Paradise Unplugged: Identifying Barriers for Female Participation on Stack Overflow

国家: USA

单位: NC State University Raleigh, NC, USA、UC San Diego La Jolla, CA, USA

主题: Open Source

作者: Denae Ford, Justin Smith , Philip J. Guo , Chris Parnin

摘要

这不是秘密,女性在编程领域比男性少。然而,在在线社区,如 Stack Overflow,这种性别差距更加极端:只有 5.8% 的贡献者是女性。在本文中,我们使用混合方法方法来确定女性在在线社区面临的贡献

障碍。通过对一系列女性用户的半结构化访谈，我们发现了 14 个障碍，阻止他们对 Stack Overflow 做出贡献，这些女性用户从无贡献者到排名前 100 的用户。然后，我们与 1470 名女性和男性开发商进行了一项调查，以确认哪些障碍是性别相关的或每个人的一般问题。女性排名五级障碍明显高于男性。其中一些包括对贡献的专业水平的怀疑，当与大量用户竞争时感到不堪重负，以及对站点特征的意识有限。但是，还有其他障碍同样影响所有 Stack Overflow 用户或受影响的特定组，如行业程序员。最后，我们描述了几个可能鼓励增加参与 Stack Overflow 社区跨性别和其他人口统计学的影响。

18、NonDex: A Tool for Detecting and Debugging Wrong

Assumptions on Java API Specifications

国家: USA

单位: Department of Computer Science, University of Illinois at
Urbana-Champaign Urbana, IL 61801, USA

主题: Specification

作者: Alex Gyori, Ben Lambeth, August Shi, Owolabi Legunsen, and
Darko Marinov

摘要

我们提供 NonDex，一个用于检测和调试 Java API 错误假设的工具。一些 API 具有欠定的规范以允许实现实现不同的目标，例如，以优化性能。当这些 API 的客户端假定强于指定的保证时，生成的客户

端代码可能会失败。例如，HashSet 的迭代顺序是不确定的，并且假定某些特定于实现的迭代顺序的代码可能失败。NonDex 有助于主动检测和调试这种错误的假设。NonDex 通过在测试执行期间随机探索未确定的 API 的不同行为来执行检测。当测试在探索期间失败时，NonDex 搜索导致失败的 API 的调用实例。NonDex 是开源的，与 Maven 完美集成，也从命令行运行。在使用 NonDex Maven 插件的实验中，我们从 GitHub 的 8 个 Java 项目中检测到 21 个新的错误，并且使用 NonDex 的调试功能，我们确定了这 21 个新错误和 54 个先前检测到的错误的基本错误假设。我们打开了 13 个请求;开发人员已经接受了 12，并且一个项目改变了连续集成配置，在每次推送时运行 NonDex。演示视频在: <https://youtu.be/h3a9ONkC59c>。

19、ARdoc: App Reviews Development Oriented Classifier (Demo)

国家: Zurich

单位: University of Zurich

主题: Android

作者: Panichella, Sebastiano; Di Sorbo, Andrea; Guzman Ortega, Emitza; Visaggio, Corrado Aaron; Canfora, Gerardo; Gall, Harald

摘要

Google Play, Apple App Store 和 Windows Phone Store 是众所周知的发布平台，用户可以下载移动应用程序，对其进行评分，并撰写有关他们正在使用的应用程序的评论评论。以前的研究表明，这些评

论包含重要信息，以帮助开发人员改进他们的应用程序。然而，分析评论是具有挑战性的，因为每天发布的大量评论，评论的非结构化性质及其不同的质量。在这个演示中，我们提出了 ARdoc，一个工具，结合三种技术：（1）自然语言分析（NLP），（2）文本分析（TA）和（3）情绪分析对于执行软件维护和演化任务非常重要。我们的定量和定性分析（涉及移动专业开发人员）表明，ARdoc 正确地将反馈用于高精度（范围在 84%和 89%之间），回忆（范围在 84%和 89%之间）的用户评论中的维护观点，FMeasure(范围在 84%和 89%之间)。在评估我们的工具时，我们还发现 ARdoc 基本上有助于提取现实世界应用程序的重要维护任务。演示网址：<https://youtu.be/Baf18V6sN8E>
演示网页：<http://www.i.uzh.ch/seal/people/panichella/tools/ARdoc.html>

20、PUMConf: A Tool to Configure Product Specific Use Case and Domain Models in a Product Line

国家：Luxembourg

单位：

（1）SnT Centre for Security, Reliability and Trust University of Luxembourg, Luxembourg

（2）International Electronics & Engineering (IEE) Contern, Luxembourg

主题：Adaptation and Change

作者：Ines Hajri, Arda Goknil, Lionel C. Briand , Thierry Stephany

摘要

我们提供 PUMConf，一种支持配置的工具，目前关注需求，并在用例驱动开发的上下文中实现有效的产品线管理。通过设计，其仅依赖于在这样的上下文中常用的工件的变异性建模（即，用例图，规范和域模型）。对于给定的产品线（PL）用例和模型，PUMConf 检查模型的一致性，交互式接收来自分析者的配置决策，自动检查决策一致性，并生成产品特定（PS）用例和域模型 PL 模型和决策。它已经在汽车领域的一个工业案例研究进行了评估。

21、Time-Travel Debugging for JavaScript/Node.js

国家：UK 、 USA

单位：University College London ， Microsoft Research

主题：Bug Detection and Debugging

作者：Earl T. Barr 、Mark Marron 、Ed Maurer 、Dan Moseley 、Gaurav Seth

摘要

在调试期间在程序的执行历史中的时间旅行使得开发者能够精确地跟踪和理解导致错误的语句和程序值的序列。为了向现实世界的开发人员提供此功能，我们开始了两年的旅程，在微软的开源 ChakraCore JavaScript 引擎和流行的 Node.js 应用程序框架中创建一个生产质量的时间旅行调试器。

22、Proteus: Computing Disjunctive Loop Summary via Path

Dependency Analysis

国家: China 、 Singapore

单位: Tianjin Key Laboratory of Advanced Networking, Tianjin

University, China 、 Nanyang Technological University, Singapore

主题: Specification

作者: Xiaofei Xie、Bihuan Chen、Yang Liu、Wei Le、Xiaohong Li

摘要

循环是用于程序分析的具有挑战性的结构,特别是当循环包含在这些路径中具有复杂交错执行的多个路径时。在本文中,我们首先提出一个多径循环分类,以了解循环执行的复杂性,这是基于循环条件的变量更新和循环路径的执行顺序。其次,我们提出一个名为 Proteus 的循环分析框架,它采用一个循环程序和一组感兴趣的变量作为输入,并总结对变量的路径敏感的循环效应。关键贡献在于使用路径相关自动机(PDA)来捕获路径之间的执行依赖性。提出了一种基于 DFS 的算法来遍历 PDA,以总结循环中所有可行执行的效果。实验结果表明 Proteus 在三个应用中是有效的: Proteus 可以 1) 计算比现有的循环绑定分析技术更精确的绑定; 2) 显著优于用于环路验证的最先进的工具;和 3) 在一秒内生成深循环的测试用例,而 KLEE 和 Pex 需要更多的时间或失败。

二、 论文分析

这部分分析、比较 93 篇论文的所属国家、主题、和作者是否是华人这三个问题。

根据统计结果显示，美国在软件领域的研究依然处于领先地位，但是中国在很多领域已经有了很大的突破，比如 Bug Detection and Debugging、Security and Privacy、Android 等。