

基于语义影响范围分析的补丁兼容性分析

(申请清华大学工程硕士学位论文)

培 养 单 位: 软 件 学 院

学 科: 软 件 工 程

研 究 生: 潘 晓 梦

指 导 教 师: 贺 飞 副 教 授

二〇一五年四月

Thesis Submitted to
Tsinghua University
in partial fulfillment of the requirement
for the professional degree of
Master of Software Engineering

by
Pan Xiaomeng
(Software Engineering)

Thesis Supervisor : Professor He Fei

April, 2015

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____

导师签名：_____

日 期：_____

日 期：_____

摘 要

我还不知道写什么，就先这样吧。

关键词：

Abstract

unknown right now

Key words: T_EX; L^AT_EX; CJK; template; thesis

目 录

第 1 章 绪论	1
1.1 研究背景和意义	1
1.1.1 本文所要解决的问题与主要工作	1
1.2 本文组织结构	2
第 2 章 相关工作	3
2.1 软件演进	3
2.2 程序间差异分析	3
2.3 程序变更影响分析	3
2.4 相关工具	3
第 3 章 分析方法	4
3.1 问题定义	4
3.1.1 补丁版本迁移	4
3.1.2 变更的语义影响分析	4
3.1.3 兼容性分析	4
3.2 应用场景	4
3.3 解决方案	4
3.4 流程	4
3.4.1 补丁版本迁移	4
3.4.2 语义影响范围分析	4
3.4.2.1 程序差异性分析	4
3.4.2.2 变更影响分析	4
3.4.3 实际流程	4
第 4 章 组合方法	5
第 5 章 兼容性分析算法	6
第 6 章 实验	7
6.1 补丁版本迁移	7
6.2 语义影响范围分析	7
6.2.1 程序间差异性分析	7
6.2.2 变更影响分析	7

6.3 兼容性分析.....	7
6.4 结论.....	7
第 7 章 结论	8
7.1 工作总结	8
7.2 未来工作	8
参考文献	9
致 谢	10
声 明	11

第 1 章 绪论

1.1 研究背景和意义

1.1.1 本文所要解决的问题与主要工作

软件维护（Software maintenance）是软件开发周期中耗时最长、开销最大的过程。随着外部环境和用户需求的不断变化，软件系统需要随之进行适应和调整，同时也需要修复在实际运行中暴露出来的问题。

补丁（Patch）就是这样一类可以用于完成修补程序漏洞、增强软件功能、改善程序性能等任务的程序。补丁在工业界中得到了广泛的实际应用，是软件维护过程的重要组成部分。

然而，补丁程序仍然具有一定的局限性，它一般只针对某个专门软件版本而开发，对于软件演进过程中获得的新版本而言，我们无法确定补丁程序是否也能适用于新版本的程序。然而补丁程序的应用一般都具有特定的目的，例如功能升级、漏洞修补等，新版本的程序中很可能仍然需要补丁程序的应用来完善自身。

为此，我们考虑这样一个问题，给定一个程序 p ，他当前具有两个版本 p_1 和 p_2 ，其中 p_2 是从 p_1 演进而来的较新版本，该程序 p 目前有一个适用于版本 p_1 的补丁 s_1 ，我们想知道该补丁 s_1 是否能够适用于程序 p_2 ？

对该问题而言，其答案可以分为多个层面来回答：

1. 语法：从语法角度出发，则该问题主要关注的是在应用过程中是否会造成语法结构上的错误，如果能够将补丁 s_1 成功应用于程序 p_2 ，则认为该补丁是可以兼容于新版本 p_2 的。
2. 语义：从语义角度出发，则单纯的语法兼容并不能够完全解答这个问题。某行修改可能会影响多处源代码，从而导致程序的行为发生变化。因而从语义层面而言，我们需要保证补丁 s_1 对程序 p_2 造成的语义影响不会波及到从版本 p_1 演进到 p_2 时所造成的语义变化。

语法层面的回答很容易就能给出，现有的版本控制系统等都能在一定程度上给出相应的答案。而语义层面的答案，就目前所知尚未有这方面的工作。

因而本文将着重从语义层面尝试去解决这个问题。该问题可以较形式化地描述如下：

定义 1.1： 具体定义

目前而言，本文的主要工作包括以下几个部分：

1. 补丁版本迁移

补丁 s_1 本来是适用于程序版本 p_1 的，如果想要适用于程序 p_2 ，可能需要进行一定的版本迁移工作。

该部分工作可以从语法层面给出兼容性的答案，并进行补丁兼容性的语法结构修正。

2. 语义影响范围分析

本文主要采用程序间差异分析、变更影响分析等手段来对界定补丁 s_1 对程序 p_2 所造成的语义影响范围。

所谓的语义影响范围可以具体定义如下，其中 $structure$ 意为程序语法结构，可以采用不同级别的程序语法结构（如 $statement$ 、 $basic\ block$ 等）进行分析，来获得不同粒度的影响范围：

(a) $change(s) = structure_i \mid structure_i$ 属于 p ，并且 $structure_i$ 发生了变更，该集合可以采用程序间差异分析获得。

(b) $impact(s) = structure_i \mid structure_i$ 属于 p ，并且 $structure_i$ 受到 $change(s)$ 的影响，该集合可以采用变更影响分析获得。

最后得到的 $impact(s)$ 即为我们所需的变更的语义影响范围。

3. 补丁兼容性分析

在有了语义影响集合 $impact(s)$ 之后，可以进行具体的兼容性分析工作。

1.2 本文组织结构

本文主要包括七个章节，第一章是绪论，介绍本文的研究背景和主要工作；第二章主要介绍与本文所述内容相关的国内外的的工作；第三章主要介绍了补丁兼容性分析的具体方法；第四章主要介绍了如何将各阶段的不同分析方法进行整合，形成具体的流程；第五章详细介绍了具体的补丁兼容性分析算法；第六章介绍了实验过程和结果；第七章主要对本文的工作进行了总结，并提出了进一步的工作方向。

第 2 章 相关工作

2.1 软件演进

2.2 程序间差异分析

2.3 程序变更影响分析

2.4 相关工具

主要介绍本文中采用到的相关工具：

- git

git 是一个分布式的版本控制系统，最初由 Linus Torvalds 在 2005 年为 Linux 内核而开发，现在已经成为最流行的版本控制系统。

与 CVS 和 SVN 等集中式的 C/S 版本控制系统不同，git 是分布式的版本库，每个本地的 git 工作目录都包含了完整的历史数据和版本追踪能力，无需网络连接或服务端。

本文中主要考虑以 git 作为版本管理系统的应用场景，类似于 GitHub，假定为项目开发了 new version 和 patch version 两个不同的分支，并使用 git 的分支合并策略实现补丁的版本迁移过程。

- beyond compare

Beyond Compare 是一款内容比较工具，可以用于文件、目录、压缩包的比较，横跨 Windows、Mac OS X、Linux 三大操作系统，可用作版本控制系统的文本比较和合并工具，例如 git。

本文中主要采用其作为 git 的文本比较和合并工具，用于解决补丁版本迁移时的冲突（conflict）问题。

- AST Differ

- jpf-regression

第 3 章 分析方法

3.1 问题定义

该问题可以拆分为若干个子问题，并分别阐述并定义之：

3.1.1 补丁版本迁移

3.1.2 变更的语义影响分析

3.1.3 兼容性分析

3.2 应用场景

3.3 解决方案

介绍补丁兼容性分析方法的整体架构：补丁版本迁移、语义影响范围分析（程序差异性分析、程序变更影响分析）、补丁兼容性分析。

3.4 流程

介绍补丁兼容性分析的整体流程：【可给出一张抽象的流程图】。

3.4.1 补丁版本迁移

3.4.2 语义影响范围分析

3.4.2.1 程序差异性分析

3.4.2.2 变更影响分析

3.4.3 实际流程

1. 采用 git 进行版本管理。
2. 将新版本和补丁版本两个分支进行合并，得到应用于新版本的补丁版本。
3. 采用 AST Differ 生成程序间差异性文件，XML 格式。
4. 采用 jpf-regression 进行变更影响分析，获取三个版本间两两的影响集合，dot 格式
5. 对两个影响集合求交集，若无交集，则兼容之，若有交集，进一步采用分析算法，看是否确实不兼容

【此时可给出一张具体的流程图】

第 4 章 组合方法

介绍如何将版本管理、程序间差异分析、变更影响分析、兼容性分析算法组合成整体的流程。

第 5 章 兼容性分析算法

介绍具体的兼容性算法，如果两个影响集合存在交集，是否确实是不兼容的。

第 6 章 实验

以 Eclipse jdt core 为例，进行实验，并分析得到的实验结果。

6.1 补丁版本迁移

6.2 语义影响范围分析

6.2.1 程序间差异性分析

6.2.2 变更影响分析

6.3 兼容性分析

6.4 结论

第 7 章 结论

7.1 工作总结

7.2 未来工作

参考文献

致 谢

感谢贺飞老师对我的悉心指导，他的言传身教令我受益匪浅，并在我完成毕设工作时给出了许多有用的意见。

最后要感谢 ThuThesis，帮助我完成了本文的写作。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：_____ 日 期：_____