

北京大学信息科学技术学院

本科生毕业论文

变异测试的动态加速技术：设计与实现

1200012741

史杨勍惟

指导老师：熊英飞

(2016 年 4 月 25 日)

摘要

变异测试是一种在通过细节改变源代码的软件测试方法，用来帮助测试者评估测试集的质量。变异测试一个很大的瓶颈在于其可扩展性。研究人员已经提出了各种不同的变异测试的加速技术，例如移除冗余的变异体等。然而，这些技术都是静态的，所以无法消除在变异体执行过程中的冗余部分。

本论文的目标是设计一个变异测试的动态加速技术：在变异测试的执行过程中对变异体进行分析，仅在变异体产生新的系统状态的时刻创建新进程来执行变异体。基于此技术，本论文在 LLVM 的框架上实现了一个 C 语言变异测试的加速工具 AccMut，并将此工具与现有的加速技术进行了对比和验证。实验表明动态加速技术加速效果显著，加速比是 Major Framework（目前最快的静态变异测试加速技术）的 X 倍。

关键词： 变异测试，动态加速，静态加速

Abstract

Mutation analysis is used to help evaluating the quality of existing software tests by modifying a program in small ways. One important bottleneck of mutation analysis is its scalability. Researchers have proposed different techniques to accelerate the mutation analysis, such as removing redundant computations in mutation analysis. However, all these techniques are static, and thus cannot remove redundancy that occurs in part of mutant execution.

The purpose of this thesis is to design a technique to accelerate the mutation analysis dynamically, which analyzes the mutants during the execution of the program and forks the execution only when a mutant leads to a new system state. Based on this technique, we developed an acceleration tool "AccMut" on C programming language on top of LLVM and compared it with other techniques. Our experiment shows that our approach can accelerate mutation analysis significantly, having a speedup up to x.xxX over Major Framework, a state-of-the-art tool of static acceleration.

Keyword: mutation analysis, dynamic acceleration, static acceleration

目录

| | |
|-----------------------------------|----------|
| 一 引言 | 5 |
| 1.1 变异测试 | 5 |
| 1.2 变异测试的瓶颈 | 5 |
| 1.3 相关工作 | 5 |
| 1.3.1 Mutation Schemata | 5 |
| 1.3.2 Weak Mutation | 5 |
| 1.3.3 Major Framework | 5 |
| 1.4 本论文的目的 | 5 |
| 二 变异测试的动态加速技术 | 6 |
| 2.1 加速原理 | 6 |
| 2.1.1 相关背景: 系统调用 Fork | 6 |
| 2.2 抽象模型 | 6 |
| 2.3 算法 | 6 |
| 2.3.1 静态算法 | 6 |
| 2.3.2 动态算法 | 6 |
| 三 工具实现 | 7 |
| 3.1 生成变异的 Pass | 7 |
| 3.2 插桩的 Pass | 7 |
| 3.3 动态分析算法的库 | 7 |
| 3.4 文件 IO 的支持 | 7 |

| | |
|-----------------------|-----------|
| 四 实验测试 | 8 |
| 4.1 实验对象 | 8 |
| 4.2 实验流程 | 8 |
| 4.3 实验结果 | 8 |
| 五 未来扩展：软件产品线测试 | 9 |
| 六 结论 | 10 |
| Reference | 11 |
| 致谢 | 12 |

一 引言

1.1 变异测试

变异测试是什么？通常来说，变异测试包括，高阶变异测试。变异测试有许多应用场景。

1.2 变异测试的瓶颈

变异测试有一个重要的瓶颈，特别慢。更长远地说，影响可扩展性。

1.3 相关工作

已有一些加速技术

1.3.1 Mutation Schemata

1.3.2 Weak Mutation

1.3.3 Major Framework

图 X 显示了 XXX 现有的加速技术有一个共同的问题，都是基于静态分析的。

1.4 本论文的目的

本论文提出了动态变异测试的相关技术。纯动态的方式对程序进行分析。本论文对进行了复现和横向对比。在本科生科研阶段，我也尝试过，不同点。

二 变异测试的动态加速技术

2.1 加速原理

不同的变异测试总的来说，通过动态的方法，加速了执行，X 的阶段。图 X 是本算法的一个示例。

2.1.1 相关背景：系统调用 Fork

XXX

2.2 抽象模型

此处将给出动态变异测试的核心算法，并将与静态变异测试的算法进行比较。

2.3 算法

2.3.1 静态算法

2.3.2 动态算法

核心算法

分类算法

三 工具实现

选择 C 语言，LLVM 上进行了实现。新增两个 Pass，插桩，库，执行，图
X

3.1 生成变异的 Pass

3.2 插桩的 Pass

3.3 动态分析算法的库

3.4 文件 IO 的支持

四 实验测试

4.1 实验对象

4.2 实验流程

4.3 实验结果

五 未来扩展：软件产品线测试

抄写熊老师

六 结论

重写一下 Intro

参考文献

- [1] Florence Lowe-Lee, “Is Korea Ready for the Demographic Revolution?” The KEI Exchange, 2009

致谢