中国科学院软件研究所学术年会'2018
暨计算机科学国家重点实验室开放周

学术论文

FSE 2017: The 12Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering

原子性缺陷的高质量自动化修复

Adaptively Generating High Quality Fixes for Atomicity Violations

蔡 彦 (yancai@ios.ac.cn)、曹玲微 等 • 中国科学院软件研究所，计算机科学国家重点实验室

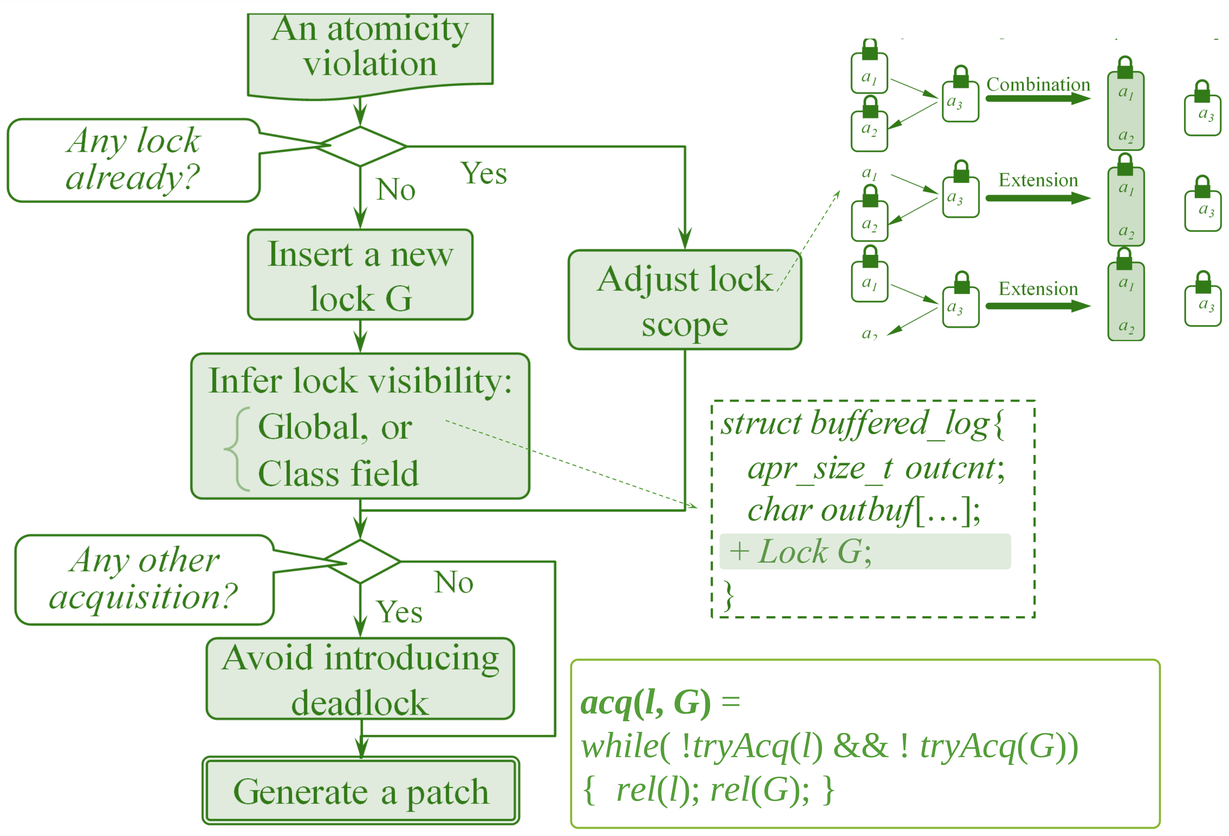
软件Bug是普遍存在的，其修复过程需要消耗大量人力和时间。现有的研究者提出了各种并发缺陷的自动修复方案，这些方案都是通过添加新锁来线性化涉及线程的执行。因此，在真实大规模程序中，这些方案要么修复失败、要么引入新的缺陷、要么引入性能缺陷。

我们提出了一种针对原子性缺陷的高质量自动修复方案*AlphaFixer*。*AlphaFixer*根据缺陷代码结构，选择调整现有保护锁或者添加相同层级的保护锁进行缺陷修复，同时结合Pre-Acquisition操作进行死锁避免。在一些真实大规模程序上的实验表明，该方案不仅能正确修复相应缺陷，避免引入较大的性能问题，同时能够提供更具可读性的修复代码。

Three Questions

- Is it necessary to introduce a gate lock for all cases?
- Can the context-aware locks be "computed" early?
- Any way to guarantee a deadlock-free fix?

AlphaFixer



re-use

Existing Locks

In many cases, locks are already used to protect variables; but the usages are incorrect. They can be re-used to fix the bug.

Otherwise, a gate lock is necessary. The gate lock can be either a Global one or a field of a class or struct, without any online synthesis, to make the fix codelines more nature to developers.

GLOBAL

STRUCT OR CLASS

BUG FREE **Guarantee**

Inserting a gate lock, the accompanied lock orders must be eliminated to avoid Introducing deadlocks.

Experiment

Summary

100% correct fixes

Better performance

# Total	#of fixed atom. violations				Avg. overhead			
	GLA	Grail	HFix	αFixer	GLA	Grail	HFix	αFixer
15	10 (67%)	10 (67%)	2 (13%)	15 (100%)	120.2%	82.9%	32.5%	21.1%

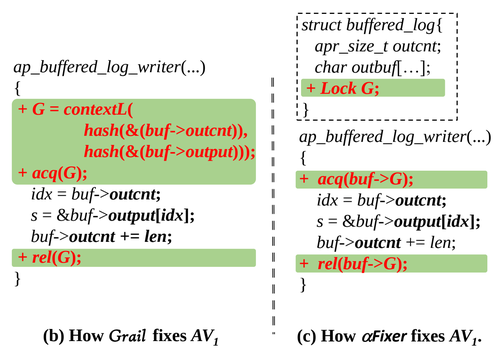
Applicable to limited number of atomicity violations (with locks)

High overhead

User Study

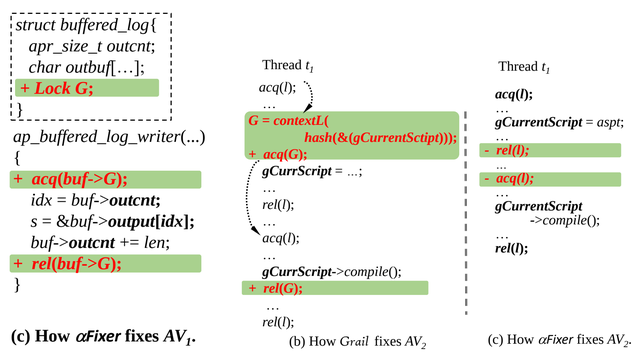
		Tool1 (Grail)	Tool2 (αFixer)	Any
AV_1	Understandability	2.5%	87.5%	10.0%
	Readability	0.0%	90.0%	10.0%
	Larger Overhead	67.5%	12.5%	20.0%
	Preferred Fix	5.0%	95.0%	0.0%
AV_2	Understandability	7.5%	92.5%	0.0%
	Readability	2.5%	85.0%	12.5%
	Larger Overhead	75.0%	10.0%	15.0%
	Preferred Fix	10.0%	82.5%	7.5%

Case Study 1



(b) How *Grail* fixes AV_1

Case Study 2



(c) How *αFixer* fixes AV_1.

(b) How *Grail* fixes AV_2

(c) How *αFixer* fixes AV_2.

Yan Cai , Lingwei Cao and Jing Zhao Adaptively Generating High Quality Fixes for Atomicity Violations.
In Proc. ESEC/FSE 2017, September 04-08, 2017, Paderborn, Germany.
http://dx.doi.org/10.1145/3106237.3106239

蔡彦
中国科学院软件研究所，计算机科学国家重点实验室 A218
http://lcs.ios.ac.cn/~yancai
yancai@ios.ac.cn
010-62661613