



Universiteit  
Leiden  
The Netherlands

**Exploring means to facilitate software debugging**  
SOLTANI, M.S.

**Citation**

SOLTANI, M. S. (2020, August 25). *Exploring means to facilitate software debugging*. Retrieved from <https://hdl.handle.net/1887/135948>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/135948>

**Note:** To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/135948> holds various files of this Leiden University dissertation.

**Author:** Soltani, M.S.

**Title:** Exploring means to facilitate software debugging

**Issue Date:** 2020-08-25

## Dutch Summary

Tegenwoordig vertrouwen we op verschillende manieren op computersystemen. Toch zijn deze systemen gevoelig voor falen. Deze mislukkingen kunnen levens verstoren, doden veroorzaken en miljarden dollars kosten. Daarom spelen testen en verificatie van software een cruciale rol bij het voorkomen van dergelijke catastrofale fouten. Omdat softwaretests en verificatieactiviteiten kostbaar en arbeidsintensief zijn, is veel moeite gedaan om zoveel mogelijk activiteiten op deze gebieden te automatiseren. In dit proefschrift is het overkoepelende doel om verschillende manieren te onderzoeken om geautomatiseerde software-foutopsporing te vergemakkelijken.

We presenteren *EvoCrash*, een zoekgebaseerde benadering van geautomatiseerde crash-reproductie. *EvoCrash* past een genetisch algoritme toe om te zoeken naar een test-case die een softwarecrash reproduceert. We hebben een grootschalige evaluatie uitgevoerd om de prestaties van de *EvoCrash*-aanpak te beoordelen en de gebieden te identificeren waar verdere verbetering nodig is.

Verder introduceren we de *IMaChecker*-aanpak, die Github bug repositories verkent, met behulp van Github API's. *IMaChecker* analyseert bovendien bugrapporten en identificeert welke elementen (bijv. Reproductiestappen) erin zijn opgenomen. Met behulp van statistische tests identificeert *IMaChecker* de impact van verschillende elementen van het bugrapport op de tijden voor het oplossen van de bugs.

Ten slotte ontwikkelen we statische analyzers die het gebruik van programmacontracten in open source programma's detecteren die zijn ontwikkeld in Java, C++ en Python. We ontwikkelen verder parsers om fixing commits te identificeren bij de hele commit historie. Met behulp van Poisson-regressietests laten we zien dat er een

negatieve correlatie bestaat tussen het gebruik van contracten en het optreden van bugs. We tonen dus een manier om bugs te voorkomen door programmacontracten te gebruiken.