

Finding Errors Before They Find You – Workshop Report

Event Details

The AC21 workshop on finding software errors was co-located with the [International Colloquium on Theoretical Aspects of Computing](#) held in Stellenbosch from 12 to 18 October 2018. The event itself was held on 15 October and the speaker list and program can be found [here](#). All the invited speakers for the workshop also attended the ICTAC program and social events to allow more options for collaborations. In addition, a number of postgraduate students from Stellenbosch were invited to speak at the workshop and then attend ICTAC as well.



Introduction

The focus of the workshop was of course on automated techniques for finding errors in software, but we left it open as to what specific topics speakers wanted to talk about. We furthermore asked for “new” work, i.e. preferably unpublished; for this reason we didn’t post the presentations on the webpage (only talk titles).

During discussions throughout the workshop it became apparent there was a topic that everybody was interested in, but nobody explicitly presented a talk on: how to test neural networks. After the Monday workshop we had a number of in-depth discussions on this subject and very good progress was made. In fact we are already organizing a new workshop in

Stellenbosch in March 2019 (“Sensemaking in the Digital Age”) to further discuss the broader topic of how to incorporate machine learning and related topics in software systems, how interpret their output and how to make sure they behave fairly and correctly.

In the following we will give a brief overview of the presentations.

Presentations

Prof Matt Dwyer (University of Virginia, USA) and Dr Antonio Filieri (Imperial College London, UK) started off with a joint presentation on how to do compositional software analysis exploiting the likelihood of program behaviors (based on probabilistic symbolic execution). This research looked both at finding errors as well as showing certain properties hold in the software.



Dr Yunho Kim (KAIST, South Korea) showed how one can effectively do unit testing, by extracting system tests that show how the units can be triggered to find errors.



Prof Ina Shaefer (Technische Universität Braunschweig, Germany) showed how one can generate correct programs. This was an interesting alternative to most of the other works that focused on finding errors.



Prof Mike Whalen (University of Minnesota) presented an approach to veritesting of Java code. This technique makes symbolic execution more scalable and thus applicable to more complex Java code.



Prof Jaco Geldenhuys (Stellenbosch University) gave a presentation on two different approaches to doing concolic execution, one based on the Java debugger interface and the other using instrumentation.

Prof Willem Visser (Stellenbosch University) showed how one can combine symbolic execution and fuzzing to find subtle errors.

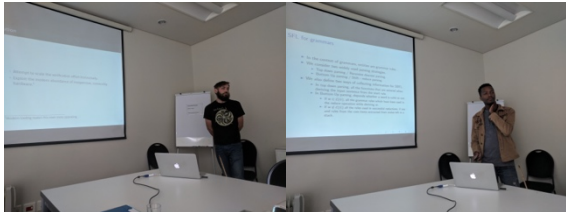
The next 4 presentations were given by postgraduate students:

Alex Leid – symbolic execution of smart contracts

Jan Taljaard – efficient constraint caching

Philip van Heerden – discovering concurrency errors in C

Moeketsi Raselimo – fault localization for grammars



Conclusion

Most importantly the workshop led to a number of new initiatives that we are very excited about.

Willem Visser (Stellenbosch) and Antonio Filieri (Imperial) started working on how to learn a grammar that a program accepts. One of the students, Philip van Heerden, decided this topic is what he wants to do for his M.Sc. We hope to have our first publication in this area soon.

Matt Dwyer (Virginia), Ina Shaefer (Braunschweig) and Willem Visser (Stellenbosch) will together consider analyzing software where machine learning plays a significant role. We started by organizing a follow-on workshop on this topic.

