UCF DEPARTMENT OF COMPUTER SCIENCE

Spring 2018 Seminar Series

Building Correct Programs

Friday, March 2nd 2018

10:00am-11:00am – HEC 450

Everyday code, written in imperative languages, is plagued by bugs: missed corner cases, logic errors, and even compiler bugs. Formal verification techniques allow us to prove that programs always respect their specifications, offering a stronger guarantee of correctness than any other approach to debugging. Several recent projects have demonstrated that we can construct bug-free software at scale, using logic, semantics, and interactive theorem proving. I will present my work in verifying two concurrent applications: a dynamic race detector with a formal guarantee that it correctly implements a race detection algorithm, and a messaging system for autonomous vehicles, which allows successful communication between sensors and control systems even in the presence of malicious components. By combining detailed models of program behavior, state-of-the art logics for memory and concurrency, and tools for constructing and checking mathematical proofs, we can formally guarantee that real-world programs are bug-free.

Dr. William Mansky

Associate Research Scholar at Princeton University

William Mansky is an associate research scholar at Princeton University, working with Andrew Appel in the Verified Software Toolchain group. His research centers on formally modeling the behavior of programming languages, especially memory and concurrency behavior, and proving correctness of real-world programs. He received his PhD from the University of Illinois at Urbana-Champaign in 2014 under Elsa L. Gunter; his thesis described a framework for formal verification of compiler optimizations. He then spent two years as a postdoc at the University of Pennsylvania, working with Steve Zdancewic on formal semantics for the LLVM intermediate representation as part of the Vellvm project.

Hosted by: Gita Sukthankar

UCF

4328 Scorpius Street Orlando, FL 32816

WWW.CS.UCF.EDU