PURDUE UNIVERSITY | Elmore Family School of Electrical and Computer Engineering

Faculty Candidate Seminar – Software Engineering

Jenna DiVincenzo
PhD Candidate, Software Engineering
Carnegie Mellon University

Wednesday, January 11, 2023
Presentation: 10:30 A.M. – 11:30 A.M.
Reception: 11:30 A.M. – 12:00 P.M.
MSEE 112

Gradual Verification: Assuring Programs Incrementally

Abstract: While software is becoming more ubiquitous in our everyday lives, so are unintended bugs. In response, static verification techniques were introduced to prove or disprove the absence of bugs in code. Unfortunately, current techniques burden users by requiring them to write inductively complete specifications involving many extraneous details. To overcome this limitation, I introduce the idea of gradual verification, which handles complete, partial, or missing specifications by soundly combining static and dynamic checking. As a result, gradual verification allows users to specify and verify only the properties and components of their system that they care about and increase the scope of verification gradually—which is poorly supported by existing tools.

In this presentation, I outline the formal foundations of gradual verification for recursive heap data structures (like lists, trees, and graphs), and the design of a gradual verifier derived from my formal work, called Gradual Co. Gradual Co is implemented on top of the Viper static verifier and supports the Co programming language—which is a safer, smaller subset of C taught at CMU. Additionally, I present the results of quantitatively evaluating Gradual Co's static and dynamic performance characteristics for thousands of partial specifications. Gradual Co on average decreases run-time overhead by 50-90% compared to dynamic verification alone and sources of overhead correspond to predictions made in prior work. Qualitatively, Gradual Co exhibits earlier error detection for incorrect specifications than static verification. I end with my planned new lines of work in gradual verification and its application to other areas, such as security and education.

Bio: Jenna DiVincenzo (previously, Jenna Wise) is a PhD candidate in Software Engineering at Carnegie Mellon University, co-advised by Jonathan Aldrich and Joshua Sunshine. She holds a B.S. in Mathematics and Computer Science from Youngstown State University (YSU). Her current work is in gradual verification and she is broadly interested in research at the intersection of formal methods, software engineering, and programming languages. She is a Google PhD Fellow and was awarded an NSF GRFP Fellowship. Previously, she interned at IBM Research, the MIT Lincoln Laboratory, and the Software Engineering Research and Empirical Studies Lab at YSU. She also previously contributed to the language designs of Penrose—which generates diagrams from mathematical prose—and Obsidian—a programming language that facilitates the development of secure blockchain applications.