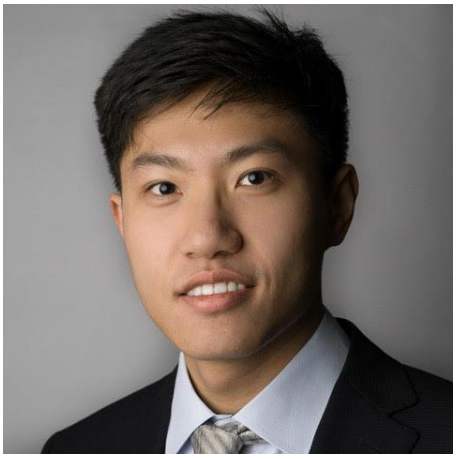
COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

DISSERTATION DEFENSE



Tony Zhang

Practical Verification of Distributed Systems: Streamlining Safety Proofs Using Invariant Taxonomies, and Verifying Latency Properties Using Symbolic Latency

Thursday, January 9, 2025
11:00am – 1:00pm
3941 Beyster
Hybrid – Zoom Passcode: 462205

ABSTRACT: Formal verification promises software that is much more robust, and even bug-free, compared to the traditional approach of testing. However, it remains a niche method, primarily due to limitations in its practicality. In this dissertation, I address two major hurdles in applying formal verification to distributed systems software, namely its ease of use and its application to broader non-functional properties.

First, central to proving the correctness of a distributed protocol is finding an inductive invariant for the protocol. Currently, automated invariant inference algorithms require developers to describe protocols using a restricted, decidable logic. If the developer wants to prove a protocol expressed without these restrictions, they must devise an inductive invariant manually. I present how we can use invariant taxonomies to simplify the task of finding such inductive invariants in the undecidable setting. This taxonomy divides invariants into Regular Invariants, which have one of a few simple low-level structures, and Protocol Invariants, which capture the higher-level host relationships that make the protocol work. Building on the insight of this taxonomy, I present two novel methodologies that streamline and automate the task of finding inductive invariants.

Second, understanding and debugging the performance of distributed systems is critical for developers in the real world, yet non-functional performance properties are currently beyond the scope of formal verification. To address this gap, we extend the application of formal verification to reason about non-functional latency properties of distributed systems. This requires carefully decoupling the formal proofs from the execution environment, formally defining latency properties, and proving them on real, distributed implementations. Our experimental evaluation shows that these bounds are a good proxy for the behavior of the deployed system and can be used to identify performance bugs in real-world systems.

CHAIR: Prof. Manos Kapritsos