Fixes, Proofs and Humans: Results and Lessons from Transforming Programs

Software is a growing part of our daily lives, but software errors remain expensive and costly. In this talk we consider three lenses for improving software quality and reducing the cost of software maintenance. We delve into not only approachable human studies (from software readability judgments to medical imaging of developers) but also rigorous and formal proof and invariant techniques (built atop static and dynamic information). We focus, however, on recent results in automated program repair, in which candidate fixes for software bugs are constructed using focused analyses and program transformations. In each case we summarize lessons learned and highlight reproducible research and key insights. Programming languages and software engineering are the interface between human developers and software systems: program understanding and transformation are powerful levers that can move the world of software quality.