New Foundations for Separation Logic

Hiep, H.A.

Summary

The research presented in this thesis concerns one of the most important questions in software engineering of our time: how can we make sure that software is free from memory safety bugs? Memory safety bugs are the major cause of common vulnerabilities and exposures, and their presence threatens the stability and security of our digital world. This question is so important that it has escalated to the highest level. In a recent White House press release (February 26, 2024), the National Cyber Director of the United States of America calls on the academic community to help solve this hard problem:[1]"addressing [this challenge] is imperative to ensuring we can secure our digital ecosystem long-term and protect the security of our Nation." The accompanying technical report advises on the use of memory safe programming languages, and prominently mentions *formal methods* as one way to achieve the highly desired freedom from bugs, including memory safety bugs.

In this thesis, formal methods are studied that are used to analyze software for its correctness, where correctness means that software satisfies its specification and incorrectness means the presence of a bug. The focus is on separation logic, a formal method designed as a scalable technique in ensuring freedom from memory safety bugs. Nowadays, separation logic is a well-established field of research: it has been widely studied academically in the past twenty years, and is successfully applied on an industry-wide scale to ensure memory safety. For example, separation logic is the technique used to prove, with mathematical certainty, that memory safe programming languages (such as Rust and Go) indeed live up to the promise that "they offer a way to eliminate, not just mitigate, entire bug classes."

In two parts, this thesis presents important scientific contributions that fill a gap in the academic literature. The first part contains the missing **completeness theorem for separation logic**, that is on par with the fundamental result by Gödel for first-order logic. Completeness is important for any formal method as it shows that the formal method can be adequately used for demonstrating every validity. The second part finally introduces **dynamic separation logic** that gives an alternative way to analyze memory safety problems, such that now it is possible to prove elementary specifications without needing extra logical techniques. This is important because it ensures 'backwards compatibility' with automated reasoning techniques that are optimized for first-order logic.

[1] `https://www.whitehouse.gov/oncd/briefing-room/2024/02/26/press-release-technical-report/`