Hazel: Live Functional Programming with Typed Holes

Cyrus Omar

Abstract:

When programming, we spend a substantial amount of our time working with program text that is not yet a formally complete program, e.g. because there are blank spots, type errors or merge conflicts at various locations. Conventional programming language definitions assign no formal meaning to structures like these, so program editors and other tools have no choice but to resort to complex and ad hoc heuristics to provide various useful language services (like code completion, type inspection, code navigation, and live programming services) without gaps in service.

We are developing a more principled approach to working with incomplete programs, rooted in contextual modal type theory and gradual type theory. We model incomplete programs as programs with holes, which (1) stand for parts of the program that are missing; and (2) serve as membranes around parts of the program that are erroneous or, in the collaborative setting, conflicted.

This hole-oriented approach is broadly relevant, but we are incorporating our results into Hazel, a web-based programming environment for an ML-like functional language that is being designed from the ground up around typed hole-driven development. Uniquely, every incomplete program that you can construct using Hazel's language of type-aware edit actions is both statically and dynamically well-defined, i.e. it has a (possibly incomplete) type, and you can run it to produce a (possibly incomplete) result. Consequently, Hazel serves as an elegant platform for research on the future of programming: language services do not exhibit gaps nor do they need to resort to ad hoc heuristics.

This talk will demonstrate Hazel, cover the theoretical and logical foundations of this work, which were described in papers at POPL 2017 and POPL 2019 (to appear), and touch on a number of ongoing projects that members of the Hazel collaboration are working on.

Bio:

Cyrus Omar received his PhD in programming languages from Carnegie Mellon University in 2017. He is now a post-doctoral researcher at University of Chicago, where he leads the Hazel collaboration (hazel.org). Previously, Cyrus was a computational neurobiologist. That experience motivated him to switch fields and focus on designing a better front-end programming experience for scientists, engineers, artists and other skilled end-users, rooted in the principles of type theory and interaction design.