

Specification Synthesis via Generalized Abduction

Aws Albarghouthi¹, Isil Dillig², Arie Gurfinkel³, and Marsha Chechik¹

¹ University of Toronto, Canada
`aws@cs.toronto.edu`, `chechik@cs.toronto.edu`

² UT Austin, US
`isil@cs.utexas.edu`

³ Carnegie Mellon University, US
`arie@cmu.edu`

Abstract

We consider the problem of verifying safety of a program in the absence of code for called procedures. Our goal is to synthesize maximal specifications of absent procedures that maintain safety of the given program. This is useful when the software to be verified contains calls to external components (e.g., libraries) that cannot be analyzed. Such components are usually (1) proprietary software whose source code is not available or (2) libraries that are too complex to be analyzed along with the client application. Thus, synthesized specifications can be used as stubs for calls to external components.

In this talk, we discuss a solution to this problem that relies on a reduction to a novel form of logical abduction, which we call generalized abduction. Generalized abduction is a powerful generalization of the classical abduction problem that allows us to abduce an individual specification (formula) for each unknown procedure as opposed to a single formula in classical abduction. The fact that a procedure may be called from more than one call site, along with the interactions between different call sites, makes this problem quite challenging. We discuss the challenges faced when solving generalized abduction problems, present an algorithm for solving generalized abduction, and outline potential uses of interpolation as a way for solving this problem.