Sound Gradual Verification with Symbolic Execution

Abstract

Gradual verification, which supports explicitly partial specifications and verifies them with a combination of static and dynamic checks, makes verification more incremental and provides earlier feedback to developers. While an abstract, weakest precondition-based approach to gradual verification was previously proven sound, the approach did not provide sufficient guidance for implementation and optimization of the required run-time checks. More recently, gradual verification was implemented using symbolic execution techniques, but the soundness of the approach (as with related static checkers based on implicit dynamic frames) was an open question. This paper puts practical gradual verification on a sound footing with a formalization of symbolic execution, optimized run-time check generation, and run time execution. We prove our approach is sound; our proof also covers a core subset of the Viper tool, for which we are aware of no previous soundness result. Our formalization enabled us to find a soundness bug in an implemented gradual verification tool and describe the fix necessary to make it sound.