Lightweight software verification with pluggable type-checking

Michael D. Ernst

University of Washington and Universidad de Buenos Aires
mernst@cs.washington.edu

Abstract. Software developers often rely on run-time exceptions to indicate bugs in their code. It would be better to use verification to prove the absence of bugs, but verification tends to be difficult to use. We propose a lightweight software verification approach, called pluggable type-checking, that is easy to use, extensible, and provides a compile-time guarantee that certain bugs are not present in the code. Pluggable type-checking permits a software developer to refine the built-in type system of a programming language to catch additional errors, such as null pointer dereferences or race conditions. This approach has been implemented for Java and is available in an open-source tool, the Checker Framework (http://checkerframework.org/). Oracle Corporation is so excited about this technology that Java 8 contains syntactic support for pluggable types. This verification approach is relevant to multiple constituencies. Researchers can build upon the framework to quickly create new program analysis tools. Previously, evaluating a new type system required building a compiler. Now, is it easier to experimentally evaluate a new type system, because the type-checker implementation is only a few lines of code. Educators can introduce software verification in a practical context, enabling students to learn by doing and bringing theory to life. The Checker Framework has been successfully used in the first or second programming class for computer science majors, and also in more advanced classes. Practitioners can use pluggable type-checking to find bugs or to prove the absence of bugs. Use of pluggable type-checking improves code quality and design, and the types act as machine-checked documentation. The Checker Framework is in daily use at corporations such as Google. Attendees will leave the tutorial with a greater appreciation of the theory and practice of pluggable type-checking. They will be prepared to use it for research, teaching, or software development.