Principles and Applications of Refinement Types

Andrew D. Gordon
Microsoft Research, Cambridge, United Kingdom

A refinement type is a type qualified by a logical constraint; an example is the type of even numbers, that is, the type of integers qualified by the is-an-even-number constraint. Although this idea has been known in the research community for some time, it has been assumed impractical because of the difficulties of constraint solving. But recent advances in automated reasoning have overturned this conventional wisdom, and transformed the idea into a practical design principle. I will present a primer on the design, implementation, and application of refinement types. I will explain:

- How a range of diverse features may be unified as instances of the general idea of refinement types.
- How a static checker for a recent systems modelling language allows us to check for security errors in server configurations; intended constraints on configurations are expressed with refinement types, so that configuration validation reduces to type checking.
- How we statically check integrity and secrecy properties of security critical code, such as implementations of cryptographic security protocols, using a system of refinement types for the F# programming language.

The lectures in this series are based on recent research with my esteemed colleagues Karthik Bhargavan, Gavin Bierman, and Cédric Fournet of MSR Cambridge, and David Langworthy of the Microsoft Connected Systems Division; much of our work relies on the excellent Z3 automated theorem prover developed by Nikolaj Bjorner and Leonardo de Moura of MSR Redmond.

I recommend reading the main part of the following technical report, which defines a system of refinement types I will describe in these lectures.

References

1. J. Bengtson, K. Bhargavan, C. Fournet, A. D. Gordon, and S. Maffeis. *Refinement Types for Secure Implementations*. Technical Report MSR–TR–2008–118, Microsoft Research, 2008.

More generally, some familiarity with OCaml or F# would be an advantage.

The web site `http://research.microsoft.com/en-us/people/adg/part.aspx` has additional links to related work.