Proof-Carrying Code with Correct Compilers

Andrew W. Appel
Princeton University
Princeton, NJ

Abstract

In the late 1990s, proof-carrying code was able to produce machine-checkable safety proofs for machine-language programs even though (1) it was impractical to prove correctness properties of source programs and (2) it was impractical to prove correctness of compilers. But now it is practical to prove some correctness properties of source programs, and it is practical to prove correctness of optimizing compilers. We can produce more expressive proof-carrying code, that can guarantee correctness properties for machine code and not just safety. We will construct program logics for source languages, prove them sound w.r.t. the operational semantics of the input language for a proved-correct compiler, and then use these logics as a basis for proving the soundness of static analyses.

E. Denney, T. Jensen (eds.); The 3rd International Workshop on Proof Carrying Code and Software Certification, pp. 2-2