Technologies for finding errors in object-oriented software

K. Rustan M. Leino

Microsoft Research, Redmond, WA, USA
http://research.microsoft.com/~leino
leino@microsoft.com


Abstract. To know if a program is correct, one needs specifications that say what the program is intended to do or intended not to do. But how does one write such specifications for object-oriented software with subtypes and dynamically-dispatched methods, and how does one reason about the semantics of such programs? These lectures give the semantics of a language with modern object-oriented features and consider several issues in the specification of object-oriented programs, including some recently formalized methodology. The lectures also include a demo of a programming tool that reasons semantically about programs and uses an automatic theorem prover to find common errors, and go through some issues in designing such a checker.

This note contains a sample of references related to my lectures at the summer school on Formal Models of Software, 1–6 September 2003, École Polytechnique de Tunisie, Tunis, Tunisia.

0 Overview papers of Extended Static Checking tools and technology


1 ESC/Java checking technology

2 Related work on full functional specification and verification


3 Related work on languages and language features

4 Related work on other checking techniques


5 Work related to my presentation of program semantics


6 Object-oriented program semantics


7 Related work on data abstraction


8 Related work on programming methodology


