Gradual Verification: Assuring Software Incrementally

Jonathan Aldrich
Software and Societal Systems Department, School of Computer Science, Carnegie Mellon University, United States

ABSTRACT Current static verification techniques do not provide good support for incrementality, making it difficult for developers to focus on specifying and verifying the properties and components that are most important. Dynamic verification approaches support incrementality, but cannot provide static guarantees. To bridge this gap, we propose gradual verification, which supports incrementality by allowing every assertion to be complete, partial, or omitted, and provides sound verification that smoothly scales from dynamic to static checking. I'll describe a system that can verify first-order specifications of programs that manipulate recursive, mutable data structures on the heap, demonstrate a prototype tool, and share some empirical results. Our approach addresses several technical challenges, such as semantically connecting iso- and equi-recursive interpretations of abstract predicates, and supporting gradual verification of heap ownership. This work thus lays the foundation for future tools that work on realistic programs and support verification within an engineering process in which cost-benefit tradeoffs can be made.

KEYWORDS Gradual verification, Dynamic and static checking, Program specification, Soundness

About the speaker

Jonathan Aldrich is a Professor of Computer Science and Software Engineering at Carnegie Mellon University, where he directs the Master of Software Engineering program. He is the coauthor (with Michael Scott) of the textbook Programming Language Pragmatics. His research combines programming languages, software engineering, and human-computer interaction to explore how the way we express software affects our ability to engineer software at scale. A particular theme of much of his work is improving software quality and programmer productivity through better ways to express structural and behavioral aspects of software design within source code. Aldrich has contributed to ownership, typestate checking, modular and gradual verification techniques, and usability in programming language and type system design. For his work specifying and verifying architecture, he received a 2006 NSF CAREER award, the 2007 Dahl-Nygaard Junior Prize, and an ICSE test of time award. Outside the university, he serves on the ACM Publications Board and is the CTO of Noteful, a startup delivering a free, fun educational app for music theory and note reading (www.noteful.net). Additional details about his research and publications can be found on his personal webpage at http://www.cs.cmu.edu/~aldrich/.

JOT reference format:
Jonathan Aldrich. *Gradual Verification: Assuring Software Incrementally*. Journal of Object Technology. Vol. 25, No. 1, 2026.
http://dx.doi.org/10.5381/jot.2026.25.1.a1