



# Jeremy Siek

Indiana University

Friday, October 19, 2018

3:00 PM

Luddy Hall, Rm. 1106

## The State of the Art in Gradual Typing

**Abstract:** Gradual typing is an approach for designing programming languages that integrate static and dynamic type checking. Gradual typing gives the programmer fine-grained control over which regions of a program are statically checked and which regions are dynamically checked. Over the last decade, there has been renewed interest in such an integration partly due to the rise in popularity of dynamic languages. They were used to create larger programs, which exposed the need for the early error detection and modularity provided by static type checking. Gradual typing provides a practical migration path for dynamically typed programs to become more statically typed. This talk will give an overview of the state of the art in gradual typing. It will describe a) the major challenges in the design and implementation of gradually typed languages, b) the research that has addressed many of these challenges, and c) the open problems that need to be solved. The challenges facing gradual typing include both theoretical questions and system design problems. On the theoretical side, there are challenges like characterizing what gradual typing is, what type soundness should mean, and whether relational parametricity can be preserved. On the system design side, there are challenges such as creating language run times that implement gradual typing in a time and space-efficient manner.

**Biography:** Jeremy Siek is an Associate Professor at Indiana University Bloomington. Jeremy's interests include programming language design, type systems, mechanized theorem proving, and optimizing compilers. Jeremy's Ph.D. thesis explored foundations for constrained templates, aka the "concepts" proposal for C++. Prior to that, Jeremy developed the Boost Graph Library, a C++ generic library for graph algorithms and data structures. Jeremy post-doc'd at Rice University where he and Walid Taha developed the idea of gradual typing: a type system that integrates both dynamic and static typing in the same programming language. Jeremy is currently working on several open questions regarding gradual typing. Is it possible to create a high-performance implementation of a gradually-typed language? Is there an intuitive specification for when a higher-order cast should fail? In 2009 Jeremy received the NSF CAREER award to fund his project: "Bridging the Gap Between Prototyping and Production". In 2010 and again in 2015, Jeremy was awarded a Distinguished Visiting Fellowship from the Scottish Informatics & Computer Science Alliance. From 2015 to the present Jeremy has been working with colleagues at Northeastern, Brown, and Maryland on the NSF-funded projects Gradual Typing Across the Spectrum and Performant Sound Gradual Typing.

