

Daniel R. Licata

- Personal Information:** E-mail: dr1@cs.cmu.edu
Web: <http://www.cs.cmu.edu/~dr1/>
Home Address: 79 Merit Ln.
Princeton, NJ 08540
Mobile Phone: +1 (412) 889-0106
- Academic Background:** **Institute for Advanced Study** 2012-2013
Member.
Post-doc for a year-long special program on Homotopy Type Theory.
- Carnegie Mellon University** 2011-2012
Teaching Post-doctoral Fellow.
Designed and delivered a new intro. course, Principles of Functional Programming.
- Carnegie Mellon University** 2004 to 2011
PhD in Computer Science. Advised by Robert Harper.
- Brown University** 2000 to 2004
Bachelor of Science in Mathematics and Computer Science.
- Awards & Fellowships:** **FoLLI E.W. Beth Dissertation Award, 2012 Winner**
CMU SCS Dissertation Award, Honorable Mention, 2011
Pradeep Sindhu Computer Science Fellowship, Carnegie Mellon University, 2009-2010.
Finalist for Computing Research Association Outstanding Undergraduate Award, 2004.
- Funding** Cowrote NSF Grant CCF-1116703: Foundations and Applications of Higher-Dimensional Type Theory, which funded part of my post-doc.
Cowrote NSF Grant CCF-0702381: Integrating Types and Verification, which funded part of my dissertation work.
- Publications:** **Dissertation**
Dependently Typed Programming with Domain-Specific Logics. February, 2011.
Committee: Robert Harper, Frank Pfenning, Karl Cray, Greg Morrisett
- Journal Articles**
Robert Harper and Daniel R. Licata. Mechanizing Metatheory in a Logical Framework. *Journal of Functional Programming*. 17(4-5), pp 613-673, July 2007.
- Conference Papers**
Calculating the Fundamental Group of the Circle in Homotopy Type Theory. Daniel R. Licata and Michael Shulman. *IEEE Symposium on Logic in Computer Science (LICS)*, June, 2013.
- Canonicity for 2-Dimensional Type Theory. Daniel R. Licata and Robert Harper. *ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*, 2012.
- 2-Dimensional Directed Type Theory. Daniel R. Licata and Robert Harper. *Mathematical Foundations of Programming Semantics (MFPS)*, 2011.

Security-Typed Programming within Dependently-Typed Programming. Jamie Morgenstern and Daniel R. Licata. *ACM SIGPLAN International Conference on Functional Programming (ICFP)*, 2010.

A Universe of Binding and Computation. Daniel R. Licata and Robert Harper. *ACM SIGPLAN International Conference on Functional Programming (ICFP)*, 2009.

Focusing on Binding and Computation. Daniel R. Licata, Noam Zeilberger, and Robert Harper. *IEEE Symposium on Logic in Computer Science (LICS)*, June 2008.

Verifying Interactive Web Programs. Daniel R. Licata and Shriram Krishnamurthi. *Automated Software Engineering*, 2004. IEEE Press.

The Feature Signatures of Evolving Programs. Daniel R. Licata, Christopher Harris, and Shriram Krishnamurthi. *Automated Software Engineering*, 2003. IEEE Press.

Refereed Workshop Papers

A Monadic Formalization of ML5. Daniel R. Licata and Robert Harper. *Workshop on Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP)*, EPTCS 34, 2010.

Positively Dependent Types. Daniel R. Licata and Robert Harper. *ACM SIGPLAN Workshop on Programming Languages Meets Program Verification (PLPV)*, January 2009.

Refereed Workshop Talks

Computing with Univalence. Daniel R. Licata and Robert Harper. *Workshop on Higher-Dimensional Algebras, Categories, and Types*, 2012.

Security-Typed Programming within Dependently Typed Programming. Jamie Morgenstern and Daniel R. Licata. *Dependently Typed Programming*, 2010.

Mechanizing a Decision Procedure for Coproduct Equality. Arbob Ahmad, Daniel R. Licata, and Robert Harper. *ACM Workshop on Mechanizing Metatheory*, 2007.

Talks for Professional Groups

Programming in Homotopy Type Theory. Daniel R. Licata and Robert Harper. *Meeting of the IFIP Working Group 2.8, Functional Programming*, 2012.

Technical Reports

Tom Murphy VII, Daniel Spoonhower, Chris Casinghino, Daniel R. Licata, Karl Crary, and Robert Harper. The Cult of the Bound Variable: The 9th Annual ICFP Programming Contest. Technical Report CMU-CS-06-163, 2006.

Daniel R. Licata and Robert Harper. A Formulation of Dependent ML with Explicit Equality Proofs. Technical Report CMU-CS-05-178, 2005.

Teaching Experience

CMU 15-150: Principles of Functional Programming

Course Designer and Lecturer, Spring 2011 - Spring 2012

Carnegie Mellon recently chose to revise its introductory computer science curriculum, to teach more verification and parallelism earlier in the curriculum. A key component of this curriculum redesign is a new introductory course on functional programming, 15-150. The course teaches students to write functional programs, to analyze their sequential and parallel time complexity, and to reason mathematically about their

correctness. In Spring 2011, I co-designed 15-150 with Robert Harper, and delivered it to a pilot audience of 80 CS majors. During the 2011-2012 school year, as the sole instructor, I continued to develop the course, delivered it to 205 students (Fall) and 225 students (Spring), and managed a course staff of 15-18 people.

CMU 15-317: Constructive Logic

Teaching Assistant, Fall 2008

With Frank Pfenning

This course, taken primarily by sophomores and juniors, is a rigorous introduction to proof theory and its applications to programming. As a TA, I wrote and graded the assignments and exams, held office hours, and led weekly recitation sections.

CMU 15-312: Principles of Programming Languages

Teaching Assistant, Spring 2006

With Robert Harper

This course, taken primarily by sophomores and juniors, is a rigorous introduction to type systems and operational semantics for functional programming, control and state effects, and concurrency. As a TA, I wrote and graded the assignments and exams, held office hours, and led weekly recitation sections.

Brown CS017/018: An Integrated Introduction to Computer Science

Head Teaching Assistant, 2002-2003 and 2003-2004; Teaching Assistant, 2001-2002

With John F. Hughes and Philip Klein

CS17-18 is a year-long introductory sequence that teaches programming in Scheme, ML, and Java as well as design and analysis of algorithms and data structures. As a TA, I held office hours, led weekly lab sections, graded, and developed new homeworks and exams. As a Head TA, I additionally developed new projects and course software, managed a course staff of seven to nine people, and collaborated with the professor to decide on course content and pedagogical techniques.

Service:

Workshop Organizer

The Twelf Tutorial, co-located with POPL 2009. Organized a 1-day workshop teaching participants to use the Twelf proof assistant to formalize the metatheory of programming languages. I have also written several tutorials on Twelf for the Twelf Wiki (twelf.org).

Program Committee Member

ACM Workshop on Programming Languages meets Program Verification, 2012.

ACM Workshop on Types in Language Design and Implementation (TLDI), 2012.

ACM Workshop on Mechanizing Metatheory (WMM), 2009.

External Reviewer

External reviewer for several journals (JFP, HOSC, TOSEM) and conferences and workshops (POPL, ICFP, LICS, TLCA, PLPV, LFMTP, AOSD, MFPS, ESOP, CPP, CSL, APLAS)

The 9th Annual ICFP Programming Contest

The ICFP Programming Contest is an annual three-day competition associated with the International Conference on Functional Programming. The 2006 contest asked participants to uncover the secrets of a (fictional) ancient society of computer scientists by solving a series of puzzles based on programming languages research. Seven hundred participants on 365 teams from all over the world competed, more than in any previous year. As one of the four primary developers of the contest, I designed puzzles, implemented contest software, and answered questions from participants.

CMU CSD PhD Admissions Committee, 2006 and 2007

In both 2006 and 2007, I was a member of a dozen-person committee of faculty and students that evaluated over 700 applicants for admission to the computer science department's PhD program.

ConCert Reading Group Organizer, 2006 to 2007

I chose research papers for a weekly reading group meeting.

**Undergraduate
Advisees:**

Joseph Lee 2012
Worked on a proof that the torus is homotopy-equivalent to the product of two circles in homotopy type theory. Applying to PhD programs this fall.

Jamie Morgenstern 2009-2010
Embedded a security-typed programming language in a proof assistant as a summer REU from the University of Chicago. Proceeded to the CS PhD program at CMU.

Arbob Ahmad 2007-2008
Worked on deciding coproduct equality in the λ -calculus using a logical technique called focusing. Proceeded to the CS PhD program at CMU.

Internships:

View Patterns for GHC Summer 2007
With Simon Peyton Jones, Microsoft Research Cambridge
During an internship at MSR Cambridge, mentored by Simon Peyton Jones, I implemented a new feature called *view patterns* in the GHC Haskell compiler. View patterns are a convenient way of pattern-matching against abstract data types.