

Curriculum Vitae

Derek Dreyer

Personal Information

Citizenship:	U.S.A.	Home Address:	Lessingstrasse 22
Birthdate:	May 2, 1980		66121 Saarbruecken
Home Phone:	+49 681 6865 441		Germany
Office Phone:	+49 681 9303 8701	Office Address:	MPI-SWS, Campus E1.5
Mobile Phone:	+49 160 9729 8151		66123 Saarbruecken
E-mail:	dreyer@mpi-sws.org		Germany
URL:	http://www.mpi-sws.org/~dreyer		

Academic Background

Carnegie Mellon University

Ph.D. in Computer Science, May 2005. 1997–2005

New York University

B.A. in Mathematics and Computer Science, Summa Cum Laude, May 1996. 1993–1996

Research Experience

Max Planck Institute for Software Systems (MPI-SWS)

Head of the “Foundations of Programming” Group.

Tenured faculty. 2013–present

Tenure-track faculty. 2008–2013

Toyota Technological Institute at Chicago (TTI-C)

Research assistant professor (3-year independent postdoc). 2005–2007

Carnegie Mellon University, Department of Computer Science

Doctoral research. 2000–2004

Thesis: Understanding and Evolving the ML Module System.

Advisors: Robert Harper, Karl Crary.

Committee: Robert Harper, Karl Crary, Peter Lee, David MacQueen.

New York University, Department of Computer Science

Undergraduate research project. 1995–1996

Project: Developing Polynomial-Time Heuristics for the Steiner Tree Problem.

Advisor: Michael Overton.

Awards, Grants, and Fellowships

Max Planck Institute for Software Systems (MPI-SWS)

ACM SIGPLAN Robin Milner Young Researcher Award	2017
ERC Consolidator Grant (5-year project, approx. 2 million euros) for <i>RustBelt: Logical Foundations for the Future of Safe Systems Programming</i>	2015
Microsoft Research PhD Scholarship (3 years of funding for David Swasey, Ph.D. student) on the topic of <i>Compositional Verification of Scalable Joins by Protocol-Based Refinement</i>	2013
Google European Doctoral Fellowship (3 years of funding for Georg Neis, Ph.D. student) on the topic of <i>Compositional Multi-Language Reasoning</i>	2012
Finalist, Microsoft Research Faculty Fellowship	2011

Carnegie Mellon University

National Defense Science and Engineering Graduate Fellowship	1997–2000
--	-----------

Professional Activities

General chair, 2019 ACM SIGPLAN International Conference on Functional Programming (**ICFP 2019**).

Associate editor, ACM Transactions on Programming Languages and Systems (**TOPLAS**), July 2017–present.

Member of the editorial board, Journal of Functional Programming (**JFP**), Feb. 2014–present.

Member-at-large (elected) and awards chair, ACM SIGPLAN Executive Committee, July 2012–June 2015.

Member, IFIP Working Group 2.8 – Functional Programming, Aug. 2014–present.

Workshop founder, ACM SIGPLAN Workshop on Higher-Order Programming with Effects (**HOPE**), 2012.

Lead organizer, Dagstuhl Seminar on “Compositional Verification Methods for Next-Generation Concurrency”, May 2015.

Workshop (co-)chair/organizer:

- 2015–16 Programming Languages Mentoring Workshop (**PLMW 2015–16**).
- 2014 Coq Workshop (**Coq 2014**).
- 2012–13 ACM SIGPLAN Workshop on Higher-Order Programming with Effects (**HOPE 2012–13**).
- 2011 ACM SIGPLAN Workshop on Types in Language Design and Implementation (**TLDI 2011**).
- 2007 ACM SIGPLAN Workshop on ML (**ML 2007**).

Program committee (PC) member:

- 2019 European Conference on Object-Oriented Programming (**ECOOP 2019**).
- 2018 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (**OOPSLA 2018**).
- 2017 ACM SIGPLAN Symposium on Principles of Programming Languages (**POPL 2017**).
- 2016 International Conference on Formal Structures for Computation and Deduction (**FSCD 2016**).
- 2015 Conference on the Mathematical Foundations of Programming Semantics (**MFPS 2015**).
- 2015 European Symposium on Programming (**ESOP 2015**).

- 2014 ACM SIGPLAN International Conference on Functional Programming (**ICFP 2014**).
- 2013 ACM SIGPLAN Workshop on ML (**ML 2013**).
- 2013 International Conference on Certified Programs and Proofs (**CPP 2013**).
- 2013 European Symposium on Programming (**ESOP 2013**).
- 2012 International Conference on Compiler Construction (**CC 2012**).
- 2011 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (**POPL 2011**).
- 2010 ACM SIGPLAN International Workshop on Foundations of Object-Oriented Languages (**FOOL 2010**).
- 2009-10 International Workshop on Modules and Libraries for Proof Assistants (**MLPA 2009-10**).
- 2009 Conference on the Mathematical Foundations of Programming Semantics (**MFPS 2009**).
- 2008 ACM SIGPLAN International Conference on Functional Programming (**ICFP 2008**).
- 2007 ACM SIGPLAN Haskell Workshop (**Haskell 2007**).
- 2007 ACM SIGPLAN International Workshop on Foundations and Developments of Object-Oriented Languages (**FOOL/WOOD 2007**).
- 2006 ACM SIGPLAN Workshop on ML (**ML 2006**).

Selection committee member:

- 2019-20 ACM SIGPLAN Robin Milner Young Researcher Award.
- 2017-18 ACM SIGPLAN John C. Reynolds Doctoral Dissertation Award.
- ICFP 2015 Student Research Competition.

PhD thesis committee member or external examiner:

- Robin Morisset, École Normale Supérieure de Paris (advisor: Francesco Zappa Nardelli), 2017.
- Edward Yang, Stanford University (advisors: David Mazières and John Mitchell), 2017.
- Azalea Raad, Imperial College London (advisor: Philippa Gardner), 2017.

External review committee (ERC) member:

- 2016 ACM SIGPLAN International Conference on Functional Programming (**ICFP 2016**).
- 2016 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (**POPL 2016**).
- 2015 ACM SIGPLAN Conference on Programming Language Design and Implementation (**PLDI 2015**).
- 2015 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (**POPL 2015**).
- 2013 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (**POPL 2013**).
- 2012 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (**POPL 2012**).
- 2009 ACM SIGPLAN Conference on Programming Language Design and Implementation (**PLDI 2009**).

Steering committee (SC) member:

- ACM SIGPLAN International Conference on Functional Programming (**ICFP**), 2016–present.

- ACM SIGPLAN Programming Languages Mentoring Workshop (**PLMW**), 2016–present.
- ACM SIGPLAN Workshop on Types in Language Design and Implementation (**TLDI**), 2011–2012.
- ACM SIGPLAN Workshop on ML, 2008–2010.

Invited speaker:

- 2018 European Symposium on Programming (**ESOP 2018**).
- 2018 ACM SIGPLAN Symposium on Principles of Programming Languages (**POPL 2018**).
- Computer Science Colloquium, Cornell University, November 2017.
- 2017 Workshop on Formal Techniques for Java-like Programs (**FTfJP 2017**).
- 2017 Conference on the Mathematical Foundations of Programming Semantics (**MFPS 2017**).
- ACM SIGPLAN Programming Languages Mentoring Workshop (**PLMW@POPL 2018, PLMW@ICFP 2017, PLMW@POPL 2017, PLMW@ICFP 2016, PLMW@POPL 2016, PLMW@POPL 2014**).
- Distinguished Lecture Series, University of Chicago, May 2016.
- 2016 South of England Regional Programming Languages Seminar (**S-REPLS 3**).
- Workshop on Certification of High-Level and Low-Level Programs, Institut Henri Poincaré thematic trimester on Semantics of Proofs and Certified Mathematics, July 2014.
- 2014 Oregon Programming Languages Summer School (**OPLSS 2014**).
- 2012 Parametricity Workshop, Glasgow, UK.
- 2011 International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice (**LFMTP 2011**).
- 2008 Conference on the Mathematical Foundations of Programming Semantics (**MFPS 2008**).

Co-editor, Special issue of the Journal of Functional Programming (JFP) for selected papers from ICFP 2014.

Workshops co-chair, 2010–11 International Conference on Functional Programming (**ICFP 2010–11**).

Staff representative (Mitarbeitervertreter) of the MPI for Software Systems in the Chemistry, Physics & Technology section of the Max Planck Society, 2010–2016.

Moderator, TYPES and TYPES/announce e-mail forums, April 2009–present.

Senior member, ACM SIGPLAN (Special Interest Group on Programming Languages).

Frequent external reviewer for a number of major conferences and journals, including POPL, ICFP, LICS, PLDI, TOPLAS, JFP, JACM, HOSC, TCS, ESOP, ECOOP, OOPSLA, CSL, PPDP, FLOPS, and APLAS.

Research Advisees

Postdocs at MPI-SWS:

- **Chung-Kil Hur** (Oct. 2010–Sep. 2012, now at Seoul National University).
- **Jacques-Henri Jourdan** (Apr. 2016–Sep. 2017, now at CNRS).
- **Neel Krishnaswami** (Sep. 2011–Sep. 2013, now at University of Cambridge).
- **Ori Lahav** (Apr. 2016–Sep. 2017, now at Tel Aviv University), co-advised by Viktor Vafeiadis.

- **Pierre-Marie Pédro** (since Oct. 2017).
- **Azalea Raad** (since July 2017), co-advised by Viktor Vafeiadis.
- **Andreas Rossberg** (Aug. 2007–Jan. 2010, now at Google Munich).
- **Aaron Turon** (Jan. 2013–April 2014, now at Mozilla Research).
Aaron Turon was recipient of the 2014 ACM SIGPLAN John C. Reynolds Doctoral Dissertation Award. Part II of his dissertation concerns our joint work published in POPL 2013 and ICFP 2013.

Doctoral students at MPI-SWS:

- **Hoang-Hai Dang** (since Apr. 2016).
- **Ralf Jung** (since Sep. 2014).
- **Jan-Oliver Kaiser** (since Sep. 2014).
- **Scott Kilpatrick** (Aug. 2010–Apr. 2016, now at Two Sigma, PhD expected 2017).
- **Georg Neis** (Nov. 2008–Aug. 2015, now at Google Munich, PhD expected 2017).
- **David Swasey** (since Sep. 2012), co-advised by Deepak Garg.
- **Joshua Yanovski** (since July 2017).
- **Beta Ziliani** (Jan. 2010–Feb. 2015, now tenure-track researcher at National University of Cordoba).
– PhD thesis: *Interactive Typed Tactic Programming in the Coq Proof Assistant*.

Interns at MPI-SWS:

- **Jecheon Kang** (Fall 2015).
- **Joseph Tassarotti** (Summer 2014).
- **Zhen Zhang** (Summer 2016).

Publications

All of my papers, including drafts of those currently under submission, are accessible from my web site at <http://www.mpi-sws.org/~dreyer/research.html>.

Journal Publications

Mtac: A Monad for Typed Tactic Programming in Coq.

Beta Ziliani, Derek Dreyer, Neelakantan R. Krishnaswami, Aleksandar Nanevski, Viktor Vafeiadis.
Journal of Functional Programming (JFP), 25, e12, July 2015.
Special issue devoted to archival versions of selected papers from ICFP 2013.

F-ing Modules.

Andreas Rossberg, Claudio Russo, Derek Dreyer.
Journal of Functional Programming (JFP), 24(5): 529–607, September 2014.
This is a significantly revised and expanded version of our TLDI 2010 paper.

How to Make Ad Hoc Proof Automation Less Ad Hoc.

Georges Gonthier, Beta Ziliani, Aleksandar Nanevski, Derek Dreyer.
Journal of Functional Programming (JFP), 23(4): 357–401, July 2013.
Special issue devoted to archival versions of selected papers from ICFP 2011.

Mixin' Up the ML Module System.

Andreas Rossberg, Derek Dreyer.

ACM Transactions on Programming Languages and Systems (TOPLAS), 35(1), Article 2, April 2013.

This is a significantly revised, corrected and expanded version of our ICFP 2008 paper.

The Impact of Higher-Order State and Control Effects on Local Relational Reasoning.

Derek Dreyer, Georg Neis, Lars Birkedal.

Journal of Functional Programming (JFP), 22(4&5): 477–528, September 2012.

Special issue devoted to archival versions of selected papers from ICFP 2010.

Non-Parametric Parametricity.

Georg Neis, Derek Dreyer, Andreas Rossberg.

Journal of Functional Programming (JFP), 21(4&5): 497–562, September 2011.

Special issue devoted to archival versions of selected papers from ICFP 2009.

Logical Step-Indexed Logical Relations.

Derek Dreyer, Amal Ahmed, Lars Birkedal.

Logical Methods in Computer Science (LMCS), 7(2:16): 1–37, June 2011.

Special issue devoted to archival versions of selected papers from LICS 2009.

Recursive Type Generativity.

Derek Dreyer.

Journal of Functional Programming (JFP), 17(4&5): 433–471, July & September 2007.

Special issue devoted to archival versions of selected papers from ICFP 2005.

Conference and Workshop Publications

RustBelt: Securing the Foundations of the Rust Programming Language.

Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, Derek Dreyer.

In 2018 ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2018).

Robust and Compositional Verification of Object Capability Patterns.

David Swasey, Deepak Garg, Derek Dreyer.

In 2017 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2017).

Recipient of **OOPSLA 2017 Distinguished Paper Award**.

Strong Logic for Weak Memory: Reasoning About Release-Acquire Consistency in Iris.

Jan-Oliver Kaiser, Hoang-Hai Dang, Derek Dreyer, Ori Lahav, Viktor Vafeiadis.

In 2017 European Conference on Object-Oriented Programming (ECOOP 2017).

Recipient of **ECOOP 2017 Distinguished Paper Award**.

Repairing Sequential Consistency in C/C++11.

Ori Lahav, Viktor Vafeiadis, Jeehoon Kang, Chung-Kil Hur, Derek Dreyer.

In 2017 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2017).

Recipient of **PLDI 2017 Distinguished Paper Award**.

The Essence of Higher-Order Concurrent Separation Logic.

Robbert Krebbers, Ralf Jung, Aleš Bizjak, Jacques-Henri Jourdan, Derek Dreyer, Lars Birkedal.

In 2017 European Symposium on Programming (ESOP 2017).

A Promising Semantics for Relaxed-Memory Concurrency.

Jeehoon Kang, Chung-Kil Hur, Ori Lahav, Viktor Vafeiadis, Derek Dreyer.

In 2017 ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2017).

Higher-Order Ghost State.

Ralf Jung, Robbert Krebbers, Lars Birkedal, Derek Dreyer.
In 2016 ACM SIGPLAN International Conference on Functional Programming (ICFP 2016).

Lightweight Verification of Separate Compilation.

Jeehoon Kang, Yoonseung Kim, Chung-Kil Hur, Derek Dreyer, Viktor Vafeiadis.
In 2016 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2016).

Pilsner: A Compositionally Verified Compiler for a Higher-Order Imperative Language.

Georg Neis, Chung-Kil Hur, Jan-Oliver Kaiser, Craig McLaughlin, Derek Dreyer, Viktor Vafeiadis.
In 2015 ACM SIGPLAN International Conference on Functional Programming (ICFP 2015).

Verifying Read-Copy-Update in a Logic for Weak Memory.

Joseph Tassarotti, Derek Dreyer, Viktor Vafeiadis.
In 2015 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2015).

Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning.

Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, Derek Dreyer.
In 2015 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2015).

GPS: Navigating Weak Memory with Ghosts, Protocols, and Separation.

Aaron Turon, Viktor Vafeiadis, Derek Dreyer.
In 2014 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2014).

Backpack: Retrofitting Haskell with Interfaces.

Scott Kilpatrick, Derek Dreyer, Simon Peyton Jones, Simon Marlow.
In 2014 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2014).

Mtac: A Monad for Typed Tactic Programming in Coq.

Beta Ziliani, Derek Dreyer, Neelakantan R. Krishnaswami, Aleksandar Nanevski, Viktor Vafeiadis.
In 2013 ACM SIGPLAN International Conference on Functional Programming (ICFP 2013).

Unifying Refinement and Hoare-Style Reasoning in a Logic for Higher-Order Concurrency.

Aaron Turon, Derek Dreyer, Lars Birkedal.
In 2013 ACM SIGPLAN International Conference on Functional Programming (ICFP 2013).

Internalizing Relational Parametricity in the Extensional Calculus of Constructions.

Neelakantan R. Krishnaswami, Derek Dreyer.
In 2013 EACSL Annual Conference on Computer Science Logic (CSL 2013).

Logical Relations for Fine-Grained Concurrency.

Aaron Turon, Jacob Thamsborg, Amal Ahmed, Lars Birkedal, Derek Dreyer.
In 2013 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2013).

The Power of Parameterization in Coinductive Proof.

Chung-Kil Hur, Georg Neis, Derek Dreyer, Viktor Vafeiadis.
In 2013 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2013).

Superficially Substructural Types.

Neelakantan R. Krishnaswami, Aaron Turon, Derek Dreyer, Deepak Garg.
In 2012 ACM SIGPLAN International Conference on Functional Programming (ICFP 2012).

The Marriage of Bisimulations and Kripke Logical Relations.

Chung-Kil Hur, Derek Dreyer, Georg Neis, Viktor Vafeiadis.
In 2012 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2012).

How to Make Ad Hoc Proof Automation Less Ad Hoc.

Georges Gonthier, Beta Ziliani, Aleksandar Nanevski, Derek Dreyer.
In 2011 ACM SIGPLAN International Conference on Functional Programming (ICFP 2011).

Separation Logic in the Presence of Garbage Collection.

Chung-Kil Hur, Derek Dreyer, Viktor Vafeiadis.
In 2011 IEEE Symposium on Logic in Computer Science (LICS 2011).

A Kripke Logical Relation Between ML and Assembly.

Chung-Kil Hur, Derek Dreyer.
In 2011 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2011).

The Impact of Higher-Order State and Control Effects on Local Relational Reasoning.

Derek Dreyer, Georg Neis, Lars Birkedal.
In 2010 ACM SIGPLAN International Conference on Functional Programming (ICFP 2010).
This paper was nominated by ACM SIGPLAN for a **CACM Research Highlight**.

F-ing Modules.

Andreas Rossberg, Claudio V. Russo, Derek Dreyer.
In 2010 ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI 2010).

A Relational Modal Logic for Higher-Order Stateful ADTs.

Derek Dreyer, Georg Neis, Andreas Rossberg, Lars Birkedal.
In 2010 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2010).

Non-Parametric Parametricity.

Georg Neis, Derek Dreyer, Andreas Rossberg.
In 2009 ACM SIGPLAN International Conference on Functional Programming (ICFP 2009).

Logical Step-Indexed Logical Relations.

Derek Dreyer, Amal Ahmed, Lars Birkedal.
In 2009 IEEE Symposium on Logic in Computer Science (LICS 2009).

State-Dependent Representation Independence.

Amal Ahmed, Derek Dreyer, Andreas Rossberg.
In 2009 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2009).

Mixin' Up the ML Module System.

Derek Dreyer, Andreas Rossberg.
In 2008 ACM SIGPLAN International Conference on Functional Programming (ICFP 2008).

A Type System for Recursive Modules.

Derek Dreyer.
In 2007 ACM SIGPLAN International Conference on Functional Programming (ICFP 2007).

Principal Type Schemes for Modular Programs.

Derek Dreyer, Matthias Blume.
In 2007 European Symposium on Programming (ESOP 2007).

Modular Type Classes.

Derek Dreyer, Robert Harper, Manuel M.T. Chakravarty.
In 2007 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2007).

Recursive Type Generativity.

Derek Dreyer.
In 2005 ACM SIGPLAN International Conference on Functional Programming (ICFP 2005).

A Type System for Well-Founded Recursion.

Derek Dreyer.

In 2004 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2004).

A Type System for Higher-Order Modules.

Derek Dreyer, Karl Cray, Robert Harper.

In 2003 ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2003).

Typed Compilation of Recursive Datatypes.

Joseph C. Vanderwaart, Derek Dreyer, Leaf Petersen, Karl Cray, Robert Harper, Perry Cheng.

In 2003 ACM SIGPLAN Workshop on Types in Language Design and Implementation (TLDI 2003).

Ph.D. Thesis

Understanding and Evolving the ML Module System.

Derek Dreyer.

Ph.D. Thesis, Carnegie Mellon University Technical Report CMU-CS-05-131, May 2005.

Teaching Experience

Max Planck Institute for Software Systems (MPI-SWS) / Saarland University

Co-instructor and course designer

Winter 2015–16

Winter 2017–18

Core graduate/undergraduate course: Semantics.

Course home page: https://courses.ps.uni-saarland.de/sem_ws1718/.

Co-taught a core course on programming language semantics with Prof. Gert Smolka of Saarland University. My half of the course focused on the use of logical-relations models of types to reason about local invariants, semantic safety, and representation independence.

Instructor and course designer

Winter 2014–15

Graduate course: Categorical Logic.

Led an advanced graduate seminar on basic category theory and applications to building models of higher-order separation logic.

Instructor and course designer

Winter 2012–13

Graduate course: Parametricity and Modular Reasoning.

Course home page: <https://wiki.mpi-sws.org/star/paramore>.

Led an advanced graduate seminar on the theory of parametricity, focusing on the use of logical-relations techniques for modular reasoning about a wide variety of semantically complex programming language features.

Co-instructor and course designer

Summer 2011

Graduate course: Concurrent Program Logics.

Course home page: <https://wiki.mpi-sws.org/star/cpl>.

Led an advanced graduate seminar, together with Viktor Vafeiadis, on Hoare-style logics for concurrent shared-memory programs.

Instructor and course designer

Winter 2010–11

Graduate course: Type Systems for Modules.

Course home page: <http://www.mpi-sws.org/~skilpat/modsem/>.

Led an advanced graduate seminar on type systems for modular programming, focusing on the design and evolution of the ML module system.

Instructor and course designer

Winter 2008–09

Graduate course: Typed Operational Reasoning.

Course home page: <http://www.mpi-sws.org/~dreyer/tor/>.

Taught a variant of the *Advanced Type Systems* graduate course that I had previously given in Winter 2006 at the University of Chicago (see below).

University of Chicago

Guest instructor

Fall 2006

Graduate/undergraduate course: Programming Languages.

Lectured for two weeks on explicit and implicit variants of the polymorphic λ -calculus, and created two homework assignments on the material covered.

Instructor and course designer

Winter 2006

Graduate course: Advanced Type Systems.

Course home page: <http://tti-c.org/dreyer/course/>.

Designed an original seminar course, in which the students learned how to apply the method of *logical relations* to prove a range of different theorems about program semantics (*e.g.*, strong normalization, decidability of typechecking, parametricity properties, and program equivalence). Lectured twice a week, and created and graded homework assignments.

Carnegie Mellon University

Teaching assistant

Spring 1999

Undergraduate course: Programming Languages.

Instructor: Robert Harper.

Created and graded exams and homework assignments, and led weekly recitation sections.

Teaching assistant

Spring 1998

Undergraduate course: Compiler Design.

Instructor: Peter Lee.

Created and graded exams and homework assignments.

New York University

Teaching assistant

Spring 1997

Undergraduate course: Introduction to Computer Science II (Data Structures).

Instructor: Samuel Marateck.

Assisted students in the computing lab and via e-mail.

Instructor

Fall 1996

Undergraduate course: Mathematical Thinking (Basic Mathematics).

Lectured three times a week, and created and graded exams and homework assignments.