

Kathleen Fisher

Professor and Chair, Computer Science Department
Tufts University (973) 610-2669
kfisher@eecs.tufts.edu
<http://www.eecs.tufts.edu/~kfisher/>
Nationality: USA

Positions

- Chair, Computer Science Department, Tufts University, September 2016 to the present.
- Professor, Computer Science Department, Tufts University, March 2011 to the present.
- Program Manager, I20, DARPA, July 2011 to July 2014.
- Consulting Professor in Computer Science, Stanford University, July 2008 to March 2011.
- Principal Member of the Technical Staff, AT&T Labs Research, April 2002 to March 2011.
- Senior Member of the Technical Staff, AT&T Labs Research, September 1996 to April 2002.

Education

- **Ph.D.** Computer Science, **Stanford University**, September 1996.
Dissertation: Type Systems for Object-Oriented Languages
Advisor: Professor John C. Mitchell
- **B.Sc.** Math & Computational Science with Distinction, **Stanford University**, June 1991.

Research Interests

Programming languages, programming language support for ad hoc data management, domain-specific programming languages, high-assurance software, probabilistic programming languages, language design, type systems, object-oriented programming.

Awards and Honors

- SIGPLAN Distinguished Service Award, 2013
- Distinguished PLDI Paper, 2012.
- SIGPLAN CACM Research Highlights Nominated Paper, 2012.
- PLDI Best Paper Award, 2011.
- ACM Fellow, 2010.
- SIGPLAN CACM Research Highlights Nominated Paper, 2008.
- ACM Distinguished Scientist, 2007.
- Best Research Paper Award, KDD, 2000.
- NSF Mathematical Sciences Postdoctoral Research Fellowship, 1996, declined.
- The University of California President's Postdoctoral Fellowship, 1996, declined.
- The Student Service Award, Stanford Computer Science Department, 1995
- Hertz Foundation Fellowship, 1994 to 1996
- Finch Fellowship, 1993
- NSF Graduate Research Fellowship, 1991 to 1994

Papers and Publications

Journals and Conferences

1. K. Fisher, J. Launchbury, and R. Richards. The HACMS Program: Using formal methods to eliminate exploitable bugs. *Philosophical Transactions A*. Forthcoming.
2. J. DiLorenzo, R. Zhang, E. Menzies, K. Fisher, and N. Foster. Incremental Forest: A DSL for efficiently managing filestores. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2016*, pages 252–271, 2016. (Acceptance rate $\frac{52}{203} = 25\%$)
3. Y. R. Wang, D. Nunez, and K. Fisher. Autobahn: Using genetic algorithms to infer strictness annotations. In *Proceedings of the 9th International Symposium on Haskell*, Haskell 2016, pages 114–126, 2016.
4. M. Greenberg, K. Fisher, and D. Walker. Tracking the flow of ideas through the programming languages literature. In *Proceedings of the Inaugural Summit on Advances in Programming Languages*, 2015.
5. T. Parr, S. Harwell, and K. Fisher. Adaptive LL(*) parsing: The power of dynamic analysis. In *Proceedings of the 2014 ACM SIGPLAN International Conference on Object-Oriented Programming Systems Languages and Applications*, 2014. (Acceptance rate $\frac{53}{185} = 28\%$)
6. P. Hawkins, A. Aiken, K. Fisher, M. Rinard, and M. Sagiv. Concurrent data representation synthesis. In *PLDI '12: Proceedings of the 2012 ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2012. Won 2012 PLDI Distinguished Paper Award. (Acceptance rate $\frac{48}{255} = 18\%$)
7. P. Hawkins, A. Aiken, K. Fisher, M. Rinard, and M. Sagiv. Reasoning about lock placements. In *European Symposium on Programming*, 2012.
8. K. Q. Zhu, K. Fisher, and D. Walker. LearnPADS++: Incremental learning of ad hoc data formats. In *PADL '12: Proceedings of the 14th International Symposium on Practical Aspects of Declarative Languages*, 2012.
9. K. Fisher, N. Foster, D. Walker, and K. Q. Zhu. Forest: A language and toolkit for programming with filestores. In *ICFP '11: Proceedings of the Sixteenth ACM SIGPLAN International Conference on Functional Programming*, 2011. (Acceptance rate $\frac{33}{112} = 29\%$)
10. T. Parr and K. Fisher. LL(*): The foundation of the ANTLR parser generator. In *PLDI '11: Proceedings of the 2011 ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2011. (Acceptance rate $\frac{55}{236} = 23\%$)
11. P. Hawkins, A. Aiken, K. Fisher, M. Rinard, and M. Sagiv. Data representation synthesis. In *PLDI '11: Proceedings of the 2011 ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2011. Won 2011 PLDI Best Paper Award. (Acceptance rate $\frac{55}{236} = 23\%$)
12. K. Fisher, Y. Mandelbaum, and D. Walker. The next 700 data description languages. *Journal of the ACM*, 57(2):1–51, 2010.

13. K. Q. Zhu, D. S. Dantas, K. Fisher, L. Jia, Y. Mandelbaum, V. Pai, and D. Walker. Language support for processing distributed ad hoc data. In *PPDP '09: Proceedings of the 11th ACM SIGPLAN Conference on Principles and Practice of Declarative Programming*, pages 243–254, New York, NY, USA, 2009. ACM. (Acceptance rate $\frac{26}{57} = 45\%$)
14. Q. Xi, K. Fisher, D. Walker, and K. Q. Zhu. Ad hoc data and the token ambiguity problem. In *PADL '09: Proceedings of the 11th International Symposium on Practical Aspects of Declarative Languages*, pages 91–106, Berlin, Heidelberg, 2009. Springer-Verlag. (Acceptance rate $\frac{18}{48} = 37\%$)
15. K. Fisher, D. Walker, K. Q. Zhu, and P. White. From dirt to shovels: Fully automatic tool generation from ad hoc data. In *POPL '08: Proceedings of the 35th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 421–434, New York, NY, USA, 2008. ACM. (Acceptance rate $\frac{35}{212} = 16\%$)
16. M. F. Fernández, K. Fisher, J. N. Foster, M. Greenberg, and Y. Mandelbaum. A generic programming toolkit for PADS/ML: First-class upgrades for third-party developers. In *PADL '08: Proceedings of the 10th International Symposium on Practical Aspects of Declarative Languages*, Berlin, Heidelberg, 2008. Springer-Verlag. (Acceptance rate $\frac{20}{44} = 45\%$)
17. Y. Mandelbaum, K. Fisher, D. Walker, M. Fernández, and A. Gleyzer. PADS/ML: A functional data description language. In *POPL '07: Proceedings of the 34th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 77–83, New York, NY, USA, 2007. ACM. (Acceptance rate $\frac{36}{198} = 18\%$)
18. K. Fisher, Y. Mandelbaum, and D. Walker. The next 700 data description languages. In *POPL '06: Proceedings of the 33rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 2–15, New York, NY, USA, 2006. ACM. (Acceptance rate $\frac{33}{167} = 19\%$)
19. K. Fisher and R. Gruber. PADS: A domain-specific language for processing ad hoc data. In *PLDI '05: Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 295–304, New York, NY, USA, 2005. ACM. (Acceptance rate $\frac{28}{135} = 20\%$)
20. N. Ramsey, K. Fisher, and P. Govereau. An expressive language of signatures. In *ICFP '05: Proceedings of the Tenth ACM SIGPLAN International Conference on Functional Programming*, pages 27–40, New York, NY, USA, 2005. ACM. (Acceptance rate $\frac{26}{87} = 29\%$)
21. C. Cortes, K. Fisher, D. Pregibon, A. Rogers, and F. Smith. Hancock: A language for analyzing transactional data streams. *ACM Transactions on Programming Languages and Systems*, 26(2):301–338, 2004.
22. K. Fisher and J. Reppy. Inheritance-based subtyping. *Information & Computation*, 177(1):28–55, 2002.
23. P. Di Blasio, K. Fisher, and C. Talcott. A control-flow analysis for a calculus of concurrent objects. *IEEE Transactions on Software Engineering*, 26(7):617–634, 2000.
24. C. Cortes, K. Fisher, D. Pregibon, and A. Rogers. Hancock: A language for extracting signatures from data streams. In *KDD '00: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 9–17, New York, NY, USA, 2000. ACM. Best paper award.

25. K. Fisher, J. H. Reppy, and J. G. Riecke. A calculus for compiling and linking classes. In *ESOP '00: Proceedings of the 9th European Symposium on Programming Languages and Systems*, pages 135–149, London, UK, 2000. Springer-Verlag. (Acceptance rate $\frac{26}{84} = 30\%$)
26. K. Fisher and J. H. Reppy. Extending Moby with inheritance-based subtyping. In *ECOOP '00: Proceedings of the 14th European Conference on Object-Oriented Programming*, pages 83–107, London, UK, 2000. Springer-Verlag.
27. K. Fisher and J. Reppy. The design of a class mechanism for Moby. In *PLDI '99: Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 37–49, New York, NY, USA, 1999. ACM. (Acceptance rate $\frac{26}{130} = 20\%$)
28. D. Bonachea, K. Fisher, A. Rogers, and F. Smith. Hancock: A language for processing very large-scale data. In *PLAN '99: Proceedings of the 2nd Conference on Domain-specific Languages*, pages 163–176, New York, NY, USA, 1999. ACM.
29. K. Fisher and J. C. Mitchell. On the relationship between classes, objects, and data abstraction. *Theory and Practice of Object Systems*, 4(1):3–25, 1998.
30. V. Bono and K. Fisher. An imperative, first-order calculus with object extension. In *ECOOP '98: Proceedings of the 12th European Conference on Object-Oriented Programming*, pages 462–497, London, UK, 1998. Springer-Verlag. (Acceptance rate $\frac{24}{124} = 19\%$)
31. P. D. Blasio and K. Fisher. A calculus for concurrent objects. In *CONCUR '96: Proceedings of the 7th International Conference on Concurrency Theory*, pages 655–670, London, UK, 1996. Springer-Verlag.
32. K. Fisher and J. C. Mitchell. The development of type systems for object-oriented languages. *Theory and Practice of Object Systems*, 1(3):189–220, 1995.
33. K. Fisher and J. C. Mitchell. A delegation-based object calculus with subtyping. In *FCT '95: Proceedings of the 10th International Symposium on Fundamentals of Computation Theory*, pages 42–61, London, UK, 1995. Springer-Verlag.
34. K. Fisher and J. C. Mitchell. Notes on typed object-oriented programming. In *TACS '94: Proceedings of the International Conference on Theoretical Aspects of Computer Software*, pages 844–885, London, UK, 1994. Springer-Verlag.
35. K. Fisher, F. Honsell, and J. C. Mitchell. A lambda calculus of objects and method specialization. *Nordic Journal of Computing*, 1(1):3–37, 1994.
36. J. C. Mitchell, F. Honsell, and K. Fisher. A lambda calculus of objects and method specialization. In *Proceedings of Eighth Annual IEEE Symposium on Logic in Computer Science, 1993. LICS '93.*, pages 26–38. IEEE Computer Society, 1993.

Workshops, Demonstrations, and Technical Reports

37. C. Cortes, K. Fisher, D. Pregibon, A. Rogers, and F. Smith. Hancock: A language for analyzing transactional data streams. In M. Garofalakis, J. Gehrke, and R. Rastogi, editors, *Data Stream Management: Processing High-Speed Data Streams*, chapter Part 4, Chapter 4, pages 387–408. Springer, 2016.
38. K. Q. Zhu, K. Fisher, and D. Walker. Incremental learning of system log formats. *SIGOPS Operating Systems Review*, 44(1):85–90, 2010. (Acceptance rate $\frac{6}{11} = 54\%$)
39. M. Bailey, K. Bruce, K. Fisher, R. Harper, and S. Reges. Report of the 2008 SIGPLAN programming languages curriculum workshop: Preliminary report. In *SIGCSE '09: Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, pages 132–133, New York, NY, USA, 2009. ACM.
40. K. Fisher. We need more than one: Why students need a sophisticated understanding of programming languages. *SIGPLAN Notices*, 43(11):62–65, 2008.
41. K. Fisher and C. Krintz. SIGPLAN Programming Language Curriculum Workshop: Workshop report summary. *SIGPLAN Notices*, 43(11):29–30, 2008.
42. K. Fisher, D. Walker, and K. Q. Zhu. LearnPADS: Automatic tool generation from ad hoc data. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1299–1302, New York, NY, USA, 2008. ACM. (Acceptance rate $\frac{30}{93} = 32\%$)
43. Y. Mandelbaum, K. Fisher, and D. Walker. A dual semantics for the data description calculus. In *Informal Proceedings of the Trends In Functional Programming Conference*, 1997.
44. D. Burke, P. White, K. Fisher, D. Walker, and K. Q. Zhu. Towards 1-click tool generation with PADS. In *Proceedings of the Challenges and Applications of Grammar Induction Workshop*, 1997.
45. M. Daly, Y. Mandelbaum, D. Walker, M. Fernández, K. Fisher, R. Gruber, and X. Zheng. PADS: An end-to-end system for processing ad hoc data. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, pages 727–729, New York, NY, USA, 2006. ACM.
46. M. Fernández, K. Fisher, R. Gruber, and Y. Mandelbaum. PADX : Querying large-scale ad hoc data with XQuery. In *Proceedings of the PLAN-X 2006 Workshop*, 2006.
47. K. Fisher and C. Chambers. Experience with author response at PLDI and ICFP 2004. *SIGPLAN Notices*, 39(12):1–6, 2004.
48. K. Fisher and J. Reppy. A typed calculus of traits. In *Proceedings of the 11th Workshop on Foundations of Object-oriented Programming*, January 2004.
49. K. Fisher and R. Gruber. PADS: Processing arbitrary data streams. In *Workshop on Management and Processing of Data Streams*, 2003. (Acceptance rate $\frac{17}{36} = 47\%$)
50. K. Fisher, R. Pucella, and J. Reppy. A framework for interoperability. In N. Benton and A. Kennedy, editors, *Proceedings of the First International Workshop on Multi-Language Infrastructure and Interoperability (BABEL'01)*, volume 59 of *Electronic Notes in Theoretical Computer Science*, New York, NY, 2001. Elsevier Science Publishers.

51. K. Fisher, C. R. Goodall, K. Högstedt, and A. Rogers. An application-specific database. In *DBPL '01: Revised Papers from the 8th International Workshop on Database Programming Languages*, pages 213–227, London, UK, 2001. Springer-Verlag.
52. K. Fisher and J. Reppy. Inheritance-based subtyping. In *Proceedings of the 7th Workshop on Foundations of Object-oriented Programming*, 2000.
53. V. Bono and K. Fisher. An imperative, first-order calculus with object extension. In *Proceedings of the 5th Workshop on Foundations of Object-oriented Programming*, 1998.
54. P. Di Blasio, K. Fisher, and C. Talcott. Analysis for concurrent objects. In *FMOODS '97: Proceedings of the IFIP TC6 WG6.1 International Workshop on Formal Methods for Open Object-based Distributed Systems*, pages 73–88, London, UK, UK, 1997. Chapman & Hall, Ltd.
55. K. Fisher. *Type Systems for Object-Oriented Programming Languages*. PhD thesis, Stanford University, 1996.
56. K. Fisher and J. C. Mitchell. Classes = Objects + Data abstraction. Research Note STAN-CS-TN-96-31, Stanford University, 1996.
57. K. Fisher and J. C. Mitchell. What is an object-oriented programming language?, 1995.

Book Chapters

1. C. Cortes, K. Fisher, D. Pregibon, A. Rogers, and F. Smith. Hancock: A language for analyzing transactional data streams. In M. Garofalakis, J. Gehrke, and R. Rastogi, editors, *Data Stream Management: Processing High-Speed Data Streams*, chapter Part 4, Chapter 4, pages 387–408. Springer, 2016.

Funding

- Synthesizing Data Wranglers. Subcontract to Princeton on DARPA contract (PI). November 2016 to October 2018.
- Prattle: Deception and anti-Reconnaissance via CyberChaff. Subcontract to Galois AF SBIR (PI). September 2016 to September 2018.
- Sirius: A Toolkit for Building First-Class Domain-Specific Languages. DARPA prime contract number FA8750-15-2-0033 (Senior Personnel). November 2014 to September 2017.
- Language Support for Data-centric Systems Monitoring. NSF award CNS 0615062 (Senior Personnel). July 2006 to July 2009.
- Automatic Tool Generation for Ad Hoc Scientific Data. NSF award IIS 0612147 (Senior Personnel). July 2006 to July 2009.
- Workshop on Modern Programming Language Curricula, NSF Award 0825525 (PI). May 2008 to April 2010.
- Real-Time Network Forensic Analysis. DARPA prime contract number FA8750-07-C-0014 (PI). January to June 2007.
- Special Projects: Travel Grants for Faculty at Minority/Female Institutions to Attend FCRC'03, NSF Award 0243337 (PI). March 2003 to February 2004.

Professional Activities

Harvey Mudd Board of Trustees: Member: July 2016 - present.

DARPA's ISAT working group: Vice Chair: 2016-present. Member: 2015 - present.

ACM Fellows Selection Committee: 2015-2019.

INRIA Review Panel: Chaired Committee of Experts for review of INRIA's Theme on Programming, Proofs, and Verification. Review occurs once every four years. Spring 2015.

NSF Committee of Visitors: 2014.

NSF Panels: 2002, 2003, 2004, 2006, 2007, 2009, 2014, 2017.

Co-Chair and Creator, Programming Language Mentoring Workshop (PLMW) Member of the steering committee (2016-present). Invited Speaker: ICFP 2015, ICFP 2016, and POPL 2017.

Computing Research Association Board: June 2009 to March 2011.

ACM Special Interest Group (SIG) Governing Board, Vice Chair for SIG Development: Elected to a two-year term, June 2008 to June 2010.

SIGPLAN: Past Chair: June 2009 to June 2012. Chair: June 2007 to June 2009. Vice Chair: June 2003 to June 2007. Member at Large: June 2001 to June 2003.

Co-Chair, SIGPLAN Undergraduate Programming Language Curriculum Workshop Envisioned and led two-day workshop with 30 participants to rethink undergraduate programming language curriculum. Co-edited final report, published in November, 2008.

Consulting Faculty, Computer Science Department, Stanford University: July 2008 to March 2011.

CRA Subcommittee on the Status of Women in Research (CRA-W) Board: Chair: October 2008 to 2011. Steering Committee Member: June 2006 to 2011. Board Member: October 2003 to the 2011. The CRA-W Board is an active board; each member is responsible for an on-going project.

Transactions on Programming Languages and Systems: Associate Editor: October 2014 to the present.

Journal of Functional Programming: Editor: June 2005 to March 2011. Editorial Board Member: April 2004 to June 2005.

Steering Committees: ICFP 2003 to 2009, 2013 to the present, HCSS 2012 to the present, (co-chair 2015, 2016), PLDI 2003 to the 2016, OOPSLA 2003 to 2015, POPL 2003 to 2009, CUFP 2007 to 2011 (chair 2007 to 2010), FOOL/WOOD 2001 to 2009 (chair 2006, 2007).

General Chair: ICFP '15.

Program Chair: PPS '16 (co-chair), HCSS '15 (co-chair), OOPSLA '11, CUFP '07 (co-chair), CUFP '06 (co-chair), ICFP '04, FOOL '01.

Program Committees: POPL '17, ML4PL '15, PLDI '15, POPL '15 (extended review committee), FARM '14, Approx '14, NFM '13, DDFP '13, PLDI '13 (extended review committee), FARM '13, XLDI '12, TLDI '12, POPL '11, ML Workshop '10, OOPSLA '10, PLDI '10 (extended review committee), IFL '09, CUFP '09, ICFP '08, ECOOP '08, PLDI '07, HOPL '07, OOPSLA '05, FOAL '05, FOAL '04, WOOD '03, PADL '02, POPL '02, PLDI '01, FMOODS '00, ICFP '99, MFPS '99, OOPSLA '98, FOOL '98, and OOPSLA '97.

Refereeing: CONCUR, ECOOP, FPCA, ICFP, LICS, OOPSLA, POPL, PLDI, *Communications of the ACM Journal of the ACM, ACM Computing Surveys, ACM Transactions on Embedded Computing Systems, ACM Transactions on Programming Languages and Systems, IEEE Transactions on Software Engineering, Information and Computation, Information and Software Technology, Journal of Functional Programming, Lisp and Symbolic Computation, Theory and Practice of Object Systems* .

Seminar coordination:

- New Jersey Programming Languages and Systems Seminar, 2001 to 2005.
- Stanford Theory Distinguished Speaker Colloquium, 1995 to 1996.

Stanford Computer Science Department Service: Ph.D. admissions committee (2010), Teaching faculty search committee (2009), Master's student advisor (2009 to the 2011), Master's admissions committee (2009), Ph.D. qualifying exam committee (2007).

Chair, FCRC'03 Educator's Grant Program: Chaired committee to provide travel grants for 100 educators at institutions with high percentages of women and minorities.

Other activities:

- Member of IFIP Working Group 2.8, 2004 to present, Chair (2015-2017).
- Member of Association for Computing Machinery (ACM).
- Member of American Association for the Advancement of Science (AAAS) since 2017.
- SIGPLAN Programming Language Achievement Award Selection Committee, 2016 and 2017.
- Member of the Grace Hopper Industrial Affiliates Advisory Committee, 2006.
- Speaker, CRA-W graduate cohort program, 2004, 2005, 2006, 2007, 2008, 2009, and 2011.
- Speaker, CRA-W mentoring workshop, 2002, 2004, 2007, and 2015.
- Speaker, CRA mentoring workshop, February 2010 and December 2010, 2012, 2014.

Tufts University Service

- DISC Advisory Board (2017-present).
- Tufts Innovation Institute, Executive Committee Member (2014-2016).
- Tufts Innovation Institute, Steering Committee Member (2014-2016).
- Tufts Thematic Working Group on Computational and Quantitative Skills and Methods (2014-2015).

Tufts Computer Science Department Service

- Chair (2016 - present).
- Computer Science Department Graduate Committee, Chair (2014 - 2016).
- Computer Science Department Admissions Committee, (2014 - 2016).

Students

PhD Mentor, Tufts University (January 2017 to present) Mentored and worked with Sam Lasser on verified parsing.

Master's Mentor, Tufts University (January 2017 to present) Mentored and worked with Ethan Pailles on incremental parsing.

Fullbright Mentor, Tufts University (January 2017 to April 2017) Mentored and worked with Ryma Abassi on the verification of a vehicle-to-vehicle network protocol.

- Post Doc Mentor, Tufts University** (September 2016 to present) Mentored and worked with Ferdinand Veseley on specifying semantics for domain-specific languages.
- PhD Mentor, Tufts University** (September 2016 to present) Mentored and worked with Jared Chandler on tools to visualize tree-structured data.
- PhD Mentor, Tufts University** (September 2016 to present) Mentored and worked with Jeanne-Marie Musca on tools to construct domain-specific languages.
- PhD Mentor, Tufts University** (September 2015 to present) Mentored and worked with Matthew Ahrens on tools to construct domain-specific languages and to build a domain-specific language for wearable computing.
- Undergraduate Mentor, Tufts University** (April 2015 to present) Mentored and worked with Remy Wang on automatically inserting strictness annotations to improve performance of Haskell programs.
- Undergraduate Mentor, Tufts University** (January 2015 to May 2016) Mentored and worked with Siddhartha Prasad on developing domain-specific debugging infrastructure for parsers as part of a COMP-0194 Directed Study course.
- Undergraduate Mentor, Tufts University** (March 2015 to September 2015) Mentored and worked with Caleb Helbling on using databases to manage programs at the level of individual functions. Received 2015 Tufts Summer Scholar award to fund this project.
- PhD Mentor, Stanford University** (September 2007 to 2012) Mentored and worked with Peter Hawkins on developing analyses to ensure memory safety of overlapping data structures in low-level C code.
- PhD Mentor, Princeton University** (September 2004 to August 2006) Mentored and worked with Yitzhak Mandelbaum on the PADS project after he completed his 2004 summer internship until he finished his Ph.D.
- Summer Students** Xuan Zheng (University of Michigan, 2005) Yitzak Mandelbaum (Princeton, 2004, 2005), Ricardo Medel (Stevens Institute of Technology, 2003), Yan Yu (USC, 2000), Fred Smith (Cornell, 1999-2002), Dan Bonachea (U.C. Berkeley, 1998),
- Master's Student Advisor, Stanford University** (Winter 2006 to Spring 2007) Supervised Pascal Perez's Master's degree with a specialization in research on an efficient type inferencing algorithm for a higher-order functional language extended with positive subtyping.

Teaching Experience

Instructor, Tufts University (Spring 2015, Spring 2016, Fall 2016)

Taught COMP105 (Programming Languages), a required course for the Computer Science major. Added recitations to the course in 2015. (Spring 2015: 75 students, Spring 2016: 77 students, Fall 2017: 105 students).

Instructor, Tufts University (Fall 2014)

Developed Comp150PLD, a new course on Programming Language Design that focused on domain-specific languages (DSLs). Students read academic papers on example DSLs and on tools for building DSLs. They each designed and built a DSL in the domain of their choice during the semester. (11 students).

Co-instructor, Stanford University (Fall 2010)

Gave six lectures in cs242, *Programming Languages*, Stanford's required masters' level course on programming language fundamentals. Topics included an introduction to functional programming, type inference, type classes, the IO Monad in Haskell, software transactional memory, and nested data parallelism.

Co-instructor, Stanford University (Fall 2009)

Gave six lectures in cs242, *Programming Languages*, Stanford's required masters' level course on programming language fundamentals. Topics included an introduction to functional programming, type inference, type classes, the IO Monad in Haskell, software transactional memory, and nested data parallelism. Student evaluation scores, based on 80 responses, were 4.05/5 for the course and 3.98/5 for me as an instructor.

Co-instructor, Stanford University (Fall 2008)

Gave six lectures in cs242, *Programming Languages*, Stanford's required masters' level course on programming language fundamentals. Topics included an introduction to functional programming, type inference, type classes, the IO Monad in Haskell, monads, and software transactional memory.

Co-instructor, Stanford University (Fall 2005)

Gave five lectures in cs242, *Programming Languages*, Stanford's required masters' level course on programming language fundamentals. Topics included scripting languages, domain-specific languages, object-oriented languages, and support for interoperability.

Teaching Fellow, Stanford University (Fall 1995)

Sole instructor for cs242, *Programming Languages*, Stanford's required masters' level course on programming language fundamentals. Approximate enrollment of 120, including 20 industry professionals taking the course through the Stanford Instructional Television Network.

Teaching Assistant, Stanford University (1993 to 1994)

Courses: *Automata and Complexity Theory*, *Programming Languages*.

Responsibilities included course design, section leading, weekly office hours, and grading.

Teaching Fellow, Stanford University (Summer 1990)

Co-instructor for accelerated introductory programming course, cs106x.

Section Coordinator, Stanford University (1990 to 1991)

Prepared and taught seminar on teaching sections.

Section Leader, Stanford University (1989 to 1990)

Prepared and taught a weekly section for beginning programming courses.

Lectures

<i>Using Formal Methods to Eliminate Exploitable Bugs</i>	
Keynote, Tufts President's Council	November 2016
Invited Talk, Workshop on Safety and Control for Artificial Intelligence	June 2016
Invited Keynote, The 8th NASA Formal Methods Symposium	June 2016
Invited Talk, Royal Society	April 2016
Department Colloquium, University of Texas A&M	January 2016
Invited Talk, PEPM	January 2016
Invited Talk, YOW! Sydney	December 2015
Invited Talk, YOW! Brisbane	December 2015
Invited Talk, YOW! Melbourne	December 2015
Department Colloquium, Brown University	November 2015
Department Colloquium, Cornell University	October 2015
Invited Talk, USENIX Security	August 2015
<i>Securing the Internet of Things</i>	
Panelist, House Congressional Committee on Homeland Security	October 2015
<i>Wrangling Complexity: Designing Systems that Work</i>	
Panelist, DARPA Wait What?	September 2015
<i>Using Topic Models to Track the Flow of Ideas in the PL Literature</i>	
ML4PL Workshop, Prague, Czech Republic	July 2015
IFIP Working Group 2.8 (Functional Programming), Kefalonia, Greece	May 2015
<i>PL&ML: Two Great Tastes that Taste Great Together</i>	
Invited Talk, DALI 2015: Data, Learning, and Inference	April 2015
Invited Talk, NIPS Workshop on Probabilistic Programming	December 2014
<i>HACMS: Provably Correct Software for Embedded Systems</i>	
Invited Talk, Stanford Center for Automotive Research	April 2015
Keynote, UK Cyber Security Research Conference	October 2014
Keynote, ACM International Conference on Functional Programming	September 2014
Google I/O	June 26, 2014
ESCAR USA	June 19, 2014
Keynote, ESCAR USA	May 30, 2014
Kick-off Meeting: Airworthiness of Complex Systems Study Group	April 30, 2014
National Security Agency	April 18, 2014
New England Advanced Cyber Security Center Annual Conference	November 12, 2013
Keynote, Second Annual Symposium, Maryland Cybersecurity Center	May 14, 2013
Layered Assurance Workshop	December 4, 2012
Keynote, ACM High Integrity Language Technology Conference	December 5, 2012
Keynote, High Confidence Software Systems Symposium	May 9, 2012
National Security Agency	January 17, 2012

The PADS Project: An Overview

Invited Talk, Brown University

November 2015

BBN Technologies, Cambridge, MA

April 5, 2011

Invited Tutorial, International Conference on Database Theory,

March 22, 2011

Uppsala, Sweden

Forest: A Language and Toolkit for Programming with Filestores

International Conference on Functional Programming, Tokyo, Japan.

September 21, 2011

Typing FileStores

IFIP Working Group 2.8 (Functional Programming), Austin, Texas.

March 10, 2011

Typing Directories

Galois, Portland, Oregon.

May 3, 2010

IFIP Working Group 2.8 (Functional Programming), Frauenchiemsee,

June 8, 2009

Germany.

Transforming Data into Information

University of Washington Department Colloquium, Seattle, Washington.

March 4, 2010

*Programming Language Ideas Escape the Lab: A Declarative Data**Description Language for Managing Ad hoc Data.*

NSA, Fort Meade, Maryland

September 30, 2010

New York University Department Colloquium, New York, New York

April 7, 2010

Northeastern University Department Colloquium, Boston, Massachusetts

April 5, 2010

Tufts University Department Colloquium, Medford, Massachusetts

April 1, 2010

University of Edinburgh Department Colloquium, Edinburgh, Scotland

January 14, 2010

University of Washington Department Colloquium, Seattle, Washington

January 7, 2010

Invited Technical Speaker, Grace Hopper Celebration of Women in

October 3, 2008

Computing, Keystone, Colorado.

From Dirt to Shovels: Inferring PADS Descriptions from ASCII Data

Invited Speaker, Computer Network Defense Research and Technology

June 22, 2009

Workshop, Annapolis Junction, Maryland.

Faculty Lunch Speaker, Stanford University, Stanford, California.

February 3, 2009

Invited Speaker, High Confidence Software Systems Conference,

March 7, 2008

Linthicum, Maryland.

Invited Speaker, Microsoft Research, Cambridge, England.

February 25, 2008

IFIP Working Group 2.8 (Functional Programming), Nesjavellir, Iceland.

July 18, 2007

*The Future of Undergraduate Programming Language Education: A**Report on the SIGPLAN Programming Language Workshop*

Invited Speaker, High Confidence Software Systems Conference,

May 18, 2009

Linthicum, Maryland.

IFIP Working Group 2.8 (Functional Programming), Park City, Utah.

June 16, 2008

PADS: A System for Managing Ad hoc Data

- Invited Lecture, University of Central Florida, Orlando, Florida. *October 18, 2007*
 Distinguished Lecture, Pomona College, Claremont, California. *November 9, 2006*
 Database Seminar, Stanford University, Stanford, California. *March 10, 2006*
 Distinguished Lecture, Kansas State University, Manhattan, Kansas. *March 7, 2006*
 Invited Lecture, IBM Research, Yorktown Heights, New York. *December 1, 2005*
 Invited Lecture, SRI, Menlo Park, California. *November 11, 2005*
 Invited Lecture, Microsoft Research, Redmond, Washington. *February 5, 2004*

Typing Ad hoc Data

- Invited Speaker, Workshop on Types in Language Design and Implementation, Nice, France. *January 16, 2007*

The Next 700 Data Description Languages

- Distinguished Lecture, Kansas State University, Manhattan, Kansas. *March 8, 2006*
 IFIP Working Group 2.8 (Functional Programming), Kalvi Manor, Estonia. *October 2, 2005*
 Invited Lecture, Oregon Summer School, Eugene, Oregon. *July 21, 2005*

Hancock: A Language for Analyzing Transactional Data Streams

- Invited Lecture, Oregon Summer School, Eugene, Oregon. *July 19, 2005*
 Invited Lecture, Harvard University, Cambridge, Massachusetts. *April 21, 2003*
 CRA-W Distinguished Lecture Series, UCSD, San Diego, California. *November 21, 2002*
 Department Colloquia, University of Arizona, Tucson, Arizona. *January 14, 2002*
 Invited Lecture, MIT, Cambridge, Massachusetts. *December 13, 2000*
 Invited Lecture, Intertrust, Mountain View, California. *November 13, 2000*
 Distinguished Lecture Series, Williams College, Williams, Massachusetts. *July 23, 2000*

Ad hoc Data: An Opportunity for Domain-Specific Languages

- IFIP Working Group 2.8 (Functional Programming), West Point, New York. *November 3, 2004*
 Invited Speaker, High Confidence Software Systems Conference, Linthicum, Maryland. *April 14, 2004*

The Design of a Class Mechanism for Moby

- Distinguished Lecture, Università di Genova, Genova, Italy. *March 24, 2000*

Foundations for Moby Classes

- Invited Lecture, POP Seminar, Carnegie Mellon University, Pittsburgh, Pennsylvania. *November 9, 1998*

On the Relationship Between Classes, Objects and Data Abstraction

Invited Lecture, Princeton University, Princeton, New Jersey.

December 4, 1997

Directions in Object-Oriented Programming Languages:

Type Systems and Language Development

Invited Lecture, University of Pennsylvania, Philadelphia, Pennsylvania.

February 3, 1997

Department Colloquia, Cornell University, Ithaca, New York.

April 30, 1996

Department Colloquia, Purdue University, West Lafayette, Indiana.

April 4, 1996

Department Colloquia, Loyola University, Chicago, Illinois.

April 3, 1996

Department Colloquia, Iowa State University, Ames, Iowa.

April 1, 1996

Invited Lecture, AT&T Bell Labs, Murray Hill, New Jersey.

March 28, 1996

Invited Lecture, University of California at Berkeley, Berkeley, California.

Fall 1995

Classes = Objects + Data Abstraction

Invited Talk, 3rd Workshop on Foundations of Object-Oriented Languages, New Brunswick, New Jersey.

May 5, 1996

Media Appearances

- “Great Debate: The Future of Artificial Intelligence— Who’s in Control,” *Origins Workshop Great Debate*. February 24, 2017. Panelist. <https://origins.asu.edu/events/future-of-artificial-intelligence>.
- “Computer Hacks of the Future, and How to Prevent Them,” *Science Friday*. February 25, 2017. Panelist. <https://www.sciencefriday.com/segments/computer-hacks-of-the-future-and-how-to-prevent-them/>.
- “DARPA: Nobody’s Safe on the Internet,” *Sixty Minutes*. February 8, 2015. Panelist. <http://www.cbsnews.com/news/darpa-dan-kaufman-internet-security-60-minutes/>.