

Curriculum Vitae

Robert Bruce Findler · Apr. 2019

Mailing Address

EECS, Northwestern University
2145 Sheridan Rd, Tech L359
Evanston, IL 60208
USA

Web, Phone

robby@eecs.northwestern.edu
eecs.northwestern.edu/~robby
Office: (847) 467-0952
Fax: (847) 491-4455

Education

- 2002 PhD, Rice University (Defense: December 2001). Dissertation: *Behavioral Software Contracts*. Advisor: Matthias Felleisen.
- 1995 BS, Carnegie Mellon University. Majors: Computer Science, Mathematics, with honors

Employment

- 2017–now Professor. Northwestern University.
- 2010–2017 Associate Professor. Northwestern University.
 - S2014 Visiting Associate Professor. National Taiwan University.
- 2009–2010 Assistant Professor. Northwestern University.
- 2002–2008 Assistant Professor. University of Chicago.
- 1/2002-7/2002 Post-doc, under Matthias Felleisen. Northeastern University.
 - S2000 IBM Watson.
- 1997–2002 Graduate Research Assistant. Rice University.
- 1995–1997 Graduate Teaching Assistant. Rice University.
 - S1994 Programmer, directed by Garth Gibson. Carnegie Mellon University.
 - S1993 Programmer, Principles of Programming Languages Group, directed by Peter Lee, Carnegie Mellon University.

Books

- 2009 Matthias Felleisen, Robert Bruce Findler, Matthew Flatt. *Semantics Engineering with PLT Redex*. MIT Press.

- 2000 Matthias Felleisen, Robert Bruce Findler, Matthew Flatt and Shriram Krishnamurthi. *How to Design Programs*. MIT Press. The entire text is also freely available on the book's website.

Copies sold as of April 2016: 16,349
(not counting the four foreign editions)
Unique web visitors per day, 1/2003—7/2009: 450
Unique web visitors per day, 1/2008—7/2009: 774
Web visitors (per day) between 2003 & 2009: 1,068,097

Distributed Software

- 1996–now DrRacket, programming environment for Racket. Freely available at <https://www.racket-lang.org/>.

Average downloads per day (unique IPs) in 2012: 256
Average downloads per day (unique IPs) in 2013: 415
Average downloads per day (unique IPs) in 2014: 448
Average downloads per day (unique IPs) in 2015: 517
Subscribers to the mailing list as of 10/2016: 1,381
Average messages/day to mailing list 9/2015—9/2016: 9

- 2004–now PLT Redex, a semantics engineer's toolkit. Freely available at <https://redex.racket-lang.org/>

Manuals

- 2008–now Matthew Flatt, Robert Bruce Findler, and PLT. *Guide: Racket*
- 2006–now Jacob Matthews and Robert Bruce Findler. *PLaneT: Automatic Package Distribution*.
- 2005–now Matthew Flatt and Robert Bruce Findler. *Slideshow: Racket Figure and Presentation Tools*.
- 2004–now Robert Bruce Findler and Casey Klein. *Redex: Debugging Operational Semantics*.
- 1998–now Robert Bruce Findler. *Plugins: Extending DrRacket*.
- 1997–now Robert Bruce Findler and Matthew Flatt. *Framework: Racket GUI Application Framework*.
- 1996–now Robert Bruce Findler and PLT. *DrRacket: Programming Environment*.

Awards

- 2018 Racket received the 2018 Programming Languages Software Award
- 2013-4 Named to the Associated Student Government Faculty and Administration Honor Roll
- 2012 Received the Most Influential ICFP Paper Award for the 2002 paper, *Contracts for Higher-Order Functions*.

Selective Conference Publications

- 2019 Florence, You, Tov, Findler, *A Calculus for Esterel: If can, can. If no can, no can..* Symposium on Principles of Programming Languages (POPL).
- 2018 Feltey, Greenman, Scholliers, Findler, St-Amour, *Collapsible Contracts: Fixing a Pathology of Gradual Typing*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA).
- 2017 St-Amour, Feltey, Florence, You, Findler, *Herbarium Rack-etensis: A Stroll Through the Woods*. International Conference on Functional Programming (ICFP).
- 2016 Moore, Dimoulas, Findler, Flatt, Chong, *Extensible Access Control with Authorization Contracts*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) (acceptance rate 47 of 176, 26.7%).
- 2016 Dimoulas, New, Findler, Felleisen, *Oh Lord, Please Don't Let Contracts Be Misunderstood*. International Conference on Functional Programming (ICFP) (acceptance rate 37 of 118, 31.3%).
- 2015 Takikawa, Feltey, Dean, Flatt, Findler, Tobin-Hochstadt, Felleisen, *Towards Practical Gradual Typing*. European Conference for Object-Oriented Programming (ECOOP) (acceptance rate 31 of 136, 22.7%).
- 2015 Fetscher, Claessen, Paćka, Hughes, Findler, *Making Random Judgments: Automatically Generating Well-Typed Terms from the Definition of a Type-System*. European Symposium on Programming (ESOP) (acceptance rate 33 of 113, 29.2%).
- 2013 Dimoulas, Findler, Felleisen, *Option Contracts*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) (acceptance rate 50 of 189, 26.4%).
- 2012 Strickland, Tobin-Hochstadt, Findler, Flatt, *Chaperones and Impersonators: Run-time Support for Reasonable Interposition*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) (acceptance rate 59 of 228, 25.8%).

- 2012 Klein, Clements, Dimoulas, Eastlund, Felleisen, Flatt, McCarthy, Rafkind, Tobin-Hochstadt, Findler, *Run Your Research: On the Effectiveness of Lightweight Mechanization*. Symposium on Principles of Programming Languages (POPL) (acceptance rate 44 of 205, 21.4%).
- 2011 Klein, McCarthy, Jacopette, Findler, *A Semantics for Context-Sensitive Reduction Semantics*. Asian Symposium on Programming Languages and Systems (APLAS) (acceptance rate 22 of 64, 34.3%).
- 2011 Hoang, Findler, Joseph, *Exploring Circuit Timing-aware Languages and Compilation*. Architectural Support for Programming Languages and Operating Systems (ASPLOS) (acceptance rate 32 of 152, 21.0%).
- 2011 Dimoulas, Findler, Flanagan, Felleisen, *Correct Blame for Contracts: No More Scapegoating*. Symposium on Principles of Programming Languages (POPL) (acceptance rate 49 of 209, 23.4%).
- 2011 Ahmed, Findler, Siek, Wadler, *Blame for All*. Symposium on Principles of Programming Languages (POPL) (acceptance rate 49 of 209, 23.4%).
- 2010 Klein, Flatt, Findler, *Random Testing for Higher-Order, Stateful Programs*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) (acceptance rate 45 of 164, 27.4%).
- 2010 Swaine, Tew, Dinda, Findler, Flatt, *Back to the Futures: Incremental Parallelization of Existing Sequential Runtime Systems*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) (acceptance rate 45 of 164, 27.4%).
- 2009 Flatt, Barzilay, Findler, *Scribble: Closing the Book on Ad Hoc Documentation Tools*. International Conference on Functional Programming (ICFP) (acceptance rate 26 of 101, 25.7%).
- 2009 Felleisen, Findler, Flatt, Krishnamurthi, *A Functional I/O System, or Fun for Freshmen Kids*. International Conference on Functional Programming (ICFP) (acceptance rate 26 of 101, 25.7%).
- 2009 Wadler, Findler, *Well-typed Programs Can't be Blamed*. European Symposium on Programming (ESOP) 1-15 (acceptance rate 26 of 98, 26.5%).

- 2007 Flatt, Yu, Findler, Felleisen, *Adding Delimited and Composable Control to a Production Programming Environment*. International Conference on Functional Programming (ICFP) 165-176 (acceptance rate 26 of 103, 25.2%).
- 2007 Kuan, MacQueen, Findler, *A Rewriting Semantics for Type Inference*. European Symposium on Programming (ESOP) 426-440 (acceptance rate 34 of 136, 25.0%).
- 2007 Matthews, Findler, *Operational Semantics for Multi-Language Programs*. Symposium on Principles of Programming Languages (POPL) 3-10 (acceptance rate 36 of 198, 18.1%).
- 2006 Flatt, Findler, Felleisen, *Scheme with Classes, Mixins, and Traits*. Asian Symposium on Programming Languages and Systems (APLAS) 270-289 (acceptance rate 20 of 70, 28.5%).
- 2006 Meunier, Findler, Felleisen, *Modular Set-Based Analysis from Contracts*. Symposium on Principles of Programming Languages (POPL) 218-231 (acceptance rate 33 of 167, 19.7%).
- 2005 Gray, Findler, Flatt, *Fine-Grained Interoperability through Contracts and Mirrors*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) 231-246 (acceptance rate 25 of 142, 17.6%).
- 2004 Goldberg, Findler, Flatt, *Super and Inner - Together at Last!*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) 116-129 (acceptance rate 27 of 173, 15.6%).
- 2004 Findler, Flatt, *Slideshow: Functional Presentations*. International Conference on Functional Programming (ICFP) 224-235 (acceptance rate 21 of 80, 26.2%).
- 2004 Flatt, Findler, *Kill-Safe Synchronization Abstractions*. Programming Language Design and Implementation (PLDI) 47-58 (acceptance rate 25 of 128, 19.5%).
- 2004 Matthews, Findler, Flatt, Felleisen, *A Visual Environment for Developing Context-Sensitive Term Rewriting Systems*. International Conference on Rewriting Techniques and Applications (RTA) 301-312 (acceptance rate 19 of 43, 44.1%).
- 2004 Findler, Flatt, Felleisen, *Semantic Casts: Contracts and Structural Subtyping in a Nominal World*. European Conference for Object-Oriented Programming (ECOOP) 364-388 (acceptance rate 25 of 132, 18.9%).
- 2003 Graunke, Findler, Krishnamurthi, Felleisen, *Modeling Web Interactions*. European Symposium on Programming (ESOP) 238-252 (acceptance rate 25 of 99, 25.2%).

- 2002 Findler, Felleisen, *Contracts for Higher-Order Functions*. International Conference on Functional Programming (ICFP) 48-59 (acceptance rate 24 of 76, 31.5%).
- 2001 Graunke, Findler, Krishnamurthi, Felleisen, *Automatically Restructuring Programs for the Web*. Automated Software Engineering (ASE) 211-222 (acceptance rate 32 of 164, 19.5%).
- 2001 Findler, Felleisen, *Contract Soundness for Object-Oriented Languages*. Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA) 1-15 (acceptance rate 27 of 145, 18.6%).
- 2001 Findler, Latendresse, Felleisen, *Behavioral Contracts and Behavioral Subtyping*. Foundations of Software Engineering (FSE) 229-236 (acceptance rate 29 of 137, 21.1%).
- 1999 Flatt, Findler, Krishnamurthi, Felleisen, *Programming Languages as Operating Systems (or, Revenge of the Son of the Lisp Machine)*. International Conference on Functional Programming (ICFP) 138-147 (acceptance rate 25 of 81, 30.8%).
- 1998 Findler, Flatt, *Modular Object-Oriented Programming with Units and Mixins*. International Conference on Functional Programming (ICFP) 94-104 (acceptance rate 30 of 77, 38.9%).
- 1997 Findler, Flanagan, Flatt, Krishnamurthi, Felleisen, *DrScheme: A Pedagogic Programming Environment for Scheme*. Programming Languages: Implementations, Logics, and Programs (PLILP) 369-388 (acceptance rate 26 of 68, 38.2%).

Book Chapter

- 2006 Krishnamurthi, Findler, Graunke, Felleisen, *Modeling Web Interactions and Errors*. Interactive Computation: The New Paradigm 255-276.

Journal Publications

- 2019 Greenman, Takikawa, New, Feltey, Findler, Vitek, Felleisen, *How to Evaluate the Performance of Gradual Type Systems*. Journal of Functional Programming (JFP).
- 2018 McCarthy, Fetscher, New, Feltey, Findler, *A Coq Library For Internal Verification of Running-Times*. Science of Computer Programming (SCP).

- 2018 Florence, Fetscher, Flatt, Temps, St-Amour, Kiguradze, West, Niznik, Yarnold, Findler, Belknap, *POP-PL: A Patient-Oriented Prescription Programming Language*. Transactions on Programming Languages and Systems (TOPLAS).
- 2017 New, Fetscher, McCarthy, Findler, *Fair Enumeration Combinators*. Journal of Functional Programming (JFP).
- 2013 Klein, Flatt, Findler, *The Racket virtual machine and randomized testing*. Journal of Higher-Order and Symbolic Computing (HOSC).
- 2012 Flatt, Culpepper, Darais, Findler, *Macros that Work Together: Compile-time bindings, partial expansion, and definition contexts*. Journal of Functional Programming (JFP) 22(02):181-216.
- 2009 Sperber, Dybvig, Flatt, van Straaten, Findler, Matthews, Kelsey, Clinger, Rees, *Revised⁶ Report on the Algorithmic Language Scheme*. Journal of Functional Programming (JFP) 19(s1):1-301.
- 2009 Matthews, Findler, *Operational Semantics for Multi-Language Programs*. Transactions on Programming Languages and Systems (TOPLAS) 31(3): 1-44.
- 2008 Matthews, Findler, *An Operational Semantics for Scheme*. Journal of Functional Programming (JFP) 18(1): 47-86.
- 2006 Findler, Flatt, *Slideshow: Functional Presentations*. Journal of Functional Programming (JFP) 16(4-5): 583-619.
- 2005 Meunier, Findler, Steckler, Wand, *Selectors Make Set-based Analysis too Hard*. Journal of Higher-Order and Symbolic Computing (HOSC) 18(3-4) 245-269, December.
- 2004 Felleisen, Findler, Flatt, Krishnamurthi, *The TeachScheme! Project: Computing and Programming for Every Student*. Computer Science Education (CSE) 14(1): 55-77, March.
- 2004 Matthews, Findler, Graunke, Krishnamurthi, Felleisen, *Automatically Restructuring Programs for the Web*. Automated Software Engineering, an International Journal (ASEj) 11(4): 337-364, October.
- 2004 Felleisen, Findler, Flatt, Krishnamurthi, *The Structure and Interpretation of the Computer Science Curriculum*. Journal of Functional Programming (JFP) 14(4): 365-378.
- 2002 Findler, Clements, Flanagan, Flatt, Krishnamurthi, Steckler, Felleisen, *DrScheme: A Programming Environment for Scheme*. Journal of Functional Programming (JFP) 12(2): 159-182, March.

- 1995 Perlin, M.W., Duggan, D.J., Davis, K., Farr, J.E., Findler, R.B., Higgins, M.J., Nowak, N.J., Evans, G.A., Qin, S., Zhang, J., Shows, T.B., James, M.R., and Richard III, C.W., *Rapid construction of integrated maps using inner product mapping: YAC coverage of human chromosome 11*. *Genomics* 28(2): 315-327.

Workshop Publications

- 2017 Tobin-Hochstadt, Felleisen, Findler, Flatt, Greenman, Kent, St-Amour, Strickland, Takikawa, *Migratory Typing: Ten Years Later*. Summit on Advances in Programming Languages (SNAPL).
- 2016 Feltey, Florence, Knutson, St-Amour, Culpepper, Flatt, Findler, Felleisen, *Languages the Racket Way: 2016 Language Workbench Challenge*. LWC@SLE Language Workbench Challenge.
- 2016 McCarthy, Fetscher, New, Feltey, Findler, *A Coq Library For Internal Verification of Running-Times*. International Symposium on Functional and Logic Programming (FLOPS).
- 2015 Florence, Fetscher, Flatt, Temps, Kiguradze, West, Niznik, Yarnold, Findler, Belknap, *POP-PL: A Patient-Oriented Prescription Programming Language*. Generative Programming: Concepts & Experience (GPCE).
- 2015 Felleisen, Findler, Flatt, Krishnamurthi, Barzilay, McCarthy, Tobin-Hochstadt, *The Racket Manifesto*. Summit on Advances in Programming Languages (SNAPL).
- 2013 Tew, Swaine, Flatt, Findler, Dinda, *Distributed Places*. Trends in Functional Programming (TFP).
- 2013 Cooper, Guha, Krishnamurthi, McCarthy, Findler, *Teaching Garbage Collection without Implementing Compilers or Interpreters*. ACM Special Interest Group on Computer Science Education (SIGCSE).
- 2012 Swaine, Fetscher, St-Amour, Findler, Flatt, *Seeing the Futures: Profiling Shared-Memory Parallel Racket*. Workshop on Functional High-Performance Computing (FHPC).
- 2011 Tew, Swaine, Flatt, Findler, Dinda, *Places: Adding Message-Passing Parallelism to Racket*. Dynamic Languages Symposium (DLS).
- 2010 Barland, Flatt, Findler, *The Design of a Functional Image Library*. Workshop on Scheme and Functional Programming (SFP).

- 2009 Klein, Findler, *Randomized Testing in PLT Redex*. Workshop on Scheme and Functional Programming (SFP).
- 2009 Ahmed, Findler, Matthews, Wadler, *Blame for all*. Workshop on Script to Program Evolution (STOP).
- 2009 Tobin-Hochstadt, Findler, *Cycles without pollution: a gradual typing poem*. Workshop on Script to Program Evolution (STOP).
- 2007 Findler, Guo, Rogers, *Lazy Contract Checking for Immutable Data Structures*. International Symposium on Implementation and Application of Functional Languages (IFL).
- 2007 Wadler, Findler, *Well-typed Programs Can't be Blamed*. Workshop on Scheme and Functional Programming (SFP) 15-26.
- 2007 Guha, Matthews, Findler, Krishnamurthi, *Relationally-Parametric Polymorphic Contracts*. Dynamic Languages Symposium (DLS) 29-40 (acceptance rate 9 of 30, 30.0%).
- 2006 Findler, Blume, *Contracts as Pairs of Projections*. International Symposium on Functional and Logic Programming (FLOPS) 226-241 (acceptance rate 17 of 50, 34.0%).
- 2005 Matthews, Findler, *An Operational Semantics for R5RS Scheme*. Workshop on Scheme and Functional Programming (SFP) 41-54.
- 2002 Felleisen, Findler, Flatt, Krishnamurthi, *The Structure and Interpretation of the Computer Science Curriculum*. Functional and Declarative Programming in Education (FDPE) 21-26.
- 2001 Meunier, Findler, Steckler, Wand, *Selectors Make Analyzing case-lambda Too Hard*. Workshop on Scheme and Functional Programming (SFP).

Unrefereed Publications

- 2018 Felleisen, Findler, Flatt, Krishnamurthi, Barzilay, McCarthy, Tobin-Hochstadt, *A Programmable Programming Language*. Communications of the ACM (CACM) March.
- 2004 Clements, Felleisen, Findler, Flatt, Krishnamurthi, *Fostering Little Languages*. Dr. Dobb's Journal (DDJ) March.
- 2004 Findler, Blume, Felleisen, *An Investigation of Contracts as Projections*. University of Chicago Computer Science Technical Report TR-2004-02.
- 2002 Findler, *Behavioral Software Contracts (dissertation)*. Rice University Computer Science Technical Report TR02-402 June.

- 2001 Findler, Latendresse, Felleisen, *Object-oriented Programming Languages Need Well-founded Contracts*. Rice University Computer Science Technical Report TR01-372 January.
- 2000 Findler, Felleisen, *Behavioral Interface Contracts for Java*. Rice University Computer Science Technical Report TR00-366 August.
- 1995 Findler, *Modular Abstract Interpreters*. Undergraduate Senior Thesis.

Keynote Talks

- 11/2016 *Redex: A Language for Lightweight Semantics Engineering* Keynote address, delivered at SLE 2016.
- 7/2016 *Pragmatics of Programming Language Design*. Keynote address, delivered at FLOLAC 2016.
- 7/2015 *Racket: a programming-language programming language*. Keynote address, delivered at Lambda Jam 2015.
- 9/2014 *Behavioral Software Contracts*. Keynote address, delivered at International Conference on Functional Programming (ICFP) 2014.
- 10/2010 *Semantics Engineering: more than just Theorem Proving*. Keynote at the International Workshop on Foundations of Object-Oriented Languages 2010.
- 8/2010 *Contracts in Racket*. Keynote at the Scheme and Functional Programming Workshop 2010.

Invited Talks and Lectures

- 7/2018 *Language-oriented Programming or Racket: the world's second best programming language* Delivered at National Taiwan University.
- 11/2016 *Macros matter: effectively building lots of programming languages*. Delivered at the University of Rochester.
- 8/2014 *Run Your Research*. Delivered at the IAMS & NCTS/TPE Seminar, National Taiwan University.
- 7/2014 *Contracts in Racket*. Delivered at the MFN Seminar, National Taiwan University.
- 7/2014 *Macros Matter*. Delivered at National Taiwan University.
- 4/2013 *Racket: a programming language programming language*. Delivered at the University of Iowa.
- 9/2012 *Macros Matter*. Delivered at Chalmers University, Sweden.

- 7/2012 *Redex in Sinaia*. Delivered at the Summer School on Language Frameworks (SSLF), Romania.
- 6/2012 *Macros matter*. Delivered at Chinese Academy of Sciences (ICT), Beijing China.
- 12/2011 *Contracts in Racket*. Delivered at Academia Sinica, Taiwan.
- 6/2010 *Expanding Parallel Programming to a Wider Audience*. HPDC 2010 panel.
- 2010 *Macros Matter*. Delivered at Flourish 2010.
- 2008 *The First Year*. Delivered at DIKU University of Copenhagen.
- 2007-2008 *DrScheme: why systems building matters*. Delivered at Chalmers Tekniska Högskola, Microsoft Research Cambridge, DIKU University of Copenhagen, Yale University, Harvard University, University of California Los Angeles, University of Washington, University of Illinois at Urbana-Champaign, Northwestern University, Indiana University, Purdue University, and Northeastern University.
- 9/2007 *PLT Redex: a semantics engineer's toolkit*. Delivered at the Scottish Programming Language Seminar, Edinburgh.
- 12/2005 *Programming the Interactive Web*. Delivered at National Taiwan Normal University.
- 7/2005 *Checking Software Properties with Contracts*. Delivered at the Summer School on Reliable Computing, University of Oregon.
- 3/2005 *Aspects and Modular Reasoning*. FOAL 2005 panel.
- 1/2005 *Extreme Contracts*. FOOL 2005 panel.
- 12/2004 *Gaming Games*. Delivered at Daniel P. Friedman's Feschrift, Indiana University.
- 12/2003 *Contracts and Subtyping*. Delivered at National Taiwan Normal University.
- 10/2003 *Late Binding for Software Contracts*. Delivered at Brown University.
- 2002-2003 *Behavioral Software Contracts*. Delivered at Worcester Polytechnic Institute, University of Oklahoma, University of California Santa Cruz, University of Oregon, University of Chicago, University of Indiana, University of Utah, and Universidad Nacional Autónoma de México.
- 2002 *Units*. Delivered at Massachusetts Institute of Technology.
- 2000 *The Architecture of DrScheme*. Delivered at Carnegie Mellon University, John Hopkins University, Brown University, and IBM Watson.

2000 *Java and Contracts*. Delivered at John Hopkins University, Worcester Polytechnic Institute, Boston University, and IBM Watson.

Teaching at Northwestern University

Co-developed *Intensive Program Design* (EECS 295). Taught Fall 2016.

Developed a lecture on undefined behavior for *Intro to Computer Systems* (EECS 213). Co-taught the course, Winter 2016.

Developed *Programming Languages* (EECS 321). Taught Winter 2010, 2011, 2012, 2013, 2014, 2015, Fall 2015.

Developed *Compilers* (EECS 322). Taught Spring 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016.

Led seminars on various PL topics, Winter 2009, Fall 2009, 2010, 2011, 2012, 2013, 2015.

Teaching at the University of Chicago

Developed *Introduction to Computer Programming I* (15100). Taught annually from 2002 through 2008.

Developed *Software Construction* and adapted it to the master's program (22001/51090). Taught in 2003, 2004 (twice), 2006, and 2008.

Taught *Introduction to Computer Systems* (154000), in Spring 2008.

Led PhD seminars on software contracts, type-directed partial evaluation, theorem proving (co-led with Dave MacQueen), and interoperability (co-led with Jacob Matthews).

PhD Students

Shu-Hung You. Started in 2016.

Spencer Florence. Started in 2015.

Dan Feltey. Started in 2015.

2015 Burke Fetscher *Automated Testing for Operational Semantics*, defended 9/17/2015. Now at Google.

2014 James Swaine *Incremental Parallelization of Existing Sequential Runtime Systems*, defended April 23, 2014. Now at Braintree.

- 2104 Giang Hoang *Code Generation for Timing Speculative Architectures*, defended May 15, 2014. Co-advised with Russ Joseph. Now at NVIDIA.
- 2007 Jacob Matthews *The Meaning of Multilanguage Programs*, defended December 4, 2007 (University of Chicago). Now at Google.

Master's Students Advised

- 2014 Max New: *Enumerating Countable Sets for Property-Based Testing*
- 2012 Yixi Zhang: *Investigation of Generational Incremental Garbage Collector*
- 2012 John Greene: *An Operation Semantics for Datalog*
- 2012 Andy Gocke: *Random Testing of Functional Programs Using Contracts*
- 2009 James Swaine: *Scheme with Futures: Incremental Parallelization in a Language Virtual Machine*
- 2009 Casey Klein *Experience with Randomized Testing in Programming Languages Metatheory* (University of Chicago).
- 2005 Jacob Matthews *Operational Semantics for Scheme via Term Rewriting* (University of Chicago).

PhD Committee Member

- 2007 Jing (George) Cao: *Introspection in Dynamically Linked Applications* (University of Chicago)
- 2006 Philippe Meunier: *Modular Set-Based Analysis from Contracts* (Northeastern University)
- 2005 Songyan Feng: *Implementation of Contracts in Wrapper Generation Tools / SWIG* (University of Chicago)

Master's Degree Committee Member

- 2007 George Kuan: *A Rewriting Semantics for Type Inference* (University of Chicago)
- 2005 Xingqi Xiao: *Toward Optimization of Concurrent ML* (University of Chicago)
- 2004 Jonathan Riehl: *Grammar Based Unit Testing for Parsers* (University of Chicago)
- 2003 Jing Cao: *Building a Scripting Interface to a Scripting Interface Generator* (University of Chicago)

Professional Activities

- 2018 General Chair for the International Conference on Functional Programming (ICFP) 2018.
- 2016 Organized the CS Education Day at Northwestern.
- 2016 Co-organized the Programming Languages Mentoring Workshop (PLMW@ICFP) 2016.
- 2016 Program Committee Member for Object Oriented Programming, Systems, Languages and Applications (OOPSLA) 2016.
- 2015 Artifact Evaluation Committee (AEC) co-chair for OOPSLA 2015.
- 2015 Co-organized a week-long PLT Redex workshop, in Salt-Lake City, Utah. 28 attendees (not including 6 staff).
- 2015–now Member of the IFIP Working Group 2.8 on Functional Programming.
- 2015 Program Committee Member for Programming Language Design and Implementation (PLDI) 2015.
- 2014 Artifact Evaluation Committee (AEC) co-chair for OOPSLA 2014.
- 2014 Program Committee Member for Principles of Programming Languages (POPL) 2015.
- 2014 Selection Committee Member for the 2014 SIGPLAN Achievement Award
- 2014 Program Committee Member for the European Conference on Object-Oriented Programming (ECOOP) 2014.
- 2013 Selection Committee Member for the Most Influential ICFP Paper Award for ICFP 2003.
- 2013 Program Committee Member for the International Workshop on the Foundations of Object-Oriented Languages (FOOL) 2013.
- 2013 Program Committee Member for the Dynamic Languages Symposium (DLS) 2013.
- 2013 Program Committee Member for the International Conference on Advanced Parallel Processing Technology (APPT) 2013.
- 2013 Program Committee Member for Onward! 2013
- 2013 Guest JFP editor for special issue on ICFP 2012
- 2012 Program Committee Chair for the International Conference on Functional Programming (ICFP) 2012.
- 2012–now CACM Research Highlights Nomination Committee member.

- 2011–now Founding member of IFIP Working Group 2.16 on Programming Language Design
- 2011 Program Committee Member for Programming Language Design and Implementation (PLDI) 2012.
- 2011 Program Committee Member for the K Workshop 2011.
- 2011 External Review Committee Member for Principles of Programming Languages (POPL) 2012.
- 2011 Program Committee Member for Trends in Functional Programming (TFP) 2011.
- 2011 External Review Committee Member for Object Oriented Programming, Systems, Languages and Applications (OOPSLA) 2011.
- 2011 Program Committee Member for Workshop on Free Composition (FREECO) 2011.
- 2011 Program Chair for the Scripts to Programs (STOP) workshop 2011.
- 2010 Program Committee Member for the Dynamic Languages Symposium (DLS) 2010.
- 2010 Program Committee Member for the International Conference on Functional Programming (ICFP) 2010.
- 2009 Program Committee Member for 21st Symposium on Implementation and Application of Functional Languages (IFL) 2009.
- 2009 Program Committee Member for Object Oriented Programming, Systems, Languages and Applications (OOPSLA) 2009.
- 2008 Program Committee Member for the Workshop on ML 2008.
- 2007 Organized the first PLT Redex workshop, in Chicago. 31 attendees.
- 2007–now Member of the ICFP Steering Committee.
- 2019–now Journal of Functional Programming (JFP) Editorial Board Member (ending December 31st, 2021)
- 2007–2017 Journal of Functional Programming (JFP) Editorial Board Member.
- 2004–2012 Scheme and Functional Programming Workshop (SFP) Steering Committee Member.
- 2008 Program Committee Member for European Conference on Object-Oriented Programming (ECOOP) 2008.
- 2007 Program Committee Member for Principles of Programming Languages (POPL) 2008.

- 2006 Organizer and Program Committee Chair for the Scheme and Functional Programming Workshop (SFP) 2006.
 - 2006 Program Committee Member for Object-Oriented Programming, Languages, Systems, and Applications (OOPSLA) 2006.
 - 2006 Program Committee Member for the International Conference on Functional Programming (ICFP) 2006.
 - 2006 Program Committee Member for Foundations of Aspect-Oriented Languages (FOAL) 2006.
 - 2004–2005 Co-organizer for Functional and Declarative Programming in Education (FDPE) 2005.
 - 2004–2005 Program Committee Member for Practical Aspects of Declarative Languages (PADL) 2005.
 - 2004 Judge and on-site coach for ACM Programming Contest.
 - 2003 Program Committee Member for the Scheme Workshop 2003.
 - 2003 Co-led 2003 SIGCSE workshop *TeachScheme!: An Innovative Approach to Introductory Programming*.
 - 2003 Program Committee Member for the International Conference on Functional Programming (ICFP) 2003.
- Served on NSF panels.

Internal Service

- F2016–now Member of the CS + X Opportunities Committee
- F2012–S2016 Member of the Distinguished Seminar Committee
- F2011–S2016 Chair of the CS Curriculum Committee
- W2009–now Member of the CS Curriculum Committee
- W2009–now Member of the CE Curriculum Committee

Numbers

- McCarthy number: 3 (Matthias Felleisen, Robert Cartwright, John McCarthy)
- Erdős number: 4 (Mitch Wand, Albert Meyer, Daniel J. Kleitman, Paul Erdős)
- Knuth number: 4 (Eli Barzilay, Robert L. Constable, Robert W. Floyd, Donald E. Knuth)
- Simon number: 4 (Matthew Flatt, Olin Shivers, Allen Newell, Herbert A. Simon)

Church number: 8 (Shriram Krishnamurthi, Kim B. Bruce, H. Jerome Keisler, K. Jon Barwise, Robert L. Vaught, William Craig, Willard Van Orman Quine, Alonzo Church)

ICFP Contest

2005 The International Conference on Functional Programming has conducted a programming contest since 1998, open to any programmer using any programming language. Most years, the contest was a regular, algorithm-oriented challenge.

In 2005, I conducted the first contest that rewarded the programmer's ability to design maintainable code. The contest was organized in two parts. The first was like an ordinary year: a single task to be completed over three days. The second task was a variation on the first, but contestants had only a single day to complete it. Well-design programs from the first phase were easy to adapt, giving their authors time to focus on exploiting the changes to the specification, but the authors of poorly designed programs spent the entire day adapting their programs to be able to follow the basics of the new specification.

The contest was featured on slashdot and attracted 370 contestants in 161 teams from 26 countries on 5 continents. I led a team of 16 people from six institutions over a period of 16 months to prepare for and determine the winners of the contest. We wrote 30k lines of code in support of the contest, and used 5.5 months of CPU time determining the winners. Personally, I spent more than the equivalent of 6 months full-time on the contest and read more than 2,400 pages of code to determine the winner of the Judge's prize.

For more details, see <http://icfpc.racket-lang.org>.

Other Activities

Chinese language studies.