Richard A. Eisenberg

rae@richarde.dev https://richarde.dev/

Research Interests

Programming languages, strongly typed functional programming, Haskell, OCaml. I want to make programs correct and elegant by construction.

Post-PhD Employment

²⁰²²-pres.</sup> Software Engineer, OCaml Language team, Jane Street Capital

^{2020–22} Principal Researcher, Tweag

^{2016–20} Assistant Professor of Computer Science, Bryn Mawr College

Education

Sep 2016 PhD in Computer & Information Science, University of Pennsylvania, Philadelphia, PA Dissertation: Dependent Types in Haskell: Theory and Practice Advisor: Stephanie Weirich

Harvard University, Cambridge, MA

- June 2003 M.S., Computer Science
- June 2003 B.A., Physics, magna cum laude with highest honors

Publications

ICFP'22 28) A. Spiwack, C. Kiss, J.-P. Bernardy, N. Wu, and **R. A. Eisenberg**. *Linearly qualified types:*

(acceptance generic inference for capabilities and uniqueness. In Proceedings of the ACM on rate: 34%) Brogramming Languages Viol. (Classic ICEP 122). ACM, 2022, Article 05, pp. 127, 104

- Programming Languages, Vol. 6, Issue ICFP (ICFP '22), ACM, 2022. Article 95, pp. 137-164.
- POPL'22 27) F. Krawiec, S. Peyton Jones, N. Krishnaswami, T. Ellis, R. A. Eisenberg, and A.
 ^(23%) Fitzgibbon. Provably Correct, Asymptotically Efficient, Higher-Order Reverse-Mode Automatic Differentiation. In Proceedings of the ACM on Programming Languages, Vol. 6, Issue POPL (POPL '22), ACM, 2022. Article 48, 30 pages.
- ICFP'21 26) R. A. Eisenberg, G. Duboc, S. Weirich, and D. Lee. An Existential Crisis Resolved: Type (33%) Inference for First-Class Existential Types. In Proceedings of the ACM on Programming Languages, Vol. 5, Issue ICFP (ICFP '21), ACM, 2021. Article 64, 29 pages.
- Haskell'21 25) G.-J. Bottu and R. A. Eisenberg. Seeking Stability by Being Lazy and Shallow: Lazy and (40%) Shallow Instantiation is User Friendly. In Proceedings of the 2021 ACM SIGPLAN Symposium on Haskell (Haskell '21), ACM, 2021. pp. 85-97.
 - POPL'21 24) P. Choudhury, H. Eades III, R. A. Eisenberg, and S. Weirich. A Graded Dependent Type (24%) System with a Usage-Aware Semantics. In Proceedings of the ACM on Programming Languages, Vol. 5, Issue POPL (POPL '21), ACM, 2021. Article 50, 32 pages.
 - ICFP'20 23) P. Downen, Z. M. Ariola, S. Peyton Jones, R. A. Eisenberg. Kinds are Calling Conventions.
 (37%) In Proceedings of the ACM on Programming Languages, Vol. 4, Issue ICFP (ICFP '20), ACM, 2020. Article 104, 29 pages.
- Haskell'20 22) R. A. Eisenberg. Stitch: The Sound Type-Indexed Type Checker (Functional Pearl). In (50%) Proceedings of the 2020 ACM SIGPLAN Symposium on Haskell (Haskell '20), ACM, 2020. pp. 39-53.

GitHub: @goldfirere

- Haskell'20 21) Y. Parès, J.-P. Bernardy, R. A. Eisenberg. Composing Effects into Tasks and Workflows. In (50%) Proceedings of the 2020 ACM SIGPLAN Symposium on Haskell (Haskell '20), ACM, 2020. pp. 80-94.
 - POPL'20 20) N. Xie, R. A. Eisenberg, B. C. d. S. Oliveira. Kind Inference for Datatypes. In Proceedings of the ACM on Programming Languages, Vol. 4, Issue POPL (POPL '20), ACM, 2020. Article 53, 28 pages.
 - POPL'20 19) M. P. Jones, J. G. Morris, R. A. Eisenberg. Partial Type Constructors; Or, Making ad hoc (29%) datatypes less ad hoc. In Proceedings of the ACM on Programming Languages, Vol. 4, Issue POPL (POPL '20), ACM, 2020. Article 40, 28 pages.
 - ICFP'19 18) S. Weirich, P. Choudhury, A. Voizard, R. A. Eisenberg. A Role for Dependent Types in (32%) Haskell. In Proceedings of the ACM on Programming Languages, Vol. 3, Issue ICFP (ICFP '19), ACM, 2019. Article 101, 29 pages.
- Haskell'18 17) R. A. Eisenberg, J. Breitner, and S. Peyton Jones. *Type Variables in Patterns*. In
 (43%) Proceedings of the 2018 ACM SIGPLAN Symposium on Haskell (Haskell '18), ACM, 2018, pp. 94-105.
- Haskell'18 16) D. Otwani and R. A. Eisenberg. The Thoralf Plugin: For Your Fancy Type Needs. In
 (43%) Proceedings of the 2018 ACM SIGPLAN Symposium on Haskell (Haskell '18), ACM, 2018, pp. 106-118.
 - ^{ICFP'17} 15) J. G. Morris and **R. A. Eisenberg**. *Constrained Type Families*. In *Proceedings of the ACM* (35%) *on Programming Languages*, Vol. 1, Issue ICFP (ICFP '17), ACM, 2017. Article 42, 28 pages.
 - ICFP'17 14) S. Weirich, A. Voizard, P. H. Azevedo de Amorim, R. A. Eisenberg. A Specification for (35%) Dependently-Typed Haskell. In Proceedings of the 22nd ACM SIGPLAN International Conference on Functional Programming (ICFP '17), ACM, 2017. Article 31, 29 pages.
 - PLDI'17 13) R. A. Eisenberg, S. Peyton Jones. Levity Polymorphism. In Proceedings of the 2017 ACM (14%) SIGPLAN Conference on Programming Language Design and Implementation (PLDI '17). ACM, 2017. pp. 525-539.
- Haskell'16 12) M. Pickering, G. Érdi, S. Peyton Jones, **R. A. Eisenberg**. *Pattern Synonyms*. In *Proceedings* of the 2016 ACM SIGPLAN Symposium on Haskell (Haskell '16), ACM, 2016. pp. 80-91.
 - JFP'16 11) J. Breitner, R. A. Eisenberg, S. Peyton Jones, S. Weirich. Safe Zero-cost Coercions for Haskell. In Journal of Functional Programming, Vol. 26. Cambridge University Press, 2016. 79 pages.
 - ESOP'16 10) R. A. Eisenberg, S. Weirich, H. Ahmed. Visible Type Application. In Programming (33%) Languages and Systems: 25th European Symposium on Programming (ESOP '16). LNCS 9632, Springer, 2016. pp. 229-254.
- Wadlerfest 9) S. Peyton Jones, S. Weirich, R. A. Eisenberg, D. Vytiniotis. A reflection on types. In A List of Successes that Can Change the World, a festschrift in honor of Phil Wadler. LNCS 9600, Springer, 2016. pp. 292-317.
- Haskell'15 8) J. Stolarek, S. Peyton Jones, R. A. Eisenberg. Injective Type Families for Haskell. In
 (39%) Proceedings of the 2015 ACM SIGPLAN Symposium on Haskell (Haskell '15), ACM, 2015.
 pp. 118-128.
- Haskell'14 7) T. Muranushi, R. A. Eisenberg. Experience Report: Type-checking Polymorphic Units for
 (43%) Astrophysics Research in Haskell. In Proceedings of the 2014 ACM SIGPLAN Symposium on Haskell (Haskell '14), ACM, 2014. pp. 31-38.
- Haskell'14 6) R. A. Eisenberg, J. Stolarek. Promoting Functions to Type Families in Haskell. In
 (43%) Proceedings of the 2014 ACM SIGPLAN Symposium on Haskell (Haskell '14), ACM, 2014.

May 2024

pp. 95-106.

- ICFP'14 5) J. Breitner, R. A. Eisenberg, S. Peyton Jones, S. Weirich. Safe Zero-cost Coercions for
 (33%) Haskell. In Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming (ICFP '14), ACM, 2014. pp. 189-202.
- POPL'14 4) R. A. Eisenberg, D. Vytiniotis, S. Peyton Jones, S. Weirich. Closed Type Families with (23%) Overlapping Equations. In Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '14), ACM, 2014. pp. 671-683.
- OOPSLA'13 3) C. DeLozier, R. A. Eisenberg, S. Nagarakatte, P.-M. Osera, M. M. K. Martin, and S.
 (26%) Zdancewic. Ironclad C++: A Library-Augmented Type-Safe Subset of C++. In Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming, Systems, Languages, & Applications (OOPSLA '13), ACM, 2013. pp. 287-304.
 - ICFP'13 2) S. Weirich, J. Hsu, and R. A. Eisenberg. System FC with Explicit Kind Equality. In
 - (30%) (2) Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming (ICFP '13), ACM, 2013. pp. 275-286.
- Haskell'12 1) R. A. Eisenberg and S. Weirich. Dependently Typed Programming with Singletons. In (41%) Proceedings of the 2012 Haskell Symposium (Haskell '12), ACM, 2012. pp. 117-130.

Technical Reports

- 2017 R. A. Eisenberg and S. Peyton Jones. Levity Polymorphism (extended version). Bryn Mawr Technical Report MS-1079, 2017.
- 2015 R. A. Eisenberg. System FC, as implemented in GHC. University of Pennsylvania Technical Report MS-CIS-15-09, 2015.
- ²⁰¹⁵ **R. A. Eisenberg**. An Overabundance of Equality: Implementing Kind Equalities into Haskell. University of Pennsylvania Technical Report MS-CIS-15-10, 2015.
- ²⁰¹⁵ J. Stolarek, S. Peyton Jones, **R. A. Eisenberg**. *Injective Type Families for Haskell (extended version)*. Politechnika Łódzka Technical Report, 2015.
- ²⁰¹⁴ **R. A. Eisenberg**, J. Stolarek. *Promoting Functions to Type Families in Haskell (extended version)*. University of Pennsylvania Technical Report MS-CIS-14-09, 2014.
- ²⁰¹⁴ J. Breitner, **R. A. Eisenberg**, S. Peyton Jones, S. Weirich. *Safe Zero-cost Coercions for Haskell* (extended version). University of Pennsylvania Technical Report MS-CIS-14-07, 2014.
- ²⁰¹³ R. A. Eisenberg, D. Vytiniotis, S. Peyton Jones, S. Weirich. *Closed Type Families with Overlapping Equations (extended version)*. University of Pennsylvania Technical Report MS-CIS-13-10, 2013.
- ²⁰¹³ P.-M. Osera, R. A. Eisenberg, C. DeLozier, S. Nagarakatte, M. M. K. Martin, S. Zdancewic. *Core Ironclad.* University of Pennsylvania Technical Report MS-CIS-13-06, 2013.
- ²⁰¹³ S. Weirich, J. Hsu, **R. A. Eisenberg**. *System FC with Explicit Kind Equality (extended version)*, University of Pennsylvania Technical Report MS-CIS-15-11, 2013.

Research-related open-source contributions

GHC core Core developer for the <u>Glasgow Haskell Compiler</u> (GHC), the main compiler for the <u>Haskell</u> developer functional programming language. Principal contributions:

Levity polymorphism, based on *Levity Polymorphism* (PLDI'17)

Kind equalities, based on Kind Equalities (ICFP'13)

Visible type application, based on Visible Type Applications (ESOP'16)

New solver for type equality, based on *Safe Zero-cost Coercions* (JFP'16) Roles, based on *Safe Zero-cost Coercions* (ICFP'14) Closed type families, based on *Closed Type Families* (POPL'14)

Haskell The *singletons* package, described in the Haskell'12 and Haskell'14 papers.

packages The units package, described in the Haskell'14 experience report.

The stitch package, described in the Haskell'20 functional pearl.

Presentations

- Sep 2023 Layout Polymorphism. ML Workshop, Seattle, WA, USA
- Apr 2023 Layout Polymorphism. IFIP Working Group 2.8, Cambridge, UK
- Sep 2022 Unboxed Types for Ocaml. ML Workshop, Ljubljana, Slovenia
- Sep 2022 Eiger: Auditable, executable, flexible legal regulation. Haskell Symp., Ljubljana, Slovenia
- Oct 2020 @rae's Thoughts on Haskell. Weekly YouTube series, <u>online</u>.
- Aug 2022
- May 2022 What Automatic Differentiation Really Means. IFIP Working Group 2.8, New York, USA
- Dec 2021 First-class existential types and their application to ergonomic programming with linear types. Invited lecture, University of Iowa, online
- Aug 2021 An Existential Crisis Resolved. ICFP, online
- Feb 2021 An Existential Crisis Resolved. IFIP Working Group 2.8, online
- Nov 2020 Simplifying Constraint Solving in GHC. Haskell eXchange, online
- Sep 2020 Partial Type Constructors. Invited talk, MuniHac, online
- Aug 2020 Stitch: the Type-Indexed Type Checker (Functional Pearl). Haskell Symp., online
- Aug 2020 Parameters of Many Flavors. Invited talk, Haskell Love, online
- June 2020 How to Extend GHC. GHC Track talk, ZuriHac, online
- May 2020 Partial Type Constructors. Invited lecture, University of Nottingham, online
- Nov 2019 Partial Type Constructors. Invited lecture, University of Cambridge, Cambridge, UK
- Oct 2019 Stitch: the Type Indexed Type-Checker. Haskell eXchange, London, UK
- June 2019 Stitch: the Type Indexed Type-Checker. Invited keynote, ZuriHac, Zürich, Switzerland
- June 2019 Tour of the GHC Front End. GHC Track talk, ZuriHac, Zürich, Switzerland
- May 2019 Generalized Newtype Compiling: Don't let you types slow you down. IFIP Working Group 2.8, Bordeaux, France
- Sep 2018 Type Variables in Patterns. Haskell Symp., St. Louis, MO, USA
- Apr 2018 Stitch: the Type Indexed Type-Checker. New York Haskell Users' Group, USA
- Sep 2017 Constrained Type Families. ICFP, Oxford, UK
- Jun 2017 Levity Polymorphism. PLDI, Barcelona, Spain
- Sep 2016 A Dependent Haskell Triptych. Haskell Implementors' Workshop, Nara, Japan
- Apr 2016 Visible Type Application. European Symposium on Programming, Eindhoven, Netherlands
- Aug 2015 Levity Polymorphism in Dependent Haskell. Haskell Implementors' Workshop, Vancouver, Canada
- May 2015 A Practical Introduction to Haskell GADTs. Invited speaker, LambdaConf, Boulder, CO, USA
- Oct 2014 Dependent Types for Haskell. Invited speaker, New York City Haskell Users' Group, USA

- Sep 2014 Dependent Haskell. Haskell Implementors' Workshop, Gothenburg, Sweden
- Sep 2014 Safe Zero-cost Coercions for Haskell. ICFP, Gothenburg, Sweden
- Jan 2014 Closed Type Families with Overlapping Equations. POPL, San Diego, CA, USA
- Sep 2013 System FC with Explicit Kind Equality. ICFP, Boston, MA, USA
- Sep 2013 GeneralizedNewtypeDeriving is now Type-safe: How Roles Save the Day. Haskell Implementors' Workshop, Boston, MA, USA
- Sep 2012 Dependently Typed Programming with Singletons. Haskell Symp., Copenhagen, Denmark

Funding

²⁰¹⁷ SHF: MEDIUM: Collaborative Research: The Theory and Practice of Dependent Types in Haskell. R. A. Eisenberg and S. Weirich (PIs). NSF 1704041, \$949,964, 7/2017-7/2021.

Honors and Awards

- 2022 Permanent Member, IFIP Working Group 2.8 on Functional Programming An invitation-only group of top functional programming researchers who meet yearly to present latest advances.
- ²⁰²¹ Distinguished Paper Award for An Existential Crisis Resolved, ICFP'21 Awarded by an additional peer-review process to fewer than 10% of papers accepted at the conference
- ²⁰²⁰ Full Member, Sigma Xi Honor Society
- ²⁰²⁰ Distinguished Paper Award for Kind Inference for Datatypes, POPL'20 Awarded by an additional peer-review process to fewer than 10% of papers accepted at the conference
- ²⁰¹⁷ Morris and Dorothy Rubinoff Award, U. of Penn. Awarded to a doctoral candidate whose dissertation may lead to innovative applications of computer technology
- 2016 Penn Prize for Excellence in Teaching by Graduate Students, U. of Penn. One of 10 graduate students across the university recognized for our teaching, nominated by undergraduates
- 2014-16 Graduate Student Fellowship, Microsoft Research One of 12 doctoral students chosen among candidates from U.S. and Canada.
- ²⁰¹³⁻¹⁴ Fellowship for Teaching Excellence, U. of Penn. Center for Teaching and Learning Nominated & selected as the graduate student departmental advocate for teaching.
- ²⁰¹¹⁻¹² John Henry Towne Fellowship, U. of Penn. School of Engineering & Applied Science Awarded to exceptionally qualified first-year doctoral students.
- 2002, 2003 John Harvard Scholarship Awarded to undergraduates in the top 5% of their class
 - ²⁰⁰¹ Harvard College Scholarship Awarded to undergraduates in the top 10% of their class

Non-profit Work

Feb 2021 - Chair of the Board of Directors: Haskell Foundation present The Haskell Foundation is a non-profit entity working to promote Haskell and functional programming, by attracting donations and funding projects of need in core Haskell infrastructure.

Pre-PhD Professional Experience

- Dec 2015 Consultant, Awake Networks, Mountain View, CA. My task was to help design an efficient Mar 2016 domain-specific language embedded in Haskell to be used in a networking security product.
- Summer '13 Research Intern, Microsoft Research, Cambridge, UK. Mentored by Simon Peyton Jones.

Summer '02 Software Design Engineer Intern, Microsoft, Redmond, WA

Summer '01 Software Engineer Intern, Actuality Systems, Inc., Reading, MA

Research Advising

Tweag G. Duboc. *Existential Quantification*. Tweag Internship, 2020-21.
G. Rudoy. *Compiling Refinement Types to Idris*. Tweag Open Source Fellow, 2020.
G.-J. Bottu. *Type Instantiation*. Tweag Internship, 2020.

Undergrad E. Feng. Verification of Dijkstra's Algorithm in Idris. Bryn Mawr College, 2019.

advisor S. Depew. Visualizing Algorithms with Android. Bryn Mawr College, 2019.

K. J.-C. Liao. *I Am The Senate: Introducing* Palpatine, a Vote Counter for Australian Senate Ballots written in Idris with Verification of Totality and Cardinality. Haverford College, 2018.
R. Xu. Comparison Between Program Verification Techniques in Dependent Haskell and Liquid Haskell. Bryn Mawr College, 2018.

X. Zhang. A Tale of Two Provers: A Comparison of Dependent Haskell and F*. Bryn Mawr College, 2018.

J. Henck. A Supercompiler for an Object-Oriented Language. Bryn Mawr College, 2017.

Undergrad M. Nguyen. *Types and Programming Languages / Type System Implementation*. Fall 2018. Ind. Study advisor M. Yacavone. *Cubical Type Theories*. Fall 2018.

E. Feng. Verifying Functional Algorithms in Idris. Fall 2018.

D. Otwani. Type-Level Finite Maps. Fall 2017 – Spring 2018.

R. Xu and X. Zhang. Verification Languages. Fall 2017.

R. Xu and X. Zhang. Types and Programming Languages. Spring 2017.

M. Yacavone. Ind. Study: Homotopy Type Theory in Coq. Spring 2017.

Summer M. Nguyen. Visible Kind Application. Summer 2018.

science research N. Adnane. Improving GHC Error Messages. Summer 2018.

advisor E. Feng and P. Thiel. Merging Term and Type Parsers. Summer 2018.

M. Yacavone. Explicit Variable Quantification. Summer 2018.

Teaching

Higher Education

Instructor, CMSC 110: Introduction to Computing (27 undergrads), fall 2016

Bryn Mawr College CMSC 113: Computer Science I (avg. 26 undergrads), fall 2017, spring 2018

CMSC 206: Introduction to Data Structures (26 undergrads), spring 2018

CMSC 231: Discrete Mathematics (21 undergrads), fall 2017

CMSC 245: Principles of Programming Languages (avg. 25 undergrads), fall 2018 (x2)

CMSC 246: Programming Paradigms (32 undergrads), spring 2017

CMSC 350: Compiler Design (16 undergrads), spring 2019

CMSC 380: Modern Functional Programming (24 undergrads), spring 2017

CMSC 399: Senior Conference (11 seniors), spring 2019

Instructor, CIS194: Haskell Programming (12 students, mostly undergrad), fall 2014

U. of Penn. CIS190: C++ Programming (19 students, mostly undergrad), fall 2012

Head TA CS50: Introduction to Computer Science (~100 undergrads), fall 2002, Harvard U.

TA CIS552: Advanced Programming (~40 students, grad & undergrad), fall 2013, U. of Penn.
 CIS120: Programming Languages & Techniques (~100 undergrads), spring 2013, U. of Penn.
 CS50: Introduction to Computer Science (~200 undergrads), fall 2001, Harvard U.
 CS50: Introduction to Computer Science (~300 undergrads), fall 2000, Harvard U.

High school

²⁰⁰⁸⁻¹¹ The American School in London, London, UK

The American School in London is a private, co-educational K-12 school with an American curriculum.

Taught computer science and math, including AP Computer Science in Java

Mentored high-school and middle-school FIRST robotics teams

Created new Digital Electronics course from scratch, including outfitting the lab

2003-08 Northfield Mount Hermon School, Mount Hermon, MA

Northfield Mount Hermon is a private, co-educational boarding high school.

Taught computer science and math, including AP Computer Science in Java

Dorm parent (3 years) and director (2 years), overseeing 40 students. Received the Parents Council Award for Excellence in Residential Life (2005).

Professional Activities

- Sep 2024 Program Committee member: ML Workshop
- Sep 2024 Program Committee member: ICFP
- Jan 2024 Organizer: Workshop on the Implementation of Type Systems (WITS)
- Dec 2023 Organizer: Jane Street Programming Languages Colloquium Regional gathering of programming language PhD students and advisors to present recent work
- Aug 2023 Program Committee member: WITS
- Jan 2023 Program Committee member: Principles of Programming Languages Conference (POPL)
- Sep '22-'26 Steering Committee member at large: ICFP
 - Jan 2022 Co-organizer: Workshop on the Implementation of Type Systems Created this new workshop, now co-located with POPL.
 - Aug 2021 Program Committee member: ML Family Workshop
 - Nov 2020 Member: Haskell Foundation Working Group
- Nov 2020 Program Committee member: Haskell eXchange
- Aug 2020 Program Committee member: Int'l Conference on Functional Programming (ICFP)
- Aug 2019 Program Committee chair: Haskell Symposium
- Aug 2019 External Review Committee member: ICFP
- Jun 2019 Invited speaker: ZuriHac, Zurich, Switzerland.
- May 2019 Invited guest/speaker: IFIP Working Group 2.8, Bordeaux, France
- Sep 2018 Program Committee co-chair: Workshop on Type-Directed Development (TyDe)
- Sep 2018 Distinguished Papers Committee member: ICFP

- Sep 2018 External Review Committee member: ICFP
- Sep 2018 Program Committee member: Implementation of Functional Languages (IFL)
- Sep 2017 Invited Panelist: Careers in Programming Languages, Prog. Lang. Mentoring Workshop
- Sep 2017 Program Chair: Haskell Implementors' Workshop
- May 2017 Organizer: GHC implementation workshop, a 2-week gathering of compiler writers
- Jan 2017 Member: GHC Steering Committee, for reviewing proposals for updates to GHC present
- Sep 2016 Program Committee member: Haskell Symposium
- Sep 2016 Program Committee member: TyDe
- Mar 2016 Invited participant: Dagstuhl Seminar Language Based Verification Tools for Functional Programs, Wadern, Germany
- Nov 2015 Member: Haskell Prime Committee, for updating the standard for the Haskell language present
- Nov 2015, Organizer: Hac Phi, a yearly weekend-long Haskell exchange, Philadelphia, PA Oct 2014
- Oct 2015 Awardee: Center for Teaching & Learning Teaching Certificate, U. of Penn. Earning this certificate requires participation in ongoing conversations about teaching and participating in a teaching observation & reflection.
- Aug 2015 Program Committee member: Haskell Implementors' Workshop, Vancouver, Canada

Outreach

- Mar 2020 Co-organize twice-yearly Tweag Fellowship program, accepting applications from global Aug 2022 contributors to open-source projects for full-time funding
 - ^{Summer} Google Summer of Code mentor, supervising the start of implementation of a dependently-²⁰¹⁸ typed core language in GHC, performed by Ningning Xie of the University of Hong Kong
 - Aug 2015 Organized and ran *Stencyl Boot Camp*, an introduction to programming for middle- and high-school students, West Tisbury, MA
 - Apr 2014 Led workshop to high school students on introduction to programming with Scratch for Women in Computer Science Day, U. of Penn.
 - Nov 2013 Presented introduction to Haskell at the Charter School of Wilmington, Wilmington, DE
 - Feb 2013 Consulted with educators at Merion Mercy Academy, a Catholic girls' high school, about starting a computer science program, Merion Station, PA
 - Feb 2013 Volunteered as Pit Coordinator at FIRST LEGO League regional championship, Philadelphia, PA