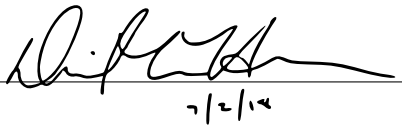
Curriculum Vitae

Notarization. I have read the following and certify that this *curriculum vitae* is a current and accurate statement of my professional record.

Signature  Date 7/2/14

I. Personal Information

UID, Last Name, First Name, Middle Name, Contact Information

UID: 113371332

Last name: Van Horn

First name: David

Middle name: Alexander

3439 A.V. Williams Building University of Maryland College Park, MD 20742

dvanhorn@cs.umd.edu

http://www.cs.umd.edu/~dvanhorn/

Academic Appointments at UMD

Assistant Professor, Computer Science, December 15, 2013 – present

Assistant Professor, UMIACS, December 15, 2013 – present

Other Employment

- Research Assistant Professor, College of Computer and Information Science, Northeastern University, September 1, 2012 – December 14, 2013
- Visiting Assistant Professor, College of Computer and Information Science, Northeastern University, September 1, 2011 – August 31, 2012
- CRA Computing Innovation Post-doctoral Fellow, College of Computer and Information Science, Northeastern University, September 1, 2009 – August 31, 2011
- Visiting Lecturer, College of Computer and Information Science, Northeastern University, September 1, 2007 – August 31, 2009

Educational Background

Ph.D. in Computer Science, 2009, Brandeis University

M.S. in Computer Science, 2006, University of Vermont

B.S. in Computer Science & Information Systems, 2003, University of Vermont

II. Research, Scholarly and Creative Activities

Names of student co-authors are annotated with superscripts to indicate being from UMD (m), NEU (n), or other (o) institutions; as being undergraduate (u), graduate (g), or post-doctoral (p) students at the time of publication. Advisees (at the time of publication) are indicated in **bold**.

Books

Books Authored

- *Realm of Racket: Learn to Program, One Game at a Time!*, Matthias Felleisen, David Van Horn, Conrad Barski, M.D., and Eight Students of Northeastern University[u,n], June 2013, 312 pp. ISBN: 978-1-59327-491-7, No Starch Press.

Articles in Refereed Journals and Conference Proceedings

- Gradual Liquid Type Inference, **Niki Vazou**[m,p], Éric Tanter, and David Van Horn. The ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'18), Boston, Massachusetts, November 2018. To appear.
- Constructive Galois Connections, **David Darais**[m,g] and David Van Horn. *Journal of Functional Programming*. To appear, 2018. (Special Issue for ICFP'16.)
- Soft Contract Verification for Higher-order Stateful Programs, **Phúc C. Nguyễn**[m,g], **Thomas Gilray**[m,p], Sam Tobin-Hochstadt, and David Van Horn. The 45th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL'18), Los Angeles, USA, January 2018.
  `https://doi.org/10.1145/3158139`
- Abstracting Definitional Interpreters (Functional Pearl), **David Darais**[m,g], **Phúc C. Nguyễn**[m,g], **Nicholas Labich**[m,g], and David Van Horn. The ACM SIGPLAN International Conference on Functional Programming (ICFP'17), Oxford, UK, September 2017.
  `https://doi.org/10.1145/3110256`
- Higher-Order Symbolic Execution for Contract Verification and Refutation, **Phúc C. Nguyễn**[m,g], Sam Tobin-Hochstadt, and David Van Horn. *Journal of Functional Programming*, 27, January 2017. (Special Issue for ICFP'14.)
  `https://doi.org/10.1017/S0956796816000216`
- A Vision for Online Verification-Validation, Matthew A. Hammer, Bor-Yuh Evan Chang, and David Van Horn. The 15th International Conference on Generative Programming: Concepts & Experience (GPCE'16), Amsterdam, Netherlands, November 2016.
  `https://doi.org/10.1145/2993236.2993255`
- Constructive Galois Connections: Taming the Galois Connection Framework for Mechanized Metatheory, **David Darais**[m,g] and David Van Horn. The ACM SIGPLAN International Conference on Functional Programming (ICFP'16), Nara, Japan, September 2016.
  `https://doi.org/10.1145/2951913.2951934`

- Pushdown Control-Flow Analysis for Free, Thomas Gilray[o,g], Steven Lyde[o,g], Michael D. Adams[o,p], and Matthew Might. The 43rd ACM SIGPLAN-SIGACT Symposium on Principles in Programming Languages (POPL'16), St. Petersburg, Florida, January 2016.
  https://doi.org/10.1145/2837614.2837631

- Incremental Computation with Names, **Matthew A. Hammer**[m,p], Joshua Dunfield, Kyle Headley[m,u], **Nicholas Labich**[m,g], Jeffrey S. Foster, and Michael Hicks. The ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'15), Pittsburgh, Pennsylvania, October 2015.
  https://doi.org/10.1145/2814270.2814305

- Galois Transformers and Modular Abstract Interpreters, **David Darais**[m,g], Matthew Might, and David Van Horn. The ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'15), Pittsburgh, Pennsylvania, October 2015.
  https://doi.org/10.1145/2858965.2814308

- Abstracting Abstract Control, **Dionna A. Glaze**[n,g] and David Van Horn. Proceedings of the 10th ACM SIGPLAN Dynamic Languages Symposium, Portland, Oregon, 2014.
  https://doi.org/10.1145/2661088.2661098

- Pruning, Pushdown Exception-Flow Analysis, Shuying Liang[o,g], Weibin Sun[o,g], Matthew Might, Andrew W. Keep[o,p], and David Van Horn. Proceedings of the 14th IEEE International Conference on Software Code Analysis and Manipulation, Victoria, British Columbia, September, 2014.
  https://doi.org/10.1109/SCAM.2014.44

- Soft Contract Verification, **Phúc C. Nguyễn**[m,g], Sam Tobin-Hochstadt, and David Van Horn. Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming, Gothenborg, Sweden, September, 2014.
  http://dx.doi.org/10.1145/2628136.2628156

- Pushdown flow analysis with abstract garbage collection, **Dionna A. Glaze**[n,g], Ilya Sergey[o,p], Christopher Earl[o,g], Matthew Might, and David Van Horn. *Journal of Functional Programming*, 24(2-3), May 2014. (Special Issue for ICFP'12.)
  http://dx.doi.org/10.1017/S0956796814000100

- Optimizing Abstract Abstract Machines, **Dionna A. Glaze**[n,g], **Nicholas Labich**[n,u], Matthew Might, and David Van Horn. Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming (ICFP'13), Boston, Massachusetts, September, 2013.
  http://dx.doi.org/10.1145/2500365.2500604

- Higher-Order Symbolic Execution via Contracts, Sam Tobin-Hochstadt and David Van Horn. The ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'12), Tuscon, Arizona, October, 2012.
  http://dx.doi.org/10.1145/2384616.2384655

- Systematic Abstraction of Abstract Machines, David Van Horn and Matthew Might. *Journal of Functional Programming*, 22(4–5), August 2012, Special Issue for ICFP 2010.
  http://dx.doi.org/10.1017/S0956796812000238

- Introspective Pushdown Analysis of Higher-order Programs, Christopher Earl[o,g], Ilya Sergey[o,g], Matthew Might, and David Van Horn. Proceedings of the 17th ACM SIGPLAN International Conference on Functional Programming (ICFP'12), Copenhagen, Denmark, September, 2012.
  http://dx.doi.org/10.1145/2364527.2364576

- A Family of Abstract Interpretations for Static Analysis of Concurrent Higher-Order Programs, Matthew Might and David Van Horn. In The 18th International Static Analysis Symposium (SAS 2011), Venice, Italy, September, 2011. Lecture Notes in Computer Science, 6887.
  http://dx.doi.org/10.1007/978-3-642-23702-7_16

- Abstracting Abstract Machines: A Systematic Approach to Higher-Order Program Analysis, David Van Horn and Matthew Might. In Communications of the ACM, Research Highlights 54(9), September, 2011.
  http://dx.doi.org/10.1145/1995376.1995400

- Abstracting Abstract Machines, David Van Horn and Matthew Might. In Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming (ICFP'10), Baltimore, Maryland, September, 2010.
  http://dx.doi.org/10.1145/1863543.1863553

- Evaluating Call-By-Need on the Control Stack, Stephen Chang[n,g], David Van Horn, and Matthias Felleisen. In Symposium on Trends in Functional Programming (TFP'10), Norman, Oklahoma, May, 2010. Winner of the **best student paper award**.
  http://dx.doi.org/10.1007/978-3-642-22941-1_1

- Resolving and Exploiting the k-CFA Paradox: Illuminating Functional vs. Object-Oriented Program Analysis, Matthew Might, Yannis Smaragdakis, and David Van Horn. In Proceedings of the ACM SIGPLAN 2010 Conference on Programming Language Design and Implementation (PLDI'10), Toronto, Canada, June, 2010.
  http://dx.doi.org/10.1145/1806596.1806631

- Deciding kCFA is complete for EXPTIME, David Van Horn and Harry G. Mairson. In Proceedings of the 13th ACM SIGPLAN International Conference on Functional Programming (ICFP'08), Victoria, British Columbia, Canada, September, 2008.
  http://dx.doi.org/10.1145/1411204.1411243

- Flow Analysis, Linearity, and PTIME, David Van Horn and Harry G. Mairson. In The 15th International Static Analysis Symposium (SAS 2008), Valencia, Spain, July, 2008. Lecture Notes in Computer Science, 5079.
  http://dx.doi.org/10.1007/978-3-540-69166-2_17

- Types and Trace Effects of Higher Order Programs, Christian Skalka, Scott Smith, and David Van Horn. *Journal of Functional Programming* 18(2), March, 2008.
  http://dx.doi.org/10.1017/S0956796807006466

- Relating Complexity and Precision in Control Flow Analysis, David Van Horn and Harry G. Mairson. In Proceedings of the Twelth ACM SIGPLAN International Conference on Functional Programming (ICFP'07), Freiburg, Germany, October, 2007.
  http://dx.doi.org/10.1145/1291151.1291166

Articles in Workshop Proceedings

- Sound and Precise Malware Analysis for Android via Pushdown Reachability and Entry-Point Saturation. Shuying Liang[o,p], Andrew W. Keep[o,p], Matthew Might, Steven Lyde[o,g], Thomas Gilray[o,g], Petey Aldous[o,g], and David Van Horn. Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices, Berlin, Germany, November 2013.
  http://dx.doi.org/10.1145/2516760.2516769

- AnaDroid: Malware Analysis of Android with User-supplied Predicates. Shuying Liang[o,p], Matthew Might, and David Van Horn. Workshop on Tools for Automatic Program Analysis, Seattle, Washington, June 2013.

- From Principles to Practice with Class in the First Year. Sam Tobin-Hochstadt and David Van Horn. International Workshop on Trends in Functional Programming in Education, Provo, Utah, May 2013. Electronic Proceedings in Theoretical Computer Science, volume 136.
  http://dx.doi.org/10.4204/EPTCS.136

- A Type and Effect System for Flexible Abstract Interpretation of Java. Christian Skalka, Scott F. Smith, and David Van Horn. The ACM Workshop on Abstract Interpretations of Object-Oriented Programs, Paris, France, January 2005.
  http://dx.doi.org/10.1016/j.entcs.2005.01.027

Conferences, Workshops, and Talks

Invited Talks

- Symbolic Execution for Higher-Order Program Verification, Vrije Universiteit Brussel, Brussels, Belgium, May 2018.

- Symbolic Execution for Higher-Order Program Verification, Princeton University, Princeton, New Jersey, May 2018.

- Symbolic Execution for Higher-Order Program Verification, University of Washington, UCLA Compilers Colloquium, Los Angeles, California, April 2018.

- Symbolic Execution for Higher-Order Program Verification University of Washington, PLSE Colloquium, Seattle, Washington, April 2018.
- Redex, Abstract Machines, and Abstract Interpretation Oregon Programming Languages Summer School (OPLSS), University of Oregon, Eugene, Oregon, July 2017. Three days of lectures.
- Verification and Refutation of Behavioral Contracts with Higher-Order Symbolic Execution, University of Chile, PLEAID Seminar, Santiago, Chile, January 2016.
- Tutorial: Introduction to Redex with Abstracting Abstract Machines, University of Chile, PLEAID Seminar, Santiago, Chile, January 2016.
- Verification and Refutation of Behavioral Contracts with Higher-Order Symbolic Execution, Johns Hopkins University, PL Seminar, Baltimore, Maryland, October 2015.
- Young Researcher Panel, ACM SIGPLAN Programming Languages Mentoring Workshop at ICFP, Vancouver, British Columbia, August 2015.
- Abstracting Abstract Machines, PLT Redex Summer School, University of Utah, Salt Lake City, Utah, July 2015.
- Verification and Refutation of Behavioral Contracts with Higher-Order Symbolic Execution, PL Wonks Seminar, Indiana University, Bloomington, Indiana, January 2015.
- Towards the Verification of Behavioral Software Contracts, Microsoft Research, RiSE Group invited lecture, Redmond, Washington, November 2012.
- Program Verification via Abstract Reduction Semantics, Invited lectures, Harvard University, Advanced Functional Language Compilation, Cambridge, Massachusetts, November 2012.
- Optimized Machines for Program Analysis, Invited lectures, Harvard University, Advanced Functional Language Compilation, Cambridge, Massachusetts, November 2012.
- Abstract Machines for Program Analysis, Invited lectures, Harvard University, Advanced Functional Language Compilation, Cambridge, Massachusetts, November 2012.

Symposia

- Soft Contract Verification, Dagstuhl Seminar on Scripting Languages and Frameworks: Analysis and Verification, Schloss Dagstuhl, Germany, July 2014.
- Analysis for Trustworthy Software, Third Annual Maryland Cybersecurity Center Symposium, College Park, Maryland, June 2014.
- Soft Contract Verification, NII Workshop on Software Contracts for Communication, Monitoring, and Security, Shonan Village, Japan, May 2014.

- Abstracting Definitional Interpreters, Mid-Atlantic Programming Languages Seminar, College Park, Maryland, April 2013.
- Verification via Abstract Reduction, NII Workshop on Automated Techniques for Automated Higher-order Program Verification, Shonan Village, Japan, September, 2011.
- The Complexity of kCFA, NII Workshop on Automated Techniques for Automated Higher-order Program Verification, Shonan Village, Japan, September, 2011.
- Modular Analysis via Abstract Reduction Semantics, New Jersey Programming Languages and Systems Symposium, Rutgers University, Piscataway, New Jersey, December.
- Pushdown Control-Flow Analysis of Higher-Order Programs, IBM Programming Languages Day, Hawthorne, New York, July.
- Abstracting Abstract Machines: Storing and Stacking Continuations, Harvard Programming Languages Seminar, Harvard University, Cambridge, Massachusetts, July, 2010.
- Abstracting Abstract Machines, New England Programming Languages and Systems Symposium, Yale University, New Haven, Connecticut, April, 2010.
- Resolving and Exploiting the k-CFA Paradox, New England Programming Languages and Systems Symposium, MIT, Cambridge, Massachusetts, December, 2009.
- Subcubic Control-Flow Analysis Algorithms, ACM Symposium in Honor of Mitchell Wand, Northeastern University, Boston, Massachusetts, August, 2009.
- The Complexity of Flow Analysis, New England Programming Languages and Systems Symposium, Harvard University, Boston, Massachusetts, November.
- The Complexity of Flow Analysis, Northeastern University, Graduate Programming Languages Seminar, Boston, Massachusetts, October, 2008.
- Relating Complexity and Precision in Control Flow Analysis, Northeastern University, Programming Languages Seminar, Boston, Massachusetts, May, 2007.
- Relating Complexity and Precision in Control Flow Analysis, IBM Programming Languages Day, Hawthorne, New York, May, 2007.
- Linearity and Program Analysis, Northeastern University, Graduate Programming Languages Seminar, Boston, Massachusetts, October.

Workshops

- Concrete Semantics for Pushdown Analysis: The Essence of Summarization. **Dionna A. Glaze**[n,g] and David Van Horn. Workshop on Higher-Order Program Analysis, New Orleans, Louisiana, June 2013. `http://arxiv.org/abs/1305.3163`

- Semantic Solutions to Program Analysis Problems. Sam Tobin-Hochstadt and David Van Horn. FIT Session, The ACM SIGPLAN 2011 Conference on Programming Language Design and Implementation (PLDI'11), San Jose, California, June 2011.
- Pushdown Control-Flow Analysis of Higher-Order Programs. Christopher Earl, Matthew Might, and David Van Horn. The 2010 Workshop on Scheme and Functional Programming (SFP'10), Montréal, Québec, August 2010.

Colloquia

- Analysis for Trustworthy Software, Computer Science Colloquium, University of Maryland, College Park, Maryland, March 2013.
- What Program Analysis Can and Cannot Do for You, Rice University CS Colloquium, Houston, Texas, March, 2011.
- What Program Analysis Can and Cannot Do for You, University of Utah CS Colloquium, Salt Lake City, Utah, February, 2011.
- The Paradox of Flow Analysis, Or: What We Talk About When We Talk About Higher-Order Flow Analysis, MIT Programming Languages Working Group, MIT, Cambridge, Massachusetts, February, 2011.
- Resolving and Exploiting the k-CFA Paradox, University of Oregon CIS Colloquium, Eugene, Oregon, April, 2010.

Other

- Synthesis from Contracts, Defense Advanced Research Projects Agency, Arlington, Virginia, March 2014.
- Program Analysis for Trustworthy Software, Laboratory for Telecommunication Sciences, College Park, Maryland, March 2014.
- Analyzing Software Contracts, DARPA Clean-slate design of Resilient Adaptive Secure Hosts, Boston, Massachusetts, December 2012.
- Raising the Level of Discourse with GnoSys, DARPA Clean-slate design of Resilient Adaptive Secure Hosts, San Diego, California, November 2012.
- Scalable Abstractions for Trustworthy Software, DARPA Automated Program Analysis for Cybersecurity, Arlington, Virginia, October, 2012.
- Low-level Analysis for High-level Assurance, DARPA Clean-slate design of Resilient Adaptive Secure Hosts, Boston, Massachusetts, October, 2011.

Professional Publications

Reports and Non-Refereed Monographs

- Theorem Proving for All: Equational Reasoning in Liquid Haskell, **Niki Vazou**[m,p], Joachim Breitner[o,p], **Will Kunkel**[m,u], David Van Horn, and Graham Hutton. In submission, 2018.
- Automated Techniques for Higher-Order Program Verification, Naoki Kobayashi, Luke Ong, and David Van Horn. Progress in Informatics, No. 10, pp.157–165, 2013.
  `http://www.nii.ac.jp/pi/n10/10_157.pdf`
- Pushdown Abstractions of Javascript, David Van Horn and Matthew Might. CoRR, abs/1109.4467, 2011.
  `http://arxiv.org/abs/1109.4467`

Sponsored Research

Grants

- NSF: SHF: Small: Collaborative Research: Online Verification-Validation, $140,009, 08/01/2016 – 08/31/2019. Lead investigator.
- NSF: REU Supplement to SHF: Small: Collaborative Research: Online Verification-Validation, $16,000, 08/01/2016 – 08/31/2019. Lead investigator.
- NSA: E-VERIFY: Establishing a Science of Security Research Lablet at the University of Maryland: Task: Trustworthy and Composable Software Systems with Contracts, $86,497.60, 03/28/2016 – 07/31/2017. Co-Investigator.
- NSF: TWC: Large: Collaborative Research: The Science and Application of Crypto-Currency, $250,492, 09/01/2015 – 06/30/2018. Lead investigator.
- DARPA: SOUCIS: Sound Over- & Under-Approximations of Complexity & Information Security, $830,104.77, 04/16/2015 – 04/15/2018. Co-Investigator.
- NSA: E-VERIFY: Establishing a Science of Security Research Lablet at the University of Maryland: Task: Trustworthy and Composable Software Systems with Contracts, $98,080, 03/28/2015 – 03/27/2016. Co-Investigator.
- NSA: E-VERIFY: Establishing a Science of Security Research Lablet at the University of Maryland: Task: Trustworthy and Composable Software Systems with Contracts, $117,332.80, 02/07/2014 – 07/31/2015. Co-Investigator.
- DARPA: Scalable and Precise Abstractions of Programs for Trustworthy Software, $541,557, 12/15/2013 – 06/30/2016. Lead investigator.
- NSF: Behavioral Software Contract Verification, $400,000. 2012 – 2013. (Northeastern University)

- NSF: CRA Computing Innovation Fellow, $267,500, 2009 – 2011. (Northeastern University)

III. Teaching, Mentoring and Advising.

Courses Taught

- Systematic Program Design II, CMSC 132A (75), Spring 2018
- Systematic Program Design I, CMSC 131A (140), Fall 2017
- Introduction to Compilers, CMSC 430 (40), Spring 2017
- Introduction to Compilers, CMSC 430 (40), Spring 2016
- Program Analysis and Understanding, CMSC 631 (20), Fall 2015
- Introduction to Compilers, CMSC 430 (40), Spring 2015
- Program Analysis and Understanding, CMSC 631 (15), Fall 2014
- Program Analysis and Understanding, CMSC 631 (20), Spring 2014
- (Northeastern University) Intro. to Programming and Computing I (247), 2007–2010.
- (Northeastern University) Intro. to Programming and Computing I, Honors (134), 2009–2011.
- (Northeastern University) Intro. to Programming and Computing II (312), 2008–2009, 2012–2013.
- (Northeastern University) Intro. to Programming and Computing II, Honors (110), 2011-2013.

Teaching Innovations

Software, Applications, Online Education, etc.

- The Class Language, a pedagogical language for teaching object-oriented programming. `https://github.com/dvanhorn/dpc`

Course or Curriculum Development

- CMSC 131A: Systematic Program Design I, Fall 2017. This course (along with 132A) is a complete redesign of the first year introductory programming cirriculum. The course is an introduction to computing and programming. Its major goal is to introduce students to the principles of systematic problem solving through programming and the basic rules of computation. This course exposes students to the fundamental techniques of program design: an approach to the creation of software that relies on systematic thought, planning, and understanding from the very beginning, at every stage and for every step. I wrote extensive notes, a complete set of labs, assignments, midterms, practice problems, practice exams, and final exams. Video recordings of all lectures, including screen captures of in-class coding, are available on Panopto.

- CMSC 132A: Systematic Program Design II, Spring 2018. This is the follow-up course to 131A. It studies the class-based program design and the design of abstractions that support the design of reusable software and libraries. It covers the principles of object oriented program design, the basic rules of program evaluation, and examines the relationship between algorithms and data structures, as well as basic techniques for analyzing algorithm complexity. I wrote extensive notes, a complete set of labs, assignments, midterms, practice problems, practice exams, and final exams. Video recordings of all lectures, including screen captures of in-class coding, are available on Panopto.

- CMSC 430: Introduction to Compilers, 2015–2017. Each semester I have taught this course, I have revised existing materials and written new midterms and exams.

- CMSC 631: Program Analysis and Understanding, 2014–2016. Upon joining the faculty at UMD, I did a from scratch redesign of the PhD-level PL and program analysis course. The course objectives include to introducing students to the complementary research areas of programming languages and program analysis and exposing students to the basic principles of research processes in computer science: how to ask/articulate questions and how to recognize elements of solutions. It covers basic theoretical ideas and practical techniques for modeling and analyzing programming languages; and leveraging those techniques to mechanically reason about programs. As part of the course development, I wrote an extensive set of course notes, designed research projects, developed lectures, exams, and programming assignments. I have signficantly revised and refined the course over three iterations.

- (Northeastern University) Intro. to Programming and Computing II, Honors, with Sam Tobin-Hochstadt, 2011. With Tobin-Hochstadt, I designed this course from scratch to offer a second semester programming course for Honors College and advanced students to take as an alternative to the standard CS II course. We developed an extensive set of course notes, a pedagogicaly oriented programming language for exploring class- and object-oriented designs, and a complete set of exams, labs, and problem sets.

- (Northeastern University) Intro. to Programming and Computing I, Honors, with Olin Shivers, 2009. With Shivers, I adapted the existing intro course for onors Collegeand advanced students to take as analternative to the standard CS I course. We developed several new units on advanced topics such as interpreters and control operators as well as accompanying labs, exams, and problem sets.

Advising: Research or Clinical

Undergraduate

- William Kunkel, 2016–2018. Now: PhD student at University of California, San Diego.

- Kyle Headley, 2014–2015. Now: PhD student at University of Colorado, Boulder.

- Rebecca MacKenzie, co-op adviser, 2014. Now: Academic IT Coordinator at Norteastern University.

Doctoral

- Phuc C. Nguyen, PhD adviser, 2014.
- Nicholas Labich, PhD adviser, 2014.
- Quentin Stiévenart, Scalable Designs for Abstract Interpretation of Concurrent Programs: Application to Actors and Shared-Memory Multi-Threading, Vrije Universiteit Brussel, PhD committee member, 2018.
- David Darais, Mechanizing Abstract Interpretation, 2017. Now: Assistant Professor in Computer Science at the University of Vermont (tenure track).
- Kristopher Micinski, Interaction-Based Privacy Policies for Mobile Apps, PhD committee member. Now: Visiting assistant professor at Haverford College.
- Piotr Mardziel, Modeling, Measuring, and Limiting Adversary Knowledge, PhD committee member, 2015. Now: Post-doc at Carnegie Mellon University.
- Dionna A. Glaze, Automating Abstract Interpretation of Abstract Machines, PhD adviser, Northeastern University, 2014. Now: Google.
- Stephen Chang, On the Relation Between Laziness and Strictness, Ph.D., Northeastern University, Committee member, 2014. Now: Associate Research Scientist at Northeastern University.
- Vincent St-Amour, Ph.D. candidate at Northeastern University, PhD committee member, 2014. Now: Warren Teaching Fellow at Northwestern University.
- Shuying Liang, Static Analysis of Android Applications, Ph.D., University of Utah, PhD committee member, 2014. Now: HP Fortify Labs.
- Letterio Galletta, Adaptivity: Linguistic Mechanisms and Static Analysis Techniques, University of Pisa, PhD thesis reviewer, 2014. Now: Post-doc at School for Advanced Studies, Lucca.

Post-doctoral

- Niki Vazou, Basili Fellow, post-doctoral adviser, 2016–2018. Soon: Assistant Professor at IMDEA Software Institute (tenure track).
- Thomas Gilray, Basili Fellow, post-doctoral adviser, 2016–2018. Soon: Assistant Professor in Computer Science at the University of Alabama, Birmingham (tenure track).
- Shiyi Wei, post-doctoral adviser, 2015–2017. Now: Assistant Professor in Computer Science at the University of Texas at Dallas (tenure track).
- Matthew Hammer, post-doctoral adviser, 2014. Now: Assistant Professor in Computer Science at the University of Colorado, Boulder (tenure track).
- Neil Toronto, post-doctoral adviser, 2014. Now: Microsoft Research, Cambridge.

IV. Service and Outreach

Editorships, Editorial Boards, and Reviewing Activities

Reviewing Activities for Journals and Presses

- ACM Computing Surveys
- ACM Transactions on Computational Logic
- ACM Transactions on Programming Languages and Systems
- Higher-Order and Symbolic Computation
- Journal of Functional Programming
- Science of Computer Programming

Reviewing Activities for Agencies and Foundations

- NSF Directorate for Computer & Information Science & Engineering (CISE), 2011.
- NSF Directorate for Computer & Information Science & Engineering (CISE), 2010.

Reviewing Activities for Conferences

- ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Program committee member, 2019.
- ACM SIGPLAN International Conference on Functional Programming (ICFP), External Review committee member, 2018.
- International Symposium on Practical Aspects of Declarative Languages (PADL), Program committee member, 2018.
- Symposium on Trends in Functional Programming (TFP), Program committee memberm 2017.
- Static Analysis Symposium (SAS), Program committee memberm 2017.
- ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Program committee member, 2017.
- Symposium on Trends in Functional Programming (TFP), Program Chair, 2016.
- European Conference on Object-Oriented Programming (ECOOP), Program committee member, 2016.
- International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), Reviewer, 2016.

- ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, External Review committee member, 2016.
- ACM SIGPLAN International Conference on Functional Programming (ICFP), Program committee member, 2015.
- ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Reviewer, 2015.
- Off the Beaten Track: New Frontiers for Programming Languages Research, Program committee member, 2015.
- European Symposium on Programming (ESOP), Program committee member, 2014.
- Workshop on Higher-Order Program Analysis, Program Chair, 2014.
- International Symposium on Practical Aspects of Declarative Languages (PADL), Program committee member, 2014.
- Workshop on Syntax and Semantics of Low-Level Languages (LOLA), Program committee member, 2014.
- Symposium on Trends in Functional Programming (TFP), Program committee member, 2014.
- International Workshop on Trends in Functional Programming in Education (TFPIE), Program committee member, 2014.
- ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Reviewer, 2014.
- International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), Reviewer, 2014.
- ACM SIGPLAN International Conference on Functional Programming (ICFP), Reviewer, 2014.
- ACM SIGPLAN Symposium on Dynamic Languages (DLS), Reviewer, 2014.
- ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), External review committee member, 2013.
- Scala Workshop, Program committee member, 2013.
- Workshop on Higher-Order Program Analysis (HOPA), Program committee member, 2013.
- Symposium on Trends in Functional Programming (TFP), Program committee member, 2013.
- ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA), Reviewer, 2012.

- Dynamic Languages Symposium (DLS), Reviewer, 2012.
- ACM SIGPLAN International Conference on Functional Programming (ICFP), Program committee member, 2011.
- Scheme and Functional Programming Workshop, Program committee member, 2011.
- European Symposium on Programming Languages (ESOP), 2011.
- ACM SIGPLAN International Conference on Functional Programming (ICFP), Reviewer, 2010.
- Scheme and Functional Programming Workshop, Program committee member, 2009.
- ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Reviewer, 2008.
- IEEE Symposium on Logic in Computer Science (LICS), Reviewer, 2007.
- EACSL Conference on Computer Science and Logic (CSL), Reviewer, 2007.

Other

- ACM SIGPLAN International Conference on Functional Programming (ICFP), Steering committee member, 2013–2015.
- ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), Workshop chair, 2013–2015.
- Workshop on Higher-Order Program Analysis (HOPA), Chair, 2014.
- Workshop on Higher-Order Program Analysis (HOPA), Steering committee member, 2014.
- NII Workshop on Automated Techniques for Higher-Order Program Verification, Chair, 2011.
- New England Programming Languages and Systems Symposium (NEPLS), Chair, 2011.
- New England Programming Languages and Systems Symposium (NEPLS), Speaker selection committee, 2012.

Committees, Professional & Campus Service

Campus Service - Department

- Graduate Student Review Committee, Chair, 2016–2018.
- Middle States Committee, 2016–2018.
- Graduate Student Review Committee, Co-Chair, 2014–2016.

- Graduate Admissions Committee, 2014–2016.
- Space Planning Committee (Iribe Center for Computer Science and Innovation), 2014–2018.
- Education Committee, 2013–2018.
- PL/SE/HCI Field Committee, 2013–2018.

V. Awards, Honors and Recognition

Research Fellowships, Prizes and Awards

- Computing Research Association, Computing Innovation Fellow, 2009–2011.

Teaching Awards

- Northeastern University Excellence in Teaching Award Nominee, 2013.

Other Special Recognition

- Communications of the ACM, Research Highlight, 2011.
- ACM Doctoral Dissertation Award Nominee, 2009.